Apostolos Antonacopoulos ·
Subhasis Chaudhuri · Rama Chellappa ·
Cheng-Lin Liu · Saumik Bhattacharya ·
Umapada Pal (Eds.)

LNCS 15322

Pattern Recognition

27th International Conference, ICPR 2024
Kolkata, India, December 1–5, 2024
Proceedings, Part XXII

22 Part XXII

ICPR 2024 INDIA

IAPR

Springer

MOREMEDIA

Lecture Notes in Computer Science 15322

Founding Editors

Gerhard Goos
Juris Hartmanis

Editorial Board Members

Elisa Bertino, *Purdue University, West Lafayette, IN, USA*
Wen Gao, *Peking University, Beijing, China*
Bernhard Steffen, *TU Dortmund University, Dortmund, Germany*
Moti Yung, *Columbia University, New York, NY, USA*

The series Lecture Notes in Computer Science (LNCS), including its subseries Lecture Notes in Artificial Intelligence (LNAI) and Lecture Notes in Bioinformatics (LNBI), has established itself as a medium for the publication of new developments in computer science and information technology research, teaching, and education.

LNCS enjoys close cooperation with the computer science R & D community, the series counts many renowned academics among its volume editors and paper authors, and collaborates with prestigious societies. Its mission is to serve this international community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings. LNCS commenced publication in 1973.

Apostolos Antonacopoulos ·
Subhasis Chaudhuri · Rama Chellappa ·
Cheng-Lin Liu · Saumik Bhattacharya ·
Umapada Pal
Editors

Pattern Recognition

27th International Conference, ICPR 2024
Kolkata, India, December 1–5, 2024
Proceedings, Part XXII

Springer

Editors

Apostolos Antonacopoulos
University of Salford
Salford, Lancashire, UK

Rama Chellappa
Johns Hopkins University
Baltimore, MD, USA

Saumik Bhattacharya
IIT Kharagpur
Kharagpur, West Bengal, India

Subhasis Chaudhuri
Indian Institute of Technology Bombay
Mumbai, Maharashtra, India

Cheng-Lin Liu
Chinese Academy of Sciences
Beijing, China

Umapada Pal
Indian Statistical Institute Kolkata
Kolkata, West Bengal, India

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-031-78311-1 ISBN 978-3-031-78312-8 (eBook)
https://doi.org/10.1007/978-3-031-78312-8

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2025

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

President's Address

On behalf of the Executive Committee of the International Association for Pattern Recognition (IAPR), I am pleased to welcome you to the 27th International Conference on Pattern Recognition (ICPR 2024), the main scientific event of the IAPR.

After a completely digital ICPR in the middle of the COVID pandemic and the first hybrid version in 2022, we can now enjoy a fully back-to-normal ICPR this year. I look forward to hearing inspirational talks and keynotes, catching up with colleagues during the breaks and making new contacts in an informal way. At the same time, the conference landscape has changed. Hybrid meetings have made their entrance and will continue. It is exciting to experience how this will influence the conference. Planning for a major event like ICPR must take place over a period of several years. This means many decisions had to be made under a cloud of uncertainty, adding to the already large effort needed to produce a successful conference. It is with enormous gratitude, then, that we must thank the team of organizers for their hard work, flexibility, and creativity in organizing this ICPR. ICPR always provides a wonderful opportunity for the community to gather together. I can think of no better location than Kolkata to renew the bonds of our international research community.

Each ICPR is a bit different owing to the vision of its organizing committee. For 2024, the conference has six different tracks reflecting major themes in pattern recognition: Artificial Intelligence, Pattern Recognition and Machine Learning; Computer and Robot Vision; Image, Speech, Signal and Video Processing; Biometrics and Human Computer Interaction; Document Analysis and Recognition; and Biomedical Imaging and Bioinformatics. This reflects the richness of our field. ICPR 2024 also features two dozen workshops, seven tutorials, and 15 competitions; there is something for everyone. Many thanks to those who are leading these activities, which together add significant value to attending ICPR, whether in person or virtually. Because it is important for ICPR to be as accessible as possible to colleagues from all around the world, we are pleased that the IAPR, working with the ICPR organizers, is continuing our practice of awarding travel stipends to a number of early-career authors who demonstrate financial need. Last but not least, we are thankful to the Springer LNCS team for their effort to publish these proceedings.

Among the presentations from distinguished keynote speakers, we are looking forward to the three IAPR Prize Lectures at ICPR 2024. This year we honor the achievements of Tin Kam Ho (IBM Research) with the IAPR's most prestigious King-Sun Fu Prize "for pioneering contributions to multi-classifier systems, random decision forests, and data complexity analysis". The King-Sun Fu Prize is given in recognition of an outstanding technical contribution to the field of pattern recognition. It honors the memory of Professor King-Sun Fu who was instrumental in the founding of IAPR, served as its first president, and is widely recognized for his extensive contributions to the field of pattern recognition.

The Maria Petrou Prize is given to a living female scientist/engineer who has made substantial contributions to the field of Pattern Recognition and whose past contributions, current research activity and future potential may be regarded as a model to both aspiring and established researchers. It honours the memory of Professor Maria Petrou as a scientist of the first rank, and particularly her role as a pioneer for women researchers. This year, the Maria Petrou Prize is given to Guoying Zhao (University of Oulu), "for contributions to video analysis for facial micro-behavior recognition and remote biosignal reading (RPPG) for heart rate analysis and face anti-spoofing".

The J.K. Aggarwal Prize is given to a young scientist who has brought a substantial contribution to a field that is relevant to the IAPR community and whose research work has had a major impact on the field. Professor Aggarwal is widely recognized for his extensive contributions to the field of pattern recognition and for his participation in IAPR's activities. This year, the J.K. Aggarwal Prize goes to Xiaolong Wang (UC San Diego) "for groundbreaking contributions to advancing visual representation learning, utilizing self-supervised and attention-based models to establish fundamental frameworks for creating versatile, general-purpose pattern recognition systems".

During the conference we will also recognize 21 new IAPR Fellows selected from a field of very strong candidates. In addition, a number of Best Scientific Paper and Best Student Paper awards will be presented, along with the Best Industry Related Paper Award and the Piero Zamperoni Best Student Paper Award. Congratulations to the recipients of these very well-deserved awards!

I would like to close by again thanking everyone involved in making ICPR 2024 a tremendous success; your hard work is deeply appreciated. These thanks extend to all who chaired the various aspects of the conference and the associated workshops, my ExCo colleagues, and the IAPR Standing and Technical Committees. Linda O'Gorman, the IAPR Secretariat, deserves special recognition for her experience, historical perspective, and attention to detail when it comes to supporting many of the IAPR's most important activities. Her tasks became so numerous that she recently got support from Carolyn Buckley (layout, newsletter), Ugur Halici (ICPR matters), and Rosemary Stramka (secretariat). The IAPR website got a completely new design. Ed Sobczak has taken care of our web presence for so many years already. A big thank you to all of you!

This is, of course, the 27th ICPR conference. Knowing that ICPR is organized every two years, and that the first conference in the series (1973!) pre-dated the formal founding of the IAPR by a few years, it is also exciting to consider that we are celebrating over 50 years of ICPR and at the same time approaching the official IAPR 50th anniversary in 2028: you'll get all information you need at ICPR 2024. In the meantime, I offer my thanks and my best wishes to all who are involved in supporting the IAPR throughout the world.

September 2024

Arjan Kuijper
President of the IAPR

Preface

It is our great pleasure to welcome you to the proceedings of the 27th International Conference on Pattern Recognition (ICPR 2024), held in Kolkata, India. The city, formerly known as ‘Calcutta’, is the home of the fabled Indian Statistical Institute (ISI), which has been at the forefront of statistical pattern recognition for almost a century. Concepts like the Mahalanobis distance, Bhattacharyya bound, Cramer–Rao bound, and Fisher–Rao metric were invented by pioneers associated with ISI. The first ICPR (called IJCPR then) was held in 1973, and the second in 1974. Subsequently, ICPR has been held every other year. The International Association for Pattern Recognition (IAPR) was founded in 1978 and became the sponsor of the ICPR series. Over the past 50 years, ICPR has attracted huge numbers of scientists, engineers and students from all over the world and contributed to advancing research, development and applications in pattern recognition technology.

ICPR 2024 was held at the Biswa Bangla Convention Centre, one of the largest such facilities in South Asia, situated just 7 kilometers from Kolkata Airport (CCU). According to ChatGPT “Kolkata is often called the ‘Cultural Capital of India’. The city has a deep connection to literature, music, theater, and art. It was home to Nobel laureate Rabindranath Tagore, and the Bengali film industry has produced globally renowned filmmakers like Satyajit Ray. The city boasts remarkable colonial architecture, with landmarks like Victoria Memorial, Howrah Bridge, and the Indian Museum (the oldest and largest museum in India). Kolkata’s streets are dotted with old mansions and buildings that tell stories of its colonial past. Walking through the city can feel like stepping back into a different era. Finally, Kolkata is also known for its street food.”

ICPR 2024 followed a two-round paper submission format. We received a total of 2135 papers (1501 papers in round-1 submissions, and 634 papers in round-2 submissions). Each paper, on average, received 2.84 reviews, in single-blind mode. For the first-round papers we had a rebuttal option available to authors.

In total, 945 papers (669 from round-1 and 276 from round-2) were accepted for presentation, resulting in an acceptance rate of 44.26%, which is consistent with previous ICPR events. At ICPR 2024 the papers were categorized into six tracks: Artificial Intelligence, Machine Learning for Pattern Analysis; Computer Vision and Robotic Perception; Image, Video, Speech, and Signal Analysis; Biometrics and Human-Machine Interaction; Document and Media Analysis; and Biomedical Image Analysis and Informatics.

The main conference ran over December 2–5, 2024. The main program included the presentation of 188 oral papers (19.89% of the accepted papers), 757 poster papers and 12 competition papers (out of 15 submitted). A total 10 oral sessions were held concurrently in four meeting rooms with a total of 40 oral sessions. In total 24 workshops and 7 tutorials were held on December 1, 2024.

The plenary sessions included three prize lectures and three invited presentations. The prize lectures were delivered by Tin Kam Ho (IBM Research, USA; King Sun

Fu Prize winner), Xiaolong Wang (University of California, San Diego, USA; J.K. Aggarwal Prize winner), and Guoying Zhao (University of Oulu, Finland; Maria Petrou Prize winner). The invited speakers were Timothy Hospedales (University of Edinburgh, UK), Venu Govindaraju (University at Buffalo, USA), and Shuicheng Yan (Skywork AI, Singapore).

Several best paper awards were presented in ICPR: the Piero Zamperoni Award for the best paper authored by a student, the BIRPA Best Industry Related Paper Award, and the Best Paper Awards and Best Student Paper Awards for each of the six tracks of ICPR 2024.

The organization of such a large conference would not be possible without the help of many volunteers. Our special gratitude goes to the Program Chairs (Apostolos Antonacopoulos, Subhasis Chaudhuri, Rama Chellappa and Cheng-Lin Liu), for their leadership in organizing the program. Thanks to our Publication Chairs (Ananda S. Chowdhury and Wataru Ohyama) for handling the overwhelming workload of publishing the conference proceedings. We also thank our Competition Chairs (Richard Zanibbi, Lianwen Jin and Laurence Likforman-Sulem) for arranging 12 important competitions as part of ICPR 2024. We are thankful to our Workshop Chairs (P. Shivakumara, Stephanie Schuckers, Jean-Marc Ogier and Prabir Bhattacharya) and Tutorial Chairs (B.B. Chaudhuri, Michael R. Jenkin and Guoying Zhao) for arranging the workshops and tutorials on emerging topics. ICPR 2024, for the first time, held a Doctoral Consortium. We would like to thank our Doctoral Consortium Chairs (Véronique Eglin, Dan Lopresti and Mayank Vatsa) for organizing it.

Thanks go to the Track Chairs and the meta reviewers who devoted significant time to the review process and preparation of the program. We also sincerely thank the reviewers who provided valuable feedback to the authors.

Finally, we acknowledge the work of other conference committee members, like the Organizing Chairs and Organizing Committee Members, Finance Chairs, Award Chair, Sponsorship Chairs, and Exhibition and Demonstration Chairs, Visa Chair, Publicity Chairs, and Women in ICPR Chairs, whose efforts made this event successful. We also thank our event manager Alpcord Network for their help.

We hope that all the participants found the technical program informative and enjoyed the sights, culture and cuisine of Kolkata.

October 2024

Umapada Pal
Josef Kittler
Anil Jain

Organization

General Chairs

Umapada Pal	Indian Statistical Institute, Kolkata, India
Josef Kittler	University of Surrey, UK
Anil Jain	Michigan State University, USA

Program Chairs

Apostolos Antonacopoulos	University of Salford, UK
Subhasis Chaudhuri	Indian Institute of Technology, Bombay, India
Rama Chellappa	Johns Hopkins University, USA
Cheng-Lin Liu	Institute of Automation, Chinese Academy of Sciences, China

Publication Chairs

Ananda S. Chowdhury	Jadavpur University, India
Wataru Ohyama	Tokyo Denki University, Japan

Competition Chairs

Richard Zanibbi	Rochester Institute of Technology, USA
Lianwen Jin	South China University of Technology, China
Laurence Likforman-Sulem	Télécom Paris, France

Workshop Chairs

P. Shivakumara	University of Salford, UK
Stephanie Schuckers	Clarkson University, USA
Jean-Marc Ogier	Université de la Rochelle, France
Prabir Bhattacharya	Concordia University, Canada

Tutorial Chairs

B. B. Chaudhuri	Indian Statistical Institute, Kolkata, India
Michael R. Jenkin	York University, Canada
Guoying Zhao	University of Oulu, Finland

Doctoral Consortium Chairs

Véronique Eglin	CNRS, France
Daniel P. Lopresti	Lehigh University, USA
Mayank Vatsa	Indian Institute of Technology, Jodhpur, India

Organizing Chairs

Saumik Bhattacharya	Indian Institute of Technology, Kharagpur, India
Palash Ghosal	Sikkim Manipal University, India

Organizing Committee

Santanu Phadikar	West Bengal University of Technology, India
SK Md Obaidullah	Aliah University, India
Sayantari Ghosh	National Institute of Technology Durgapur, India
Himadri Mukherjee	West Bengal State University, India
Nilamadhaba Tripathy	Clarivate Analytics, USA
Chayan Halder	West Bengal State University, India
Shibaprasad Sen	Techno Main Salt Lake, India

Finance Chairs

Kaushik Roy	West Bengal State University, India
Michael Blumenstein	University of Technology Sydney, Australia

Awards Committee Chair

Arpan Pal	Tata Consultancy Services, India

Sponsorship Chairs

P. J. Narayanan	Indian Institute of Technology, Hyderabad, India
Yasushi Yagi	Osaka University, Japan
Venu Govindaraju	University at Buffalo, USA
Alberto Bel Bimbo	Università di Firenze, Italy

Exhibition and Demonstration Chairs

Arjun Jain	FastCode AI, India
Agnimitra Biswas	National Institute of Technology, Silchar, India

International Liaison, Visa Chair

Balasubramanian Raman	Indian Institute of Technology, Roorkee, India

Publicity Chairs

Dipti Prasad Mukherjee	Indian Statistical Institute, Kolkata, India
Bob Fisher	University of Edinburgh, UK
Xiaojun Wu	Jiangnan University, China

Women in ICPR Chairs

Ingela Nystrom	Uppsala University, Sweden
Alexandra B. Albu	University of Victoria, Canada
Jing Dong	Institute of Automation, Chinese Academy of Sciences, China
Sarbani Palit	Indian Statistical Institute, Kolkata, India

Event Manager

Alpcord Network

Track Chairs – Artificial Intelligence, Machine Learning for Pattern Analysis

Larry O’Gorman	Nokia Bell Labs, USA
Dacheng Tao	University of Sydney, Australia
Petia Radeva	University of Barcelona, Spain
Susmita Mitra	Indian Statistical Institute, Kolkata, India
Jiliang Tang	Michigan State University, USA

Track Chairs – Computer and Robot Vision

C. V. Jawahar	International Institute of Information Technology (IIIT), Hyderabad, India
João Paulo Papa	São Paulo State University, Brazil
Maja Pantic	Imperial College London, UK
Gang Hua	Dolby Laboratories, USA
Junwei Han	Northwestern Polytechnical University, China

Track Chairs – Image, Speech, Signal and Video Processing

P. K. Biswas	Indian Institute of Technology, Kharagpur, India
Shang-Hong Lai	National Tsing Hua University, Taiwan
Hugo Jair Escalante	INAOE, CINVESTAV, Mexico
Sergio Escalera	Universitat de Barcelona, Spain
Prem Natarajan	University of Southern California, USA

Track Chairs – Biometrics and Human Computer Interaction

Richa Singh	Indian Institute of Technology, Jodhpur, India
Massimo Tistarelli	University of Sassari, Italy
Vishal Patel	Johns Hopkins University, USA
Wei-Shi Zheng	Sun Yat-sen University, China
Jian Wang	Snap, USA

Track Chairs – Document Analysis and Recognition

Xiang Bai	Huazhong University of Science and Technology, China
David Doermann	University at Buffalo, USA
Josep Llados	Universitat Autònoma de Barcelona, Spain
Mita Nasipuri	Jadavpur University, India

Track Chairs – Biomedical Imaging and Bioinformatics

Jayanta Mukhopadhyay	Indian Institute of Technology, Kharagpur, India
Xiaoyi Jiang	Universität Münster, Germany
Seong-Whan Lee	Korea University, Korea

Metareviewers (Conference Papers and Competition Papers)

Wael Abd-Almageed	University of Southern California, USA
Maya Aghaei	NHL Stenden University, Netherlands
Alireza Alaei	Southern Cross University, Australia
Rajagopalan N. Ambasamudram	Indian Institute of Technology, Madras, India
Suyash P. Awate	Indian Institute of Technology, Bombay, India
Inci M. Baytas	Bogazici University, Turkey
Aparna Bharati	Lehigh University, USA
Brojeshwar Bhowmick	Tata Consultancy Services, India
Jean-Christophe Burie	University of La Rochelle, France
Gustavo Carneiro	University of Surrey, UK
Chee Seng Chan	Universiti Malaya, Malaysia
Sumohana S. Channappayya	Indian Institute of Technology, Hyderabad, India
Dongdong Chen	Microsoft, USA
Shengyong Chen	Tianjin University of Technology, China
Jun Cheng	Institute for Infocomm Research, A*STAR, Singapore
Albert Clapés	University of Barcelona, Spain
Oscar Dalmau	Center for Research in Mathematics, Mexico

Tyler Derr	Vanderbilt University, USA
Abhinav Dhall	Indian Institute of Technology, Ropar, India
Bo Du	Wuhan University, China
Yuxuan Du	University of Sydney, Australia
Ayman S. El-Baz	University of Louisville, USA
Francisco Escolano	University of Alicante, Spain
Siamac Fazli	Nazarbayev University, Kazakhstan
Jianjiang Feng	Tsinghua University, China
Gernot A. Fink	TU Dortmund University, Germany
Alicia Fornes	CVC, Spain
Junbin Gao	University of Sydney, Australia
Yan Gao	Amazon, USA
Yongsheng Gao	Griffith University, Australia
Caren Han	University of Melbourne, Australia
Ran He	Institute of Automation, Chinese Academy of Sciences, China
Tin Kam Ho	IBM, USA
Di Huang	Beihang University, China
Kaizhu Huang	Duke Kunshan University, China
Donato Impedovo	University of Bari, Italy
Julio Jacques	University of Barcelona and Computer Vision Center, Spain
Lianwen Jin	South China University of Technology, China
Wei Jin	Emory University, USA
Danilo Samuel Jodas	São Paulo State University, Brazil
Manjunath V. Joshi	DA-IICT, India
Jayashree Kalpathy-Cramer	Massachusetts General Hospital, USA
Dimosthenis Karatzas	Computer Vision Centre, Spain
Hamid Karimi	Utah State University, USA
Baiying Lei	Shenzhen University, China
Guoqi Li	Chinese Academy of Sciences, and Peng Cheng Lab, China
Laurence Likforman-Sulem	Institut Polytechnique de Paris/Télécom Paris, France
Aishan Liu	Beihang University, China
Bo Liu	Bytedance, USA
Chen Liu	Clarkson University, USA
Cheng-Lin Liu	Institute of Automation, Chinese Academy of Sciences, China
Hongmin Liu	University of Science and Technology Beijing, China
Hui Liu	Michigan State University, USA

Jing Liu	Institute of Automation, Chinese Academy of Sciences, China
Li Liu	University of Oulu, Finland
Qingshan Liu	Nanjing University of Posts and Telecommunications, China
Adrian P. Lopez-Monroy	Centro de Investigacion en Matematicas AC, Mexico
Daniel P. Lopresti	Lehigh University, USA
Shijian Lu	Nanyang Technological University, Singapore
Yong Luo	Wuhan University, China
Andreas K. Maier	FAU Erlangen-Nuremberg, Germany
Davide Maltoni	University of Bologna, Italy
Hong Man	Stevens Institute of Technology, USA
Lingtong Min	Northwestern Polytechnical University, China
Paolo Napoletano	University of Milano-Bicocca, Italy
Kamal Nasrollahi	Milestone Systems, Aalborg University, Denmark
Marcos Ortega	University of A Coruña, Spain
Shivakumara Palaiahnakote	University of Salford, UK
P. Jonathon Phillips	NIST, USA
Filiberto Pla	University Jaume I, Spain
Ajit Rajwade	Indian Institute of Technology, Bombay, India
Shanmuganathan Raman	Indian Institute of Technology, Gandhinagar, India
Imran Razzak	UNSW, Australia
Beatriz Remeseiro	University of Oviedo, Spain
Gustavo Rohde	University of Virginia, USA
Partha Pratim Roy	Indian Institute of Technology, Roorkee, India
Sanjoy K. Saha	Jadavpur University, India
Joan Andreu Sánchez	Universitat Politècnica de València, Spain
Claudio F. Santos	UFSCar, Brazil
Shin'ichi Satoh	National Institute of Informatics, Japan
Stephanie Schuckers	Clarkson University, USA
Srirangaraj Setlur	University at Buffalo, SUNY, USA
Debdoot Sheet	Indian Institute of Technology, Kharagpur, India
Jun Shen	University of Wollongong, Australia
Li Shen	JD Explore Academy, China
Chen Shengyong	Zhejiang University of technology and Tianjin University of Technology, China
Andy Song	RMIT University, Australia
Akihiro Sugimoto	National Institute of Informatics, Japan
Qianru Sun	Singapore Management University, Singapore
Arijit Sur	Indian Institute of Technology, Guwahati, India
Estefania Talavera	University of Twente, Netherlands

Wei Tang	University of Illinois at Chicago, USA
Joao M. Tavares	Universidade do Porto, Portugal
Jun Wan	NLPR, CASIA, China
Le Wang	Xi'an Jiaotong University, China
Lei Wang	Australian National University, Australia
Xiaoyang Wang	Tencent AI Lab, USA
Xinggang Wang	Huazhong University of Science and Technology, China
Xiao-Jun Wu	Jiangnan University, China
Yiding Yang	Bytedance, China
Xiwen Yao	Northwestern Polytechnical University, China
Xu-Cheng Yin	University of Science and Technology Beijing, China
Baosheng Yu	University of Sydney, Australia
Shiqi Yu	Southern University of Science and Technology, China
Xin Yuan	Westlake University, China
Yibing Zhan	JD Explore Academy, China
Jing Zhang	University of Sydney, Australia
Lefei Zhang	Wuhan University, China
Min-Ling Zhang	Southeast University, China
Wenbin Zhang	Florida International University, USA
Jiahuan Zhou	Peking University, China
Sanping Zhou	Xi'an Jiaotong University, China
Tianyi Zhou	University of Maryland, USA
Lei Zhu	Shandong Normal University, China
Pengfei Zhu	Tianjin University, China
Wangmeng Zuo	Harbin Institute of Technology, China

Reviewers (Competition Papers)

Liangcai Gao
Mingxin Huang
Lei Kang
Wenhui Liao
Yuliang Liu
Yongxin Shi
Da-Han Wang
Yang Xue
Wentao Yang
Jiaxin Zhang
Yiwu Zhong

Reviewers (Conference Papers)

Aakanksha Aakanksha
Aayush Singla
Abdul Muqeet
Abhay Yadav
Abhijeet Vijay Nandedkar
Abhimanyu Sahu
Abhinav Rajvanshi
Abhisek Ray
Abhishek Shrivastava
Abhra Chaudhuri
Aditi Roy
Adriano Simonetto
Adrien Maglo
Ahmed Abdulkadir
Ahmed Boudissa
Ahmed Hamdi
Ahmed Rida Sekkat
Ahmed Sharafeldeen
Aiman Farooq
Aishwarya Venkataramanan
Ajay Kumar
Ajay Kumar Reddy Poreddy
Ajita Rattani
Ajoy Mondal
Akbar K.
Akbar Telikani
Akshay Agarwal
Akshit Jindal
Al Zadid Sultan Bin Habib
Albert Clapés
Alceu Britto
Alejandro Peña
Alessandro Ortis
Alessia Auriemma Citarella
Alexandre Stenger
Alexandros Sopasakis
Alexia Toumpa
Ali Khan
Alik Pramanick
Alireza Alaei
Alper Yilmaz
Aman Verma
Amit Bhardwaj
Amit More
Amit Nandedkar
Amitava Chatterjee
Amos L. Abbott
Amrita Mohan
Anand Mishra
Ananda S. Chowdhury
Anastasia Zakharova
Anastasios L. Kesidis
Andras Horvath
Andre Gustavo Hochuli
André P. Kelm
Andre Wyzykowski
Andrea Bottino
Andrea Lagorio
Andrea Torsello
Andreas Fischer
Andreas K. Maier
Andreu Girbau Xalabarder
Andrew Beng Jin Teoh
Andrew Shin
Andy J. Ma
Aneesh S. Chivukula
Ángela Casado-García
Anh Quoc Nguyen
Anindya Sen
Anirban Saha
Anjali Gautam
Ankan Bhattacharyya
Ankit Jha
Anna Scius-Bertrand
Annalisa Franco
Antoine Doucet
Antonino Staiano
Antonio Fernández
Antonio Parziale
Anu Singha
Anustup Choudhury
Anwesan Pal
Anwesha Sengupta
Archisman Adhikary
Arjan Kuijper
Arnab Kumar Das

Arnav Bhavsar
Arnav Varma
Arpita Dutta
Arshad Jamal
Artur Jordao
Arunkumar Chinnaswamy
Aryan Jadon
Aryaz Baradarani
Ashima Anand
Ashis Dhara
Ashish Phophalia
Ashok K. Bhateja
Ashutosh Vaish
Ashwani Kumar
Asifuzzaman Lasker
Atefeh Khoshkhahtinat
Athira Nambiar
Attilio Fiandrotti
Avandra S. Hemachandra
Avik Hati
Avinash Sharma
B. H. Shekar
B. Uma Shankar
Bala Krishna Thunakala
Balaji Tk
Balázs Pálffy
Banafsheh Adami
Bang-Dang Pham
Baochang Zhang
Baodi Liu
Bashirul Azam Biswas
Beiduo Chen
Benedikt Kottler
Beomseok Oh
Berkay Aydin
Berlin S. Shaheema
Bertrand Kerautret
Bettina Finzel
Bhavana Singh
Bibhas C. Dhara
Bilge Gunsel
Bin Chen
Bin Li
Bin Liu
Bin Yao
Bin-Bin Jia
Binbin Yong
Bindita Chaudhuri
Bindu Madhavi Tummala
Binh M. Le
Bi-Ru Dai
Bo Huang
Bo Jiang
Bob Zhang
Bowen Liu
Bowen Zhang
Boyang Zhang
Boyu Diao
Boyun Li
Brian M. Sadler
Bruce A. Maxwell
Bryan Bo Cao
Buddhika L. Semage
Bushra Jalil
Byeong-Seok Shin
Byung-Gyu Kim
Caihua Liu
Cairong Zhao
Camille Kurtz
Carlos A. Caetano
Carlos D. Martã-Nez-Hinarejos
Ce Wang
Cevahir Cigla
Chakravarthy Bhagvati
Chandrakanth Vipparla
Changchun Zhang
Changde Du
Changkun Ye
Changxu Cheng
Chao Fan
Chao Guo
Chao Qu
Chao Wen
Chayan Halder
Che-Jui Chang
Chen Feng
Chenan Wang
Cheng Yu
Chenghao Qian
Cheng-Lin Liu

Chengxu Liu
Chenru Jiang
Chensheng Peng
Chetan Ralekar
Chih-Wei Lin
Chih-Yi Chiu
Chinmay Sahu
Chintan Patel
Chintan Shah
Chiranjoy Chattopadhyay
Chong Wang
Choudhary Shyam Prakash
Christophe Charrier
Christos Smailis
Chuanwei Zhou
Chun-Ming Tsai
Chunpeng Wang
Ciro Russo
Claudio De Stefano
Claudio F. Santos
Claudio Marrocco
Connor Levenson
Constantine Dovrolis
Constantine Kotropoulos
Dai Shi
Dakshina Ranjan Kisku
Dan Anitei
Dandan Zhu
Daniela Pamplona
Danli Wang
Danqing Huang
Daoan Zhang
Daqing Hou
David A. Clausi
David Freire Obregon
David Münch
David Pujol Perich
Davide Marelli
De Zhang
Debalina Barik
Debapriya Roy (Kundu)
Debashis Das
Debashis Das Chakladar
Debi Prosad Dogra
Debraj D. Basu
Decheng Liu
Deen Dayal Mohan
Deep A. Patel
Deepak Kumar
Dengpan Liu
Denis Coquenet
Désiré Sidibé
Devesh Walawalkar
Dewan Md. Farid
Di Ming
Di Qiu
Di Yuan
Dian Jia
Dianmo Sheng
Diego Thomas
Diganta Saha
Dimitri Bulatov
Dimpy Varshni
Dingcheng Yang
Dipanjan Das
Dipanjyoti Paul
Divya Biligere Shivanna
Divya Saxena
Divya Sharma
Dmitrii Matveichev
Dmitry Minskiy
Dmitry V. Sorokin
Dong Zhang
Donghua Wang
Donglin Zhang
Dongming Wu
Dongqiangzi Ye
Dongqing Zou
Dongrui Liu
Dongyang Zhang
Dongzhan Zhou
Douglas Rodrigues
Duarte Folgado
Duc Minh Vo
Duoxuan Pei
Durai Arun Pannir Selvam
Durga Bhavani S.
Eckart Michaelsen
Elena Goyanes
Élodie Puybareau

Emanuele Vivoli
Emna Ghorbel
Enrique Naredo
Enyu Cai
Eric Patterson
Ernest Valveny
Eva Blanco-Mallo
Eva Breznik
Evangelos Sartinas
Fabio Solari
Fabiola De Marco
Fan Wang
Fangda Li
Fangyuan Lei
Fangzhou Lin
Fangzhou Luo
Fares Bougourzi
Farman Ali
Fatiha Mokdad
Fei Shen
Fei Teng
Fei Zhu
Feiyan Hu
Felipe Gomes Oliveira
Feng Li
Fengbei Liu
Fenghua Zhu
Fillipe D. M. De Souza
Flavio Piccoli
Flavio Prieto
Florian Kleber
Francesc Serratosa
Francesco Bianconi
Francesco Castro
Francesco Ponzio
Francisco Javier Hernández López
Frédéric Rayar
Furkan Osman Kar
Fushuo Huo
Fuxiao Liu
Fu-Zhao Ou
Gabriel Turinici
Gabrielle Flood
Gajjala Viswanatha Reddy
Gaku Nakano

Galal Binamakhashen
Ganesh Krishnasamy
Gang Pan
Gangyan Zeng
Gani Rahmon
Gaurav Harit
Gennaro Vessio
Genoveffa Tortora
George Azzopardi
Gerard Ortega
Gerardo E. Altamirano-Gomez
Gernot A. Fink
Gibran Benitez-Garcia
Gil Ben-Artzi
Gilbert Lim
Giorgia Minello
Giorgio Fumera
Giovanna Castellano
Giovanni Puglisi
Giulia Orrù
Giuliana Ramella
Gökçe Uludoğan
Gopi Ramena
Gorthi Rama Krishna Sai Subrahmanyam
Gourav Datta
Gowri Srinivasa
Gozde Sahin
Gregory Randall
Guanjie Huang
Guanjun Li
Guanwen Zhang
Guanyu Xu
Guanyu Yang
Guanzhou Ke
Guhnoo Yun
Guido Borghi
Guilherme Brandão Martins
Guillaume Caron
Guillaume Tochon
Guocai Du
Guohao Li
Guoqiang Zhong
Guorong Li
Guotao Li
Gurman Gill

Haechang Lee
Haichao Zhang
Haidong Xie
Haifeng Zhao
Haimei Zhao
Hainan Cui
Haixia Wang
Haiyan Guo
Hakime Ozturk
Hamid Kazemi
Han Gao
Hang Zou
Hanjia Lyu
Hanjoo Cho
Hanqing Zhao
Hanyuan Liu
Hanzhou Wu
Hao Li
Hao Meng
Hao Sun
Hao Wang
Hao Xing
Hao Zhao
Haoan Feng
Haodi Feng
Haofeng Li
Haoji Hu
Haojie Hao
Haojun Ai
Haopeng Zhang
Haoran Li
Haoran Wang
Haorui Ji
Haoxiang Ma
Haoyu Chen
Haoyue Shi
Harald Koestler
Harbinder Singh
Harris V. Georgiou
Hasan F. Ates
Hasan S. M. Al-Khaffaf
Hatef Otroshi Shahreza
Hebeizi Li
Heng Zhang
Hengli Wang
Hengyue Liu
Hertog Nugroho
Hieyong Jeong
Himadri Mukherjee
Hoai Ngo
Hoda Mohaghegh
Hong Liu
Hong Man
Hongcheng Wang
Hongjian Zhan
Hongxi Wei
Hongyu Hu
Hoseong Kim
Hossein Ebrahimnezhad
Hossein Malekmohamadi
Hrishav Bakul Barua
Hsueh-Yi Sean Lin
Hua Wei
Huafeng Li
Huali Xu
Huaming Chen
Huan Wang
Huang Chen
Huanran Chen
Hua-Wen Chang
Huawen Liu
Huayi Zhan
Hugo Jair Escalante
Hui Chen
Hui Li
Huichen Yang
Huiqiang Jiang
Huiyuan Yang
Huizi Yu
Hung T. Nguyen
Hyeongyu Kim
Hyeonjeong Park
Hyeonjun Lee
Hymalai Bello
Hyung-Gun Chi
Hyunsoo Kim
I-Chen Lin
Ik Hyun Lee
Ilan Shimshoni
Imad Eddine Toubal

Imran Sarker
Inderjot Singh Saggu
Indrani Mukherjee
Indranil Sur
Ines Rieger
Ioannis Pierros
Irina Rabaev
Ivan V. Medri
J. Rafid Siddiqui
Jacek Komorowski
Jacopo Bonato
Jacson Rodrigues Correia-Silva
Jaekoo Lee
Jaime Cardoso
Jakob Gawlikowski
Jakub Nalepa
James L. Wayman
Jan Čech
Jangho Lee
Jani Boutellier
Javier Gurrola-Ramos
Javier Lorenzo-Navarro
Jayasree Saha
Jean Lee
Jean Paul Barddal
Jean-Bernard Hayet
Jean-Philippe G. Tarel
Jean-Yves Ramel
Jenny Benois-Pineau
Jens Bayer
Jerin Geo James
Jesús Miguel García-Gorrostieta
Jia Qu
Jiahong Chen
Jiaji Wang
Jian Hou
Jian Liang
Jian Xu
Jian Zhu
Jianfeng Lu
Jianfeng Ren
Jiangfan Liu
Jianguo Wang
Jiangyan Yi
Jiangyong Duan
Jianhua Yang
Jianhua Zhang
Jianhui Chen
Jianjia Wang
Jianli Xiao
Jianqiang Xiao
Jianwu Wang
Jianxin Zhang
Jianxiong Gao
Jianxiong Zhou
Jianyu Wang
Jianzhong Wang
Jiaru Zhang
Jiashu Liao
Jiaxin Chen
Jiaxin Lu
Jiaxing Ye
Jiaxuan Chen
Jiaxuan Li
Jiayi He
Jiayin Lin
Jie Ou
Jiehua Zhang
Jiejie Zhao
Jignesh S. Bhatt
Jin Gao
Jin Hou
Jin Hu
Jin Shang
Jing Tian
Jing Yu Chen
Jingfeng Yao
Jinglun Feng
Jingtong Yue
Jingwei Guo
Jingwen Xu
Jingyuan Xia
Jingzhe Ma
Jinhong Wang
Jinjia Wang
Jinlai Zhang
Jinlong Fan
Jinming Su
Jinrong He
Jintao Huang

Jinwoo Ahn
Jinwoo Choi
Jinyang Liu
Jinyu Tian
Jionghao Lin
Jiuding Duan
Jiwei Shen
Jiyan Pan
Jiyoun Kim
João Papa
Johan Debayle
John Atanbori
John Wilson
John Zhang
Jónathan Heras
Joohi Chauhan
Jorge Calvo-Zaragoza
Jorge Figueroa
Jorma Laaksonen
José Joaquim De Moura Ramos
Jose Vicent
Joseph Damilola Akinyemi
Josiane Zerubia
Juan Wen
Judit Szücs
Juepeng Zheng
Juha Roning
Jumana H. Alsubhi
Jun Cheng
Jun Ni
Jun Wan
Junghyun Cho
Junjie Liang
Junjie Ye
Junlin Hu
Juntong Ni
Junxin Lu
Junxuan Li
Junyaup Kim
Junyeong Kim
Jürgen Seiler
Jushang Qiu
Juyang Weng
Jyostna Devi Bodapati
Jyoti Singh Kirar

Kai Jiang
Kaiqiang Song
Kalidas Yeturu
Kalle Åström
Kamalakar Vijay Thakare
Kang Gu
Kang Ma
Kanji Tanaka
Karthik Seemakurthy
Kaushik Roy
Kavisha Jayathunge
Kazuki Uehara
Ke Shi
Keigo Kimura
Keiji Yanai
Kelton A. P. Costa
Kenneth Camilleri
Kenny Davila
Ketan Atul Bapat
Ketan Kotwal
Kevin Desai
Keyu Long
Khadiga Mohamed Ali
Khakon Das
Khan Muhammad
Kilho Son
Kim-Ngan Nguyen
Kishan Kc
Kishor P. Upla
Klaas Dijkstra
Komal Bharti
Konstantinos Triaridis
Kostas Ioannidis
Koyel Ghosh
Kripabandhu Ghosh
Krishnendu Ghosh
Kshitij S. Jadhav
Kuan Yan
Kun Ding
Kun Xia
Kun Zeng
Kunal Banerjee
Kunal Biswas
Kunchi Li
Kurban Ubul

Lahiru N. Wijayasingha
Laines Schmalwasser
Lakshman Mahto
Lala Shakti Swarup Ray
Lale Akarun
Lan Yan
Lawrence Amadi
Lee Kang Il
Lei Fan
Lei Shi
Lei Wang
Leonardo Rossi
Lequan Lin
Levente Tamas
Li Bing
Li Li
Li Ma
Li Song
Lia Morra
Liang Xie
Liang Zhao
Lianwen Jin
Libing Zeng
Lidia Sánchez-González
Lidong Zeng
Lijun Li
Likang Wang
Lili Zhao
Lin Chen
Lin Huang
Linfei Wang
Ling Lo
Lingchen Meng
Lingheng Meng
Lingxiao Li
Lingzhong Fan
Liqi Yan
Liqiang Jing
Lisa Gutzeit
Liu Ziyi
Liushuai Shi
Liviu-Daniel Stefan
Liyuan Ma
Liyun Zhu
Lizuo Jin
Longteng Guo
Lorena Álvarez Rodríguez
Lorenzo Putzu
Lu Leng
Lu Pang
Lu Wang
Luan Pham
Luc Brun
Luca Guarnera
Luca Piano
Lucas Alexandre Ramos
Lucas Goncalves
Lucas M. Gago
Luigi Celona
Luis C. S. Afonso
Luis Gerardo De La Fraga
Luis S. Luevano
Luis Teixeira
Lunke Fei
M. Hassaballah
Maddimsetti Srinivas
Mahendran N.
Mahesh Mohan M. R.
Maiko Lie
Mainak Singha
Makoto Hirose
Malay Bhattacharyya
Mamadou Dian Bah
Man Yao
Manali J. Patel
Manav Prabhakar
Manikandan V. M.
Manish Bhatt
Manjunath Shantharamu
Manuel Curado
Manuel Günther
Manuel Marques
Marc A. Kastner
Marc Chaumont
Marc Cheong
Marc Lalonde
Marco Cotogni
Marcos C. Santana
Mario Molinara
Mariofanna Milanova

Markus Bauer
Marlon Becker
Mårten Wadenbäck
Martin G. Ljungqvist
Martin Kampel
Martina Pastorino
Marwan Torki
Masashi Nishiyama
Masayuki Tanaka
Massimo O. Spata
Matteo Ferrara
Matthew D. Dawkins
Matthew Gadd
Matthew S. Watson
Maura Pintor
Max Ehrlich
Maxim Popov
Mayukh Das
Md Baharul Islam
Md Sajid
Meghna Kapoor
Meghna P. Ayyar
Mei Wang
Meiqi Wu
Melissa L. Tijink
Meng Li
Meng Liu
Meng-Luen Wu
Mengnan Liu
Mengxi China Guo
Mengya Han
Michaël Clément
Michal Kawulok
Mickael Coustaty
Miguel Domingo
Milind G. Padalkar
Ming Liu
Ming Ma
Mingchen Feng
Mingde Yao
Minghao Li
Mingjie Sun
Ming-Kuang Daniel Wu
Mingle Xu
Mingyong Li
Mingyuan Jiu
Minh P. Nguyen
Minh Q. Tran
Minheng Ni
Minsu Kim
Minyi Zhao
Mirko Paolo Barbato
Mo Zhou
Modesto Castrillón-Santana
Mohamed Amine Mezghich
Mohamed Dahmane
Mohamed Elsharkawy
Mohamed Yousuf
Mohammad Hashemi
Mohammad Khalooei
Mohammad Khateri
Mohammad Mahdi Dehshibi
Mohammad Sadil Khan
Mohammed Mahmoud
Moises Diaz
Monalisha Mahapatra
Monidipa Das
Mostafa Kamali Tabrizi
Mridul Ghosh
Mrinal Kanti Bhowmik
Muchao Ye
Mugalodi Ramesha Rakesh
Muhammad Rameez Ur Rahman
Muhammad Suhaib Kanroo
Muming Zhao
Munender Varshney
Munsif Ali
Na Lv
Nader Karimi
Nagabhushan Somraj
Nakkwan Choi
Nakul Agarwal
Nan Pu
Nan Zhou
Nancy Mehta
Nand Kumar Yadav
Nandakishor Nandakishor
Nandyala Hemachandra
Nanfeng Jiang
Narayan Hegde

Narayan Ji Mishra
Narayan Vetrekar
Narendra D. Londhe
Nathalie Girard
Nati Ofir
Naval Kishore Mehta
Nazmul Shahadat
Neeti Narayan
Neha Bhargava
Nemanja Djuric
Newlin Shebiah R.
Ngo Ba Hung
Nhat-Tan Bui
Niaz Ahmad
Nick Theisen
Nicolas Passat
Nicolas Ragot
Nicolas Sidere
Nikolaos Mitianoudis
Nikolas Ebert
Nilah Ravi Nair
Nilesh A. Ahuja
Nilkanta Sahu
Nils Murrugarra-Llerena
Nina S. T. Hirata
Ninad Aithal
Ning Xu
Ningzhi Wang
Niraj Kumar
Nirmal S. Punjabi
Nisha Varghese
Norio Tagawa
Obaidullah Md Sk
Oguzhan Ulucan
Olfa Mechi
Oliver Tüselmann
Orazio Pontorno
Oriol Ramos Terrades
Osman Akin
Ouadi Beya
Ozge Mercanoglu Sincan
Pabitra Mitra
Padmanabha Reddy Y. C. A.
Palaash Agrawal
Palaiahnakote Shivakumara
Palash Ghosal
Pallav Dutta
Paolo Rota
Paramanand Chandramouli
Paria Mehrani
Parth Agrawal
Partha Basuchowdhuri
Patrick Horain
Pavan Kumar
Pavan Kumar Anasosalu Vasu
Pedro Castro
Peipei Li
Peipei Yang
Peisong Shen
Peiyu Li
Peng Li
Pengfei He
Pengrui Quan
Pengxin Zeng
Pengyu Yan
Peter Eisert
Petra Gomez-Krämer
Pierrick Bruneau
Ping Cao
Pingping Zhang
Pintu Kumar
Pooja Kumari
Pooja Sahani
Prabhu Prasad Dev
Pradeep Kumar
Pradeep Singh
Pranjal Sahu
Prasun Roy
Prateek Keserwani
Prateek Mittal
Praveen Kumar Chandaliya
Praveen Tirupattur
Pravin Nair
Preeti Gopal
Preety Singh
Prem Shanker Yadav
Prerana Mukherjee
Prerna A. Mishra
Prianka Dey
Priyanka Mudgal

Qc Kha Ng
Qi Li
Qi Ming
Qi Wang
Qi Zuo
Qian Li
Qiang Gan
Qiang He
Qiang Wu
Qiangqiang Zhou
Qianli Zhao
Qiansen Hong
Qiao Wang
Qidong Huang
Qihua Dong
Qin Yuke
Qing Guo
Qingbei Guo
Qingchao Zhang
Qingjie Liu
Qinhong Yang
Qiushi Shi
Qixiang Chen
Quan Gan
Quanlong Guan
Rachit Chhaya
Radu Tudor Ionescu
Rafal Zdunek
Raghavendra Ramachandra
Rahimul I. Mazumdar
Rahul Kumar Ray
Rajib Dutta
Rajib Ghosh
Rakesh Kumar
Rakesh Paul
Rama Chellappa
Rami O. Skaik
Ramon Aranda
Ran Wei
Ranga Raju Vatsavai
Ranganath Krishnan
Rasha Friji
Rashmi S.
Razaib Tariq
Rémi Giraud
René Schuster
Renlong Hang
Renrong Shao
Renu Sharma
Reza Sadeghian
Richard Zanibbi
Rimon Elias
Rishabh Shukla
Rita Delussu
Riya Verma
Robert J. Ravier
Robert Sablatnig
Robin Strand
Rocco Pietrini
Rocio Diaz Martin
Rocio Gonzalez-Diaz
Rohit Venkata Sai Dulam
Romain Giot
Romi Banerjee
Ru Wang
Ruben Machucho
Ruddy Théodose
Ruggero Pintus
Rui Deng
Rui P. Paiva
Rui Zhao
Ruifan Li
Ruigang Fu
Ruikun Li
Ruirui Li
Ruixiang Jiang
Ruowei Jiang
Rushi Lan
Rustam Zhumagambetov
S. Amutha
S. Divakar Bhat
Sagar Goyal
Sahar Siddiqui
Sahbi Bahroun
Sai Karthikeya Vemuri
Saibal Dutta
Saihui Hou
Sajad Ahmad Rather
Saksham Aggarwal
Sakthi U.

Salimeh Sekeh
Samar Bouazizi
Samia Boukir
Samir F. Harb
Samit Biswas
Samrat Mukhopadhyay
Samriddha Sanyal
Sandika Biswas
Sandip Purnapatra
Sanghyun Jo
Sangwoo Cho
Sanjay Kumar
Sankaran Iyer
Sanket Biswas
Santanu Roy
Santosh D. Pandure
Santosh Ku Behera
Santosh Nanabhau Palaskar
Santosh Prakash Chouhan
Sarah S. Alotaibi
Sasanka Katreddi
Sathyanarayanan N. Aakur
Saurabh Yadav
Sayan Rakshit
Scott McCloskey
Sebastian Bunda
Sejuti Rahman
Selim Aksoy
Sen Wang
Seraj A. Mostafa
Shanmuganathan Raman
Shao-Yuan Lo
Shaoyuan Xu
Sharia Arfin Tanim
Shehreen Azad
Sheng Wan
Shengdong Zhang
Shengwei Qin
Shenyuan Gao
Sherry X. Chen
Shibaprasad Sen
Shigeaki Namiki
Shiguang Liu
Shijie Ma
Shikun Li
Shinichiro Omachi
Shirley David
Shishir Shah
Shiv Ram Dubey
Shiva Baghel
Shivanand S. Gornale
Shogo Sato
Shotaro Miwa
Shreya Ghosh
Shreya Goyal
Shuai Su
Shuai Wang
Shuai Zheng
Shuaifeng Zhi
Shuang Qiu
Shuhei Tarashima
Shujing Lyu
Shuliang Wang
Shun Zhang
Shunming Li
Shunxin Wang
Shuping Zhao
Shuquan Ye
Shuwei Huo
Shuyue Lan
Shyi-Chyi Cheng
Si Chen
Siddarth Ravichandran
Sihan Chen
Siladittya Manna
Silambarasan Elkana Ebinazer
Simon Benaïchouche
Simon S. Woo
Simone Caldarella
Simone Milani
Simone Zini
Sina Lotfian
Sitao Luan
Sivaselvan B.
Siwei Li
Siwei Wang
Siwen Luo
Siyu Chen
Sk Aziz Ali
Sk Md Obaidullah

Sneha Shukla
Snehasis Banerjee
Snehasis Mukherjee
Snigdha Sen
Sofia Casarin
Soheila Farokhi
Soma Bandyopadhyay
Son Minh Nguyen
Son Xuan Ha
Sonal Kumar
Sonam Gupta
Sonam Nahar
Song Ouyang
Sotiris Kotsiantis
Souhaila Djaffal
Soumen Biswas
Soumen Sinha
Soumitri Chattopadhyay
Souvik Sengupta
Spiros Kostopoulos
Sreeraj Ramachandran
Sreya Banerjee
Srikanta Pal
Srinivas Arukonda
Stephane A. Guinard
Su O. Ruan
Subhadip Basu
Subhajit Paul
Subhankar Ghosh
Subhankar Mishra
Subhankar Roy
Subhash Chandra Pal
Subhayu Ghosh
Sudip Das
Sudipta Banerjee
Suhas Pillai
Sujit Das
Sukalpa Chanda
Sukhendu Das
Suklav Ghosh
Suman K. Ghosh
Suman Samui
Sumit Mishra
Sungho Suh
Sunny Gupta

Suraj Kumar Pandey
Surendrabikram Thapa
Suresh Sundaram
Sushil Bhattacharjee
Susmita Ghosh
Swakkhar Shatabda
Syed Ms Islam
Syed Tousiful Haque
Taegyeong Lee
Taihui Li
Takashi Shibata
Takeshi Oishi
Talha Ahmad Siddiqui
Tanguy Gernot
Tangwen Qian
Tanima Bhowmik
Tanpia Tasnim
Tao Dai
Tao Hu
Tao Sun
Taoran Yi
Tapan Shah
Taveena Lotey
Teng Huang
Tengqi Ye
Teresa Alarcon
Tetsuji Ogawa
Thanh Phuong Nguyen
Thanh Tuan Nguyen
Thattapon Surasak
Thibault Napolãon
Thierry Bouwmans
Thinh Truong Huynh Nguyen
Thomas De Min
Thomas E. K. Zielke
Thomas Swearingen
Tianatahina Jimmy Francky Randrianasoa
Tianheng Cheng
Tianjiao He
Tianyi Wei
Tianyuan Zhang
Tianyue Zheng
Tiecheng Song
Tilottama Goswami
Tim Büchner

Tim H. Langer
Tim Raven
Tingkai Liu
Tingting Yao
Tobias Meisen
Toby P. Breckon
Tong Chen
Tonghua Su
Tran Tuan Anh
Tri-Cong Pham
Trishna Saikia
Trung Quang Truong
Tuan T. Nguyen
Tuan Vo Van
Tushar Shinde
Ujjwal Karn
Ukrit Watchareeruetai
Uma Mudenagudi
Umarani Jayaraman
V. S. Malemath
Vallidevi Krishnamurthy
Ved Prakash
Venkata Krishna Kishore Kolli
Venkata R. Vavilthota
Venkatesh Thirugnana Sambandham
Verónica Maria Vasconcelos
Véronique Ve Eglin
Víctor E. Alonso-Pérez
Vinay Palakkode
Vinayak S. Nageli
Vincent J. Whannou De Dravo
Vincenzo Conti
Vincenzo Gattulli
Vineet Padmanabhan
Vishakha Pareek
Viswanath Gopalakrishnan
Vivek Singh Baghel
Vivekraj K.
Vladimir V. Arlazarov
Vu-Hoang Tran
W. Sylvia Lilly Jebarani
Wachirawit Ponghiran
Wafa Khlif
Wang An-Zhi
Wanli Xue
Wataru Ohyama
Wee Kheng Leow
Wei Chen
Wei Cheng
Wei Hua
Wei Lu
Wei Pan
Wei Tian
Wei Wang
Wei Wei
Wei Zhou
Weidi Liu
Weidong Yang
Weijun Tan
Weimin Lyu
Weinan Guan
Weining Wang
Weiqiang Wang
Weiwei Guo
Weixia Zhang
Wei-Xuan Bao
Weizhong Jiang
Wen Xie
Wenbin Qian
Wenbin Tian
Wenbin Wang
Wenbo Zheng
Wenhan Luo
Wenhao Wang
Wen-Hung Liao
Wenjie Li
Wenkui Yang
Wenwen Si
Wenwen Yu
Wenwen Zhang
Wenwu Yang
Wenxi Li
Wenxi Yue
Wenxue Cui
Wenzhuo Liu
Widhiyo Sudiyono
Willem Dijkstra
Wolfgang Fuhl
Xi Zhang
Xia Yuan

Xianda Zhang
Xiang Zhang
Xiangdong Su
Xiang-Ru Yu
Xiangtai Li
Xiangyu Xu
Xiao Guo
Xiao Hu
Xiao Wu
Xiao Yang
Xiaofeng Zhang
Xiaogang Du
Xiaoguang Zhao
Xiaoheng Jiang
Xiaohong Zhang
Xiaohua Huang
Xiaohua Li
Xiao-Hui Li
Xiaolong Sun
Xiaosong Li
Xiaotian Li
Xiaoting Wu
Xiaotong Luo
Xiaoyan Li
Xiaoyang Kang
Xiaoyi Dong
Xin Guo
Xin Lin
Xin Ma
Xinchi Zhou
Xingguang Zhang
Xingjian Leng
Xingpeng Zhang
Xingzheng Lyu
Xinjian Huang
Xinqi Fan
Xinqi Liu
Xinqiao Zhang
Xinrui Cui
Xizhan Gao
Xu Cao
Xu Ouyang
Xu Zhao
Xuan Shen
Xuan Zhou

Xuchen Li
Xuejing Lei
Xuelu Feng
Xueting Liu
Xuewei Li
Xueyi X. Wang
Xugong Qin
Xu-Qian Fan
Xuxu Liu
Xu-Yao Zhang
Yan Huang
Yan Li
Yan Wang
Yan Xia
Yan Zhuang
Yanan Li
Yanan Zhang
Yang Hou
Yang Jiao
Yang Liping
Yang Liu
Yang Qian
Yang Yang
Yang Zhao
Yangbin Chen
Yangfan Zhou
Yanhui Guo
Yanjia Huang
Yanjun Zhu
Yanming Zhang
Yanqing Shen
Yaoming Cai
Yaoxin Zhuo
Yaoyan Zheng
Yaping Zhang
Yaqian Liang
Yarong Feng
Yasmina Benmabrouk
Yasufumi Sakai
Yasutomo Kawanishi
Yazeed Alzahrani
Ye Du
Ye Duan
Yechao Zhang
Yeong-Jun Cho

Yi Huo
Yi Shi
Yi Yu
Yi Zhang
Yibo Liu
Yibo Wang
Yi-Chieh Wu
Yifan Chen
Yifei Huang
Yihao Ding
Yijie Tang
Yikun Bai
Yimin Wen
Yinan Yang
Yin-Dong Zheng
Yinfeng Yu
Ying Dai
Yingbo Li
Yiqiao Li
Yiqing Huang
Yisheng Lv
Yisong Xiao
Yite Wang
Yizhe Li
Yong Wang
Yonghao Dong
Yong-Hyuk Moon
Yongjie Li
Yongqian Li
Yongqiang Mao
Yongxu Liu
Yongyu Wang
Yongzhi Li
Youngha Hwang
Yousri Kessentini
Yu Wang
Yu Zhou
Yuan Tian
Yuan Zhang
Yuanbo Wen
Yuanxin Wang
Yubin Hu
Yubo Huang
Yuchen Ren
Yucheng Xing
Yuchong Yao
Yuecong Min
Yuewei Yang
Yufei Zhang
Yufeng Yin
Yugen Yi
Yuhang Ming
Yujia Zhang
Yujun Ma
Yukiko Kenmochi
Yun Hoyeoung
Yun Liu
Yunhe Feng
Yunxiao Shi
Yuru Wang
Yushun Tang
Yusuf Osmanlioglu
Yusuke Fujita
Yuta Nakashima
Yuwei Yang
Yuwu Lu
Yuxi Liu
Yuya Obinata
Yuyao Yan
Yuzhi Guo
Zaipeng Xie
Zander W. Blasingame
Zedong Wang
Zeliang Zhang
Zexin Ji
Zhanxiang Feng
Zhaofei Yu
Zhe Chen
Zhe Cui
Zhe Liu
Zhe Wang
Zhekun Luo
Zhen Yang
Zhenbo Li
Zhenchun Lei
Zhenfei Zhang
Zheng Liu
Zheng Wang
Zhengming Yu
Zhengyin Du

Zhengyun Cheng
Zhenshen Qu
Zhenwei Shi
Zhenzhong Kuang
Zhi Cai
Zhi Chen
Zhibo Chu
Zhicun Yin
Zhida Huang
Zhida Zhang
Zhifan Gao
Zhihang Ren
Zhihang Yuan
Zhihao Wang
Zhihua Xie
Zhihui Wang
Zhikang Zhang
Zhiming Zou
Zhiqi Shao
Zhiwei Dong
Zhiwei Qi
Zhixiang Wang
Zhixuan Li
Zhiyu Jiang
Zhiyuan Yan
Zhiyuan Yu
Zhiyuan Zhang
Zhong Chen
Zhongwei Teng
Zhongzhan Huang
Zhongzhi Yu
Zhuan Han
Zhuangzhuang Chen
Zhuo Liu
Zhuo Su
Zhuojun Zou
Zhuoyue Wang
Ziang Song
Zicheng Zhang
Zied Mnasri
Zifan Chen
Žiga Babnik
Zijing Chen
Zikai Zhang
Ziling Huang
Zilong Du
Ziqi Cai
Ziqi Zhou
Zi-Rui Wang
Zirui Zhou
Ziwen He
Ziyao Zeng
Ziyi Zhang
Ziyue Xiang
Zonglei Jing
Zongyi Xu

Contents – Part XXII

SMFuse: Two-Stage Structural Map Aware Network for Multi-focus Image Fusion

Tianyu Shen[1], Hui Li[1](✉), Chunyang Cheng[1], Zhongwei Shen[2], and Xiaoning Song[1]

[1] International Joint Laboratory on Artificial Intelligence of Jiangsu Province, School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China
lihui.cv@jiangnan.edu.cn

[2] School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou, China

Abstract. Multi-focus image fusion (MFIF) explores the positioning and reorganization of the focused parts from the input images. Focused and defocused parts have similar representations in color, contour and other appearance information, which degrades the fusion quality due to the influence of these redundant information. Currently, most MFIF methods have not identified an effective way to remove redundant information before fusion stage. Thus, in this paper, we introduce a structural map extraction strategy for multi-focus image fusion. Compared to the source image, structural map reduces redundant information, and the clearer parts of the image retain more abundant structural features. Consequently, the differences between focused part and defocused part become more pronounced based on the extracted structural map. Specifically, the proposed fusion method adopts a two-stage training strategy. Firstly, the structural map is extracted by the proposed structural map extraction network (SMENet) from the source images. Secondly, the structural map is thus applied to train the decision map generation network (DMGNet) to obtain the decision map which is utilized to generate the final fusion image. Qualitative and quantitative experiments on three public datasets demonstrate the superiority of the proposed method, compared with the advanced image fusion algorithms.

Keywords: Image fusion · Deep learning · Two-stage strategy · Structural map · Decision map

1 Introduction

Due to the limitation of image sensors, it is difficult to capture fully focused images [1] with a single camera, which leads to the emergence of the MFIF task. The fusion image generated by the MFIF task is solely generated from the content within the source images, distinguishing MFIF from other fusion tasks. As an example, the fusion of infrared and visible images [2] necessitates the extraction of distinct features from the source images and the recombination of

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 1–17, 2025.
https://doi.org/10.1007/978-3-031-78312-8_1

the fused image with varying weights. Nevertheless, the fundamental objective of MFIF tasks is to discriminate between different focal regions and subsequently reintegrate them. The key of this task lies in accurately identifying the focused and defocused areas.

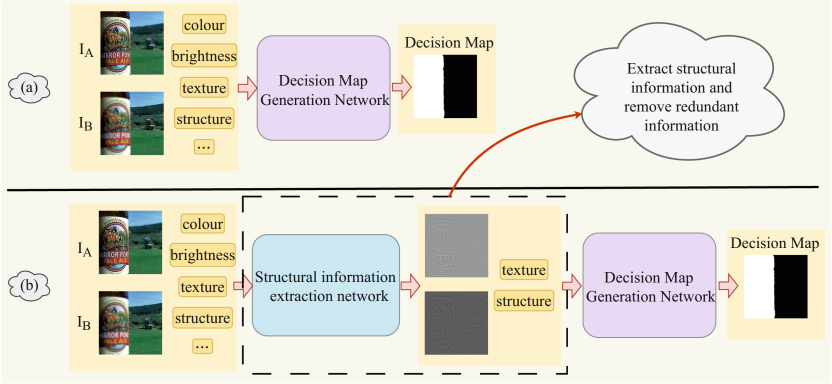

(a)
I_A
I_B
colour
brightness
texture
structure
...
Decision Map Generation Network
Decision Map
Extract structural information and remove redundant information
(b)
I_A
I_B
colour
brightness
texture
structure
...
Structural information extraction network
texture
structure
Decision Map Generation Network
Decision Map

Fig. 1. (a) Generating decision map solely based on source image information. (b) The first stage involves extracting structural map from the source images to remove redundant information. In the second stage, structural maps are utilized to generate decision map guiding the fusion process.

In the early years, MFIF methods concentrated on spatial domain and transform domain operations. The transformation domain-based MFIF methods include common non-subsampled contour transform (NSCT) [3], non-subsampled shear transform (NSST) [4], and sparse representation (SR) based methods [5]. These methods accomplish comprehensive fusion tasks in an alternate fusion space through transformation and inverse transformation and also successfully mitigate issues related to discontinuity or blocking effects. However, they did not adequately address spatial consistency and the fused image obtained through inverse transformation may also introduce unwanted noise information.

The spatial domain-based MFIF methods [6] directly integrate the information of multi-focus images into the fused image to address the above drawback. These methods complete the fusion task at the pixel level, block level, and region level by weighted averaging the source image. The spatial domain algorithms inherently consider spatial consistency and relies on solving an energy function to assign weights.

However, such methods are not suitable for solving fusion problems in complex scenes due to their time-consuming nature. Therefore, with the development of deep learning, scholars have proposed deep learning-based MFIF methods [7–9]. Deep learning methods for MFIF mainly fall into two categories: regression-based and classification-based. The former may lose some information from the source images, while the latter preserves more information from the source images at the pixel level. As depicted in Fig. 1(a), most scholars conduct focus attribute classification training directly based on source images. However, besides the focus attribute, source images contain other redundant information such as color and brightness, which are not conducive to decision map generation.

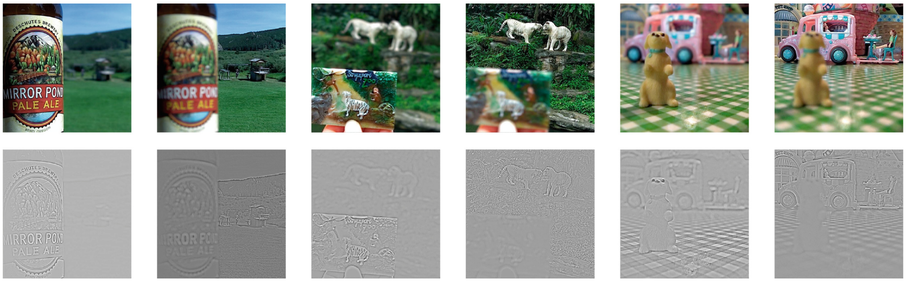


Fig. 2. Comparison between source image and structural map. The first line is the source image, and the second line is the corresponding structural map.

To reduce redundant information in the source images, enhance the contrast between focused and defocused images, and obtain better decision map, this paper proposes a two-stage fusion network that is constructed based on structural map, as shown in Fig. 1(b). The extraction strategy for structural map comes from other visual tasks, such as super-resolution [10] and video compression [11]. Compared to the source images, the structural map generated by the SMENet eliminate redundant information and emphasize high-frequency details, such as structural and textural elements, within the focused regions. The increased structural information disparity between focused and defocused areas has been demonstrated to be beneficial for the subsequent process of generating enhanced decision map. After obtaining structural map, the DMGNet is employed, thereby overcoming the limitations associated with manual rules.

As depicted in Fig. 2, the structural details in focused areas are more pronounced than defocused part. As a result, there is a significant disparity of the structural information between the focused and the defocused regions. The greater the discernible distinctions in gradient and structural characteristics between the focused and defocused regions, the more favorable the conditions for generating an accurate decision map through deep learning.

The main contributions of this paper are given as follows:

- A two-stage fusion network is proposed to deliver more robust fusion results. The first stage generates the structural map of the input image, removing redundant information from the source images. The second stage utilizes the structural map to generate a more stable decision map.
- A new multi-scale feature extraction network has been introduced into image fusion task. Specifically, the multi-scale module extracts multiple features at different scales which is utilized to ultimately reconstruct structural map and decision map.
- Qualitative and quantitative experiments on multiple MFIF benchmarks have demonstrated the superiority of the proposed method.

2 Related Works

This section briefly introduces methods based on deep learning. These methods are divided into regression based methods and classification based methods.

2.1 Regression based MFIF Methods

Regression methods directly output fused images. The framework includes three parts: feature extraction, feature fusion, and image reconstruction. U2Fusion [12] uses a feature extractor with dense connections to obtain features from source images and trains the reconstruction module to generate fusion results. MUFusion [13] utilizes multiple dense extraction blocks and two scale reconstruction blocks to extract significant features. Then, it reconstructs these features to obtain fusion results. Some scholars use adversarial generative networks to directly generate fused images. FuseGAN [14] introduced conditional GAN to play adversarial game to generate fused image. MFF-GAN [15] conducted adversarial games under joint gradient constraints to generate fused images.

Deep learning methods based on regression (end-to-end image fusion) achieve fusion results without the handcrafted fusion rules. However, these methods also have some drawbacks: (1) Due to the unstable training process, this type of method does not have good generalization performance; (2) The reconstructed images cannot truly reflect the information of source images.

2.2 Classification based MFIF Methods

Classification based methods have similar frameworks with spatial domain methods. Specifically, networks are trained to classify focused and unfocused areas. In 2017, CNN [7] was first introduced into this field. In 2020, Ma et al. designed an unsupervised model named SESF [8] based on Densefuse. Unlike CNN, SESF uses a spatial frequency method to determine the focusing attribute of pixel values. In addition to spatial frequency, some scholars have also adopted simple manually designed rules to distinguish. UNIFusion [16] inputs the low-frequency and high-frequency images obtained from the filtering operation into the encoder, uses the l1 norm to obtain the gradient perception image, and finally uses the maximum strategy to obtain the decision map.

However, manually designed classification rules cannot accurately classify pixel value focusing attributes. The learning ability of the network is naturally suitable for pixel value-focused attribute classification tasks. GEU-Net [17] uses a U-shaped network to segment the focusing and defocusing regions.

Recently, some scholars have proposed obtaining fusion results with zero samples. ZMFF [9] achieved zero sample learning to obtain decision map through extracted prior information, but it is very time-consuming.

Compared with regression methods Classification based methods usually achieve better fusion performance. This is because the fusion images are completely derived from the source images and does not introduce new noise. An ideal decision map is the key for Classification based methods.

Most existing methods simply feed the source images to the neural network for training. This is limited in magnifying differences between focused and defocused areas. In order to highlight the difference between focused and defocused areas, this paper extracts structural map from the source images. The structural map of the focused area has more complete structural information, which forms a significant difference from the defocused area. The subjective picture is shown in Fig. 2. Based on this difference, the learning ability of the neural network is used to learn the ideal decision map and finally obtain good fusion results. Specifically, this paper considers using two-stage training to solve this problem. The first stage of training resulted in an ideal structural map. The second stage uses the structural map obtained in the first stage as the training dataset to obtain the decision map.

3 Proposed Method

Firstly, the whole network architecture is described. Then, details of the training strategy are provided. Finally, the post-processing module is introduced. Since the decision map output in the second stage can be seen as a segmentation task, it is natural to consider using the U-Net structure, which has achieved good performance in semantic segmentation. This structure also exhibits advantages in extracting multi scale features, thus adopting the similar architecture for the SMENet and DMGNet.

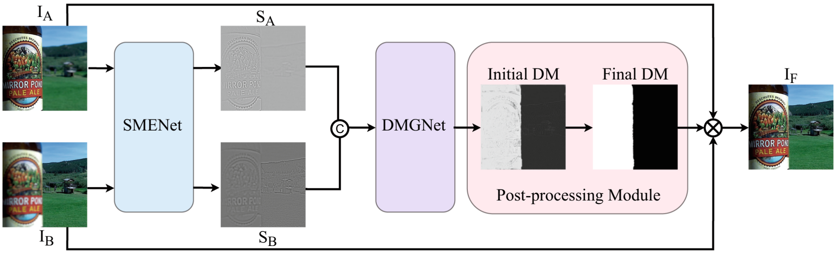


Fig. 3. The framework of the proposed fusion method.I_A is a close-up focused image, and I_B is a long-range focused image.S_A and S_B are two structural maps with different focus areas.I_F is the final fused image.

3.1 The Network Architecture

The focus of an image is reflected in the high-frequency structural information such as edges and details. When the amount of structural information collected is little, this part of the image is defocused. Therefore, we locate the focused region of the image based on structural information. The overall framework of our proposed method is shown in Fig. 3. Three modules are designed in the framework: the Structural Map Extraction Network (SMENet), the Decision Map Generation Network (DMGNet) and the post-processing module.

SMENet Network Structure SMENet is used to extract structural information from the input images.

In order to extract the structural information and high-frequency information in the source images, U-shaped network architecture is used to extract and integrate the structural information of multiple scales in the source images.

Fig. 4 (A) is the detailed structure of SMENet. H and W represent the height and width of inputs. The two numbers after H and W are the number of input and output channels. The kernel size of all the convolution layers is set to 3×3. In the convolutional layers, the gradual decrease of H and W indicates the down-sampling operation (5 down-sampling layers), while the gradual increase resolution indicates the up-sampling.

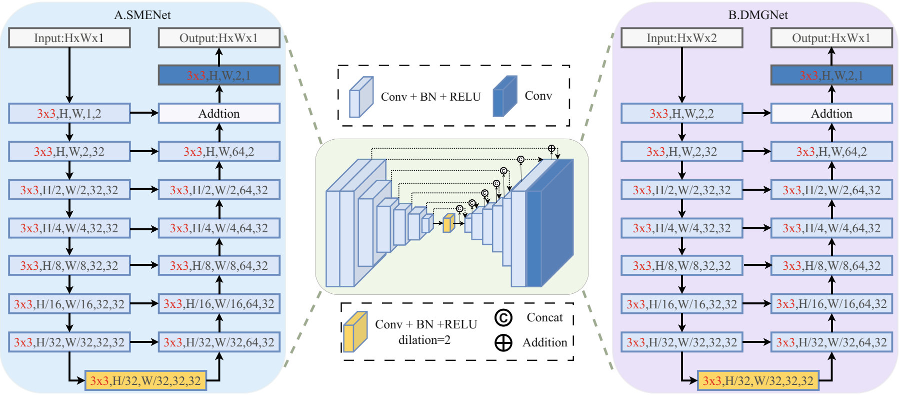


Fig. 4. Proposed network structure

The pooling operation and convolution operation at different scales are aimed at extracting multi-scale structural features from input feature map with arbitrary spatial resolution, extracting multi-scale features from gradually down-sampling feature maps, and encoding them into high-resolution feature map through gradually up-sampling, concatenation, and convolution. This way reduces the detail loss caused by large-scale up-sampling operations.

DMGNet Network Structure Based on the structural map extracted by SMENet, DMGNet extracts decision map through pixel-level conversion for positioning the focused part.

DMGNet exhibits a U-shaped network akin to that of SMENet. The distinction between SMENet and DMGNet resides solely in the architecture of the initial convolutional layer. While SMENet takes an image as input and generates the corresponding structural map, DMGNet, in contrast, requires concatenating two structural maps with different focal settings to the network and producing a decision map. As a result, the number of channels for these two models is 1 and 2, respectively. Fig. 4 (B) is the detailed structure of DMGNet.

3.2 The Training Phase of SMENet

Firstly, the clear natural image is blurred using Gaussian blur to obtain the blurred images. The blurred image is used as the target image. Following the residual concept, we compute the structural similarity between the clear source image subtracted from the network output and the blurred image.

When the loss value converges, the output of SMENet encapsulates a greater amount of structural information. Ultimately, SMENet excels in extracting structural detail features from images.

The training process of SMENet is shown in Fig. 5. The input images are converted into grayscale. y is a clear (all-focused) image. Firstly, a blurred image x is obtained by applying the Gaussian filter into y. Based on experience [17], the Gaussian blur parameter with a standard deviation of 2 and a size of 7×7 is set. Then, y is fed to SMENet to obtain the structural map $F(y)$. Here, F represents the SMENet model.

The loss function of this network is calculated based on the structural similarity loss (SSIM) [18], which measures the error between the $y - F(y)$ and the blurred image x. The use of structural similarity loss is to enable the network to extract richer structural information. The structural similarity loss (L_{ssim}) is formulated as follows,

$$L_{ssim} = 1 - SSIM(y - F(y, \gamma), x) \tag{1}$$

where γ represents the trained weights in SMENet.

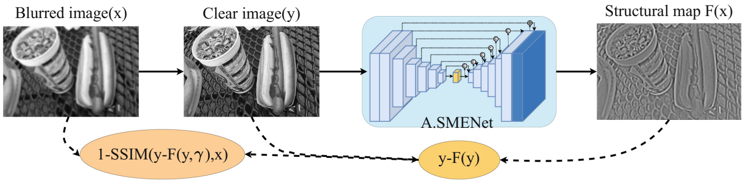


Fig. 5. An illustration of the SMENet training phase.

3.3 The Training Phase of DMGNet

DMGNet takes two cascaded structural maps output by SMENet as input, and ultimately generates an initial decision map. The training process of DMGNet is shown in Fig. 6.

Firstly, the paired structural maps $\{S_A, S_B\}$ obtained by the SMENet are concatenated as the inputs of DMGNet, we use F' to represent the DMGNet model. The (Ground Truth)GT of this network is a binary image gt. To acquire a precise and seamless initial decision map, we employ the mean square error (MSE) loss as the loss function.

The loss function of DMGNet training process is denoted as follows,

$$L_{MSE} = ||gt - F'(S_A, S_B, \alpha)||_2^2 \tag{2}$$

where α represents the trained weights, S_A and S_B represent the two structural maps obtained by the proposed SMENet.

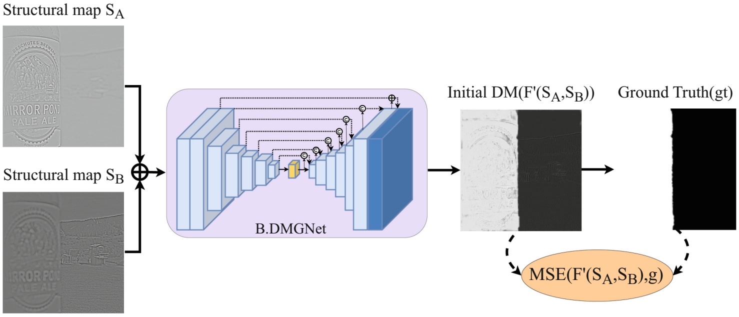


Fig. 6. The training process of the decision map generation network (DMGNet).

3.4 Post Processing Module and Fusion

In this section, we will introduce how to process the initial decision map to obtain the final decision map and obtain the fused image. The overall framework is shown in Fig. 7.

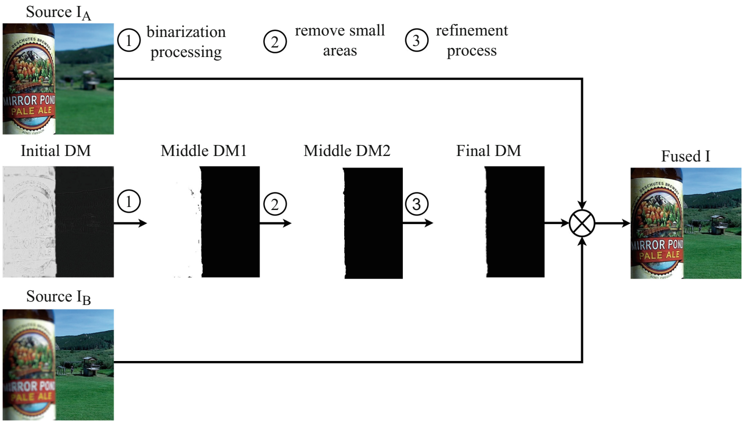


Fig. 7. An illustration of the post-processing module.

The first step is binarization processing. The binary image $D(x, y)$ is generated by setting the threshold value of the initial decision map generated by DMGNet [1]. This process is represented as

$$D(x,y) = \begin{cases} 1 \; S(x,y) > 0.5 \\ 0 \quad otherwise \end{cases} \tag{3}$$

where $S(x, y)$ is the initial decision map generated by the DMGNet.

[1] The threshold is set to 0.5 [1].

Next is the small domain removal operation. Although the binary map is almost complete, there are still a few 'holes'. To solve this problem, a morphological filtering technique is used to invert small holes with an area smaller than the manual set threshold [2].

Afterwards, refine the decision map. After using sliding windows[3] to detect the boundaries of different focus areas in the decision map, guided filtering technology is used to redistribute the weights of the boundary areas to reduce visual artifacts while maintaining the weights of other pixels.

Finally, we use the $D'(x,y)$ to represent the final decision map. I_A and I_B are the source image A and source image B, respectively. The fused image I_F is obtained using the weighted average operation, which is defined as:

$$R(x,y) = I_A(x,y) \cdot D'(x,y) + I_B(x,y) \cdot (1 - D'(x,y)) \tag{4}$$

4 Experiments

In this section, the experimental settings, fusion results analysis, and ablation experiments will be introduced.

4.1 Experimental Settings

In the first training stage, the MS-COCO [19] dataset is utilized to train the structural map extraction network. Specifically, the training set is composed of 40000 images blurred by the Gaussian filtering. The clear images (original image of MS-COCO) are regarded as the Ground Truth (GT). These images are converted to grey-scale and cropped to 288 × 288. The batch size and epoch are set to 16 and 50, respectively. Adam optimizer is used to train our network. Bilinear interpolation is used in the upsampling operation.

In the second training stage, we select a public image segmentation dataset benchmark [20] which contains high-quality natural images and corresponding masks for the segmentation task. 10000 original RGB images are converted into grey images and then Gaussian blur (with the standard deviation of 2 and window size of 7×7) is added on the target area through the example segmentation label to generate defocused parts.

Finally, 10,000 pairs of synthesized multi-focus images and corresponding accurate GT-focused maps are obtained. The focused and blurred images are fed into the SMENet, and the corresponding paired structure information features are obtained and concatenated as the input of the DMGNet. The batch size and epoch are also set as 16 and 60.

[2] The threshold is set to $0.002 \times H \times W$. Among them, H and W are the height and width of the source image, respectively.

[3] The window size of the filter is set to 5, and the smoothness is set to 1.

In order to objectively evaluate the fusion performance, we conduct experiments on three multi-focus image fusion datasets. The involved datasets include Lytro [21], MFI-WHU [15], and MFFW [22] [4]. These images are composed of rich content, such as people, plants, toys, animals, etc.

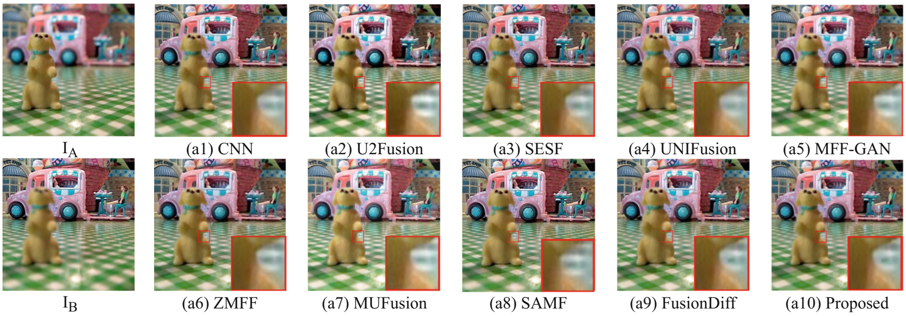


Fig. 8. Visual comparison of various image fusion methods on Lytro dataset.

4.2 Fusion Results Analysis

Seven state-of-the-art image fusion algorithms are selected as comparison algorithms, including the multi-focus image fusion with a deep convolutional neural network(CNN, 2017) [7]; a uniform fusion network (U2Fusion, 2022) [12]; an unsupervised deep model for multi-focus image fusion(SESF, 2021) [8]; an unsupervised generative adversarial network with adaptive and gradient joint constraints for multi-focus image fusion(MFF-GAN, 2021) [15]; a lightweight unified image fusion network(UNIFusion, 2021) [16]; zero-shot multi-focus image fusion(ZMFF, 2023) [9]; a general unsupervised image fusion network based on memory unit(MUFusion, 2023) [13];small-area-aware multi-focus image fusion(SAMF, 2024) [23];multi-focus image fusion based on denoising diffusion probability model(FusionDiff, 2024) [24].

In order to comprehensively evaluate and analyze the performance of MFIF methods, we selected six different metrics to conduct the quantitative experiments. Information theory-based measurement QMI [25] and QNCIE [26] as important metrics of image fusion quality evaluation, effectively evaluate the fusion results from two aspects of information retention and information difference. Image feature-based measurement QAB/F [27] and QG [28] measure the quality and practicability of image fusion technology from the transmission and retention of edge information and the overall visual effect. QE [29] is mainly

[4] Due to space limitations, the qualitative and quantitative experimental results on the MFFW dataset have been included in the supplementary materials.

evaluated based on the retention of edge information. Specifically, it evaluates the fusion quality by measuring the retention of edge information in the fused image. QCB [30] is used to evaluate the quality of fused images. It measures the visual comfort of fused images by quantifying the degree of deblocking effect.

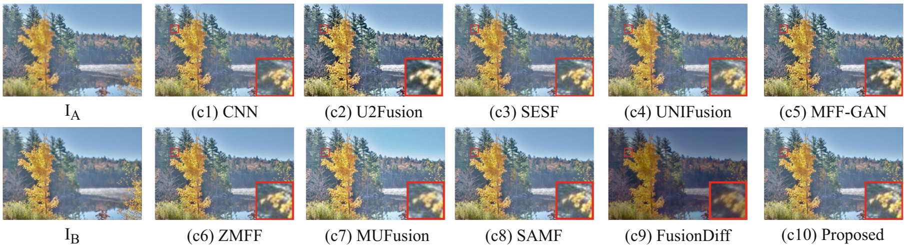


Fig. 9. Visual comparison of various image fusion methods on MFI-WHU dataset.

Subjective Evaluation As shown in Fig. 8-Fig. 9, the examples are selected from Lytro and MFI-WHU. The key parts have been marked with red boxes.

From Fig. 8, it can be seen that various methods have similar results, but the method introduced in this paper has clearer lines on the contour of the fused image. As there are no complex real-world issues in the Lytro dataset, the majority of methods have achieved visually ideal fusion images on Lytro.

In Fig. 9, it can be seen that the result of this paper, especially 'leaves', is superior to other methods in terms of structure, texture, and clarity. The clearer fusion images also validate the relatively higher accuracy of the decision maps obtained in this paper. This is attributed to the significant differences in detail between focused and unfocused areas on the structural maps derived from SMENet. U2fusion, MFF-GAN and FusionDiff are generative deep learning methods. The fused image introduces interference information that does not exist in the original images. It can be seen that the fusion results show a certain degree of color deviation.

Objective Evaluation The results of the evaluation metrics on two public datasets are shown in Table 1.

The highest performance on QG and QAB/F proves that our fusion results preserve as much gradient and visual information as possible in the source images. In most indicators, the method in this paper ranks first or second, which shows that the fused image obtained by this method has the highest comprehensive quality.

Table 1. Comparison of objective metrics on different datasets.(**Bold**:Best, Red:Second Best, Blue:Third Best)

	Lytro						MFI-WHU					
Methods	QMI	QNCIE	QAB/F	QG	QE	QCB	QMI	QNCIE	QAB/F	QG	QE	QCB
CNN	1.1095	0.8402	0.7520	0.7081	0.9034	0.7995	1.1767	0.8455	0.7299	0.7362	0.8471	**0.8269**
U2Fusion	0.7694	0.8220	0.6086	0.5150	0.7880	0.5689	0.6975	0.8193	0.5444	0.5023	0.6714	0.5161
SESF	1.1169	0.8406	0.7517	0.7088	**0.9048**	0.7992	1.1734	0.8457	0.7222	0.7321	0.8451	0.8219
MFF-GAN	0.8031	0.8236	0.6587	0.5629	0.8513	0.6452	1.1065	0.8399	0.7184	0.7212	0.8306	0.8054
UNIFusion	1.0117	0.8340	0.7377	0.6856	0.8727	0.7592	0.7704	0.8222	0.6364	0.5837	0.7737	0.6330
ZMFF	0.8838	0.8271	0.7031	0.6313	0.8912	0.7412	0.7911	0.8228	0.6318	0.5850	0.8028	0.7101
MUFusion	0.8005	0.8233	0.6676	0.5774	0.8429	0.6417	0.7226	0.8201	0.5987	0.5464	0.7356	0.6633
FusionDiff	0.7838	0.8227	0.5794	0.5004	0.6299	0.5967	0.1049	0.8088	0.0785	0.0943	0.0139	0.2922
SAMF	1.1191	0.8411	0.7511	0.7067	0.9046	0.7951	**1.1956**	**0.8474**	0.7278	0.7348	0.8480	0.8247
Ours	**1.1277**	**0.8414**	**0.7527**	**0.7108**	0.8997	**0.8006**	1.1863	0.8458	**0.7342**	**0.7365**	**0.8482**	0.8256

4.3 Ablation Studies

In this section[5], we conduct ablation experiments to demonstrate the effectiveness of the components in the proposed method. In general, our fusion model boosts the fusion performance from 3 perspectives: (1) Generating decision map in a two-stage training manner; (2) Structural map and decision map are all obtained by deep learning. (3) The structure of SMENet and DMGNet;

Fig. 10 shows the visualization results of the ablation experiments. Specifically, Fig. 10 (a) and Fig. 10 (b) represent the source images; Fig. 10 (c) represents one stage training to generate fusion results; Fig. 10 (d) represents that without deep learning, only Gaussian filtering and subtraction are used to get the structure map; Fig. 10 (e) represents the second stage replaced by using spatial frequency methods [1] to generate decision map and ultimately generate fusion

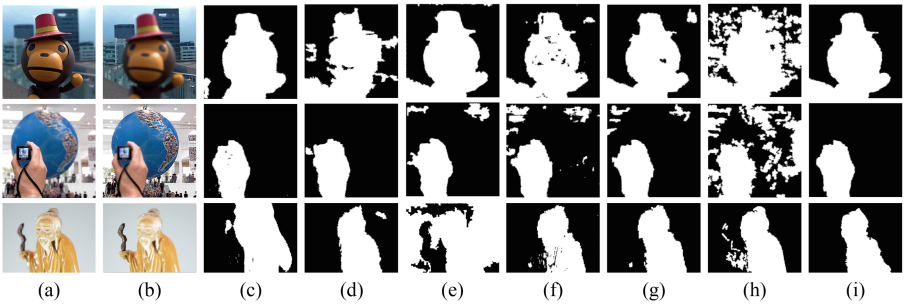


Fig. 10. (a) - (b) Source images, (c) One stage, (d) Effectiveness of SMENet, (e) Effectiveness of DMGNet, (f) Multi-scale effectiveness, (g) Effectiveness of skip connections, (h) Effectiveness of dense connections, (i) Ours

[5] Due to page constraints, the ablation experiments on the MFFW and MFI-WHU datasets are included in the supplementary materials.

images; Fig. 10 (f) represents the removal of up-sampling and down-sampling operations to verify the rationality of multi-scale; Fig. 10 (g) represents the experimental results of removing skip connections; and Fig. 10 (h) represents the replacement of the original skip connections with dense connections; Fig. 10 (i) represents this paper.

Two-stage Training Strategy In order to verify whether two-stage training is better than one stage, we merged the training of the two stages into one stage and retrained the net.

By objectively measuring the fusion results, as shown in Table 2, the outcomes obtained from the two-stage training surpass those from one stage.

While the differences in metrics may not be pronounced, the two-stage approach exhibits a clear advantage on the decision map. As illustrated in Fig. 10, where 'c' denotes the results of one-stage training and 'i' represents the outcomes of this work, the superiority of the two-stage method can be attributed to the provision of richer supervisory information during the first training stage, resulting in more precise generation of structural map.

Table 2. Performance comparison between the ablation experiments and our experiment.(**Bold**:Best)

		QMI	QNCIE	QAB/F	QG	QE	QCB
One-stage Training Strategy	c	1.1249	0.8411	0.7513	0.7079	0.9031	0.8000
Effectiveness of SMENet and DMGNet	d	1.0324	0.8367	0.6885	0.6392	0.7916	0.7587
	e	1.1263	0.8411	0.7518	0.7075	**0.9060**	**0.8015**
The effectiveness of multi-scale and network connectivity methods	f	1.1194	0.8407	0.7480	0.7060	0.8920	0.7952
	g	1.1274	0.8412	0.7516	0.7096	0.9019	0.8000
	h	1.0812	0.8393	0.6376	0.6383	0.5452	0.7360
OURS	i	**1.1279**	**0.8413**	**0.7521**	**0.7102**	0.9005	0.8003

Effectiveness of SMENet and DMGNet In order to verify the rationality of SMENet, in the first stage, each input image is processed with Gaussian filtering and then the structural map is obtained by making a difference with the original image. The second stage remains unchanged. In order to verify the rationality of DMGNet, the first stage remains unchanged, and the second stage calculates the spatial frequency to obtain the decision map.

As shown in Table 2, compared with our method, manually obtaining the structural map and decision map cannot produce better results. Fig. 11 shows the qualitative results that the structural maps obtained by the linear filtering operation (Gaussian filtering) have a lot of noise, which subsequently disrupts the creation of decision map. Fig. 10 (e) shows that decision map obtained by the spatial frequency method is unstable and unable to distinguish regions with indistinct differences in focus attributes. The decision map of the third image can confirm this point. Therefore, SMENet and DMGNet cannot be replaced.

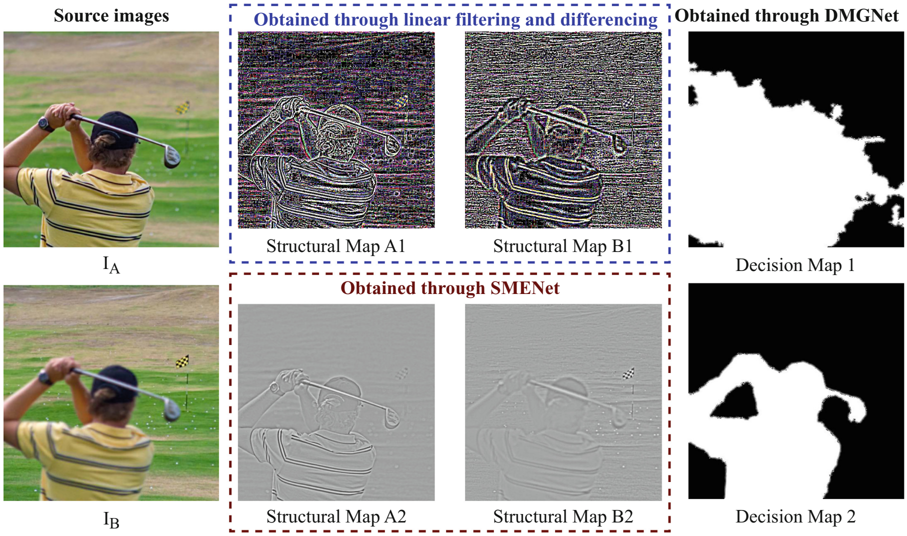


Fig. 11. I_A and I_B are the source images. *Structural map* $A1$ and $B1$ were obtained by Gaussian filtering and differencing. *Structural map* $A2$ and $B2$ were obtained by SMENet. These structural maps are trained to obtain *Decision Map* 1 and 2 respectively in the DMGNet.

The effectiveness of multi-scale and network connectivity methods In order to investigate the effectiveness of multi-scale, all up-sampling and down-sampling operations in the network are removed and fused images are generated. This ablation experiment corresponds to 'f' in Table 2 and Fig. 10. Compared to before the removal of up-sampling/down-sampling operations, all metrics show a noticeable decline, and the quality of the generated decision map significantly deteriorates.

Nowadays, there are various ways to connect convolutional blocks in neural networks. The most common operations are skip connections and dense connections. To verify the rationality of the structure, ablation experiments are conducted. As shown in Fig. 10 and Table 2, 'g' and 'h' respectively represent the experimental results after removing skip connections and dense connections. Comparing the metrics, it can be observed that the absence of skip connections or solely using dense connections significantly reduces the performance of the fused image. It is more evident from the generated decision map that there is a significant error in the decision map without the design of up-sampling and down-sampling and skip connections.

Combining the above three sets of ablation experiments, it is demonstrated that the infrastructure of SMENet and DMGNet effectively extracts features at different scales and can lead to better fusion results.

5 Conclusion

In the task of multi-focus image fusion, how to use source images to obtain high-quality decision map is a key issue. However, there is a lot of redundant information such as brightness and color in the source images that interfere with the generation of decision map.

To enhance the contrast between focused and defocused regions and reduce redundant information in source images, in this paper, a two-stage image fusion framework is introduced, termed as SMFuse. The network architecture comprises multiple convolutional layers of varying scales aimed at capturing structural information from images across different scales. In the first stage, a multi-scale feature extraction network structure is utilized to extract structural information from the source images, reducing redundant information. In the second stage, the structural information obtained from the first stage is used to train and generate decision map.

Compared with the state-of-the-art methods, the proposed method achieves better fusion performance in the subjective evaluation and objective evaluation. However, due to the blurring of the focus area boundary, the decision map will lead to a certain gradient dispersion phenomenon in the fused image. How to further improve the performance of feature extraction and reduce the occurrence of gradient discretization remains to be further studied. In addition, the method proposed in this paper is a two-stage fusion architecture, the time cost is high. How to lighten the model needs further research.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (62202205,62306203), the National Social Science Foundation of China(21 & ZD166), the Natural Science Foundation of Jiangsu Province, China(BK20221535), and the Fundamental Research Funds for the Central Universities (JUSRP123030).

References

1. Wang, Z., Li, X., Duan, H., Zhang, X.: A self-supervised residual feature learning model for multifocus image fusion. IEEE Trans. Image Process. **31**, 4527–4542 (2022)
2. Li, X., Li, Y., Chen, H., Peng, Y., Chen, L., Wang, M.: Ritfusion: Reinforced interactive transformer network for infrared and visible image fusion. IEEE Trans. Instrum. Meas. **73**, 1–16 (2024)
3. Da Cunha, A.L., Zhou, J., Do, M.N.: The nonsubsampled contourlet transform: Theory, design, and applications. IEEE Trans. Image Process. **15**(10), 3089–3101 (2006)
4. Easley, G., Labate, D., Lim, W.-Q.: Sparse directional image representations using the discrete shearlet transform. Appl. Comput. Harmon. Anal. **25**(1), 25–46 (2008)
5. Zhang, Q., Wang, F., Luo, Y., Han, J.: Exploring a unified low rank representation for multi-focus image fusion. Pattern Recogn. **113**, 107752 (2021)
6. Liu, Yu., Liu, S., Wang, Z.: Multi-focus image fusion with dense sift. Information Fusion **23**, 139–155 (2015)

7. Yu Liu, Xun Chen, Hu Peng, and Zengfu Wang. Multi-focus image fusion with a deep convolutional neural network. *Information Fusion*, 36:191–207, 2017.
8. Boyuan Ma, Yu., Zhu, X.Y., Ban, X., Huang, H., Mukeshimana, M.: Sesf-fuse: an unsupervised deep model for multi-focus image fusion. Neural Comput. Appl. **33**, 5793–5804 (2021)
9. Xingyu, H., Jiang, J., Liu, X., Ma, J.: Zmff: Zero-shot multi-focus image fusion. Information Fusion **92**, 127–138 (2023)
10. Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. Structure-preserving super resolution with gradient guidance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7769–7778, 2020
11. Yuan Tian, Guo Lu, Yichao Yan, Guangtao Zhai, Li Chen, and Zhiyong Gao. A coding framework and benchmark towards low-bitrate video understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024
12. Han, X., Ma, J., Jiang, J., Guo, X., Ling, H.: U2fusion: A unified unsupervised image fusion network. IEEE Trans. Pattern Anal. Mach. Intell. **44**(1), 502–518 (2022)
13. Cheng, C., Tianyang, X., Xiao-Jun, W.: Mufusion: A general unsupervised image fusion network based on memory unit. Information Fusion **92**, 80–92 (2023)
14. Guo, X., Nie, R., Cao, J., Zhou, D., Mei, L., He, K.: Fusegan: Learning to fuse multifocus image via conditional generative adversarial network. IEEE Trans. Multimedia **21**(8), 1982–1996 (2019)
15. Zhang, H., Le, Z., Shao, Z., Han, X., Ma, J.: Mff-gan: An unsupervised generative adversarial network with adaptive and gradient joint constraints for multi-focus image fusion. Information Fusion **66**, 40–53 (2021)
16. Cheng, C., Xiao-Jun, W., Tianyang, X., Chen, G.: Unifusion: A lightweight unified image fusion network. IEEE Trans. Instrum. Meas. **70**, 1–14 (2021)
17. Xiao, B., Bocheng, X., Bi, X., Li, W.: Global-feature encoding u-net (geu-net) for multi-focus image fusion. IEEE Trans. Image Process. **30**, 163–175 (2021)
18. Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004
19. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
20. Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013
21. Nejati, M., Samavi, S., Shirani, S.: Multi-focus image fusion using dictionary-based sparse representation. Information Fusion **25**, 72–84 (2015)
22. Xiao, B., Bocheng, X., Bi, X., Li, W.: Global-feature encoding u-net (geu-net) for multi-focus image fusion. IEEE Trans. Image Process. **30**, 163–175 (2021)
23. Xilai Li, Xiaosong Li, Haishu Tan, and Jinyang Li. Samf: Small-area-aware multifocus image fusion for object detection. *ArXiv*, abs/2401.08357, 2024
24. Fusiondiff: Multi-focus image fusion using denoising diffusion probabilistic models. *Expert Systems with Applications*, 238:121664, 2024
25. Pingfan Yan Guihong, Q., Zhang, D.: Information measure for performance of image fusion. Electron. Lett. **38**, 3 (2002)
26. Qiang Wang, Yi Shen, and Jing Jin. 19 - performance evaluation of image fusion techniques. *Image Fusion*, pages 469–492, 2008

27. C.S. Xydeas and V. Petrovi. Objective image fusion performance measure. 2000
28. Costas, S.: Xydeas and Vladimir S. Petrovic. Objective pixel-level image fusion performance measure. In: Dasarathy, B.V. (ed.) Sensor Fusion: Architectures. Algorithms, and Applications IV, volume 4051, pp. 89–98. International Society for Optics and Photonics, SPIE (2000)
29. G. Piella and H. Heijmans. A new quality metric for image fusion. In *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, volume 3, pages III–173, 2003
30. Yin Chen and Rick S. Blum. A new automated quality assessment algorithm for image fusion. *Image and Vision Computing*, 27(10):1421–1432, 2009. Special Section: Computer Vision Methods for Ambient Intelligence

Despeckling SAR Images Using CNN-Based Approach Incorporating GAN and Gradient Estimation

Anirban Saha(✉) and Suman Kumar Maji

Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna 801106, Bihar, India
{anirban_2021cs13,smaji}@iitp.ac.in

Abstract. Synthetic Aperture Radar (SAR) technology stands at the forefront of capturing and processing Earth's surface visuals due to its widespread acceptance across various organizations. However, the presence of unwanted random granular interference, commonly referred to as “speckle," poses a significant challenge in SAR data processing. Addressing this challenge, known as “despeckling," is crucial for extracting clear SAR visuals. This article introduces a novel CNN-based approach for despeckling SAR visuals contaminated with speckle. Our proposed model integrates a Generative Adversarial Network (GAN) module to estimate the distribution of contaminating speckle components from the input SAR data. Concurrently, a gradient estimator module captures the crisp changes in textural information within the input data. Subsequently, the input SAR data, the estimated noise distribution, and the extracted gradient undergo further processing through a deep convolutional module to generate a clean SAR visual. Unlike traditional methods that focus solely on learning the residual noisy component or the clean data, our proposed despeckling model learns the degradation pattern caused by noisy components while emphasizing gradient information, thereby capturing critical minute information. Experimental results demonstrate that our methodology significantly enhances despeckling performance compared to existing technologies in the literature. This research presents a promising step forward in advancing SAR visual despeckling techniques, with implications for improved data quality and interpretation in various applications.

Keywords: Synthetic Aperture Radar (SAR) · Convolutional Neural Network (CNN) · Generative Adversarial Network (GAN) · SAR Despeckling · SAR denoising · SAR Image Restoration

1 Introduction

Before the advent of machine learning (ML) and deep learning (DL), the field of SAR image despeckling predominantly relied on filter-based techniques to

Supported by the Science and Engineering Research Board (SERB) (Government of India) Core Research under Grant CRG/2022/002010.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 18–32, 2025.
https://doi.org/10.1007/978-3-031-78312-8_2

address noise issues. Notably, the "Lee Filter" demonstrated its effectiveness in reducing noise, especially in uniform sections of SAR images but encountered challenges when applied to non-uniform areas [19]. To overcome this limitation, the "Enhanced Lee Filter" was introduced, proving its effectiveness in both uniform and non-uniform scenarios [18]. The "Frost Filter" introduced adaptability by analyzing local image statistics [9], while the "Kaun Filter" improved upon the Lee Filter by averaging pixel intensity within a defined window [17]. However, these filters exhibited reduced efficiency when dealing with images that had varying variances across different regions. To address this challenge, a controlled filtration technique was proposed, delivering superior denoising results [20]. In the 21st century, significant advancements have occurred in SAR image denoising strategies. An early approach presented in [10] bears resemblance to a speckle filtering method that combines the Stationary Wavelet Transform (SWT) with an averaging smoothing process. Another technique shares similarities with the Coherence Reduction Speckle Noise (CRSN) algorithm [13], which heavily relies on the coherent principles underlying SAR imaging. An improved despeckling technique based on the non-local means approach was introduced, enhancing SAR image quality by comparing and averaging similar patches with statistical correlations, thereby leveraging image redundancy and self-similarity [7]. A similar filtration approach, as described in [2], examines the performance of adaptive stack filters. Another method incorporates adaptive filter parameters obtained from unscented Kalman filter output sampling [16]. The study conducted by Torres et. al. [31] also showcased the effectiveness of employing an NLM-based method in conjunction with a statistical test that relies on stochastic divergences. Additionally, a dual-phase approach for SAR image denoising was proposed, involving an initial stage of hard thresholding applied to directionally smoothed output, followed by a final step using a hybrid Gaussian-Laplacian filter to enhance processed images [29]. In conjunction with the findings presented in [15], the performance of despeckling was showcased through the utilization of a hybrid filter operating in the global thresholding domain encompassing both spatial and frequency aspects. Moreover approach such as those outlined in [22] is employed to derive meaningful characteristics from a distorted image affected by speckle noise. Subsequently, an improved, noise-reduced rendition of the image is inferred using the informative gradient. However, despite their advantages, these filtration methods exhibit significant limitations. The primary issue lies in excessive smoothing, which leads to noticeable blurring and a loss of detail along sharp edges. Moreover, they introduce blocking artifacts, unintended ghostly textural artifacts, and a residual presence of speckle components in the denoised output.

In the late 2010s, DL-based denoising methods made significant progress in SAR image enhancement. In [36], an approach using a residual neural network with shortcut connections and atrous convolutions was applied, harnessing nonlinearity for denoising. An innovative approach, discussed in [32], introduced a sophisticated cost function for effective SAR denoising while preserving intricate details. Another study, referenced in [5], improved SAR denoising by utilizing various convolutions in the despeckling network. [8] presented a conventional

10-layer despeckling network emphasizing a novel cost function incorporating statistical characteristics and optical properties. Additionally, [24] proposed log transformation for input analysis and a flexible CNN model for speckle removal. They evaluated the model's performance with and without iteration-based nonlinear correction. [11] prioritized preserving texture-level details, integrating a unit for accurate texture mapping, and training a module for noise removal in both uniform and non-uniform areas. [21] addressed the challenge of obtaining noisy-ground truth image pairs by developing a DL-based denoising technique employing residual learning, yielding substantial image quality improvements. [6] explored effective SAR noise handling using a Fisher-Trippett despeckling module with log transformation, demonstrating remarkable efficacy through visual and quantitative assessments. Several recent learning-based SAR despeckling methodologies have been documented in articles [28,30]. These approaches have substantially enhanced despeckling performance while maintaining intricate details inherent in the input imagery.

This article introduces a novel CNN-based approach to SAR visual despeckling, presenting distinctive contributions that address key limitations of traditional methods:

1. **Integration of Generative Adversarial Network (GAN) Module:** We propose the incorporation of a GAN module, a cutting-edge deep learning technique, to estimate the distribution of speckle components. By leveraging the power of GANs, our model can effectively capture complex distorting patterns within SAR imagery, enhancing the accuracy of speckle removal.
2. **Inclusion of Gradient Estimator Module:** Our approach incorporates a gradient estimator module alongside the GAN module. This module focuses on capturing textural information, thus enabling the model to preserve intricate details often lost or distorted in traditional despeckling methods. By emphasizing gradient data, our model ensures a more faithful representation of the underlying scene.
3. **Unique Approach to Despeckling:** Our model takes a unique despeckling approach, unlike conventional methods that typically target noise reduction or clean data generation. It learns the specific degradation pattern caused by speckles while simultaneously emphasizing gradient information. This dual focus significantly improves despeckling performance, leading to enhanced data quality and interpretation.
4. **Promise of Enhanced Data Quality:** Through our research, we anticipate substantial advancements in SAR imagery processing. By effectively mitigating speckle artifacts while preserving crucial details, our model promises to elevate the quality and interpretability of SAR data across various applications, ranging from environmental monitoring to disaster response and urban planning.

In summary, our work represents a crucial advancement in SAR visual despeckling, offering a comprehensive solution that leverages deep learning techniques to overcome longstanding challenges in the field.

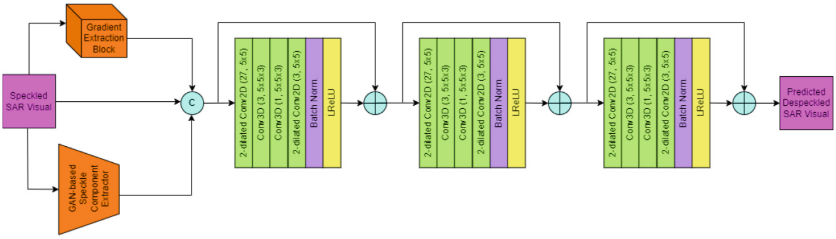

Gradient Extraction Block
Speckled SAR Visual
GAN-based Speckle Component Extractor
C
2-dilated Conv2D (27, 5x5)
Conv3D (3, 5x5x3)
Conv3D (1, 5x5x3)
2-dilated Conv2D (3, 5x5)
Batch Norm.
LReLU
2-dilated Conv2D (27, 5x5)
Conv3D (3, 5x5x3)
Conv3D (1, 5x5x3)
2-dilated Conv2D (3, 5x5)
Batch Norm.
LReLU
2-dilated Conv2D (27, 5x5)
Conv3D (3, 5x5x3)
Conv3D (1, 5x5x3)
2-dilated Conv2D (3, 5x5)
Batch Norm.
LReLU
Predicted Despeckled SAR Visual

Fig. 1. Architectural Overview of the proposed model.

2 Proposed Methodology

2.1 Problem Definition and Approach

The presence of noise in acquired Synthetic Aperture Radar (SAR) data exhibits a multiplicative nature concerning the pixel values. This noise interference is formally expressed by Equation 1, where X, Y, and $\mathfrak{S}$ represent the intended clear visual, the raw captured noisy visual, and the contaminating speckle distribution, respectively.

$$Y = X \times \mathfrak{S} \tag{1}$$

The detrimental effect of frequency interference, causing disruptive visual noise, is known to conform to a characteristic following the Gamma distribution within the visual-spatial domain. The properties of this distribution concerning the data are elucidated by the probability distribution delineated in Equation 2.

$$\rho_n(\mathfrak{S}) = \frac{L^L n^{L-1} \exp\left(-nL\right)}{\Gamma(L)}; \qquad n \geq 0, L \geq 1 \tag{2}$$

The primary aim of a despeckling model is to alleviate the multiplicative granular structure, thereby converting the visual representation denoted as Y into an estimated representation $\hat{X}$ that closely resembles the original X [27].

The proposed methodology aims to extract the $\mathfrak{S}$ distribution rather than learning the conventional data transformation from Y to X. The model is then trained with this extracted information to comprehend how these noisy components impact different segments of X, facilitating the prediction of the closest approximation, $\hat{X}$. The suggested model is visually depicted in Figure 1. It incorporates a generative adversarial network (GAN) module specifically designed and trained to estimate the speckle distribution from Y and generate $\hat{\mathfrak{S}}$. Simultaneously, a gradient-based feature estimator module computes $\nabla(Y)$, capturing the sharp features in Y. Subsequently, $\hat{\mathfrak{S}}$ and $\nabla(Y)$ are integrated with Y and further processed to extract the relationships between these inputs to estimate $\hat{X}$.

2.2 GAN-based Speckle Estimator

This section delineates the objective of the module, which is to utilize a generative adversarial network (GAN) to extract the stochastic, detailed speckle

Table 1. Details of used notations.

Notation	**Description**
$\kappa^{\{f,w\}}$	'f' number of kernels each with dimension 'w'
$\circledast_{j,s}$	'j' dimensional convolution with stride 's'
$\circledast^{k}_{j,s}$	'k' dilated 'j' dimensional convolution with stride 's'
N	Batch Normalization
$R^{(l)}$	Leaky ReLU
$\odot$	Concatenation
$(\bullet)^{Z_j}$	'j' dimensional zero padding

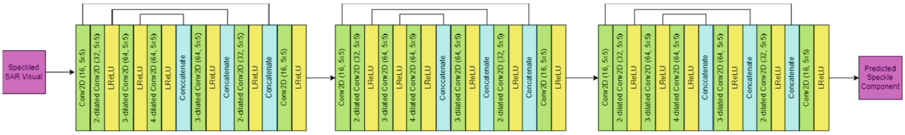

Speckled SAR Visual
Conv2D (16, 5x5)
2-dilated Conv2D (32, 5x5)
LReLU
3-dilated Conv2D (64, 5x5)
LReLU
4-dilated Conv2D (64, 5x5)
LReLU
Conccatenate
3-dilated Conv2D (64, 5x5)
LReLU
Concatenate
2-dilated Conv2D (32, 5x5)
LReLU
Concatenate
Conv2D (16, 5x5)
LReLU
Conv2D (16, 5x5)
2-dilated Conv2D (32, 5x5)
LReLU
3-dilated Conv2D (64, 5x5)
LReLU
4-dilated Conv2D (64, 5x5)
LReLU
Conccatenate
3-dilated Conv2D (64, 5x5)
LReLU
Concatenate
2-dilated Conv2D (32, 5x5)
LReLU
Concatenate
Conv2D (16, 5x5)
LReLU
Conv2D (16, 5x5)
2-dilated Conv2D (32, 5x5)
LReLU
3-dilated Conv2D (64, 5x5)
LReLU
4-dilated Conv2D (64, 5x5)
LReLU
Conccatenate
3-dilated Conv2D (64, 5x5)
LReLU
Concatenate
2-dilated Conv2D (32, 5x5)
LReLU
Concatenate
Conv2D (16, 5x5)
LReLU
Predicted Speckle Component

Fig. 2. Architectural Overview of the Generator Component within GAN-based Speckle Component Extractor module of the proposed model.

distribution impacting the obtained raw data. It delves into a comprehensive examination of the design intricacies pertaining to both the generator and discriminator modules, which play a vital role in enhancing the generative potential of the system. Table 1 enumerates the significance of different notations employed in the subsequent sections.

Generator: The effective approximation of the undesired granular structure is achieved using a deep convolutional module. This module follows a sequential model, visually represented in Figure 2, comprising three instances of submodules with similar layer configurations, denoted as G^1_7, G^2_7, and G^3_7.

Each generative instance sequentially processes input through seven interconnected blocks, incorporating skip-connections and merge layers at specific points. The initial block employs a convolutional operation, detailed in Equation 3.

$$G^i_1 = \begin{cases} \kappa^{\{16,5\times5\}} \circledast_{2,1} Y, & \text{if } i = 1 \\ \kappa^{\{16,5\times5\}} \circledast_{2,1} G^{i-1}_7, & \text{Otherwise} \end{cases} \tag{3}$$

Subsequent blocks utilize dilated convolutional operations of increasing rates: 2-dilated, 3-dilated, and 4-dilated, each followed by a leaky ReLU activation, as represented by Equations 4, 5, and 6.

$$G^i_2 = R^{(l)}\left(\kappa^{\{32,5\times5\}} \circledast^2_{2,1} G^i_1\right) \tag{4}$$

$$G^i_3 = R^{(l)}\left(\kappa^{\{64,5\times5\}} \circledast^3_{2,1} G^i_2\right) \tag{5}$$

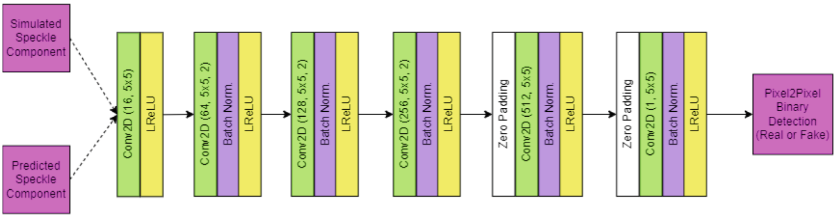


Fig. 3. Architectural Overview of the Discriminator Component within GAN-based Speckle Component Extractor module of the proposed model.

$$G_4^i = R^{(l)}\left(\kappa^{\{64,5\times5\}} \circledast_{2,1}^{4} G_3^i\right) \tag{6}$$

The remaining blocks in each instance operate on a merged input from the preceding layer and skip connections, employing 3-dilated, 2-dilated, and standard 2-D convolutional operations, followed by a leaky ReLU activation. Equations 7, 8, and 9 outline these operations.

$$G_5^i = R^{(l)}\left(\kappa^{\{64,5\times5\}} \circledast_{2,1}^{3} (G_4^i \odot G_3^i)\right) \tag{7}$$

$$G_6^i = R^{(l)}\left(\kappa^{\{32,5\times5\}} \circledast_{2,1}^{2} (G_5^i \odot G_2^i)\right) \tag{8}$$

$$G_7^i = R^{(l)}\left(\kappa^{\{16,5\times5\}} \circledast_{2,1} (G_6^i \odot G_1^i)\right) \tag{9}$$

The output of the last module, G_7^3, corresponds to the predicted $\hat{\mathfrak{S}}$ (Equation 10).

$$\hat{\mathfrak{S}} = G_7^3 \tag{10}$$

While training this generator module to extract the speckle distribution, it is essential to consider integrating a carefully designed discriminator module. This discriminator should effectively distinguish between pixel data generated by the generator and genuine pixel data representing the speckle structure.

Discriminator: The discriminator component employs a sequence comprising six consecutive blocks, each processing data through a series of complex network layers. This process ultimately yields a two-dimensional binary output, distinguishing pixel values as either real or generated by the generator. Figure 3 depicts a schematic representation of this architectural arrangement. The specific number of filters utilized by the convolutional layer within each block can be determined from the set defined in Equation 11.

$$F_D = [16, 64, 128, 256, 512, 1] \tag{11}$$

As the architectural design shows, the initial layer consists of a 2D convolutional layer paired with a leaky ReLU activation layer, as shown in Equation 12.

$$D^1 = R^{(l)}\left(\kappa^{\{F_D^1,5\times5\}} \circledast_{2,1} D^0\right) \tag{12}$$

Following this initial block, three similar blocks follow, each employing a sequential arrangement of 2D convolution, batch normalization, and a leaky ReLU activation layer. The functionality of these blocks can be inferred from Equation 13.

$$D^i = R^{(l)}\left(N\left(\kappa^{\{F_D^i,5\times5\}} \circledast_{2,2} D^{i-1}\right)\right); \qquad i \in \{2,3,4\} \tag{13}$$

Ultimately, the processing sequence in the last two blocks mirrors that of the preceding three blocks, with the addition of an initial layer involving 2D zero padding at each block. Consequently, the mathematical formulation governing the operation of these blocks undergoes a transformation, as described in Equation 14.

$$D^i = R^{(l)}\left(N\left(\kappa^{\{F_D^i,5\times5\}} \circledast_{2,1} \left(D^{i-1}\right)^{Z_2}\right)\right); \qquad i \in \{5,6\} \tag{14}$$

The final result of this module is a pixel-by-pixel representation that distinguishes between genuine and counterfeit, thereby balancing the trade-off with the generator module.

2.3 Gradient Extraction Block

This module holds a pivotal function in assessing the degree of significant changes observed across diverse geographical regions or the intrinsic structural attributes within the provided data. The determination of these metrics involves computing both horizontal ($\nabla_h(Y)$) and vertical ($\nabla_v(Y)$) gradient information. The horizontal gradient ($\nabla_h(Y)$) is obtained by convolving the input data with the kernel κ'_h as specified in equation 15. Similarly, the vertical gradient ($\nabla_v(Y)$) is computed through convolution with the kernel κ'_v as outlined in equation 16.

$$\nabla_h(Y) = \kappa'_h \circledast_{2,1} Y; \qquad \kappa'_h = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{15}$$

$$\nabla_v(Y) = \kappa'_v \circledast_{2,1} Y; \qquad \kappa'_v = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{16}$$

Subsequently, the comprehensive gradient ($\nabla(Y)$) of the input image is determined by performing element-wise matrix operations on the previously calculated horizontal and vertical gradients, as described in equation 17.

$$\nabla(Y) = \sqrt{[\nabla_h(Y)]^2 + [\nabla_v(Y)]^2} \tag{17}$$

2.4 Mapping of Despeckled Visual

The primary objective of this module is to establish the correlation between input data and the derived speckle and gradient information, ultimately associating it with its corresponding despeckled outcome. The initial input, symbolized as O^0, undergoes sequential processing through four restoration sub-modules. The core operation within this module involves concatenating the input visual data, estimated speckle distribution, and gradient information, as illustrated in Equation 18.

$$O^0 = (Y \odot \hat{\mathfrak{S}} \odot \nabla(Y)) \tag{18}$$

The initial outcome, designated O^0, undergoes sequential processing through three similar sub-modules. These sub-modules primarily entail a predetermined combination of 2D dilated and 3D convolutional operations. The operations in these sub-modules can be broadly categorized into four distinct procedures. Initially, the processing involves filtration using a 2D convolutional operation with a 2-dilated kernel, as depicted in Equation 19.

$$O_1^i = \kappa^{\{27,5\times 5\}} \circledast_{2,1}^{2} O^{i-1}; \qquad i \in \{1,2,3,4\} \tag{19}$$

Subsequently, the data undergoes processing through two consecutive series of 3D convolutional blocks, as represented by Equations 20 and 21 corresponding to the two sets of convolutional blocks, respectively.

$$O_2^i = \kappa^{\{3,5\times 5\times 3\}} \circledast_{3,1} O_1^i; \qquad i \in \{1,2,3,4\} \tag{20}$$

$$O_3^i = \kappa^{\{1,5\times 5\times 3\}} \circledast_{3,1} O_2^i; \qquad i \in \{1,2,3,4\} \tag{21}$$

The final sequence of operations within this sub-module comprises a stacked configuration consisting of a 2D convolutional block with a dilation factor of 2, followed by a batch normalization block, a leaky ReLU activation, and ultimately an additive merge layer that combines with the input of this sub-module. The entire operation can be analyzed mathematically by referencing Equation 22.

$$O^i = O^{i-1} \oplus \left(R^{(l)}\left(N\left(\kappa^{\{3,5\times 5\}} \circledast_{2,1}^{2} O_3^i\right)\right)\right); \qquad i \in \{1,2,3,4\} \tag{22}$$

The processed outcome of the last sub-module, denoted as O^3, corresponds to the estimated despeckled visual, as represented in Equation 23.

$$\hat{X} = O^3 \tag{23}$$

3 Experimental Analysis

This section serves as an authoritative evaluation of the potential of the proposed system, assessing enhancements in image quality through scrutiny of both visual and performance-related metrics. Comparative analysis has been conducted against contemporary state-of-the-art models, including PPB [7], SAR-BM3D [25], SAR-CNN [3], SAR-IDCNN [33], SAR-DRN [36], Nb2Nb [14], and

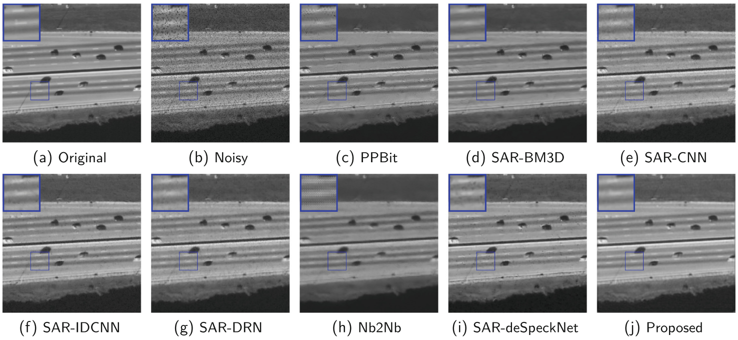


Fig. 4. Result over subset of synthetic Freeway data contaminated with look 16 speckle.

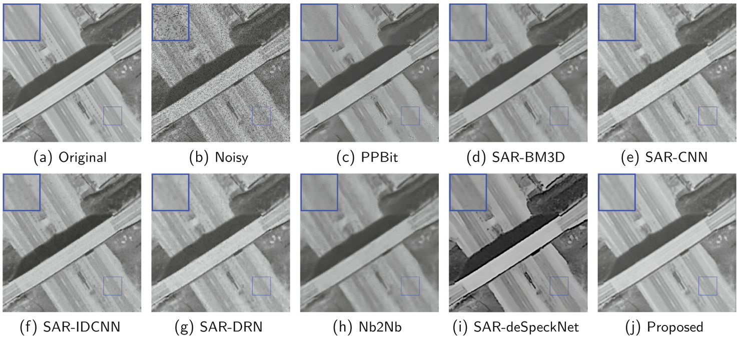
SAR-deSpeckNet [23]. The efficacy of the system has been appraised utilizing both simulated and empirically validated real-world data.

This model utilizes a predefined segment of the UCMerced LandUse Dataset [35] for its training regimen. The selected training data is corrupted by speckle noise, generated following a Gamma Distribution with a randomly chosen look level ranging from 1 to 20. This process aims to construct a training dataset comprising three main components: intentionally introduced noisy data, artificially generated noise, and clean data. Subsequently, each subset within this training dataset trains specific sub-modules designed for particular tasks. Furthermore, the training process occurs in two stages, with the training data being partitioned into patches of dimensions 32×32.

Initially, the GAN-based Speckle estimator module is trained using artificially generated noise and noisy data from the training set. This training employs a batch gradient optimizer and a conventional binary cross-entropy-based GAN loss. Throughout this phase, the learning rate is maintained at 10^{-4} for 10,000 training steps to fine-tune the model. Following this, the pre-trained GAN module is integrated with other adjustable parameters within the proposed system, initiating the second training phase. Here, clear data functions as the ground truth, while noisy data from the training set is utilized as input. The model undergoes training for 35 epochs employing the Mean Squared Error (MSE) loss function and the ADAM optimizer. Throughout this phase, the learning rate is sustained at 10^{-4}.

3.1 Simulated Data Experimentation

Two randomly selected classes from the UC Merced Land-use dataset were deliberately corrupted with varying noise levels. Subsequently, the efficacy of the proposed method in reducing noise was demonstrated using these images, contrasting with existing models documented in the literature. Specifically, these images were sourced from the dataset's "Freeway" and "Overpass" categories.


(a) Original (b) Noisy (c) PPBit (d) SAR-BM3D (e) SAR-CNN

(f) SAR-IDCNN (g) SAR-DRN (h) Nb2Nb (i) SAR-deSpeckNet (j) Proposed

Fig. 5. Result over subset of synthetic Overpass data contaminated with look 8 speckle.

Table 2. Assessment Outcomes $\{Mean(\mu) \pm Std.(\sigma)\}$ across Simulated Data Affected by Different Looks of Speckle Noise. The red shading denotes the top-performing outcome, while the blue shading indicates the second-best outcome.

Data	Looks	Metric	PPB	SAR-BM3D	SAR-CNN	SAR-IDCNN	SAR-DRN	NB2NB	SAR-deSpeckNet	Proposed
Freeway	*L = 1*	**PSNR**	14.8340 ± 1.1564	16.3021 ± 1.1474	24.2691 ± 1.6121	23.3147 ± 0.9890	26.2807 ± 1.7713	21.2076 ± 2.4772	26.4516 ± 2.3952	28.2807 ± 2.2306
		SSIM	0.3426 ± 0.0634	0.5786 ± 0.0899	0.6108 ± 0.0675	0.5653 ± 0.0728	0.6918 ± 0.0815	0.4921 ± 0.0725	0.7144 ± 0.0696	0.8528 ± 0.0675
	L = 2	**PSNR**	17.5305 ± 1.1173	18.3866 ± 1.1348	24.0314 ± 1.7182	23.8681 ± 0.8338	27.1844 ± 1.8362	22.2063 ± 2.2091	27.7447 ± 2.5758	28.3386 ± 2.1922
		SSIM	0.4911 ± 0.0673	0.6560 ± 0.0891	0.5898 ± 0.0784	0.5902 ± 0.0591	0.7260 ± 0.0812	0.5630 ± 0.0759	0.7384 ± 0.0625	0.8755 ± 0.053
	L = 4	**PSNR**	19.9690 ± 1.0938	20.6014 ± 1.1454	26.1572 ± 2.3169	28.5499 ± 1.6752	28.3738 ± 1.8656	23.2642 ± 2.4262	28.7118 ± 2.6002	30.4928 ± 2.7341
		SSIM	0.6115 ± 0.0660	0.7324 ± 0.0799	0.7093 ± 0.0873	0.7830 ± 0.0565	0.7698 ± 0.0726	0.6227 ± 0.0792	0.7551 ± 0.0556	0.8976 ± 0.0359
	L = 8	**PSNR**	22.2361 ± 1.0775	22.9064 ± 1.1337	26.8597 ± 1.5851	28.2572 ± 1.2039	29.3257 ± 1.8452	24.0221 ± 2.4984	29.6355 ± 2.5825	30.7041 ± 2.3406
		SSIM	0.7008 ± 0.0615	0.7987 ± 0.0665	0.7155 ± 0.0681	0.8122 ± 0.0472	0.8018 ± 0.0620	0.6728 ± 0.0825	0.7682 ± 0.0491	0.8796 ± 0.0636
	L = 16	**PSNR**	24.4589 ± 1.0923	25.2140 ± 1.1353	28.2609 ± 2.8215	31.3048 ± 1.9964	30.1736 ± 1.8911	24.7875 ± 2.6626	30.7219 ± 2.6905	32.2295 ± 3.0938
		SSIM	0.7721 ± 0.0527	0.8502 ± 0.0500	0.7777 ± 0.0722	0.8726 ± 0.0481	0.8248 ± 0.0582	0.7138 ± 0.0826	0.7793 ± 0.0432	0.9128 ± 0.0312
Overpass	*L = 1*	**PSNR**	15.1345 ± 1.6135	16.6063 ± 1.5983	23.5625 ± 2.0563	22.9077 ± 1.1315	25.3018 ± 1.8781	20.1988 ± 2.5958	25.2182 ± 2.0324	26.4947 ± 2.6529
		SSIM	0.3643 ± 0.0547	0.5918 ± 0.0634	0.6108 ± 0.0772	0.5682 ± 0.0573	0.6907 ± 0.0630	0.4942 ± 0.0566	0.7044 ± 0.0578	0.8179 ± 0.041
	L = 2	**PSNR**	17.7893 ± 1.5589	18.7177 ± 1.5980	23.7553 ± 1.9091	23.4548 ± 0.7649	26.3376 ± 1.8453	21.4748 ± 2.3488	26.6726 ± 2.1356	27.456 ± 2.2645
		SSIM	0.5133 ± 0.0581	0.6743 ± 0.0615	0.6024 ± 0.0784	0.5903 ± 0.0418	0.7271 ± 0.0601	0.5621 ± 0.0565	0.7316 ± 0.0537	0.8641 ± 0.0501
	L = 4	**PSNR**	20.1989 ± 1.5151	20.9589 ± 1.6053	25.9189 ± 2.3338	27.9159 ± 1.5778	27.7050 ± 1.8471	22.7398 ± 2.3424	27.7381 ± 2.1128	28.6308 ± 2.3208
		SSIM	0.6307 ± 0.0577	0.7496 ± 0.0585	0.7219 ± 0.0840	0.7819 ± 0.0386	0.7734 ± 0.0554	0.6206 ± 0.0591	0.7494 ± 0.0497	0.8588 ± 0.0434
	L = 8	**PSNR**	22.4429 ± 1.4908	23.2477 ± 1.6007	26.6152 ± 1.9989	27.8802 ± 1.1833	28.7674 ± 1.8671	23.5494 ± 2.2968	28.7490 ± 2.0800	28.9507 ± 2.1267
		SSIM	0.7159 ± 0.0533	0.8107 ± 0.0518	0.7218 ± 0.0661	0.8149 ± 0.0414	0.8063 ± 0.0507	0.6674 ± 0.0605	0.7637 ± 0.0448	0.8537 ± 0.0451
	L = 16	**PSNR**	24.6658 ± 1.4892	25.5174 ± 1.5926	25.0218 ± 6.1970	30.7870 ± 1.8406	29.6346 ± 1.8674	24.3848 ± 2.4579	29.9174 ± 2.2301	30.6196 ± 2.4921
		SSIM	0.7837 ± 0.0464	0.8577 ± 0.0414	0.7803 ± 0.0752	0.8735 ± 0.0360	0.8290 ± 0.0464	0.7046 ± 0.0625	0.7759 ± 0.0402	0.8895 ± 0.0478

Conventional metrics for assessing visual quality, such as the Structural Similarity (SSIM) [34] index and the Peak Signal-to-Noise Ratio (PSNR) [12], were employed for the quantitative analysis of the results.

The visuals depicted in Figures 4 and 5 present a visual examination of data pertaining to the "Freeway" and "Overpass" categories within the UC Merced Land-Use dataset. The data underwent perturbation with noise levels of 16 and 8 for the respective classes, followed by processing through various despeckling techniques. The visual analysis reveals that certain methods, such as PPB, SAR-BM3D, and NB2NB, resulted in excessive data smoothing. Conversely, approaches such as SAR-CNN, SAR-IDCNN, SAR-DRN, and SAR-deSpeckNet aimed at preserving finer details from the input data but exhibited noticeable levels of undesired speckle. In contrast, the proposed method appears to strike a harmonious balance between preserving intricate features and mitigating speckle, as its visual output demonstrates superior preservation of lines and patterns.

Table 2 presents a comprehensive assessment of quantitative analysis. It exhibits the mean Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity

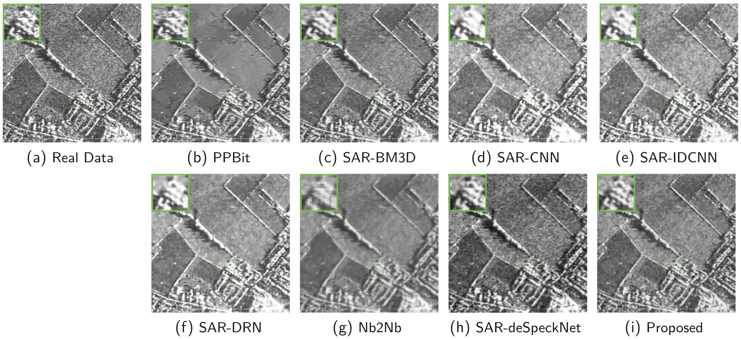


Fig. 6. Result over Real data [Bedfordshire].

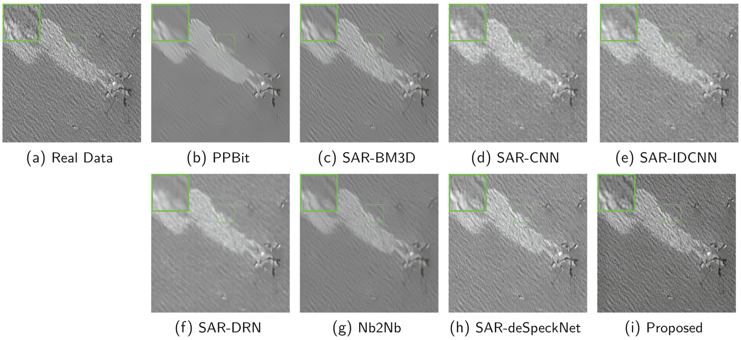


Fig. 7. Result over Real data [OilRigExplosion].

Table 3. Evaluation Results over Real Data (Metric Calculation Window Size - 13)

extbfData	**Metric**	**PPB**	**SAR-BM3D**	**SAR-CNN**	**SAR-IDCNN**	**SAR-DRN**	**NB2NB**	**SAR-deSpeckNet**	**Proposed**
Bedfordshire	**ENL**	3.4951	3.2339	2.9550	3.0562	3.9921	4.0545	2.1548	4.3609
	EPI	0.9554	0.9538	0.8867	0.9153	0.9163	0.9102	0.9617	0.9627
	TCR	0.1329	0.1061	0.9914	0.9896	0.8053	0.3289	0.5415	0.0010
OilRigExplosion	**ENL**	4.0629	4.0143	3.9801	4.1194	4.3431	4.2341	2.5917	4.3735
	EPI	0.8020	0.8756	0.7228	0.8445	0.7283	0.7896	0.9185	0.9210
	TCR	0.9399	0.0144	0.1866	0.9557	0.4222	0.9664	0.3730	0.0010
ThreeGorgesDam	**ENL**	4.3251	4.1982	4.1355	4.1783	4.3504	3.7859	3.7901	4.4079
	EPI	0.9858	0.9893	0.9844	0.9882	0.9790	0.9774	0.9914	0.9922
	TCR	0.0419	0.5528	0.9208	0.8499	0.9877	0.9888	0.5129	0.0041

Index (SSIM) values for the Freeway and Overpass data subsets within the UC Merced dataset, each consisting of 100 samples. The results indicate that irrespective of the input data or noise level, the proposed model consistently yields superior results in terms of PSNR and SSIM.

3.2 Real Data Experimentation

The empirical study used three different SAR data, capturing both quantitative and visual outputs for analysis. The first data is an X-band amplitude visual of

Bedfordshire, southeast England, obtained by the United Kingdom's DRA SAR system. The second data showed a synthetic aperture radar (SAR) image of an oil explosion that occurred in the Gulf of Mexico on May 4, 2015, as recorded by Pleiades satellite imagery. Finally, the third data was gathered on October 21, 2009, utilizing TerraSAR-X sensors, and it relates to the region of China around the Three Gorges Dam.

Figure 6 and 7 illustrate the visual analysis of outcomes generated by various models utilizing real SAR data as input. While examining the Bedfordshire visual, it is observed that models like PPBit, SAR-DRN, and Nb2Nb have yielded excessively smoothed outputs. Moreover, models such as SAR-BM3D, SAR-CNN, SAR-IDCNN, SAR-deSpeckNet, and SAR-CNN have made compromises on speckle removal to preserve finer details. Furthermore, upon comparing images from the OilRigExplosion, it is evident that certain models, including PPBit, SAR-BM3D, SAR-DRN, and Nb2Nb, have produced excessively smoothed results, while SAR-CNN, SAR-IDCNN, and SAR-deSpeckNet have retained some speckle information alongside minute details. In all instances, the proposed model has demonstrated promising outcomes.

In the quantitative evaluation process, three quantitative metrics were employed to assess various aspects of the processed output. These metrics include the Equivalent number of looks (ENL) [26], edge-preserving index (EPI) [4], and target-to-cluster ratio (TCR) [1]. The results of these performance metrics are detailed in Table 3. Consistent with the superiority of our proposed model observed in visual comparisons, it is evident that the model demonstrates superior metric outcomes across nearly all criteria. This indicates that the suggested model effectively maintains the original SAR image's sharp edges and intricate features while successfully mitigating undesired speckle artifacts.

4 Conclusion

In summary, our study introduces a new method for SAR despeckling that holds great promise for practical use. Through thorough experimentation, we have demonstrated the effectiveness of our approach in enhancing visual representations and evaluating image quality using real-world data from various SAR sensors. Despite a noticeable processing delay of around 300 to 350 milliseconds for images of size 256 × 256, our model's performance remains commendable, considering the computational complexity inherent in deep convolutional architectures. Our innovative model incorporates several convolutional sub-modules, each fulfilling a specific role. Additionally, our model demonstrates adaptability to different regional characteristics, validated extensively through experiments on simulated and real SAR data. The findings confirm the superiority of our proposed model over established state-of-the-art SAR despeckling techniques, highlighting its potential for practical implementation and further advancements in research within the field.

References

1. Argenti, F., Lapini, A., Bianchi, T., Alparone, L.: A tutorial on speckle reduction in synthetic aperture radar images. IEEE Geoscience and Remote Sensing Magazine **1**(3), 6–35 (2013). https://doi.org/10.1109/MGRS.2013.2277512
2. Buemi, M.E., Jacobo, J., Mejail, M.: Sar image processing using adaptive stack filter. Pattern Recognition Letters **31**(4), 307–314 (2010).https://doi.org/10.1016/j.patrec.2009.02.008, 20th SIBGRAPI: Advances in Image Processing and Computer Vision
3. Chierchia, G., Cozzolino, D., Poggi, G., Verdoliva, L.: Sar image despeckling through convolutional neural networks. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). pp. 5438–5441 (2017).https://doi.org/10.1109/IGARSS.2017.8128234
4. Chumning, H., Huadong, G., Changlin, W.: Edge preservation evaluation of digital speckle filters. In: IEEE International Geoscience and Remote Sensing Symposium. vol. 4, pp. 2471–2473 vol.4 (2002).https://doi.org/10.1109/IGARSS.2002.1026581
5. Cozzolino, D., Verdoliva, L., Scarpa, G., Poggi, G.: Nonlocal sar image despeckling by convolutional neural networks. In: IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium. pp. 5117–5120 (2019).https://doi.org/10.1109/IGARSS.2019.8897761, https://ieeexplore.ieee.org/document/8897761
6. Dalsasso, E., Yang, X., Denis, L., Tupin, F., Yang, W.: Sar image despeckling by deep neural networks: from a pre-trained model to an end-to-end training strategy. Remote Sensing **12**(16) (2020https://doi.org/10.3390/rs12162636, https://www.mdpi.com/2072-4292/12/16/2636
7. Deledalle, C.A., Denis, L., Tupin, F.: Iterative weighted maximum likelihood denoising with probabilistic patch-based weights. IEEE Trans. Image Process. **18**(12), 2661–2672 (2009). https://doi.org/10.1109/TIP.2009.2029593
8. Ferraioli, G., Pascazio, V., Vitale, S.: A novel cost function for despeckling using convolutional neural networks. In: 2019 Joint Urban Remote Sensing Event (JURSE). pp. 1–4 (2019).https://doi.org/10.1109/JURSE.2019.8809042, https://ieeexplore.ieee.org/document/8809042
9. Frost, V.S., Stiles, J.A., Shanmugan, K.S., Holtzman, J.C.: A model for radar images and its application to adaptive digital filtering of multiplicative noise. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-4**(2), 157–166 (1982).https://doi.org/10.1109/TPAMI.1982.4767223, https://ieeexplore.ieee.org/document/4767223
10. Gnanadurai, D., Sadasivam, V.: Undecimated wavelet based speckle reduction for sar images. Pattern Recogn. Lett. **26**(6), 793–800 (2005). https://doi.org/10.1016/j.patrec.2004.09.034
11. Gu, F., Zhang, H., Wang, C.: A two-component deep learning network for sar image denoising. IEEE Access **8**, 17792–17803 (2020). https://doi.org/10.1109/ACCESS.2020.2965173, https://ieeexplore.ieee.org/document/8954707
12. Horé, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: 2010 20th International Conference on Pattern Recognition. pp. 2366–2369 (2010).https://doi.org/10.1109/ICPR.2010.579
13. qi Huang, S., zhi Liu, D., qing Gao, G., jian Guo, X.: A novel method for speckle noise reduction and ship target detection in sar images. Pattern Recognition **42**(7), 1533–1542 (2009).https://doi.org/10.1016/j.patcog.2009.01.013
14. Huang, T., Li, S., Jia, X., Lu, H., Liu, J.: Neighbor2neighbor: A self-supervised framework for deep image denoising. IEEE Trans. Image Process. **31**, 4023–4038 (2022). https://doi.org/10.1109/TIP.2022.3176533

15. Iwin Thanakumar Joseph, S., Sasikala, J., Sujitha Juliet, D., Velliangiri, S.: Hybrid spatio-frequency domain global thresholding filter (hsfgtf) model for sar image enhancement. Pattern Recognition Letters **146**, 8–14 (2021).https://doi.org/10.1016/j.patrec.2021.02.023
16. Jojy, C., Nair, M.S., Subrahmanyam, G.R.K.S., R., R.: Discontinuity adaptive non-local means with importance sampling unscented kalman filter for de-speckling sar images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **6**(4), 1964–1970 (2013).https://doi.org/10.1109/JSTARS.2012.2231055, https://ieeexplore.ieee.org/document/6376246
17. Kuan, D.T., Sawchuk, A.A., Strand, T.C., Chavel, P.: Adaptive noise smoothing filter for images with signal-dependent noise. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-7**(2), 165–177 (1985).https://doi.org/10.1109/TPAMI.1985.4767641, https://ieeexplore.ieee.org/document/4767641
18. Lee, J.S.: Refined filtering of image noise using local statistics. Computer Graphics and Image Processing **15**(4), 380–389 (1981). https://doi.org/10.1016/S0146-664X(81)80018-4, https://www.sciencedirect.com/science/article/pii/S0146664X81800184
19. Lee, J.S.: Speckle analysis and smoothing of synthetic aperture radar images. Computer Graphics and Image Processing **17**(1), 24–32 (1981). https://doi.org/10.1016/S0146-664X(81)80005-6, https://www.sciencedirect.com/science/article/pii/S0146664X81800056
20. LOPES, A., NEZRY, E., TOUZI, R., LAUR, H.: Structure detection and statistical adaptive speckle filtering in sar images. International Journal of Remote Sensing **14**(9), 1735–1758 (1993). https://doi.org/10.1080/01431169308953999, https://doi.org/10.1080/01431169308953999
21. Ma, X., Wang, C., Yin, Z., Wu, P.: Sar image despeckling by noisy reference-based deep learning method. IEEE Transactions on Geoscience and Remote Sensing **58**(12), 8807–8818 (2020). https://doi.org/10.1109/TGRS.2020.2990978, https://ieeexplore.ieee.org/abstract/document/9091002
22. Maji, S.K., Thakur, R.K., Yahia, H.M.: Structure-preserving denoising of sar images using multifractal feature analysis. IEEE Geosci. Remote Sens. Lett. **17**(12), 2100–2104 (2020). https://doi.org/10.1109/LGRS.2019.2963453
23. Mullissa, A.G., Marcos, D., Tuia, D., Herold, M., Reiche, J.: despecknet: Generalizing deep learning-based sar image despeckling. IEEE Trans. Geosci. Remote Sens. **60**, 1–15 (2022). https://doi.org/10.1109/TGRS.2020.3042694
24. Pan, T., Peng, D., Yang, W., Li, H.C.: A filter for sar image despeckling using pre-trained convolutional neural network model. Remote Sensing **11**(20) (2019https://doi.org/10.3390/rs11202379, https://www.mdpi.com/2072-4292/11/20/2379
25. Parrilli, S., Poderico, M., Angelino, C.V., Verdoliva, L.: A nonlocal sar image denoising algorithm based on llmmse wavelet shrinkage. IEEE Trans. Geosci. Remote Sens. **50**(2), 606–616 (2012). https://doi.org/10.1109/TGRS.2011.2161586
26. Ren, W., Song, J., Tian, S., Zhang, X.: Estimation of the equivalent number of looks in sar images based on singular value decomposition. IEEE Geosci. Remote Sens. Lett. **12**(11), 2208–2212 (2015). https://doi.org/10.1109/LGRS.2015.2457334
27. Saha, A., Maji, S.K., Yahia, H.: A review on despeckling of the earth's surface visuals captured by synthetic aperture radar. In: Swarnkar, S., Patra, J.P., Tran, T.A., Bhushan, B., Biswas, S. (eds.) Multimedia Data Processing and Computing, pp. 1–20. CRC Press, 1st edn. (2023).https://doi.org/10.1201/9781003391272-1
28. Saha, A., K. R., A., Maji, S.K.: Pcenet: Deep sar despeckling network using parallel convolutional encoding modules. IEEE Geoscience and Remote Sensing Letters **21**, 1–5 (2024https://doi.org/10.1109/LGRS.2023.3344684

29. Sujitha, A.G., Vasuki, D.P., Deepan, A.A.: Hybrid laplacian gaussian based speckle removal in sar image processing. Journal of Medical Systems **43**(7), 222 (2019). https://doi.org/10.1007/s10916-019-1299-0, https://doi.org/10.1007/s10916-019-1299-0
30. Thakur, R.K., Maji, S.K.: Agsdnet: Attention and gradient-based sar denoising network. IEEE Geoscience and Remote Sensing Letters **19**, 1–5 (2022).https://doi.org/10.1109/LGRS.2022.3166565
31. Torres, L., Sant'Anna, S.J., da Costa Freitas, C., Frery, A.C.: Speckle reduction in polarimetric sar imagery with stochastic distances and nonlocal means. Pattern Recogn. **47**(1), 141–157 (2014). https://doi.org/10.1016/j.patcog.2013.04.001
32. Vitale, S., Ferraioli, G., Pascazio, V.: A new ratio image based cnn algorithm for sar despeckling. In: IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium. pp. 9494–9497 (2019).https://doi.org/10.1109/IGARSS.2019.8899245, https://ieeexplore.ieee.org/document/8899245
33. Wang, P., Zhang, H., Patel, V.M.: Sar image despeckling using a convolutional neural network. IEEE Signal Process. Lett. **24**(12), 1763–1767 (2017). https://doi.org/10.1109/LSP.2017.2758203
34. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861
35. Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems. p. 270-279. GIS '10, Association for Computing Machinery, New York, NY, USA (2010).https://doi.org/10.1145/1869790.1869829
36. Zhang, Q., Yuan, Q., Li, J., Yang, Z., Ma, X.: Learning a dilated residual network for sar image despeckling. Remote Sensing **10**(2) (2018).https://doi.org/10.3390/rs10020196

SalFoM: Dynamic Saliency Prediction with Video Foundation Models

Morteza Moradi[1](✉), Mohammad Moradi[1], Francesco Rundo[2],
Concetto Spampinato[1], Ali Borji[3], and Simone Palazzo[1]

[1] University of Catania, Catania, Italy
{morteza.moradi,mohammad.moradi}@phd.unict.it,
{concetto.spampinato,simone.palazzo}@unict.it
[2] STMicrolectronics, ADG Central R & D, Catania, Italy
francesco.rundo@st.com
[3] Quintic AI, San Francisco, CA, USA

Abstract. Recent advancements in video saliency prediction (VSP) have shown promising performance in emulating the human visual system, which is the primary goal of VSP. However, current state-of-the-art models employ spatio-temporal transformers trained on limited datasets, hindering their generalizability and adaptation to downstream tasks. The benefits of vision foundation models present a potential solution to improve the VSP process. However, adapting image foundation models to the video domain presents significant challenges in modeling scene dynamics and capturing temporal information. To address these challenges, and as the first initiative to design a VSP model based on video foundation models, we introduce SalFoM, a novel encoder-decoder video transformer architecture. Our model employs UnMasked Teacher (UMT) as feature extractor and presents a heterogeneous decoder which features a locality-aware spatio-temporal transformer and integrates local and global spatio-temporal information from various perspectives to produce the final saliency map. Our qualitative and quantitative experiments on the challenging VSP benchmark datasets of DHF1K, Hollywood-2 and UCF-Sports demonstrate the superiority of our proposed model in comparison with the state-of-the-art methods.

Keywords: Video Saliency Prediction · Video Foundation Model · Human Attention Prediction

1 Introduction

Video saliency prediction, which models the focus of attention of the human visual system when observing dynamic scenes, has gained increasing attention

A. Borji and S. Palazzo—Equal supervision.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78312-8_3.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 33–48, 2025.
https://doi.org/10.1007/978-3-031-78312-8_3

in recent years [36], driven by the growing demand for video content understanding and analysis across various application domains. Numerous deep learning-based strategies have been explored to enhance the accuracy and performance of these methods, aiming to reach human-level scene recognition. Among the different approaches, state-of-the-art methods that have shown the best results to date utilize spatio-temporal transformers as encoder parts. However, since these networks are often not trained on massive datasets, their generalizability and adaptability for downstream tasks are limited.

The emergence of foundation models [4] offers a solution to this fundamental challenge, as these models are trained on vast and diverse datasets, encompassing a large variability and gaining high generalizability without the need for re-training. In line with the core goal of foundation models to achieve human-like intelligence and understanding, viable solutions for video saliency prediction can leverage the capabilities of video foundation models (VFMs).

Most of current designs of VFMs are built upon robust image foundation models (IFMs), such as CLIP-ViP [33], based on CLIP [23]. Although this approach is cost-effective, as it builds on pre-trained static features, it also presents significant challenges due to the nature of IFMs, that overlook temporal and motion-related features. Therefore, such models may not be fully suitable for adaptation in various video understanding tasks, including video saliency prediction. To fill this gap, in this work we employ UnMasked Teacher (UMT) [15], a pure video foundation model that retains spatio-temporal features of video content, aiming to handle various video-centric tasks such as video-text retrieval and action recognition.

In this work, we design a novel dynamic saliency prediction model that is empowered by a video foundation model based encoder. To fully exploit the expressive power of spatio-temporal representations extracted by the encoder, we introduce a decoder architecture that is composed of three different intermediate branches, each of them reconstructing features from different perspectives. More specifically, one of the branches employs spatio-temporal transformers [17] to extract long-range spatio-temporal relationships, operating at the same resolution of the encoded features; the second branch extracts local spatio-temporal representations, gradually reducing the temporal resolution and compensating for the lack of global information by a feature fusion mechanism with the first branch; the final branch focuses instead on the spatial relations between scene elements, collapsing the temporal dimension and producing high-resolution features to guide the synthesis of the output saliency map, while at the same time incorporating information from the previous two branches. We conduct extensive quantitative and qualitative experiments on the standard VSP dataset, namely DHF1K, Hollywood-2 and UCF-Sports; our findings uncover the superiority of our model over the state-of-the-art models.

To summarize our contributions:

- We propose the first video saliency prediction model based on a pure video foundation model to capture spatio-temporal features of video content.

- We introduce a novel heterogeneous decoder network that employs locality-aware spatio-temporal attention to better process encoder features.
- We show the superiority of the performance of our model on the most challenging video saliency dataset, DHF1K, compared to the state-of-the-art VSP models.

2 Related Work

2.1 Video Saliency Prediction

Deep learning-based video saliency prediction, as explored in [26], has recently become a prominent method for modeling human gaze in dynamic scenes. The primary goal of video saliency prediction (VSP) is to mimic the human visual system and create patterns of attention allocation for video frames. One practical application of these models is predicting a driver's focus of attention in traffic scenarios [32], which is crucial for decision-making processes.

One of the most effective VSP models in the pre-transformer era is STSANet [29], which addresses the challenge of understanding long-range temporal relationships in videos. It features a 3D fully convolutional network as its backbone and is structured as a four-branch network. The network utilizes spatio-temporal self-attention (STSA) modules to capture spatio-temporal dependencies and employs attentional multi-scale fusion modules to integrate the extracted features. In the audio-visual VSP domain, TSPF-Net [7] addresses saliency modeling by designing a feature pyramid network that integrates scale, space and time. The network hierarchically decoded features at various levels of the pyramid, considering the impact of spatial and temporal features at different scales. Following a different approach, HD2S [2] constructs multiple intermediate saliency maps at various levels of abstraction, which are then integrated to produce the final saliency map. This design aims to incorporate both general and data-specific features into the saliency prediction process.

VSFT [18] is the first to employ transformer architectures in VSP, focusing on forecasting saliency for unseen future frames. Unlike the CNN-based models mentioned before, VSFT uses a self-attention mechanism to capture both short- and long-range relationships between video frames. Its decoder combines the encoded features using proposed cross-attention guidance blocks (CAGB) to capture spatio-temporal correlations. Another transformer-based model, THTD-Net [22], stands out as a lightweight solution for VSP, where most of the temporal information is processed in the decoding stage. Opting for a single-branch decoder without reducing the encoded features before decoding, the model manages to use fewer parameters, compared to attention-based or multi-branch approaches. Despite its relative simplicity, THTD-Net demonstrated performance on par with state-of-the-art solutions. Another transformer-based model, TMFI-Net [36], consists of a semantic-guided encoder and a hierarchical decoder. The encoder captures spatio-temporal features and provides semantic contextual information, integrating this information in a top-down manner using a feature pyramid structure. Before decoding, a multi-dimensional attention (MA) module is used to enhance the spatio-temporal features.

In this work, we present the first dynamic saliency prediction model that is empowered by a video foundation model for feature encoding. Moreover, unlike other approaches, where different decoding branches generally focus on varying spatial scales with a late feature fusion mechanism (e.g., [2]), our model processes and combines encoded features from different temporal viewpoints to gradually reconstruct crucial features for the ultimate saliency map prediction.

2.2 Video Foundation Models

Although foundation models for vision [1] have become increasingly prominent in recent years, the majority of published works and practical efforts have focused on image foundation models, as seen in [34,35]. The growing need to understand and analyze the pervasive and continuously-generated video content, ranging from social media and sports videos to traffic and surveillance footage, has spurred the research community to harness the power of foundation models for video-based tasks. However, the development of such models faces two significant challenges: the scarcity of a large and diverse video dataset and the substantial computational costs involved to train such models.

To overcome these obstacles, research has shifted towards the creation of video foundation models (VFMs) that build upon image foundation models (IFMs). Leveraging the strength and versatility of established large image models, such as CLIP [23], several vision models have been modified to use IFMs for addressing downstream video-related tasks. InternVideo [28] proposes a general video foundation model by introducing the concept of a unified video representation. This model utilizes UniformerV2 [14] as its encoder and is built upon CLIP. In an effort to devise an efficient approach for translating masked modeling to videos, VideoMAE [24] presents a straightforward design with reduced computational costs by employing an asymmetric encoder-decoder structure. Building upon this, VideoMAEv2 [25] adopts a dual masking strategy to enhance the original model, making it more efficient for pre-training.

Although IFMs have facilitated the development of large video models in certain respects, such models still struggle with handling the temporal dynamics that are inherent to video data. To deal with such issues, UMT, as the first attempt to design a native large video model, opted for an innovative strategy to both make the training process efficient and preserve temporal information. It utilizes CLIP-ViT as an Unmasked Teacher to train a vanilla spatio-temporal ViT from scratch for masked video modeling. UMT preserves the spatial architecture of the teacher model for processing each frame individually. Furthermore, it leverages spatio-temporal attention to facilitate interaction among the unmasked tokens. This strategy not only enables handling limited data scale for video understanding tasks but also accelerates convergence and significantly enhances the model's capability to capture temporal information across frames.

3 Methodology

The progressive development of vision foundation models, that aims at building models with high generalizability for being adapted to various downstream tasks, motivated us to design a dynamic saliency prediction model based on a video foundation model (Figure 1), namely Unmasked Teacher (UMT), as the encoder part of our network. To the best of our knowledge, this is the earliest work that incorporates a purely spatio-temporal foundation model into the video saliency prediction's workflow. Moreover, we designed a heterogeneous multi-branch decoder, consisting of both dynamic and static branches, that is intended to include spatial and temporal information in the process of generating saliency map for an input video frame from different perspectives.

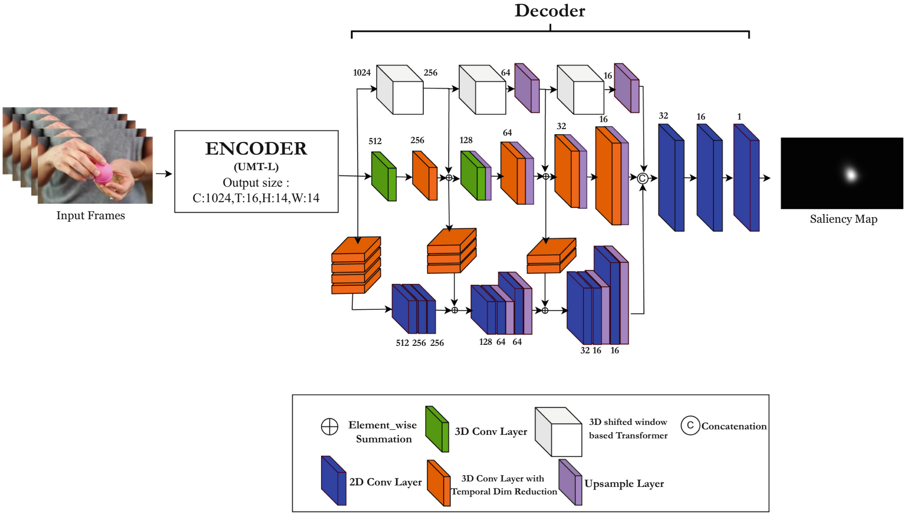


Fig. 1. Summary of the proposed dynamic saliency prediction framework. The model utilizes a video foundation model called UMT-L as its encoder, while the decoding phase comprises three different intermediate branches aimed at progressively refining and reconstructing essential features for producing the ultimate saliency map. In the end, all intermediate features are combined to generate the final saliency map.

3.1 Video Foundation Model–based Feature Encoder

In our architecture, we employ UMT [15] as our network's encoder. In contrast to other video foundation models, that are directly adapting image foundation models, UMT is trained by using CLIP-ViT [23] as Unmasked Teacher to train a vanilla spatio-temporal ViT from scratch. Training is carried out using a self-supervised teacher-student knowledge distillation approach, by masking out and predicting most of the video tokens with low semantics and aligning the unmasked tokens with a linear projection to the corresponding ones from the teacher, in order to handle the limited data scale and effectively utilize the video data. Spatio-temporal attention [3] is exploited to facilitate the interaction between all the unmasked tokens with each other. The architecture does not use temporal downsampling, which ensures that tokens can be aligned frame by frame.

In our framework, we employ the pre-trained large version of the UMT, referred to as UMT-L, in the feature encoding stage. After removing the classification head, the model can be treated as a feature extractor, which receives an input video $\mathbf{V} \in \mathbb{R}^{T \times H \times W \times 3}$ and provides features $\mathbf{F} \in \mathbb{R}^{T \times h \times w \times f}$, with $h = \frac{H}{16}$, $w = \frac{W}{16}$ and $f = 1024$. More details about model design are provided in the supplementary materials.

3.2 Multiperspective Heterogeneous Decoder

The design of the proposed multiperspective heterogeneous decoder is based on the principle of capturing and integrating diverse aspects of spatio-temporal information encoded by the VFM feature encoder. The rationale behind this approach is to ensure that the final saliency map is a comprehensive representation of the most salient features in both space and time, which is crucial for attention modeling in videos. To this aim, the proposed decoder is designed according to the following insights and principles:

- *Gradual temporal dimension reduction*: Inspired by existing strategies, that shows the benefits of keeping the temporal resolution as close as possible to the original input [22], our approach avoids abrupt loss of temporal information by gradually reducing the temporal dimension.
- *Channel dimension reduction*: We exploit the strong expressive power of features provided by the VFM encoder, and hypothesize that the objective of the decoder is to find a suitable feature analysis and interaction modality, rather than *extracting* more complex features. Hence, we encourage our model to distill the essential features by reducing the channel dimension from a high-dimensional space to a more compact representation, facilitating efficient computation and potentially improving generalization.
- *Heterogeneous spatio-temporal feature decoding*: The first intermediate branch of our decoder network (Transformer-based Complementary Feature Extraction, TCFE) is designed to capture spatio-temporal relationships and encode them into feature maps. The second branch (Dynamic Feature Decoding, DFD) focuses, instead, on maintaining temporally-rich information and extracting detailed local features. While the locality of this operation limits the discovery of useful long-range spatio-temporal patterns, it allows to gradually increase and recover the original input resolution. The third branch (Static Feature Decoding, SFD), finally, abstracts the temporal effects to focus on spatial information, recognizing that not all temporal information is equally relevant for saliency prediction.

- *Feature Fusion*: The final feature fusion stage integrates features from all branches, allowing the network to leverage the diverse perspectives captured by each branch. This integrated representation is then processed through 2D convolutional layers to produce the final saliency map.

Formally, Let $\mathbf{F} \in \mathbb{R}^{t \times h \times w \times f}$ be the set of spatio-temporal features obtained from the transformer feature extractor, with t, h, w, and f representing the temporal, height, width, and feature dimensions, respectively. The objective is to map $\mathbf{F}$ to a saliency map $\mathbf{S} \in \mathbb{R}^{H \times W}$, where H and W are the dimensions of the original frame.

Let us model the first branch of the decoder, i.e. the TCFE subnetwork, as a sequence of N layers, producing features $\boldsymbol{\Theta} = \{\boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_N}\}$. At this perspective of the mode, features $\boldsymbol{\Theta}$ are intended to extract long-range spatio-temporal relationships, operating at the same resolution as input features $\mathbf{F}$, acting only on the channel dimension. For this reason, we do not let $\boldsymbol{\Theta}$ be affected by higher-resolution features extracted by other branches, as the advantage of the latter, i.e., the higher detail, would be inevitably lost due to downsampling to the $t \times h \times w$ resolution.

The second branch of the decoder, DFD, is made up of the same number of layers as the first, and extracts features $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N\}$. This branch is dedicated to extracting local spatio-temporal features information, increasing the spatial resolution to recover details while gradually reducing the temporal dimension to retain as much of the video dynamics as possible. In order to compensate for the lack of global analysis in this branch, we integrated features from the first branch at corresponding position of the layer cascade, through proper upsampling. In detail, we compute feature $\boldsymbol{\phi}_i$, with $i > 1$, as follows:

$$\boldsymbol{\phi}_i = f_i\left(\boldsymbol{\phi}_{i-1}\right) \oplus \sigma_i\left(\boldsymbol{\theta}_i\right), \tag{1}$$

where f_i is the transformation applied at the i-th layer of the branch, and σ_i is the appropriate down-sampling function.

The third branch of the decoder, SFD, extracts features $\boldsymbol{\Gamma} = \{\gamma_1, \ldots, \gamma_N\}$; the main property of this branch is that it collapses the temporal dimension of its input features, summarizing them into a single channel and focusing instead on spatial relations between scene elements. To this aim, each layer g_i receives a temporally-collapsed version of the input frames, using a learned transformation τ_i, and produces upscaled features, with the same resolution as the corresponding ones from $\boldsymbol{\Phi}$. In particular, the input to the first layer, instead of being the set of encoder features $\mathbf{F}$, becomes $\tau_1\left(\mathbf{F}\right)$; similarly, subsequent layers perform a similar operation on feature extracted from the second branch, such that:

$$\gamma_i = g_i\left(\gamma_{i-1}\right) \oplus \tau_i\left(\boldsymbol{\phi}_i\right). \tag{2}$$

The reason for integrating features $\boldsymbol{\Phi}$ from the second branch into $\boldsymbol{\Gamma}$ is twofold: first, $\boldsymbol{\Phi}$ features internally encode global information from $\boldsymbol{\Theta}$, which ought to be taken into account to compensate for the removal of the temporal dimension; second, since the third branch focuses on spatial relations, it makes sense to provide it with the largest resolution available at that stage.

Finally, output features from each branch are processed by a late fusion layer to produce the final saliency map $\mathbf{S}$ as:

$$\mathbf{S} = o\left(\left[\sigma_n\left(\boldsymbol{\theta}_N, \boldsymbol{\phi}_N, \boldsymbol{\gamma}_N\right]\right)\right), \tag{3}$$

with o being the transformation applied by the fusion layer and $[\cdot]$ denoting the concatenation operator.

3.3 Training Objective

Given an input video sequence $\mathbf{V} \in \mathbb{R}^{T\times H\times W\times 3}$ and the ground-truth saliency map $\mathbf{G} \in \mathbb{R}^{H\times W}$ for the last frame of the clip, the objective of the model is to estimate a saliency map $\mathbf{S} \in \mathbb{R}^{H\times W}$ for the corresponding frame. Our training objective $\mathcal{L}$ is inspired from [36] and defined as follows:

$$\mathcal{L}(\mathbf{S}, \mathbf{G}) = \mathcal{L}_{\text{KL}}(\mathbf{S}, \mathbf{G}) + \mathcal{L}_{\text{CC}}(\mathbf{S}, \mathbf{G}) \tag{4}$$

The $\mathcal{L}_{\text{KL}}$ loss term treats the predicted and ground-truth saliency maps as two probability distribution, and estimates their distance by means of the Kullback-Leibler divergence:

$$\mathcal{L}_{\text{KL}}(\mathbf{S}, \mathbf{G}) = \sum_x \mathbf{G}(x) \log \frac{\mathbf{G}(x)}{\mathbf{S}(x)} \tag{5}$$

where x scans pixel locations.

The $\mathcal{L}_{\text{CC}}$ loss term computes the correlation coefficient between the saliency maps, considering them as random variables:

$$\mathcal{L}_{\text{CC}}(\mathbf{S}, \mathbf{G}) = -\frac{\text{cov}(\mathbf{S}, \mathbf{G})}{\rho(\mathbf{S})\rho(\mathbf{G})} \tag{6}$$

where $\text{cov}(\mathbf{S}, \mathbf{G})$ is the covariance of $\mathbf{S}$ and $\mathbf{G}$ and $\rho(\cdot)$ is the standard deviation operator.

4 Experiments

In this section, we present our comprehensive experiments designed to showcase the superiority of our proposed model, SalFoM. We specifically describe three benchmark datasets for Video Scene Parsing (VSP) in Section 4.1. Following that, we elaborate on the experimental setup, analyze the results, and conduct an ablation study in Sections 4.2, 4.3, and 4.4, respectively.

4.1 Datasets

We extensively evaluate our model's performance on three commonly used benchmark datasets: DHF1K [27], UCF-Sports [19], and Hollywood-2 [20].

DHF1K, the largest eye-tracking dataset for dynamic fixation prediction, consists of 1,000 annotated videos divided into train (600), validation (100), and test (300) sets. Ground truths for the test set are not released, so quantitative evaluations are provided by the dataset's curators. The Hollywood-2 dataset [20] contains 1,707 video clips extracted from 69 Hollywood movies, grouped by 12 action categories. Unlike the DHF1K dataset, this dataset employs a task-driven viewing approach for video annotation. The annotations were gathered from three different perspectives: context recognition by four observers, free viewing by three observers, and action recognition by twelve observers. The UCF-Sports dataset [19], which is derived from the UCF sports action dataset, consists of 150 videos spanning nine sports classes. It is annotated using the same task-driven viewing methodology as the Hollywood-2 dataset. Following the approach in [30], we utilized 103 videos for training and 47 videos for testing.

4.2 Experimental Setup

When training on DHF1K, we initialize the encoder of our network from the pretrained weights from UMT-L/16 on the Kinetics-400 dataset [11]; when training on Hollywood-2 and UCF-sports, instead, we initialize the encoder using weights obtained after training on DHF1K. In both cases, encoder parameters are fine-tuned at training time. We train our model using a batch size of 1, and employing the Adam optimizer [12] for gradient descent, with an initial learning rate of 10^{-5}.

At each training iteration, the network processes 16 consecutive video frames with a spatial resolution of 224×224, and predicts the saliency map for the last frame of the video sequence. We implement early stopping based on the performance on the target dataset's validation set.

At inference time, we generate saliency maps for all video frames using a sliding window approach, as utilized by [21]. To ensure sufficient temporal context for the initial frames of a clip, we reverse the order of the frames. To assess our network's performance, we used three location-based metrics, Shuffled AUC (S-AUC), AUC-Judd (AUC-J), and Normalized Scanpath Saliency (NSS), as well as two distribution-based metrics, Linear Correlation Coefficient (CC) and Similarity Metric (SIM) [5].

4.3 Result Analysis

We quantitatively compare the performance of our model with state-of-the-art VSP models that have demonstrated the best results on the DHF1K benchmark[1]: SalEMA [16], STRA-Net [13], TASED-Net [21], SalSAC [31], UNISAL [9], ViNet [10], HD2S [2], VSFT [10], TSFP-Net [8], STSANet [30], TMFI-Net [36], and THTD-Net [22]. The results presented in Table 1 demonstrate that on the DHF1K dataset, SalFoM outperforms other state-of-the-art models across almost all evaluation metrics (the only exception being SalEMA on SIM, although it

[1] https://mmcheng.net/videosal.

Table 1. Quantitative comparison of different models on DHF1K dataset. The top score in each metric is in bold.

Models	**DHF1K**				
	AUC-J	SIM	S-AUC	CC	NSS
SalEMA	0.890	**0.466**	0.667	0.449	2.574
STRA-Net	0.895	0.355	0.663	0.458	2.558
TASED-Net	0.895	0.361	0.712	0.470	2.667
SalSAC	0.896	0.357	0.697	0.479	2.673
UNISAL	0.901	0.390	0.691	0.490	2.776
ViNet	0.908	0.381	0.729	0.511	2.872
HD2S	0.908	0.406	0.700	0.503	2.812
VSFT	0.910	0.410	0.720	0.518	2.977
TSFP-Net	0.911	0.392	0.723	0.516	2.966
STSANet	0.912	0.382	0.722	0.528	3.010
THTD-Net	0.915	0.406	0.729	0.547	3.138
TMFI-Net	0.915	0.406	0.730	0.546	3.146
SalFoM (Ours)	**0.922**	0.420	**0.735**	**0.569**	**3.353**

performs significantly worse on the other metrics), notably surpassing the top-ranked TMFI-Net model. Moreover, qualitative examples reported in Figure 2 confirm that the proposed approach, when compared with state-of-the-art VSP models, aligns better with ground-truth annotations, focusing on single points of interest, whereas other methods distribute their attention towards less salient portions of the video. Finally, as proposed by Bylinskii et al. [6], evaluating the capability and performance of saliency prediction models in identifying high-level concepts, such as face recognition, in visual scenes can serve as an indicator of their effectiveness. We qualitatively perform such analysis and report some samples in Figure 3, again showing the alignment between our model's predictions and the ground truth.

Results on Hollywood-2 and UCF-Sports are presented in Table 2. We can see that our model achieves comparable results to state-of-the-art methods on the Hollywood-2 dataset, while its performance drops in certain metrics when assessed on the UCF-Sports dataset. This pattern of performance drop is not unique to our model but is also observed in other leading methods, and its causes can be led to certain characteristics of those datasets [2], including task-driven observations and center bias. As illustrated in Figure 4, our model, SalFoM, effectively detects the human subject, representing the most salient region, yet it does not precisely match the ground truth. This discrepancy arises because the annotations accompanying the Hollywood-2 and UCF-Sports datasets primarily emphasize actions rather than salient regions. Additionally, the lower performance on these datasets can be attributed to the scarcity of diverse spatio-temporal data, particularly notable in Hollywood-2, which comprises numerous

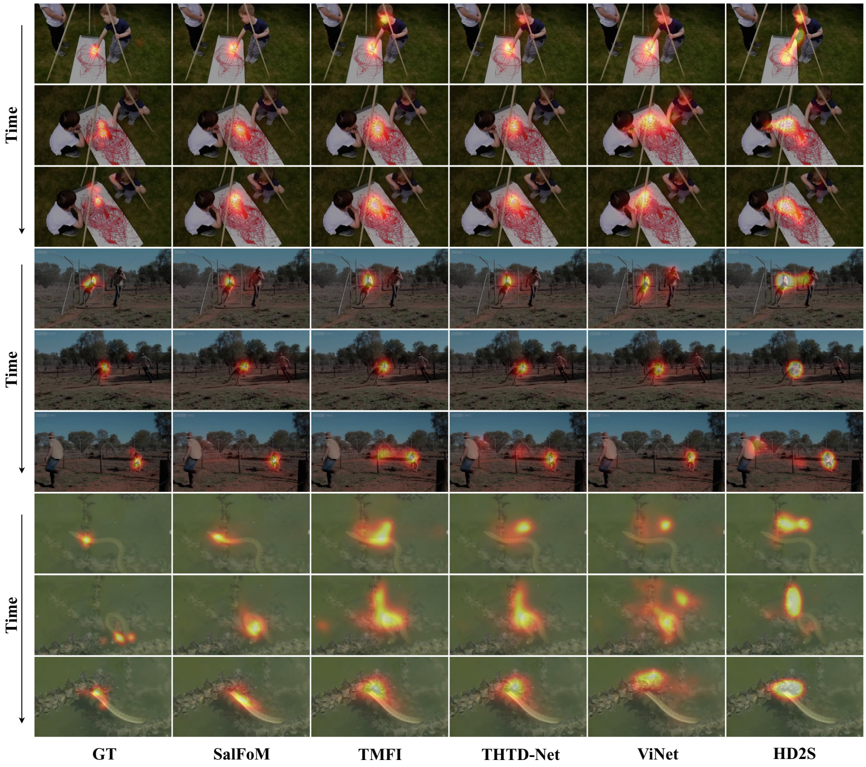


Fig. 2. Qualitative comparison of the performance of different VSP models on DHF1K.

short videos. An additional challenge arises from the relatively small size of the UCF-Sports dataset when compared to DHF1K and Hollywood-2. In our case, the limited size of the dataset contributes to overfitting of the model and negatively impacts its ability to generalize. Additional model evaluations on DHF1K are reported in the supplementary materials.

4.4 Ablation Study

In this section, we assess the impact of various components within our model by designing and evaluating different model variants on DHF1K, utilizing the validation set as the test set. The results are reported in Table 3. We first explore the influence of the encoder employed, by replacing UMT-L/16 with a Video Swin Transformer, similarly to TMFI-Net, and with variants of UMT that process 8 frames instead of 16. We accordingly configure our model's decoder parameters to align with the different temporal sizes of the encoded features. Results show that non-VFM-based encoders fail to yield satisfactory features, and that reducing the input frames of a VFM degrades performance.

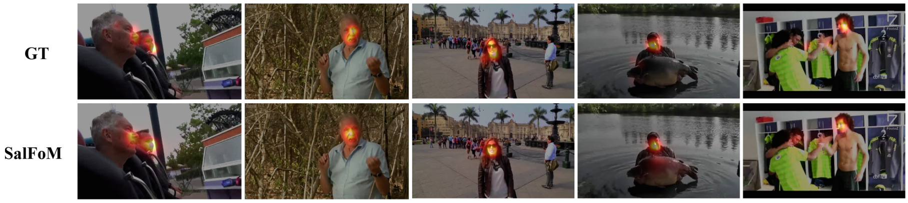


Fig. 3. Qualitative evaluation of the performance of SalFoM for predicting saliency of faces against ground truths, on DHF1K.

Table 2. Quantitative comparison of different models on Hollywood-2 and UCF-Sports datasets. The top score in each metric is in bold.

Models	**Hollywood-2**				**UCF-Sports**			
	AUC-J	SIM	CC	NSS	AUC-J	SIM	CC	NSS
SalEMA	0.919	0.487	0.613	3.186	0.906	0.431	0.544	2.638
STRA-Net	0.923	0.536	0.662	3.478	0.910	0.479	0.593	3.018
TASED-Net	0.918	0.507	0.646	3.302	0.899	0.469	0.582	2.920
SalSAC	0.931	0.529	0.670	3.356	0.926	0.534	0.671	3.523
UNISAL	0.934	0.542	0.673	3.901	0.918	0.523	0.644	3.381
ViNet	0.930	0.550	0.693	3.73	0.924	0.522	0.673	3.62
HD2S	0.936	0.551	0.670	3.352	0.904	0.507	0.604	3.114
VSFT	0.936	0.577	0.703	3.916	-	-	-	-
TSFP-Net	0.936	0.571	0.711	3.910	0.923	0.561	0.685	3.698
STSANet	0.938	0.579	0.721	3.927	**0.936**	0.560	0.705	**3.908**
THTD-Net	0.939	0.585	0.726	3.965	0.933	**0.565**	**0.711**	3.840
TMFI-Net	**0.940**	**0.607**	**0.739**	**4.095**	**0.936**	**0.565**	0.707	3.863
SalFoM (Ours)	0.935	0.583	0.709	3.902	0.928	0.516	0.631	3.543

Table 3. Ablation study: assessing the impact of SalFoM model components on validation set of DHF1K.

Model	**CC**	**NSS**	**SIM**	**AUC-J**
Encoder: VidSwin-S	0.512	2.917	0.379	0.916
Encoder: UMT-L (8 Frames)	0.552	3.169	0.418	0.924
Encoder: UMT-B (8 Frames)	0.527	2.935	0.400	0.915
Decoder: single TCFE branch	0.560	3.245	0.433	0.926
Decoder: single DFD branch	0.564	3.288	0.425	0.926
Decoder: single SFD branch	0.560	3.220	0.408	0.926
Decoder: SFD + DFD branches	0.564	3.291	0.429	0.927
Decoder: SFD + TCFE branches	0.563	3.240	0.415	0.918
Decoder: DFD + TCFE branches	0.564	3.294	0.433	0.927
SalFoM (ours)	**0.565**	**3.299**	**0.436**	**0.928**

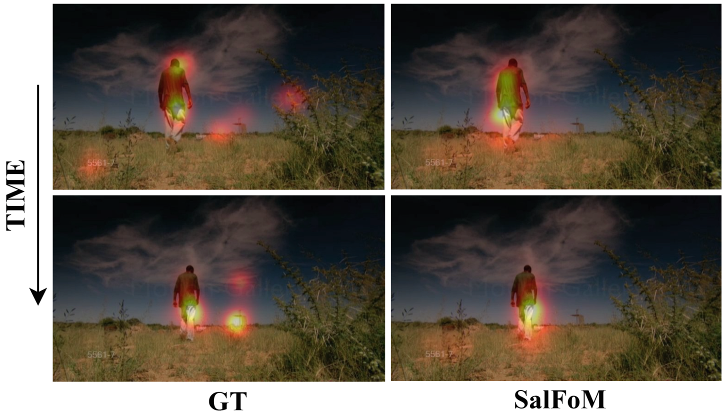


Fig. 4. A sample of failure case on UCF-Sports dataset, due to the task-driven annotation methodology.

After showing the superiority of the UMT-based feature encoder, we carry out additional experiments to demonstrate the importance of the proposed decoder strategy. In particular, we evaluate the performance of variants of our decoder which use either a single or a combination of two (out of three) branches. While all configurations are able to achieve satisfactory results, the full combination of all three decoding branches yields the best values for the employed metrics.

5 Conclusion

In this work, we present SalFoM, a video saliency prediction model that incorporates UMT, an innovative video foundation model, into our network architecture. This integration enables the model to effectively capture both temporal (scene dynamics) and spatial (object-related) information. For the decoder component of SalFoM, we introduce a three-branch structure that includes a locality-aware spatio-temporal transformer branch, a branch based on 3D convolutional layers, and another based on 2D convolutional layers. This configuration is designed to merge local and global spatio-temporal signals from diverse perspectives to generate the final saliency map. Our experiments demonstrate that SalFoM outperforms existing VSP models on several standard evaluation metrics. Future work could investigate the use of knowledge distillation to leverage SalFoM's capabilities in designing a more lightweight model for applications such as driver attention modeling.

Acknowledgments. The work of Simone Palazzo, who has contributed to the design of the model architecture and the evaluation procedure, has been supported by MUR in the framework of PNRR Mission 4, Component 2, Investment 1.1, PRIN, under project RESILIENT, CUP E53D23016360001.

References

1. Awais, M., Naseer, M., Khan, S., Anwer, R.M., Cholakkal, H., Shah, M., Yang, M.H., Khan, F.S.: Foundational models defining a new era in vision: A survey and outlook. arXiv preprint arXiv:2307.13721 (2023)
2. Bellitto, G., Proietto Salanitri, F., Palazzo, S., Rundo, F., Giordano, D., Spampinato, C.: Hierarchical domain-adapted feature learning for video saliency prediction. Int. J. Comput. Vision **129**, 3216–3232 (2021)
3. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: ICML. vol. 2 (2021)
4. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)
5. Bylinskii, Z., Judd, T., Oliva, A., Torralba, A., Durand, F.: What do different evaluation metrics tell us about saliency models? IEEE Trans. Pattern Anal. Mach. Intell. **41**(3), 740–757 (2018)
6. Bylinskii, Z., Recasens, A., Borji, A., Oliva, A., Torralba, A., Durand, F.: Where Should Saliency Models Look Next? In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 809–824. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_49
7. Chang, Q., Zhu, S.: Temporal-spatial feature pyramid for video saliency detection. arXiv preprint arXiv:2105.04213 (2021)
8. Chang, Q., Zhu, S.: Temporal-Spatial Feature Pyramid for Video Saliency Detection. arXiv e-prints arXiv:2105.04213 (May 2021). 10.48550/arXiv.2105.04213
9. Droste, R., Jiao, J., Noble, J.A.: Unified Image and Video Saliency Modeling. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12350, pp. 419–435. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58558-7_25
10. Jain, S., Yarlagadda, P., Jyoti, S., Karthik, S., Subramanian, R., Gandhi, V.: Vinet: Pushing the limits of visual modality for audio-visual saliency prediction. In: IROS 2021. pp. 3520–3527 (2021)
11. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset. CoRR **abs/1705.06950** (2017)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Lai, Q., Wang, W., Sun, H., Shen, J.: Video saliency prediction using spatiotemporal residual attentive networks. IEEE Trans. Image Process. **29**, 1113–1126 (2019)
14. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Wang, L., Qiao, Y.: Uniformerv2: Unlocking the potential of image vits for video understanding. In: CVPR 2023. pp. 1632–1643 (2023)
15. Li, K., Wang, Y., Li, Y., Wang, Y., He, Y., Wang, L., Qiao, Y.: Unmasked teacher: Towards training-efficient video foundation models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 19948–19960 (October 2023)
16. Linardos, P., Mohedano, E., Nieto, J.J., O'Connor, N.E., Giro-i Nieto, X., McGuinness, K.: Simple vs complex temporal recurrences for video saliency prediction. arXiv preprint arXiv:1907.01869 (2019)
17. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3202–3211 (2022)

18. Ma, C., Sun, H., Rao, Y., Zhou, J., Lu, J.: Video saliency forecasting transformer. IEEE Trans. Circuits Syst. Video Technol. **32**(10), 6850–6862 (2022). https://doi.org/10.1109/TCSVT.2022.3172971
19. Mathe, S., Sminchisescu, C.: Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(7), 1408–1424 (2014)
20. Mathe, S., Sminchisescu, C.: Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(7), 1408–1424 (2014)
21. Min, K., Corso, J.J.: Tased-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection. In: ICCV 2019 (2019)
22. Moradi., M., Palazzo., S., Spampinato., C.: Transformer-based video saliency prediction with high temporal dimension decoding. In: VISAPP 2024. SCITEPRESS (2024). https://doi.org/10.5220/0012422800003660
23. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML 2021. pp. 8748–8763 (2021)
24. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. NeurIPS 2022 (2022)
25. Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., Qiao, Y.: Videomae v2: Scaling video masked autoencoders with dual masking. In: CVPR 2023. pp. 14549–14560 (2023)
26. Wang, W., Shen, J., Xie, J., Cheng, M.M., Ling, H., Borji, A.: Revisiting video saliency prediction in the deep learning era. IEEE Trans. Pattern Anal. Mach. Intell. **43**(1), 220–237 (2019)
27. Wang, W., Shen, J., Xie, J., Cheng, M.M., Ling, H., Borji, A.: Revisiting video saliency prediction in the deep learning era. IEEE Trans. Pattern Anal. Mach. Intell. **43**(1), 220–237 (2019)
28. Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., et al.: Internvideo: General video foundation models via generative and discriminative learning. arXiv preprint arXiv:2212.03191 (2022)
29. Wang, Z., Liu, Z., Li, G., Wang, Y., Zhang, T., Xu, L., Wang, J.: Spatio-temporal self-attention network for video saliency prediction. IEEE Transactions on Multimedia (2021)
30. Wang, Z., Liu, Z., Li, G., Wang, Y., Zhang, T., Xu, L., Wang, J.: Spatio-temporal self-attention network for video saliency prediction. IEEE Transactions on Multimedia (2021)
31. Wu, X., Wu, Z., Zhang, J., Ju, L., Wang, S.: Salsac: A video saliency prediction model with shuffled attentions and correlation-based convlstm. In: AAAI 2020. pp. 12410–12417 (2020)
32. Xia, Y., Zhang, D., Kim, J., Nakayama, K., Zipser, K., Whitney, D.: Predicting driver attention in critical situations. In: ACCV 2018. pp. 658–674 (2019)
33. Xue, H., Sun, Y., Liu, B., Fu, J., Song, R., Li, H., Luo, J.: Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. arXiv preprint arXiv:2209.06430 (2022)
34. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., Wu, Y.: Coca: Contrastive captioners are image-text foundation models. arXiv preprint arXiv:2205.01917 (2022)
35. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al.: Florence: A new foundation model for computer vision. arXiv preprint arXiv:2111.11432 (2021)

36. Zhou, X., Wu, S., Shi, R., Zheng, B., Wang, S., Yin, H., Zhang, J., Yan, C.: Transformer-based multi-scale feature integration network for video saliency prediction. IEEE Trans. Circuits Syst. Video Technol. **33**(12), 7696–7707 (2023)

Velocity Field-Based Surveillance Video Frame Deletion Detection Using Siamese Network

Yang Su[1,2](✉), ShunQuan Tan[2,3], and Jiwu Huang[1,2]

[1] College of Information Engineering, Shenzhen University, Shenzhen, China
2205433002@email.szu.edu.cn

[2] Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen Key Laboratory of Media SecurityGuangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China

[3] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

Abstract. Surveillance videos play a crucial role in providing evidences. However, the deletion of even a few frames can significantly impact the interpretation of events, while the deletion can be performed easily using video editing softwares without leaving visual traces. This paper introduces a novel method to detect video frame deletion based on velocity field characteristics. The main idea is to convert long videos into a feature sequence, which appears as a sequence containing outlier values for videos with frame deletions, and as a sequence without outlier values for original videos. The proposed approach employs a siamese network to distinguish frame deletions between adjacent frames. A video can be transformed into a feature sequence through the siamese network, and the resulting sequence is fed into a binary classifier for final classification. Experimental results demonstrate the effectiveness of our proposed approach, even for longer videos (e.g., 2000 frames) with minimal frame deletions (e.g., 1 frame).

Keywords: Passive Forensics · Siamese Neural Network · Velocity Field · Inter-Frame Forgery Detection

1 Introduction

As video editing software becomes more widespread, ordinary individuals can now manipulate video content more easily. This trend poses a significant threat to surveillance videos, which serve as essential evidences in event investigations. Malicious attackers can effortlessly delete crucial frames from surveillance video in an attempt to conceal the occurrence of events[5,6,10,17]. For instance, deleting all frames that include a specific individual from a video could erroneously indicate their non-presence in the depicted events, as shown in Figure 1. Such

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 49–61, 2025.
https://doi.org/10.1007/978-3-031-78312-8_4

straightforward tampering actions can lead to severe consequences, compromising the integrity of evidence and the traceability of events. As a result, interests in research on detection methods of video frame deletion have been steadily increasing.

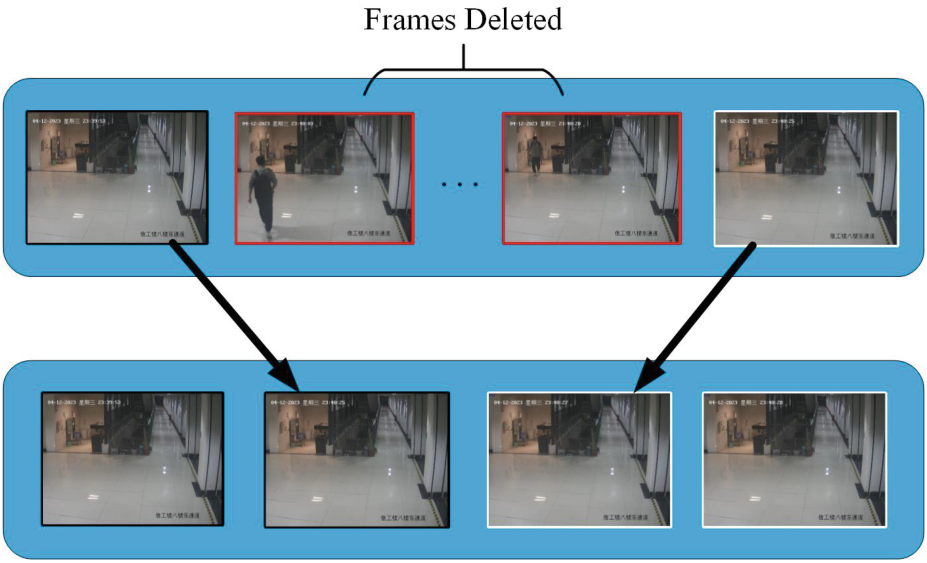


Fig. 1. Example diagram of video frame deletion, the red box represents the deleted video frames.

Video forensic methods can be categorized into active and passive approaches. Active methods require prior information added before video propagation, such as watermarks or digital signatures[9,18], while passive methods rely on discovering tampering traces without the need for prior information. Since general videos typically do not contain prior information, passive methods are more suitable for practical applications[13,15]. Some detection algorithms of frame-deletion videos had been proposed based on manual features, which generally rely on the continuity of certain features, such as velocity field[20], motion residuals[8], and so on. Some researchers[1,4] have proposed that the traditional methods often yield features vulnerable to post-processing, resulting in a decrease in detection accuracy. On the contrary, deep learning networks can extract complex, high-dimensional features that are robust and effective in representing the necessary information.

Long et al. assigned a score to a test video by passing it through the pre-trained network to detect frame deletions[14]. Bakas et al. also utilized a deep learning network for detecting frame-deleted videos, incorporating improvements by adding a pixel-wise difference layer before the deep learning network[2]. Fadl et al. integrated the spatial and temporal information of video frames into a single image, and subsequently utilized a pre-trained deep learning network along with the Structural Similarity Index to assess whether frame deletions exist in the video[4]. Gong et al. utilized enhanced residual feature images to separately train two deep learning networks, and subsequently detected whether frame deletions exist in the video[11].

Despite these methods demonstrating competitive results in various benchmark tests, current deep learning-based approaches still face certain limitations. Firstly, the majority of existing methods typically involve sequentially detecting video segments, such as analyzing a dozen frames at a time. For longer videos, it is necessary to divide the video into several segments for detection. Despite the high accuracy of detection for each video segment, if any segment is incorrectly labeled as a deleted-frame video, the entire original video will be labeled as a deleted-frame video. Secondly, these methods typically require a minimum of around 10 deleted frames for detection, rendering them impractical for scenarios involving only one frame deletion-an extreme but plausible occurrence. These limitations hinder their broader application. Therefore, there is an urgent need for further research to develop more accurate detection methods under these circumstances.

To address the aforementioned issues, we propose a novel method for detecting frame deletions in videos based on the continuity feature of the velocity field[20]. It is worth noting that, in [20], only the feature of the mutation of the sum of velocity field components was utilized. In contrast, this paper employs a deep learning network to extract more features. Our approach transforms the velocity field into images, referred to as velocity field images. A siamese network[3] is employed to extract similarity features between consecutive frames and frames with deletions from the velocity field images. This allows us to extract more velocity field features and transform a long video into a sequence containing temporal information. Subsequently, we employ a binary classifier to classify these sequences, enabling the detection of frame deletions in long videos. The main contributions of our work can be summarized as follows:

- Leveraging a siamese network, we transform videos into similarity matrices of adjacent frame velocity field images. This enables the system to detect the extreme case of long videos with a minimum of one frame deletion (e.g., deleting one frame in a 2000-frame video).
- By only training on original videos and videos with one frame deleted, and testing on videos with different amounts of deleted frames, we ensure convenient access to the training data.

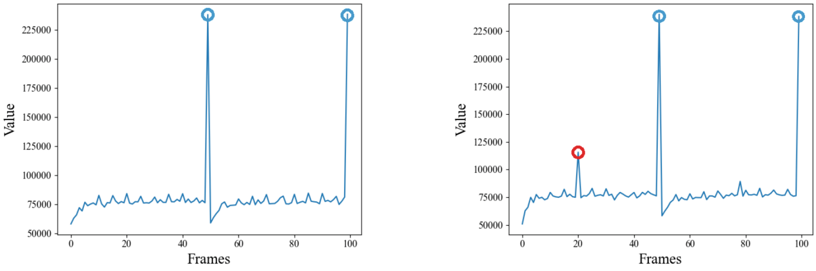
2 Preliminaries

In the context of videos, a velocity field refers to the speed and direction of pixel motion between different frames. It can be obtained by comparing adjacent video frames and estimating the displacement caused by their time separation. Any inter-frame operations, such as frame deletion and repetition, can magnify this displacement[20]. We use the optical flow field to represent the velocity field, and its calculation method is given below[7].

$$P_p(x, y) \sim P_n(x + F(x, y)[0], y + F(x, y)[1]) \tag{1}$$

This formula represents predicting the pixel values in the subsequent image based on the preceding image. $P_p(x,\ y)$ represents the pixel value at coordinates $(x,\ y)$ in the previous frame image. $P_n(x\ +\ F(x,\ y)[0],\ x\ +\ F(x,\ y)[1])$ represents the pixel value at the predicted position in the current frame image obtained through the optical flow field. $F(x,\ y)$ represents the optical flow field at coordinates $(x,\ y)$.

Figure 2 illustrates the trends in the sum of the magnitude of the velocity field for both original and frame-deleted videos(one frame was deleted after frame 20). The example videos in the figures were encoded using H.264 with a GOP (Group of Pictures) size of 50. It can be observed that in the original video, significant increases in magnitude occur only when I-frames (key frames) appear. In contrast, the frame-deleted video exhibits significant increases in magnitude not only at I-frames but also around frame 20. This behavior is due to the fact that I-frames are fully encoded frames, while frames after an I-frame are predicted frames, which may have significant differences. And one frame was deleted after frame 20, leading to a sudden change.



(a) (b)

Fig. 2. The trends in the sum of the magnitudes of the horizontal component of the velocity field for both original and frame-deleted videos((a) represents the original trend, while (b) represents the trend with one frame deleted. The blue circles represent I-frame in the velocity field, while the red circles indicate the presence of deleted frames).

3 Proposed Method

The schematic diagram of the proposed video frame deletion detection system is illustrated in Figure 3. It can be observed that the system is divided into two steps. Step 1 serves to train a siamese network using velocity field images, this step aims to enable the siamese network to distinguish between consecutive frames and frames with deletions. Step 2 employs the pre-trained siamese network to transform videos into similarity feature matrices. The two inputs of the siamese network are temporally adjacent velocity field images. After processing all frames of the entire video, a similarity matrix of velocity field images containing temporal information is obtained. Then a binary classifier is utilized for decision-making.

3.1 Preprocessing

The approach in [20] solely utilized the amplitude information of the velocity field. However, the velocity field encompasses not only amplitude but also directional information. Transforming the velocity field into an image allows the preservation of both of these aspects simultaneously. We save the obtained velocity field matrices[1] as images, as shown in Figure 4. The left side of the figure displays frame image, while the right side displays velocity field image.

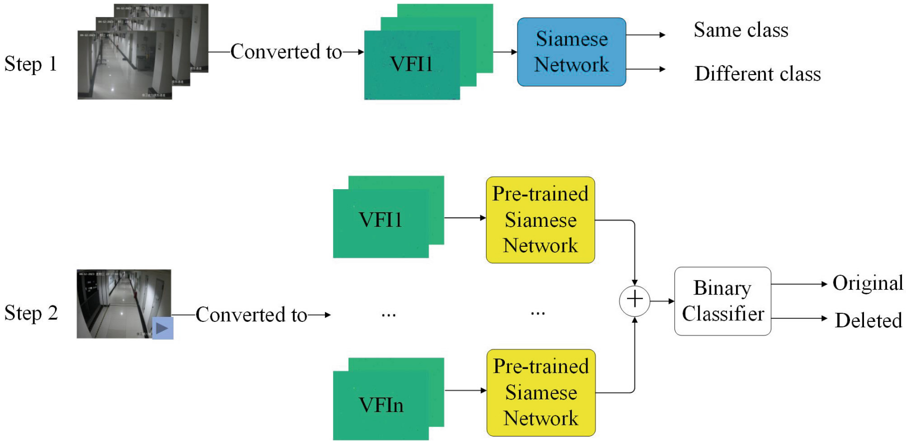


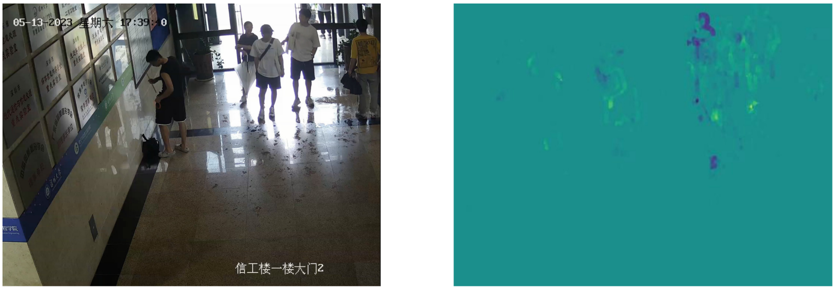
Fig. 3. The schematic diagram of the proposed video frame deletion detection system(*VFIn* represents the velocity field image).

3.2 Feature extraction

In this paper, we adopt VGG16[19] with pre-trained weights[2] on ImageNet to construct the siamese network, although theoretically, other types of deep learning networks can also be used.

[1] 'calcOpticalFlowFarneback' function from the cv2 library in the Python environment used to get the velocity field matrices.

[2] Download link: https://download.pytorch.org/models/vgg16-397923af.pth



(a) Frame image

(b) Velocity filed image

Fig. 4. Frame image and velocity filed image

Videos that have undergone frame deletion inevitably need to be re-compressed. Therefore, consecutive frames can be categorized into two situations: consecutive frames in the original video and consecutive frames in the video with frame deletions, where the latter has undergone recompression. So the velocity field images are categorized into three categories: original consecutive frames, re-compressed consecutive frames, and non-consecutive frames(frames deletion occurs between adjacent frames).

The number of frame deletion affects the degree of mutation in the velocity field. Specifically, the more frames deleted, the greater the mutation in the velocity field; while the fewer frames deleted, the smaller the degree of mutation. However, the mutation does not disappear simply because the number of deleted frames is small. We train the siamese network using videos with one deleted frame and original videos. If the system can distinguish between videos with one deleted frame and the original videos, it can similarly differentiate videos with a larger number of deleted frames. This approach makes it easier to collect training data. Additionally, it makes the system more practical because in real-world scenarios, frame deletions can occur with any positive value.

For siamese network, it is common practice to train the network with pairs of images from the same category and different categories. This involves pairing original consecutive frames, re-compressed consecutive frames, and non-consecutive frames, training them as belonging to the same class or different classes. However, in our approach, non-consecutive frames are not treated as the same category during training. In practical situations, there is a lack of continuity in velocity field features between non-consecutive frames, because it is improbable for two segments of frame-deleted videos to be consecutive.

During the training of the same category, two inputs of the siamese network are velocity field images that are temporally adjacent. For training different categories, re-compressed consecutive frames are paired with either non-consecutive frames(one frame was deleted between two consecutive frames) or original video consecutive frames as the two inputs of the siamese network, the inputs are randomly sampled from the dataset. The adoption of this training approach is based on the fact that temporally adjacent original video frames are always the most similar. However, due to varying numbers of deleted frames, the differences between original video frames and frames with deletions are diverse.

This approach allows us to effectively differentiate between original consecutive frames, re-compressed consecutive frames, and non-consecutive frames. After training the network, we use the output from the second-to-last layer of the network as the similarity feature of velocity field images, as shown in Figure 5. We extract similarity features frame by frame from the long video, resulting in a feature sequence.

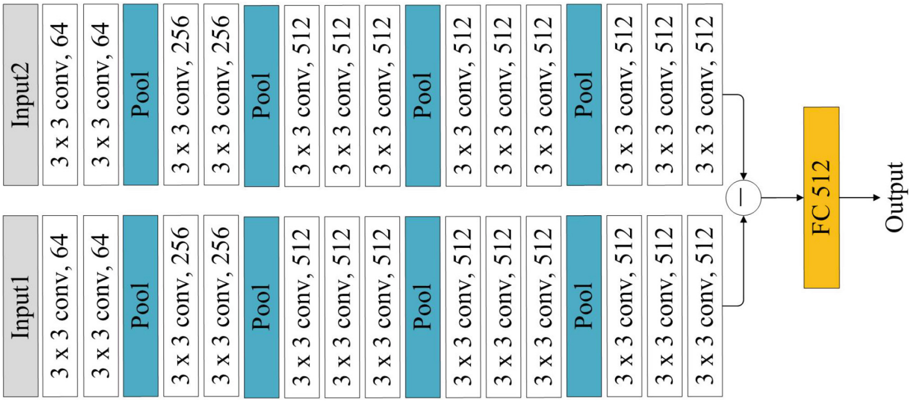
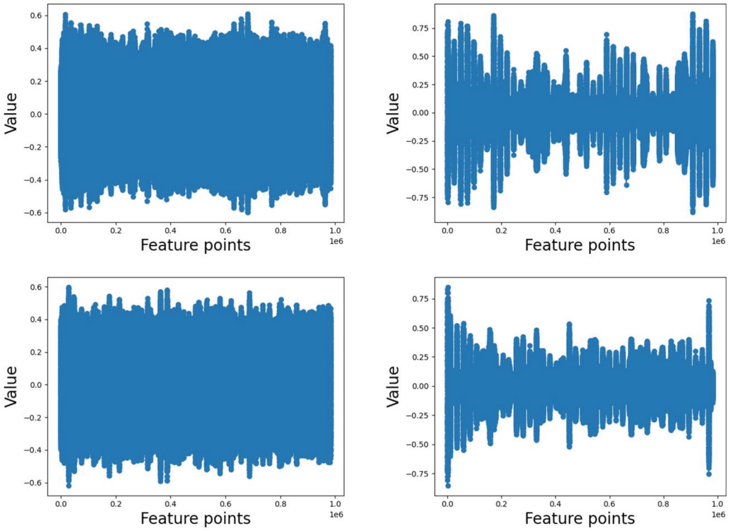


Fig. 5. The architecture of the siamese network designed to extract similarity features from velocity field images.

We have drawn the feature sequence into an image, as shown in Figure 6. From a visual perspective, it can be seen that there is a clear difference between the original video and the frame-deleted video, with the former's feature sequence fluctuating relatively smoothly and the latter's fluctuating greatly.

3.3 Classification

After the aforementioned operations, the video is transformed into a time sequence containing similarity features of the velocity field images. These time series are divided into two categories: normal time sequences generated from the original videos and abnormal time sequences containing outlier values generated from the frame-deleted videos. In theory, any binary classifier can be utilized to determine the presence of frame deletion in a video. This paper employs a random forest classifier. At each node split in each decision tree, the random forest selects a subset of features from the sequence for splitting. This helps ensure that different decision trees focus on different aspects of the sequence, enhancing the model's generalization ability. When classifying a new sequence, each decision tree independently classifies the sequence. The final classification result is determined based on the majority vote of all trees, i.e., adopting the class with the most votes, reducing the risk of overfitting[16].



(a)

(b)

Fig. 6. Illustrative examples of feature sequences, where (a) depicts the feature sequence of the original video, and (b) illustrates the feature sequence of the video with a deleted frame.

4 Experiment

4.1 Video Dataset

As far as we know, there is currently no dedicated dataset available for inter-frame forensics in videos. We created our own dataset using surveillance videos from our laboratory. There are three types of surveillance cameras used: Hikvision DS-8616N-I8(referred to as H1), Hikvision DS-2CD1301D-I(referred to as H2) and Dahua DH-HAV-HDW1120E(referred to as D).

H1 has the most cameras, with 6 cameras capturing 6 different scenes, while H2 has 2 cameras capturing 2 scenes, and D has only one camera capturing one scene. We segmented the videos into multiple video clips for training and testing based on these different scenes. Their parameters are listed in Table 1, and the scenes they captured are shown in the Figure 7.

Table 1. Parameters of the surveillance videos(VCM stands for Video Camera Model.)

VCM	Encoding Format	Resolution	Frame Rate	GOP	Number of Cameras
H1	H.264	2048*1536	25	50	6
H2	H.264	1280*720	25	50	2
D	H.264	1280*720	25	50	1

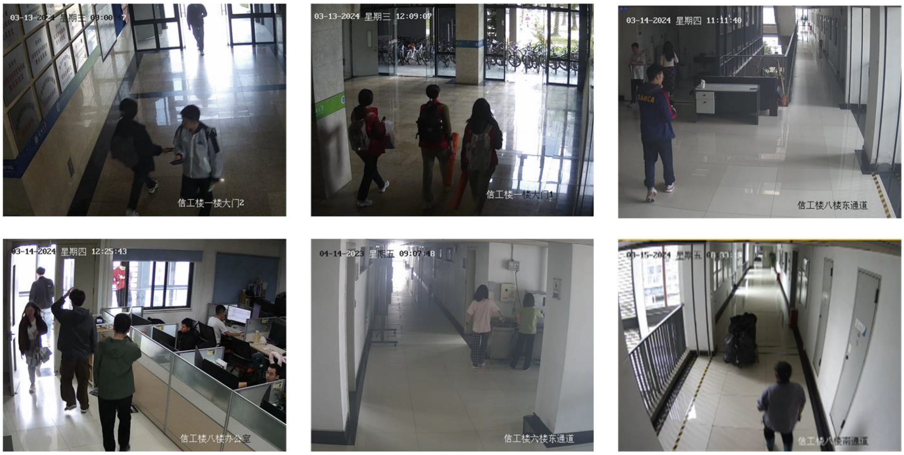

信工楼一楼大门2
03-13-2024 星期三 12:09:07
信工楼一楼大门1
03-14-2024 星期四 11:11:40
信工楼八楼东通道
03-14-2024 星期四 12:25:43
信工楼八楼办公室
信工楼六楼东通道

Fig. 7. Video Examples.

We trained the siamese network using three scenes captured by H1. The random forest classifier was trained and tested using videos obtained from the same brand and model. Specifically:

- For the H1 camera, we used three scenes for training and additional three scenes for testing.
- For the H2 camera, we used one scene for training and additional scene for testing.
- For the D camera, we trained and tested on two video segments from the same scene but with a one-month time difference.

The deleted-frame videos were re-encoded using the same encoding format and parameters as the original videos[3]. The number of velocity field images used for training the siamese network were as follows: 20,000 frames of original consecutive frames, 19,990 frames of re-compressed consecutive frames, and 19,590 frames of non-consecutive frames (frames with one deleted in between). The number of video segments used for training and testing random forest classifier is specified in Table 2, where each video segment contains 2000 frames.

The training parameters for the siamese network are as follows: input size is 512x512, learning rate is 1e-2, batch size is 8, the optimizer is SGD, and trained for 80 epochs on a Tesla P100 GPU with 16GB of VRAM.

We use accuracy to measure the final detection performance, which is the proportion of the correctly detected instances to the total number of videos.

4.2 Performance

Training the siamese network using H1 videos and training random forest classifier with H1, H2 and D videos, and testing with videos of the corresponding

[3] Utilizing the built-in ffmpeg tool in the Python environment.

Table 2. The detail of video segments used for training and testing the random forest classifier

Video Camera Model		H1	H2	D
Number of video for training	Original	30	20	20
	Delete 1	30	20	20
Number of videos for testing	Original	30	20	20
	Delete 1	30	20	20
	Delete 3	30	20	20
	Delete 5	30	20	20
	Delete 10	30	20	20

brand/model, the performance is presented in Table 3. From Table 3, it can be observed that the system can effectively distinguish between original videos and frame-deleted videos.

We selected the methods from [20], [4] and [12] for performance comparison. For fairness, we trained the method in reference [4] and [12] using both the original videos and videos with one frame deleted, and conducted testing on different scenes. As shown in Table 3, the method from reference [4] is prone to erroneously classifying deleted-frame videos as original videos, whereas the method from reference [12] tends to misclassify most original videos as deleted-frame videos. Furthermore, the method from reference [20] demonstrates subpar overall performance. The method proposed in this paper outperforms the methods in the other three papers, showing the state-of-art detection results.

Table 3. The final experimental results(VCM stands for Video Camera Model, while DFN represents Deleted Frame Number. Training the siamese network with H1 videos and separately training the random forest classifier with H1, H2, and D videos)

VCM		H1					H2					D				
DFN		0	1	3	5	10	0	1	3	5	10	0	1	3	5	10
Accuracy	[4]	0.93	0.13	0.53	0.60	0.60	1.00	0.20	0.55	0.50	0.55	1.00	0.15	0.50	0.60	0.55
	[20]	0.43	0.40	0.53	0.57	0.57	0.35	0.50	0.60	0.55	0.65	0.45	0.45	0.50	0.60	0.65
	[12]	0.33	1.00	1.00	1.00	1.00	0.20	1.00	1.00	0.10	1.00	0.25	1.00	1.00	1.00	1.00
	Ours	1.00	0.93	0.93	0.97	1.00	1.00	0.90	0.90	0.90	0.95	1.00	0.95	1.00	1.00	1.00

4.3 Ablation Study

To investigate the contribution of each module (i.e., the siamese network and the random forest classifier) to the final detection performance, we conducted ablation studies. We conducted video frame deletion detection separately using

Table 4. The results of the ablation studies(Only SN represents using only the siamese network, and Only RFC represents using only the random forest classifier, Complete represents using all modules).

Video Camera Model		H1				
Deleted frame number		0	1	3	5	10
Accuracy	Only SN	0	1.00	1.00	1.00	1.00
	Only RFC	1.00	0.90	0.90	0	0
	Complete	1.00	0.90	0.95	0.95	1.00

the siamese network and the random forest classifier. The detection results are presented in Table 4.

When using only the siamese network for detection, videos from six scenes captured by H1 were used, three scenes for training, one scene for validation, and another two scenes for testing. The parameters and data used in the training are the same as those in Section 4.1. The validation and test sets are structured similarly, with each category containing 4,000 images, including original video frames, re-compressed consecutive frames, and frames with deletions in between. The classification accuracy for the above three categories can reach 82.9%. When applied directly to long video detection, the siamese network categorizes all original videos as manipulated videos.

We resized the velocity field images to 512x512, then flattened them and input them into the random forest classifier. The training data consisted of three scenes captured by H1, each with 20 video segments. Additionally, it only includes original videos and videos with one frame deleted. The test data included two additional scenes captured by H1, each with 20 video segments comprising original videos and videos with 1, 3, 5, and 10 frames deleted. As shown in Table 4, the random forest classifier exhibited relatively poor generalization performance and failed to detect videos with 5 and 10 frames deleted. It can be observed that combining the siamese network with the random forest classifier achieved the best results.

5 Conclusion

This paper presents a method for detecting frame deletion in videos. The approach relies on the continuity feature of the velocity field, where frame deletions result in a discontinuity in the velocity field sequence. It uses a siamese network to extract complex features from the velocity field images, transforming the videos into velocity field feature sequences. Subsequently, random forest classifier is employed to distinguish between original videos and videos with frames deleted. Experimental results demonstrate the effectiveness of the proposed method, enabling the detection of long videos with extreme frame deletions(e.g., deleting one frame in a 2000-frame video).

As the future work, we will investigate whether our system can be applied to other types of inter-frame tampering operations, such as frame replacement, frame insertion, etc. Additionally, we will expand the training dataset and use transfer learning to enhance the system's generalization capability.

References

1. Akhtar, et al.: Digital video tampering detection and localization: review, representations, challenges and algorithm. Mathematics **10**(2), 168 (2022)
2. Bakas, J., Naskar, R.: A digital forensic technique for inter–frame video forgery detection based on 3d cnn. In: International Conference on Information Systems Security. pp. 304–317. Springer (2018)
3. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-Convolutional Siamese Networks for Object Tracking. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 850–865. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_56
4. Fadl, S., Han, Q., Li, Q.: Cnn spatiotemporal features and fusion for surveillance video forgery detection. Signal Processing: Image Communication **90**, 116066 (2021)
5. Fadl, S., Han, Q., Qiong, L.: Inter-frame forgery detection based on differential energy of residue. IET Image Proc. **13**(3), 522–528 (2019)
6. Fadl, S., Han, Q., Qiong, L.: Exposing video inter-frame forgery via histogram of oriented gradients and motion energy image. Multidimension. Syst. Signal Process. **31**(4), 1365–1384 (2020). https://doi.org/10.1007/s11045-020-00711-6
7. Farnebäck, G.: Two-Frame Motion Estimation Based on Polynomial Expansion. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 363–370. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45103-X_50
8. Feng, C., Xu, Z., Jia, S., Zhang, W., Xu, Y.: Motion-adaptive frame deletion detection for digital video forensics. IEEE Trans. Circuits Syst. Video Technol. **27**(12), 2543–2554 (2016)
9. Ghimire, S., Choi, J.Y., Lee, B.: Using blockchain for improved video integrity verification. IEEE Trans. Multimedia **22**(1), 108–121 (2019)
10. Gironi, A., Fontani, M., Bianchi, T., Piva, A., Barni, M.: A video forensic technique for detecting frame deletion and insertion. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6226–6230 (2014)
11. Gong, H.Y., Hui, F.C., Dan, B.D.: Iref: Improved residual feature for video frame deletion forensics. In: 2022 4th International Conference on Data Intelligence and Security (ICDIS). pp. 248–253. IEEE (2022)
12. Gowda, R., Pawar, D.: Deep learning-based forgery identification and localization in videos. SIViP **17**(5), 2185–2192 (2023)
13. Kono, K., Yoshida, T., Ohshiro, S., Babaguchi, N.: Passive video forgery detection considering spatio-temporal consistency. In: Proceedings of the Tenth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2018) 10. pp. 381–391. Springer (2020)
14. Long, C., Smith, E., Basharat, A., Hoogs, A.: A c3d-based convolutional neural network for frame dropping detection in a single video shot. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1898–1906. IEEE (2017)

15. Nguyen, X.H., et al.: Detecting video inter-frame forgeries based on convolutional neural network model. International Journal of Image, Graphics and Signal Processing **10**(3), 1–12 (2020)
16. Parmar, A., Katariya, R., Patel, V.: A review on random forest: An ensemble classifier. In: International conference on intelligent data communication technologies and internet of things (ICICI) 2018. pp. 758–763. Springer (2019)
17. Shelke, N.A., Kasana, S.S.: Multiple forgeries identification in digital video based on correlation consistency between entropy coded frames. Multimedia Systems pp. 1–14 (2022)
18. Shi, Y., Qi, M., Yi, Y., Zhang, M., Kong, J.: Object based dual watermarking for video authentication. Optik **124**(19), 3827–3834 (2013)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations (ICLR 2015). Computational and Biological Learning Society (2015)
20. Wu, Y., Jiang, X., Sun, T., Wang, W.: Exposing video inter-frame forgery based on velocity field consistency. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 2674–2678. IEEE (2014)

Vehicle Detection Performance in Nordic Region

Hamam Mokayed[1](✉), Rajkumar Saini[1], Oluwatosin Adewumi[1],
Lama Alkhaled[1], Björn Backe[1], Palaiahnakote Shivakumara[2], Olle Hagner[3],
and Yan Chai Hum[4]

[1] Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden
{hamam.mokayed,rajkumar.saini,oluwatosin.adewumi,
lama.alkhaled,bjorn.backe}@ltu.se

[2] School of Science, Engineering and Environment, Salford University, Salford, UK
S.Palaiahnakote@salford.ac.uk

[3] Smartplanes, Jävre, Sweden
olle.hagner@smartplanes.se

[4] Mechatronics and Biomedical Engineering, Universiti Tunku Abdul Rahman,
Petaling Jaya, Selangor, Malaysia
humyc@utar.edu.my

Abstract. This paper addresses the critical challenge of vehicle detection in the harsh winter conditions in the Nordic regions, characterized by heavy snowfall, reduced visibility, and low lighting. Due to their susceptibility to environmental distortions and occlusions, traditional vehicle detection methods have struggled in these adverse conditions. The advanced proposed deep learning architectures brought promise, yet the unique difficulties of detecting vehicles in Nordic winters remain inadequately addressed. This study uses the Nordic Vehicle Dataset (NVD), which contains UAV (unmanned aerial vehicle) images from northern Sweden, to evaluate the performance of state-of-the-art vehicle detection algorithms under challenging weather conditions. Our methodology includes a comprehensive evaluation of single-stage, two-stage, segmentation-based, and transformer-based detectors against the NVD. We propose a series of enhancements tailored to each detection framework, including data augmentation, hyperparameter tuning, transfer learning, and Specifically implementing and enhancing the Detection Transformer (DETR). A novel architecture is proposed that leverages self-attention mechanisms with the help of MSER (maximally stable extremal regions) and RST (Rough Set Theory) to identify and filter the region that model long-range dependencies and complex scene contexts. Our findings not only highlight the limitations of current detection systems in the Nordic environment but also offer promising directions for enhancing these algorithms for improved robustness and accuracy in vehicle detection amidst the complexities of winter landscapes. The code and the dataset are available at https://nvd.ltu-ai.dev.

Keywords: Vehicle detection · Nordic region · DETR, MSER, Roughset, YOLO (You only look once), Faster-RCNN (regions with convolutional neural networks), SSD (Single Shot MultiBox), U-Net

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 62–77, 2025.
https://doi.org/10.1007/978-3-031-78312-8_5

1 Introduction

Vehicle detection systems are crucial for many applications, including traffic management and autonomous navigation. Yet, their performance in adverse weather conditions, especially in the harsh winter of the Nordic regions, presents significant challenges. The consistent snowfall, reduced visibility, and low lighting conditions inherent to these areas complicate vehicle detection tasks, demanding an evaluation and enhancement of current detection methods to ensure reliability and robustness [1]. contributions in this paper include:

Evaluation and enhancement of State-of-the-Art Algorithms: We assess the performance of various vehicle detection frameworks, including single-stage, two-stage, segmentation-based, and transformer-based architectures, using the Nordic Vehicle Dataset (NVD).

Enhancement of DETR: We implement and enhance the Detection Transformer (DETR) model, leveraging its self-attention mechanisms to handle the complex scene contexts and long-range dependencies inherent in the Nordic winter environment.

Public Availability of Resources: We make the code and dataset publicly available to facilitate further research and development in this domain.

By focusing on these contributions, our study aims to advance vehicle detection technologies for challenging weather conditions, providing a foundation for future research into adaptive, context-aware detection systems capable of maintaining high performance across diverse and dynamic environments.

1.1 Related Work

Initially, traditional non-deep learning techniques were employed for vehicle detection. These conventional methods, however, struggled with image distortions, vehicle occlusions, and variations in illumination, resulting in limited accuracy and applicability to specific scenarios. For instance, Tsai et al. (2007) [2] discussed the limitations of normalized color and edge map techniques under varying light conditions. Similarly, Felzenszwalb et al. (2008)[3] highlighted the challenges faced by multiscale deformable part models in maintaining detection accuracy amid occlusions. Mokayed et al. (2014)[4] proposed an enhanced traditional edge detection method to identify vehicles and license plate numbers in images captured by drones. This approach aimed to achieve real-time processing on constrained devices while addressing challenges related to lighting, viewing angles, and occlusions. As a result, their accuracy remained limited, constraining their applicability to specific scenarios [5]. With the advent of deep learning, various approaches aimed to enhance vehicle detection by introducing more advanced network architectures. For example, Geiger et al. (2012)[6]. introduced the KITTI vision benchmark suite, which provided a comprehensive dataset and benchmark for evaluating the performance of different detection algorithms under real-world conditions. Howard et al. (2017)[7]. developed MobileNets, efficient convolutional neural networks designed for mobile vision applications, which significantly improved the computational efficiency and accuracy of detection tasks. Hu et al. (2021) [8] utilized visual attention cues to enhance vehicle detection and tracking, focusing on improving detection accuracy in complex environments.Despite extensive research in the broader field of object detection, many general-purpose classifiers struggle to achieve competitive performance in vehicle detection benchmarks. This struggle is due to the unique challenges posed by vehicle detection, including significant light variations, dense occlusions, and size disparities. These challenges prompted us to explore the performance of various vehicle detection architectures under such demanding conditions. A search of available UAV datasets validates the originality of our study by improving vehicle detectors using the NVD dataset. Existing datasets primarily address orientation and scale issues in clear weather, such as the VisDrone dataset [9] focusing on general object detection, and the UAV project dataset limited to foggy conditions. Other datasets, like Mimos[4,10] and CARISSMA [11] focus on different objectives or lack UAV perspectives. The Video-Traffic dataset[12] emphasizes traffic information without considering weather impacts. Our research uniquely aims for comprehensive vehicle detection under challenging weather conditions, leveraging a precisely annotated dataset. The Nordic Vehicle Dataset (NVD) [5] serves to address and highlight these challenges. Capturing 22 aerial videos across northern Sweden's snowy terrain, the NVD provides a comprehensive view of the difficulties faced in vehicle detection from unmanned aerial vehicles (UAVs).

1.2 Vehicle Detectors

Numerous techniques have been devised previously to tackle the complexities associated with vehicle detection, particularly concerning small-sized vehicles and scenarios involving multiple vehicles within images in different weather conditions. Vehicle detectors have evolved with significant advancements in accuracy, speed, and robustness. The most prominent types of vehicle detectors include single-stage detectors, two-stage detectors, and transformer-based detectors, each with distinct architectures and operational mechanisms [13].
1. Single stage (or single pass): these model architectures use the neural network in predicting the bounding boxes and class probabilities of objects in one step for a full image input [14,15]. Examples of such models include the popular You Only Look Once (YOLO) [16], Fast Detection [17], and 3D-DETNet [18]. They tend to have high inference speeds [15]. These models have been applied in vehicle detection tasks [18,19]. The introduction of the SotA YOLO, now in version 9 [20], marked a milestone in computer vision. The latest version combines two novel concepts: Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Network (GELAN).
2. Double stage (or double pass): these model architectures generate candidate regions before a second stage of pooling operation to classify these regions [15,21]. Examples of these models are Region-based Convolutional Neural Network (R-CNN) [14] and Faster R-CNN [22], and FPN [23]. They tend to have high recognition accuracy and are also used for vehicle detection [15,24]. The Faster R-CNN

model introduced a Region Proposal Network (RPN), which shares convolutional features of the full input image with the detection network.
3. Segmentation-based detectors: these models' architectures are types of neural network models designed for pixel-wise classification, which means it assigns a class label to each pixel in an image, effectively segmenting the image into different regions based on the detected objects or features. Unlike traditional object detectors that produce bounding boxes around objects, segmentation-based detectors provide detailed spatial understanding and precise localization of objects within an image. The U-Net architecture is a prime example of this approach, utilizing an encoder-decoder structure with skip connections to capture both contextual and fine-grained details. This makes segmentation-based detectors particularly useful for tasks requiring high-resolution spatial accuracy[25]
4. Transformers-based detectors: these detection models use self-attention mechanisms, allowing each part of the image to be considered in the context of every other part. This enhances the model's ability to understand complex scenes and the relationships between different objects within them. This global processing capability is particularly beneficial in densely populated scenes or scenarios where objects' context and relative positioning are crucial for accurate detection. Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data. The Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers [26]. Examples of these models, Detection Transformer (DETR) [27], Detection Transformer-Spatial Pyramid Pooling (DETR-SPP) [28],demformable DETR [29], and Swin Transformer [30].

2 Proposed Method

The foundation of the proposed method for evaluating vehicle detection performance in Nordic environments hinges on the strategic selection of the Nordic Vehicle Dataset (NVD) as a rigorous testing ground. With its comprehensive compilation of challenging scenarios collected from the Nordic region, the NVD provides a unique opportunity to validate, tune, and enhance various state-of-the-art (SOTA) vehicle detection algorithms. This methodological approach is designed to encompass a broad spectrum of contemporary detection frameworks, categorically spanning single-stage, two-stage, segmentation-based, and transformer-based detectors, each known for their distinct operational paradigms and performance characteristics, refer to Fig 1.

2.1 Dataset Selection: Nordic Vehicle Dataset (NVD)

The NVD's rich repository of UAV images, meticulously captured over the snowy landscapes of northern Sweden, provides a comprehensive view of the difficulties faced in vehicle detection from unmanned aerial vehicles (UAVs) in harsh weather. With altitudes ranging from 120 to 250 meters, the dataset encompasses a variety of snow and cloud conditions across 8,450 annotated frames featuring

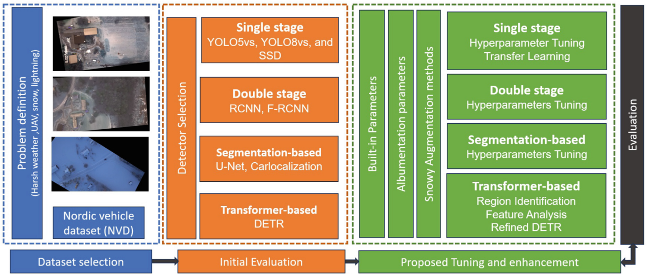


Fig. 1: Proposed Method.

26,313 cars. The diverse video resolutions, frame rates, and varying Ground Sample Distance (GSD) metrics offer a detailed representation of vehicles against the challenging backdrop of Nordic winters [5].

2.2 Evaluation Framework

The method entails a structured evaluation framework in which a diverse array of SOTA vehicle detection algorithms will be systematically tested against NVD.

1. Single-Stage Detectors: YOLO5vs, YOLO8vs, and SSD (Single Shot MultiBox Detector) will be assessed for their detection capabilities in the face of the NVD's challenging conditions.
2. Two-Stage Detectors: RCNN and Faster R-CNN (F-RCNN) will undergo rigorous testing to determine its effectiveness in accurately identifying vehicles amidst heavy occlusions and variable snow cover.
3. Segmentation-based Detectors: U-Net and calLocalization[31] methods will be adjusted and assessed to evaluate their effectiveness in precisely detecting vehicles under conditions of significant occlusion and varying levels of snow cover.
4. Transformer-Based Detectors: DETR will be evaluated for their ability to model long-range dependencies and complex scene contexts.

2.3 Performance Tuning and Enhancement

Critical to this proposed method is the iterative process of tuning and improving algorithms based on their performance metrics against the NVD. This involves not only the adjustment of hyperparameters but also the potential integration of novel steps specifically tailored to overcome the identified challenges by NVD. Data augmentation, a crucial preprocessing step, was performed for all the different types of the detectors. We initially used the Albumentations library to simulate various weather conditions (snow, rain, fog) offline. This process included pixel-level transformations and maintaining bounding box accuracy, but it required significant disk space and processing time. To overcome this, we transitioned to built-in online augmentation, which is more efficient. Some

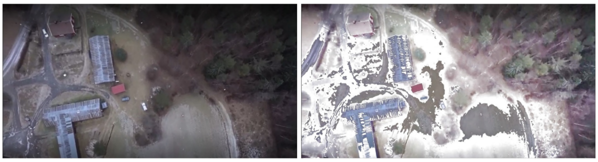


Fig. 2: Applied Snow Augmentation.

hyperparameters that we have set, which affect data augmentation, are listed below, but the entire set can be accessed through the code available on Github.

- fl_gamma: 0.0 - focal loss gamma.
- hsv_h: 0.015 - image HSV-Hue augmentation (fraction)
- hsv_s: 0.7 - image HSV-Saturation augmentation (fraction)
- degrees: 45.0 - image rotation (+/- deg)
- perspective: 0.001 - image perspective (+/- fraction), range 0-0.001.

Albumentation library parameters are also tuned to improve different models robustness. Different parameters as blur, median blur, and other transformations are tuned to simulate various snowy conditions and enhance the training dataset's diversity as below.

- Blur - p=0.01, blur_limit=(3, 7).
- MedianBlur - p=0.01, blur_limit=(3, 7)
- ToGray - p=0.01
- CLAHE -p=0.01, clip_limit=(1, 4.0), tile_grid_size=(8, 8)

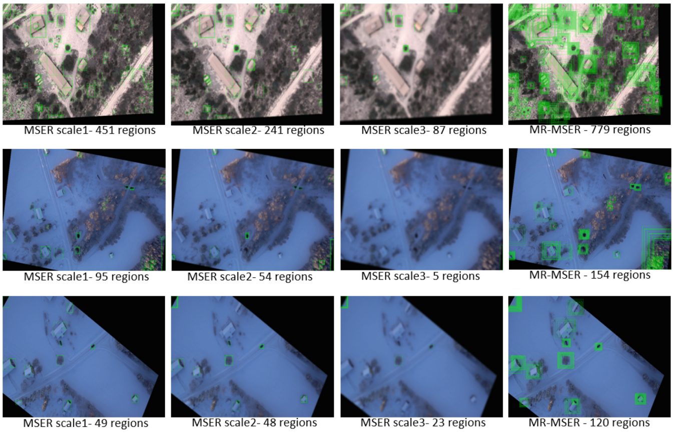
Additional augmentation methods, specifically designed for snowy conditions, have been proposed to enhance the quality of training data in such environments. These techniques include the addition of features such as snow overlays and snowflakes, which can be adjusted to simulate various levels of snowfall and accumulation. This approach aims to mimic the progressive accumulation of snow, introducing a dynamic aspect to the training data set, as clarified in Fig 2.

1. Single-Stage Detectors: Improving the performance of SSD and YOLO in general, regardless of the model version, involves a series of traditional techniques and strategies. In this work, we will focus on the following [32,33].
   (a) Hyperparameter tuning is essential for adjusting different parameters. Adjust learning rate, batch size, and anchor box dimensions are implemented to better fit the characteristics of vehicles in snowy environments. For instance, use lower learning rates to refine the model and avoid overshooting in the fine-tuning phase.
   (b) Transfer learning is an effective practice used to utilize pretrained models on larger datasets. To implement transfer learning for enhancing YOLO's detection accuracy in snowy conditions for vehicles, start by initializing the YOLO model with weights from a pre-trained dataset (COCO). This

provides a solid foundation for feature extraction capabilities. Next, compile a domain-specific dataset such as the NVD (Nordic Vehicle Dataset), which includes images of vehicles captured in various snowy conditions. The NVD dataset contains a diverse collection of vehicle images specifically designed to address the challenges of detecting vehicles in Nordic winter conditions, including heavy snowfall, low-light environments, and varying levels of snow coverage on vehicles. By fine-tuning the YOLO model on this dataset, you can adapt it to recognize the unique features and patterns associated with vehicles in snowy settings.

2. Two-Stage Detectors: Improving the performance of these detectors involves a series of traditional techniques and strategies. In this work, we use a low learning rate (e.g., 0.001) for stable convergence and adaptive learning rate schedulers for dynamic adjustments. Customize anchor box scales and aspect ratios using K-means clustering to address occlusions, and adjust the number of proposals from the Region Proposal Network (RPN) for speed and accuracy. Fine-tune the Non-Maximum Suppression (NMS) threshold to reduce false positives, configure the Region of Interest (RoI) pooling layer for scale variations, and optimize batch sizes and training epochs with early stopping to prevent overfitting. These strategies enhance F-RCNN's accuracy in snowy environments[34].
3. Segmentation-Based Detectors: This study aims to fine-tune the parameters of two different segmentation-based detectors and assess their performance on the proposed enhanced augmented dataset.
4. Transformer-Based Detectors: This paper primarily focuses on introducing enhancements to the Detection Transformers (DETR) model to improve its performance. In this regard, the following steps will be presented:

(a) Initial Region Identification under Adverse Conditions:
Initially, a robust algorithm called maximally stable extremal regions (MSER) is proposed, which generates a novel set of image components known as extremal regions. These regions are characterized by two features derived from the projection transformation of image coordinates and the monotonic transformation of image intensities [35]. Affine invariant feature descriptors are computed on a grayscale image, and although MSER's robustness varies from multiple measurement regions derived from invariant constructs from extremal regions [36], certain regions exhibit distinct characteristics that are notably larger and potentially useful for establishing preliminary correspondences [35]. As MSER can generate numerous blobs of varying sizes, accommodating original image resolution detection as well as different resolutions stemming from long distances or blurred (coarse) images, it leads to a loss of image details and connections between different regions and their neighbors [36]. MSER's strength lies in its capability to maintain invariance to scale changes in the scene image across different resolutions, thus stabilizing vehicle regions. The image's resolution is obtained using a scale pyramid (without Gaussian filtering), encompassing one octave per scale and a total of three scales ranging from the finest image (input image) to the coarser, blurred image. This process results in the creation of mul-



MSER scale1- 451 regions MSER scale2- 241 regions MSER scale3- 87 regions MR-MSER - 779 regions

MSER scale1- 95 regions MSER scale2- 54 regions MSER scale3- 5 regions MR-MSER - 154 regions

MSER scale1- 49 regions MSER scale2- 48 regions MSER scale3- 23 regions MR-MSER - 120 regions

Fig. 3: Initial Region Identification by MSER and MR-MSER.

tiresolution maximally stable extremal regions, denoted as MR-MSER, which are subsequently applied to NVD's images [35] as clarified in Fig. 3).

To enhance region detection recall, an augmentation process is applied to MR-MSER. The augmentation procedure is implemented for each MR-MSER region. Each MR-MSER region of varying sizes is enclosed within a rectangular bounding box that remains centered. This bounding box is then transformed into a square shape by expanding its area by 30% and 60% in three distinct dimensions. These different square dimensions are resized to a 28×28 pixel image patch. Each image patch undergoes random rotation within the range of [$\pi/4$, $\pi/4$] four times, facilitating model training [37]. The regions generated by applying augmented MR-MSER over NVD's images are shown in Fig. 3.

(b) Feature Analysis and Reduction in Obscured Environments.

The purpose of using the rough set approach is knowledge discovery and approximation of sets using granular information. By applying RST, one can reduce the dimensionality of the feature space and generate decision rules that are crucial for distinguishing between vehicle and non-vehicle regions. This step enhances the computational efficiency and effectiveness of the subsequent detection process. Given a confidence map, the process for granularization splits the input image window into multiple windows with a resolution of each sub-window ($g = 4$). The main purpose is to classify pixel values into vehicle and nonvehicle approximations. Let a set of objects be U. There is also an indiscernibility relation R $\subseteq$ U* U that refers to the central concept of rough set theory. In the indiscernibility relation, the values of the object are identical, considering a subset of the related attributes. In other words, it is an equivalence relation where all identical values of the object are elementary. Hence, R can also be considered an equivalence relation. Let X be a subset of U with two possibilities: either is crisp, which is explicit with respect to R if the boundary region of X is empty, or is rough, which is in-explicit with respect to R if the boundary region of X is nonempty using RST to characterize the set U as possible for lower approximation and upper approximation, and boundary region of set X.

$$\underline{R}(x) \ = \ \bigcup_{x \in U} R(x) : R(x) \subseteq X \tag{1}$$

$$\text{R-upper approximation of X: } \overline{R}(x) = \bigcup_{x \in U} R(x) : R(x) \cap X \neq \phi \tag{2}$$

$$\text{R-boundary region of X: } \overline{RN}_R(X) = \overline{R}(X) \ - \ \underline{R}(X) \tag{3}$$

Rough entropy (RE) is introduced to avoid imprecision to find the optimum threshold as precisely as possible. The rough entropy threshold (RET) as the reference for the threshold in the binarization approach in the grayscale image, which was obtained by using a sliding window with a nonoverlapping granule window in m×n, is set as a 2×2 window size. RET can be defined as [38].

$$RE_T = -\frac{\exp}{2}\left[R_{OT} log_{\exp}(R_{OT}) + R_{BT} log_{\exp}(R_{BT})\right] \tag{4}$$

where $R_{OT} = 1 - \frac{|\overline{O}_T|}{|\underline{O}_T|}$ is the roughness of the object, $R_{BT} = 1 - \frac{|\overline{B}_T|}{|\underline{B}_T|}$ is the roughness of the background, $|\overline{O}_T|$ and $|\underline{O}_T|$ are the cardinality of the sets $\overline{O}_T$ and $\underline{O}_T$ for a given image depending on the value T, and $|\overline{B}_T|$ and $|\underline{B}_T|$ are the cardinality of the sets $\overline{B}_T$ and $\underline{B}_T$ for a given image depending on the value T. The principle of reducing the roughness of both the object and background and maximizing RE_T is computed for every T representing the object and background regions, respectively (0,. . . ,T) and (T+1,. . . ,L-1). The optimum threshold is selected for the maximum RE_T to provide the object-background segmentation given by the definition of T^*.

$$T^* = \arg\max_T RE_T \tag{5}$$

Maximizing the rough entropy RE_T to obtain the required threshold implies minimizing both the object roughness and background roughness such that this method is an object enhancement/extraction method [38].

The confidence map is generated to show the final regions selected after applying RST as the filtration layer. This confidence map is constructed by utilizing confidence values from each stacked regions. Regions with higher intensity in the confidence map are indicative of potential vehicle components. The outcome of the generated augmented confidence map is shown in Fig. 4.

(c) Refined DETR Detection in Complex Contexts

Maximally Stable Extremal Regions (MSER) refined by rough set, a method renowned for its robustness in detecting coherent regions in images, presents a promising solution to enhance DETR's capabilities in these complex visual environments. By integrating refined MSER with DETR, the improved model can leverage the strength of MSER in efficiently segmenting and identifying stable regions within images, even under severe weather distortions, as shown in Fig. 5.

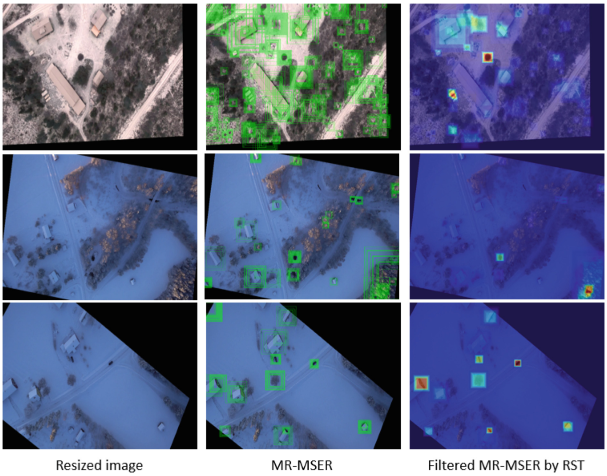


Fig. 4: Filtered region Identification by RST and MR-MSER.

This fusion aims to provide a more resilient feature extraction mechanism, allowing DETR to better recognize and localize objects partially or fully obscured by snow.

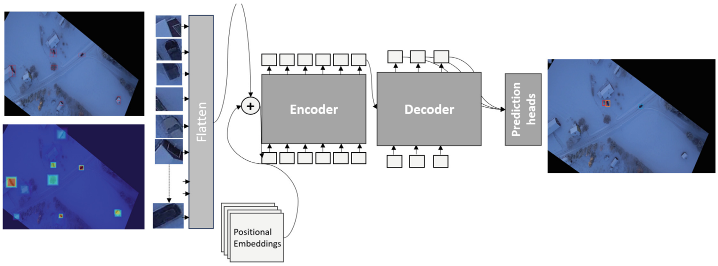


Fig. 5: Refined DETR performance.

3 Experimental Results

This section presents the results obtained from experiments conducted using different detectors before and after implementing specific performance enhancements customized for each detector. The experiments initially showcase the performance of single-stage detectors without any enhancements, as illustrated in Fig. 6, along with the corresponding accuracy detailed in Table 1. Subsequently, the enhancement techniques proposed for single detectors, as described in the methodology, are implemented to demonstrate improvements in detector performance, as depicted in Fig. 7, Table 2.

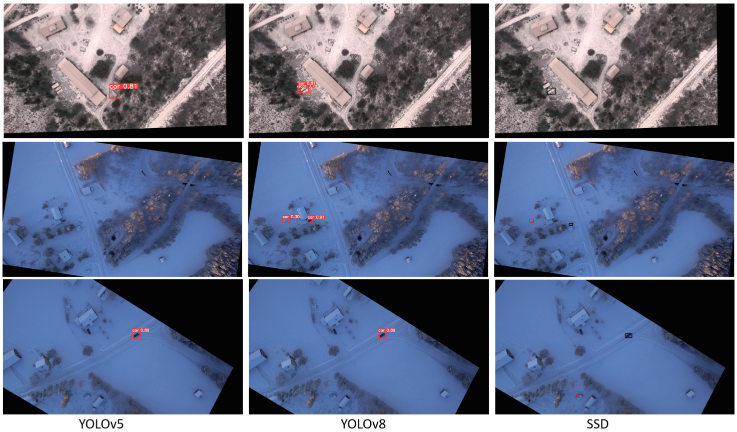


Fig. 6: Single detectors performance without enhancement.

1. Single-Stage Detectors

Table 1: Model accuracy without enhancement

Model	Precision	Recall	mAP50	mAP50-95
YOLOv5s	69.00%	32.10%	53.20%	31.70%
YOLOv8s	72.40%	28.00%	45.80%	22.80%
SSD	31.20%	18.00%	26.8%	12.4%

Table 2: Model accuracy by implementing the proposed methodology

Model	Precision	Recall	mAP50	mAP50-95
YOLOv5s	70.6%	48.2%	56.0%	33.80%
YOLOv8s	77.1%	34.60%	50.7%	24.22%
SSD	39.60%	25.8%	33.75%	20.2%

The study assessed the effectiveness of a dual-stage detector, exemplified by RCNN, and F-RCNN, both prior to and after the implementation of suggested enhancements, as illustrated in Fig. 8 and Table 3.

2. Two-Stage Detectors Following this, we assessed the efficacy of segmentation-based approaches by employing a U-Net model[25] alongside a fractional B-spline wavelet transform called CarLocalization, utilizing a specially designed U-Net architecture [31], as shown in Fig. 9, Table 4.

Finally, we evaluated the performance of both DETR and an enhanced version of DETR incorporating MR-MSER and rough set theory, as described in Fig. 10, Table 5.

Table 3: RCNN performance and F-RCNN without/with enhancement

Model	Precision	Recall	mAP50	mAP50-95
RCNN	5.7%	11.4%	7.4%	2.50%
F-RCNN	8.4%	13.3%	10.15%	4.30%
Enhanced RCNN	10.2%	18.6%	12.6%	7.2%
Enhanced F-RCNN	22.40%	26.3%	23.25%	12.10%

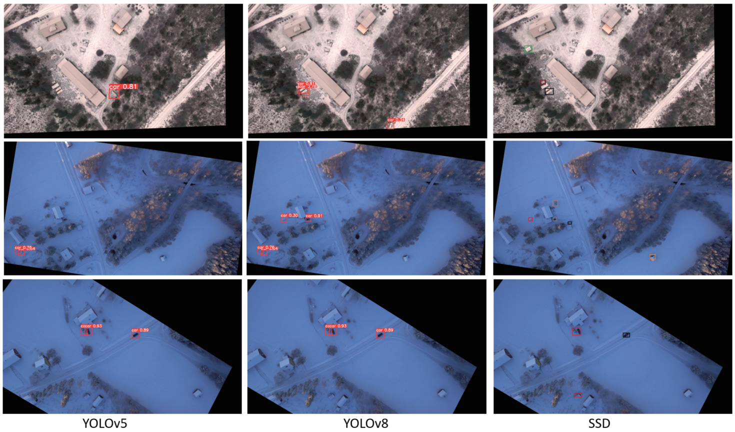


Fig. 7: Single detectors performance with the proposed methodology.

Table 4: Segmentation-based Performance

Model	Precision	Recall	mAP50	mAP50-95
U-Net	72.5%	50.50%	56.8%	32.3%
CarLocalization	74.7%	54.8%	60.4%	38.5%

3. Transformer-Based Detectors

Table 5: DETR Performance

Model	Precision	Recall	mAP50	mAP50-95
DETR	80.4%	62.50%	74.8%	52.3%
refined DETR	85.4%	70.2%	79.4%	58.6%

Our detailed analysis highlights why certain detectors perform better than others under the challenging conditions of the Nordic winter. Single-stage detectors as YOLO struggle with occlusions and snow cover, while two-stage detectors like Faster R-CNN and RCNN offer the worst performance among the others , due to their region proposal mechanisms. U-Net, known for its segmentation capabilities, effectively delineates vehicle regions even in challenging environments, contributing to enhanced detection accuracy. Transformer-based detectors like DETR excel in complex scenarios due to their advanced attention mechanisms. The refined DETR, which uses Rough Set Theory (RST) and Maximally Stable Extremal Regions (MR-MSER) for region nomination and filtration, demonstrated the best performance among the evaluated detectors.

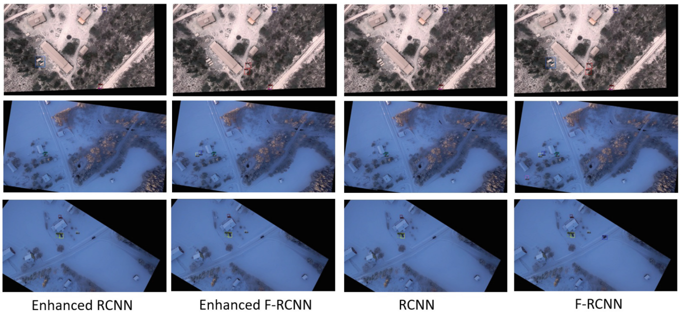


Fig. 8: Double detectors performance without/with enhancement.

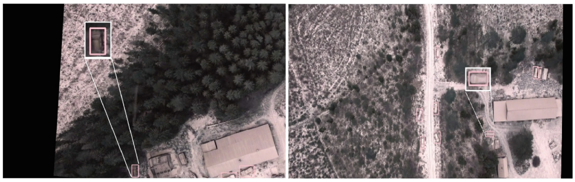


Fig. 9: Segmentation-based detectors

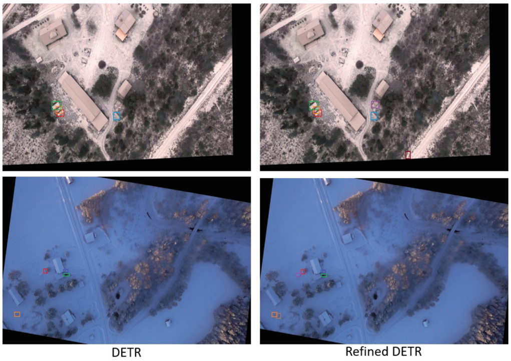


Fig. 10: DETR vs Refined DETR

4 Conclusion

This study comprehensively examined vehicle detection algorithms under the extreme and variable conditions of Nordic winters. By utilizing the Nordic Vehicle Dataset (NVD), we rigorously evaluated a diverse array of state-of-the-art vehicle detection frameworks, including single-stage, two-stage, segment-based, and transformer-based architectures. Our analysis revealed significant challenges faced by these algorithms when confronted with the unique environmental factors of the Nordic landscape, such as vehicles fully covered by snow and variable illumination. Despite these challenges, we demonstrated notable improvements in the performance of these detection systems through systematic tuning and enhancements tailored to this demanding environment. Key strategies, including data augmentation, hyperparameter adjustment, and transfer learning, were pivotal in enhancing detection accuracy and robustness. Specifically, for transformer-based frameworks like DETR, the implementation of region identification using MSER and a filtering layer grounded in Rough Set theory significantly improved detection performance. An important ethical consideration of our work is the potential for surveillance misuse. While our advancements in vehicle detection can greatly benefit traffic management and autonomous navigation, it is crucial to ensure that these technologies are implemented with respect for privacy and civil liberties. Policies and regulations must be in place to prevent misuse and ensure that the deployment of such technologies is transparent and ethically responsible.

In conclusion, this study advances vehicle detection technologies for challenging weather conditions and lays the groundwork for future research into adaptive, context-aware detection systems capable of maintaining high performance across diverse and dynamic environments. We hope our findings inspire further innovations in the development and application of vehicle detection systems.

References

1. Sakhare, K.V., Tewari, T., Vyas, V.: Review of vehicle detection systems in advanced driver assistant systems. Archives of Computational Methods in Engineering **27**(2), 591–610 (2020)
2. Tsai, L.-W., Hsieh, J.-W., Fan, K.-C.: Vehicle detection using normalized color and edge map. IEEE Trans. Image Process. **16**(3), 850–864 (2007)
3. Felzenszwalb, P., McAllester, D., Ramanan, D., “A discriminatively trained, multiscale, deformable part model,” in,: IEEE conference on computer vision and pattern recognition. Ieee **2008**, 1–8 (2008)
4. H. Mokayed, L. K. Meng, H. H. Woon, and N. H. Sin, “Car plate detection engine based on conventional edge detection technique,” in *The International Conference on Computer Graphics, Multimedia and Image Processing (CGMIP2014). The Society of Digital Information and Wireless Communication*, 2014
5. H. Mokayed, A. Nayebiastaneh, K. De, S. Sozos, O. Hagner, and B. Backe, “Nordic vehicle dataset (nvd): Performance of vehicle detectors using newly captured nvd from uav in different snowy weather conditions.” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5313–5321
6. Geiger, A., Lenz, P., Urtasun, R., “Are we ready for autonomous driving? the kitti vision benchmark suite,” in,: IEEE conference on computer vision and pattern recognition. IEEE **2012**, 3354–3361 (2012)
7. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.04861, 2017
8. F. Hu, G. Venkatesh, N. E. O’Connor, A. F. Smeaton, and S. Little, “Utilising visual attention cues for vehicle detection and tracking,” in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 5535–5542

9. Y. Cao, Z. He, L. Wang, W. Wang, Y. Yuan, D. Zhang, J. Zhang, P. Zhu, L. Van Gool, J. Han *et al.*, “Visdrone-det2021: The vision meets drone object detection challenge results,” in *Proceedings of the IEEE/CVF International conference on computer vision*, 2021, pp. 2847–2854
10. Mokayed, H., Shivakumara, P., Woon, H.H., Kankanhalli, M., Lu, T., Pal, U.: A new dct-pcm method for license plate number detection in drone images. Pattern Recogn. Lett. **148**, 45–53 (2021)
11. Rothmeier, T., Huber, W., “Let it snow: On the synthesis of adverse weather image data,” in,: IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE **2021**, 3300–3306 (2021)
12. Liu, K., Mattyus, G.: Fast multiclass vehicle detection on aerial images. IEEE Geosci. Remote Sens. Lett. **12**(9), 1938–1942 (2015)
13. X. Zhao, Y. Ma, D. Wang, Y. Shen, Y. Qiao, and X. Liu, “Revisiting open world object detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2023
14. R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587
15. Lohia, A., Kadam, K.D., Joshi, R.R., Bongale, A.M.: Bibliometric analysis of one-stage and two-stage object detection. Libr. Philos. Pract. **4910**, 34 (2021)
16. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788
17. M. A. Hossain, M. I. Hossain, M. D. Hossain, N. T. Thu, and E.-N. Huh, “Fast-d: When non-smoothing color feature meets moving object detection in real-time,” *IEEE Access*, vol. 8, pp. 186 756–186 772, 2020
18. S. Li and F. Chen, “3d-detnet: a single stage video-based vehicle detector,” in *Third International Workshop on Pattern Recognition*, vol. 10828. SPIE, 2018, pp. 60–66
19. Wang, H., Yu, Y., Cai, Y., Chen, X., Chen, L., Li, Y.: Soft-weighted-average ensemble vehicle detection method based on single-stage and two-stage deep learning models. IEEE Transactions on Intelligent Vehicles **6**(1), 100–109 (2020)
20. C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, “Yolov9: Learning what you want to learn using programmable gradient information,” arXiv preprint arXiv:2402.13616, 2024
21. C. Meng, H. Bao, and Y. Ma, “Vehicle detection: A review,” in *Journal of Physics: Conference Series*, vol. 1634, no. 1. IOP Publishing, 2020, p. 012107
22. S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, 2015
23. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125
24. Yang, M.Y., Liao, W., Li, X., Cao, Y., Rosenhahn, B.: Vehicle detection in aerial images. Photogrammetric Engineering & Remote Sensing **85**(4), 297–304 (2019)
25. I. Y. Tanasa, D. H. Budiarti, Y. Guno, A. S. Yunata, M. Wibowo, A. Hidayat, F. N. Purnamastuti, A. Purwanto, G. Wicaksono, and D. D. Domiri, “U-net utilization on segmentation of aerial captured images,” in *2023 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*. IEEE, 2023, pp. 107–112

26. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017
27. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European conference on computer vision*. Springer, 2020, pp. 213–229
28. K. SP and P. Mohandas, “Detr-spp: a fine-tuned vehicle detection with transformer,” *Multimedia Tools and Applications*, pp. 1–22, 2023
29. X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable detr: Deformable transformers for end-to-end object detection,” arXiv preprint arXiv:2010.04159, 2020
30. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022
31. Mokayed, H., Ulehla, C., Shurdhaj, E., Nayebiastaneh, A., Alkhaled, L., Hagner, O., Hum, Y.C.: Fractional b-spline wavelets and u-net architecture for robust and reliable vehicle detection in snowy conditions. Sensors **24**(12), 3938 (2024)
32. Maity, M., Banerjee, S., Chaudhuri, S.S., “Faster r-cnn and yolo based vehicle detection: A survey,” in,: 5th international conference on computing methodologies and communication (ICCMC). IEEE **2021**, 1442–1447 (2021)
33. Q. M. Chung, T. D. Le, T. V. Dang, N. D. Vo, T. V. Nguyen, and K. Nguyen, “Data augmentation analysis in vehicle detection from aerial videos,” in *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*. IEEE, 2020, pp. 1–3
34. Mo, N., Yan, L.: Improved faster rcnn based on feature amplification and oversampling data augmentation for oriented vehicle detection in aerial images. Remote Sensing **12**(16), 2558 (2020)
35. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image Vis. Comput. **22**(10), 761–767 (2004)
36. Donoser, M., Bischof, H., “Efficient maximally stable extremal region (mser) tracking,” in,: IEEE computer society conference on computer vision and pattern recognition (CVPR’06), vol. 1. Ieee **2006**, 553–560 (2006)
37. A. Zamberletti, I. Gallo, and L. Noce, “Augmented text character proposals and convolutional neural networks for text spotting from scene images,” in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, 2015, pp. 196–200
38. Pal, S.K., Shankar, B.U., Mitra, P.: Granular computing, rough entropy and object extraction. Pattern Recogn. Lett. **26**(16), 2509–2517 (2005)

An Edge-Assisted Mural Image Inpainting Approach Leveraging Aggregated Contextual Transformations

Bojie Tang, Linxi Hong, Qing Xie(✉), Tao Guo, and Xiaolei Du

School of Computer Science and Artificial Intelligence, Wuhan 430010, China
felixxq@whut.edu.cn

Abstract. Mural image restoration involves repairing damaged sections of murals to attain desirable visual outcomes. In recent years, the development of mural restoration algorithms has emerged as a key area of interest, driven by the need to preserve murals as valuable artifacts of human historical heritage. Despite their significance, murals have suffered various degrees of deterioration over time. The limited availability of mural-specific datasets and the complexity of mural textures pose significant challenges for contemporary image restoration algorithms, rendering them less effective in mural restoration tasks. To address this, we have compiled a dataset encompassing 3,492 murals and introduced a novel mural image restoration approach, the Edge Assistance and Aggregated Contextual Transformations GAN (EAAOT-GAN). This approach is structured around two phases: edge generation and image restoration. Initially, it generates complete edges of murals, followed by the restoration of the entire mural images through the integration of these edges. Comparative analysis with leading image restoration techniques demonstrates that our method competes favorably with the most advanced mural restoration models, as evidenced by both qualitative and quantitative evaluations.

Keywords: Mural image restoration · Mural dataset · Edge assistance · Aggregated contextual transformations

1 Introduction

Murals, embodying centuries of cultural heritage, are invaluable for their historical, artistic, and archaeological significance. However, these artifacts often suffer from deterioration, such as cracking and color fading, underscoring the urgency of their preservation and restoration. Traditional approaches to mural restoration, involving direct manual intervention, pose risks of inefficiency and potential further damage. The advent of computer vision and deep neural network technologies offers promising alternatives for mural conservation.

B. Tang and L. Hong—Equal Contribution.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 78–96, 2025.
https://doi.org/10.1007/978-3-031-78312-8_6

Recent advancements have seen the application of sophisticated algorithms, including partial convolution techniques for irregular damage repair and structure-aware restoration methods. Despite the efficacy of these algorithms in general image restoration, mural restoration poses unique challenges due to the scarcity of mural-specific datasets, the complexity of damage patterns, and the inherent difficulty in fully recapturing the original aesthetics of murals.

This study introduces a novel two-stage mural restoration methodology leveraging edge assistance and aggregated contextual transformation to navigate these challenges. Initially, an edge generation model delineates the mural's damaged edges, aiming to replicate the original contours closely. This edge map guides the subsequent restoration phase, where a combination of the original image and the edge map, alongside histogram loss, refines the restoration process, addressing the prevalent issue of blurriness.

Our approach is inspired by and builds upon recent innovations in edge-focused restoration and texture synthesis, integrating elements such as AOT Blocks[1] for enhanced context understanding and texture generation. This methodology aims to achieve higher fidelity in restoring high-resolution mural images, supported by a curated dataset of 3492 mural images and the incorporation of histogram loss[2] to mitigate shape blurriness.

This work's principal contributions are outlined as follows:

- First, we collected scattered known mural datasets[3][4] to construct a mural dataset comprising 3492 mural images, dedicated to the research of mural conservation and restoration algorithms.
- Secondly, we prioritize DexiNed[5] edge generation as the initial step to enhance image restoration quality and obtain high-quality edge maps.
- Furthermore, we innovatively apply AOT Block[1] to mural image restoration, enhancing the model's abilities in context reasoning and texture synthesis.
- Lastly, we introduces histogram loss[2] as a constraint in the second stage to address the issue of shape blurriness encountered in high-resolution mural image restoration.

2 Related Works

2.1 Edge Detection

Edge detection is a pivotal task in computer vision and image processing, aimed at delineating object boundaries within images. This task underpins critical applications such as object recognition, scene understanding, and image segmentation. Edge detection methodologies are categorized into classical algorithms, machine learning (ML)-based approaches, and deep learning (DL)-based techniques.

Classical algorithms, exemplified by Canny's detector[6], leverage local image intensity variations to pinpoint edges. The Canny detector, a cornerstone in this domain, employs a multi-step process including Gaussian filtering, gradient calculation, non-maximum suppression, and double thresholding, with non-maximum suppression remaining a widely adopted technique in contemporary algorithms.

ML-based methods have evolved alongside machine learning, employing classifiers like Support Vector Machines (SVM) and Random Forests to identify edge pixels[7]. These approaches excel in automatically adapting to image features, thus offering robust performance across varied datasets. Nevertheless, their efficacy is contingent on the availability of high-quality training data and the efficiency of feature extraction, often at the expense of considerable computational resources.

DL-based methods capitalize on the advancements in Convolutional Neural Networks (CNN) to propose algorithms that surpass traditional techniques in complexity and accuracy[8][9]. Holistically-Nested Edge Detection (HED) exemplifies this approach, achieving end-to-end edge detection by harmonizing features at multiple scales. Besides, U-Net's innovative skip connections, which directly pass feature maps of the same resolution between the encoder and decoder, have achieved high-precision image segmentation performance. DL-based methods demonstrate unparalleled proficiency in capturing complex image features, thereby enhancing the precision and reliability of edge detection.

2.2 Image Inpainting

Image restoration, pivotal in computer vision, aims to repair images' missing areas, with significant implications for image editing applications like object removal. In the last decade, deep learning has propelled the development of innovative image restoration techniques. These approaches are dichotomized into non-learning and learning-based methods.

Non-learning methods include diffusion and patch-based techniques. Diffusion-based methods leverage neighboring pixel information for content filling. Although such algorithms [10] gradually "diffuse" the information of surrounding pixels into the damaged area, they face challenges with large or complexly textured damages due to their reliance on local information and neglect of global image structure. Conversely, patch-based methods, effective for textured images, replicate information from analogous image regions. Criminisi et al.[11] prioritization strategy for repair sequence and Barnes et al.[12] PatchMatch algorithm exemplify advancements in reducing computational demands and improving efficiency, though they occasionally produce inconsistent results.

Learning-based methods, notably those employing deep learning, have significantly advanced image restoration by learning from extensive image data to produce more accurate and natural repairs for complex textures and structures. Context Encoders[13] by Pathak et al. and the DeepFill v2[14] model by Nazeri et al. illustrate the shift towards employing advanced neural network architectures, such as encoder-decoder frameworks[13], dilated convolutions[15], and Generative Adversarial Networks (GANs), for improved restoration quality. These methods efficiently address irregularly shaped missing areas and enhance detail accuracy through innovative network layers and attention mechanisms, underscoring the synergy between edge detection and image restoration in recent advancements.

2.3 Mural Inpainting

Mural image restoration, a specialized domain within image restoration, has garnered considerable attention due to its unique challenges and cultural significance. While employing standard image restoration techniques offers a direct path to addressing mural damage, the unique characteristics of mural datasets, including their imperfection and the complexity of their texture structures, often render such applications less effective.

Advancements in mural restoration have been driven by adaptations and improvements upon existing image restoration algorithms. Notably, building upon the foundation of EdgeConnect[16], Ciortan et al.[17] introduced a structure-guided restoration technique, whereas Li et al.[3] presented a methodology that leverages line drawings to incrementally restore damaged murals, resulting in more lifelike restorations. Additionally, drawing inspiration from Liu et al.'s partial convolution concept[18], Chen et al.[19] applied it within a sliding window framework for mural restoration, and Wang et al.[20] developed a Thangka mural restoration technique utilizing multi-scale adaptive partial convolutions. Despite these advancements, the balance between structure preservation and color restoration in severely damaged areas remains a critical challenge, highlighting the need for further research and innovation in this field.

3 Methodology

High-resolution image restoration, especially when it involves inferring content and generating textures for missing areas, poses significant challenges. This section introduces our proposed mural restoration method, EAAOT-GAN, which leverages edge assistance and aggregated contextual transformations to address these issues. The method encompasses an overview of the system architecture, the design of the edge generation network, and the specifics of the image restoration network.

3.1 Overview

The architecture and operational framework of EAAOT-GAN are illustrated in Fig. 1, encompassing two primary stages: edge generation and image restoration. Each stage operates within the Generative Adversarial Network (GAN) framework, comprising a generator network and a discriminator network.

Suppose G_1 and D_1 represent the generator and discriminator of the edge generator, respectively, and G_2 and D_2 represent the generator and discriminator of the image completion network.

During the edge generation phase, the input $input_1$ to the edge generator G$_1$ consists of the masked mural image's edge map $\tilde{E}_{gt}$, the grayscale image $\tilde{I}_{gray}$ post-mask application, and the mask M itself. This input undergoes two downsampling stages, becoming I_{result}, which is one-sixteenth the size of the original image. Then, I_{result} passes through eight residual blocks[21], producing 256 feature maps of the image. Subsequently, these feature maps are upsampled twice to reconstruct a comprehensive edge map. This final edge map undergoes evaluation by a discriminator designed with a 70×70 PatchGAN[22][23] architecture to verify its authenticity.

In the image restoration phase, the inputs to the image generator G$_2$ include the damaged mural $\tilde{I}_{gt}$, the edge map E_{pred} generated by G$_1$ and the mask M. These inputs, collectively referred to as $input_2$, are first subjected to two downsampling steps, resulting in I'_{result}, which is one-sixteenth of the original image size. Following this, I'_{result} passes through four AOT-Blocks[1], yielding 256 feature maps of the image.These feature maps are then upsampled twice, culminating in the generation of a complete image I_{pred}. The authenticity of I_{pred} is assessed by a discriminator D$_2$, which utilizes an SM-PatchGAN[1] architecture to verify the image's genuineness.

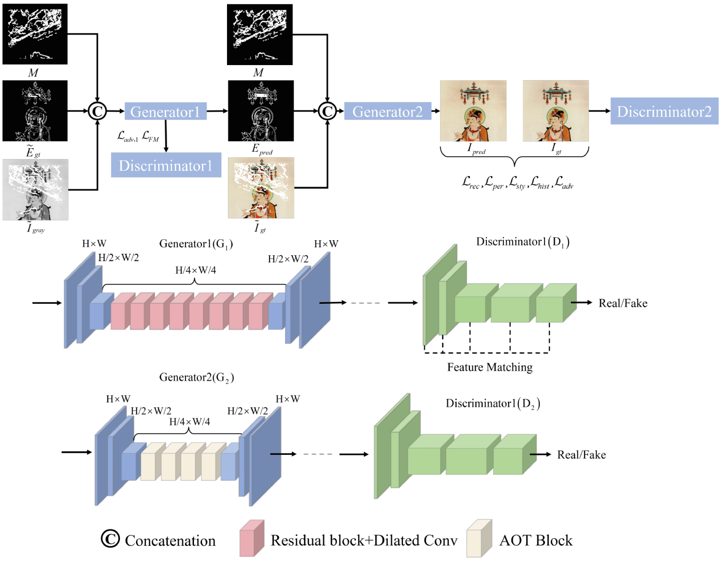


Fig. 1. The architecture of EAAOT-GAN is presented in two main sections: the system overview and the detailed network structure. This model integrates an Edge Generation Network (EGN) and an Image Inpainting Network (IIN). Initially, the EGN reconstruct the full edge image. Subsequently, IIN generates the fully restored mural image.

3.2 Edge Generation Network

This section is dedicated to discussing the Edge Generation Adversarial Network, a crucial component of EAAOT-GAN. Inspired by EdgeConnect[16], the pivotal role of edge information in image inpainting is acknowledged for its substantial contribution to improving restoration quality. Consequently, edge generation constitutes the foundational stage of the EAAOT-GAN framework.

Edge detection Recognizing the enhanced repair capabilities introduced by incorporating edge information, as demonstrated by the use of the Canny edge detector[6] in EdgeConnect, we prioritize generating accurate mural image edges as the initial step in mural restoration. However, the Canny edge detector exhibits limitations, particularly in capturing the complex textures of mural images, often resulting in significant edge information loss, as illustrated in Fig. 2(b).

The acquisition of precise edge information is vital for improving image restoration effectiveness. DexiNed, with its advanced multi-scale feature extraction and fine-grained feature fusion techniques, excels in delineating edges with higher accuracy, making it ideally suited for mural images, as evidenced in Fig. 2(c). Nonetheless, the edges generated by DexiNed are not inherently binary, posing challenges for direct application in the edge generation network and complicating the mural edge restoration process. To circumvent this issue, we employ thresholding to binarize DexiNed's output, as depicted in Fig. 2(d). This approach not only produces optimal edge images but also simplifies the edge restoration task.

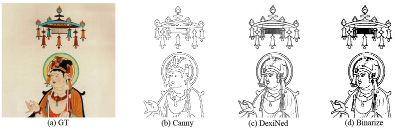

(a) GT
(b) Canny
(c) DexiNed
(d) Binarize

Fig. 2. The original mural image is as displayed in (a), with edges extracted by the Canny edge generator illustrated in (b), edges extracted by the DexiNed edge generator depicted in (c), and the binary edge map shown in (d).

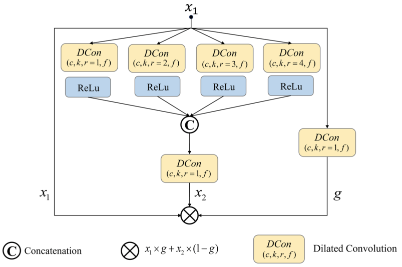


Fig. 3. The structure of AOT-Block. The numbers inside orange blocks denote as input channels, filter sizes, dilation rates and output channels.

Edge generation As widely recognized, M assists the model in concentrating on the obscured or damaged areas, and $\tilde{E}_{gt}$ directly offers prior knowledge regarding the locations of image edges.Inspired by previous works[16][24], we believe that having only M and $\tilde{E}_{gt}$ as inputs to the edge generator is insufficient. Consequently, we incorporate $\tilde{I}_{gray}$ into the edge generator's inputs because grayscale images offer significant insights into variations in image brightness, enabling the edge generator to restore edges with greater precision.

Given that convolution can increase the receptive field of the network without additional parameters and computational costs, residual blocks are effective in alleviating the issue of gradient disappearance. Like EdgeConnect[16], this paper's edge generator uses dilated convolution and residual blocks for repairing image edges, enabling it to effectively capture multi-scale information of image edges, have a wide receptive field, and yet retain accurate spatial positioning capabilities.

To conclude, the output E_{pred} from the edge generator can be expressed as

$$E_{pred} = G_1(\tilde{I}_{gray}, \tilde{E}_{gt}, M) \tag{1}$$

where, $\tilde{E}_{gt} = E_{gt} \odot (1 - M)$ and E_{gt} represent the edge map generated by DexiNed, $\odot$ represents the Hadamard product, $\tilde{I}_{gray}$ similarly.

Since edge maps only provide detailed information of mural images, directly using $\tilde{E}_{gt}$ as the input for D_1 does not effectively distinguish between E_{gt} and $\tilde{E}_{gt}$. To overcome this issue, we introduce I_{gray} that provides overall image information, using E_{gt} and I_{gray}, $\tilde{E}_{pred}$ and I_{gray} as inputs for D_1, to determine whether $\tilde{E}_{pred}$ is a real edge map. By doing so, the performance of D_1 is improved, which in turn promotes better edge map restoration by G_1.

Loss function and Optimization objectives In the edge generation adversarial network, we train the network using adversarial loss and feature matching loss[25].

The adversarial loss is defined as

$$\begin{aligned} \mathcal{L}_{adv,1} = \mathbb{E}_{(\mathbf{E}_{gt}, \mathbf{I}_{gray})} [\log D_1(\mathbf{E}_{gt}, \mathbf{I}_{gray})] \\ + \mathbb{E}_{\mathbf{I}_{gray}} \log [1 - D_1(\mathbf{E}_{pred}, \mathbf{I}_{gray})] \end{aligned} \tag{2}$$

where, $\mathbb{E}$ represents expectation, $D_1(\cdot)$ represents the probability of being classified as real. The symbols E_{gt} and I_{gray}, among others, simply represent inputs.

Feature matching loss is a training technique used in GANs aimed at improving the training process of the generator, making the images it generates more realistic. By introducing feature matching loss, the difference between generated and real images at the feature level can be minimized. This not only enhances pixel-level similarity but also improves the consistency of high-level features, producing edge maps that are closer to E_{pred}.

This loss compares the activations in the intermediate layers (or feature layers) of the discriminator between real and generated images, defined as

$$\mathcal{L}_{FM} = \mathbb{E}\left[\sum_{i=1}^{L} \frac{1}{N_i} \left\| D_1^{(i)}(\mathbf{E}_{gt}) - D_1^{(i)}\left(\mathbf{E}_{pred}\right) \right\|_1\right] \tag{3}$$

where, L represents the final activation layer of the discriminator, N_i represents the number of elements in the i'th activation layer, and $D_1^{(i)}$ is the activation of the i'th layer of the discriminator. According to experiments by Nazeri et al.[16], we know that applying Spectral normalization (SN)[26] to G_1 and D_1, by scaling the weight matrices according to their respective largest singular values, further stabilizes training.

In summary, the optimization objective of the edge generation network is

$$\underset{G_1}{\min}\underset{D_1}{\max}\mathcal{L}_{G_1} = \underset{G_1}{\min}\left(\lambda_{adv,1}\max\nolimits_{D_1}\left(\mathcal{L}_{adv,1}\right) + \lambda_{FM}\mathcal{L}_{FM}\right) \tag{4}$$

wherein, $\lambda_{adv,1}$ and λ_{FM} are constants, set at 1 and 10 respectively, consistent with the parameters used in[16].

3.3 Image Inpainting Network

In this section, we will detail the construction of the image restoration generative adversarial network.

The E_{pred} generated in the edge generation network can provide accurate boundary information for objects and shapes in the image.

With $\tilde{I}_{gt}$, E_{pred} and M as inputs for the image generator, the output I_{pred} from image generator G_2 is expressed as

$$I_{pred} = G_2(\tilde{I}_{gt}, E_{pred}, M) \tag{5}$$

where, $\tilde{I}_{gt} = I_{gt} \odot (1 - M) + M$.

AOT Block The integration of the AOT Block within the image restoration phase significantly enhances the model's context reasoning ability. Unlike standard residual blocks, the AOT Block aggregates multiple context transformations, allowing for a more nuanced inference of output pixel values from diverse perspectives without a corresponding increase in model parameters or computational costs. The AOT Block possesses the ability to capture distant, information-rich image backgrounds and intricate patterns essential for high-resolution image restoration[1]. Consequently, utilizing the AOT Block in mural image restoration facilitates comprehensive recovery of complex textures absent in mural photographs, thereby replacing standard residual blocks during this phase.

When processing input x_1 with a channel count of 256 through the AOT Block, as shown in Fig. 3,the operation unfolds in three stages:

1) Dilated Convolution: x_1 is partitioned into four subsets, each with 64 channels, through dilated convolution, employing four distinct sub-convolution kernels at varying dilation rates to produce intermediate results.
2) Transformation and Series Processing: The intermediate results are serially transformed to yield x_2, integrating the varied perspectives from the sub-convolution kernels.
3) Feature Fusion: x_1 and x_2 undergo fusion, facilitated by a gated residual connection. This connection first computes a spatially-variant gating value via standard convolution and sigmoid operations. The fusion is executed according to $x_3 = x_1 \times g + x_2 \times (1 - g)$,effectively updating features within the missing area while retaining detail in the surrounding regions.

This approach leverages the AOT Block's advanced capabilities for feature aggregation, ensuring the meticulous preservation and enhancement of details throughout the restoration process.

Loss function and Optimization objectives In the image restoration adversarial network, we optimize the training of the network using reconstruction loss[1], perceptual loss[27], style loss[28], adversarial loss[29], and histogram loss[2].

In terms of reconstruction loss[1], we minimize the L_1 distance on a per-pixel basis to ensure the accuracy of the reconstruction, denoted as

$$L_{rec} = \left\| I_{gt} - G_2(\tilde{I}_{gt}, E_{pred}, M) \right\|_1 \tag{6}$$

Regarding perceptual loss[27], we aim to minimize the L_1 distance between feature maps and real images to enhance the accuracy of perceptual reconstruction, expressed as

$$L_{per} = \sum_i \frac{\|\phi_i(I_{gt}) - \phi_i(I_{comp})\|_1}{N_i} \tag{7}$$

where, ϕ_i is the feature map at the ith layer in a pretrained network (for example, VGG19[30]), N_i is the number of elements in ϕ_i, and I_{comp} is the fusion of the restored image I_{pred} with the original mural I_{gt}, denoted as

$$I_{comp} = I_{gt} \odot M + I_{pred} \odot (1 - M) \tag{8}$$

In terms of style loss[28], we minimize the L_1 distance of the Gram matrix of the deep features between the restored image and the real image, denoted as

$$L_{sty} = \mathbb{E}_i \left[\left\| \boldsymbol{\phi}_i(I_{gt})^T \boldsymbol{\phi}_i(I_{gt}) - \boldsymbol{\phi}_i(I_{comp})^T \boldsymbol{\phi}_i(I_{comp}) \right\|_1 \right] \tag{9}$$

Regarding adversarial loss[29], we aim to minimize the L_2 distance between the judgment of I_{comp} by discriminator D_2 and a constant 1, expressed as

$$L_{adv}^G = \mathbb{E}_{I_{comp} \sim p_{I_{comp}}} \left[\left(D(I_{comp}) - 1 \right)^2 \right] \tag{10}$$

The aforementioned four types of loss functions are widely used in most known image restoration models[1][16][31], but there still exist issues of instability in texture synthesis and blurriness in the shape of the restoration results. Histogram loss[2] can effectively address these issues, and we first calculate the histogram matching result $R\left(I_{pred_i}\right)$ between the restored image I_{pred} and the original mural. Afterwards, the minimization of the L_2 norm between I_{pred} and $R\left(I_{pred_i}\right)$ is expressed as

$$L_{hist} = \sum_{l=1}^{L} \gamma_l \| I_{pred_i} - R\left(I_{pred_i}\right) \|_2 \tag{11}$$

In summary, the optimization objective of the image restoration network is

$$L = \lambda_{adv} L_{adv}^{G_2} + \lambda_{rec} L_{rec} + \lambda_{per} L_{per} + \lambda_{sty} L_{sty} + \lambda_{hist} L_{hist} \tag{12}$$

In our experiments, consistent with the experimental parameters used in[1][3], the parameters adopted during training are $\lambda_{adv} = 0.01, \lambda_{rec} = 1, \lambda_{per} = 0.1, \lambda_{sty} = 250, \lambda_{hist} = 0.0005$.

The image discriminator uses the SM-PatchGAN[1] proposed by AOT-GAN to improve the training of the discriminator and to force it to focus more on the central part of the missing area rather than the boundaries, a soft patch-level mask is used in training. The generator adversarial loss is represented as

$$\begin{aligned} L_{adv}^{D} = {} & \mathbb{E}_{I_{comp} \sim p_{I_{comp}}} \left[(D(I_{comp}) - \sigma(M))^2 \right] \\ & + \mathbb{E}_{I_{gt} \sim p_{data}} \left[(D(I_{gt}) - 1)^2 \right] \end{aligned} \tag{13}$$

where, σ represents the operation of processing the mask during training, such as downsampling.

With this, we have completed the entire introduction of this method.

4 Experiments

4.1 Datasets

We have collected known scattered mural datasets[3][4], where available[1][2], and compiled a mural dataset comprising 3492 mural images, mainly featuring characters, Buddha statues, and wrathful deities in murals. We used 3392 images for training and 100 images for testing.

For edge map generation, the DexiNed edge detector was employed. The generated edge maps were subsequently binarized, with the effects of varying binarization thresholds explored to optimize model performance, as detailed in Section 4.6's ablation study.

The free-form mask dataset proposed by Liu et al.[18] has been proven effective in improving the training outcomes of repair models and has been widely adopted by recent repair methods[1][3][16]. We also use this mask dataset for training and testing the EAAOT-GAN.

[1] https://github.com/qinnzou/mural-image-inpainting

[2] https://github.com/WHUT-DCRC/Thangka

4.2 Training Strategy

The training of EAAOT-GAN comprises two distinct stages. Initially, the EGN undergoes training with a batch size of 8 for 50,000 iterations. Early in this phase, generating high-quality edge maps is challenging, rendering any immediate transition to the subsequent stage premature. Upon completing the first stage, the IIN training commences, also with a batch size of 8 and 50,000 iterations. .Throughout the training and evaluation phases, images are uniformly cropped and resized to 512 × 512. Notably, the IIN achieves convergence at approximately 75 epochs.

EAAOT-GAN is deployed on PyTorch 1.10.0 and Cuda 11.3, and trained on two NVIDIA RTX 3090 GPUs. Both the edge generation network and the image repair network are trained using the ADAM optimizer[32] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rates for both the edge generator G_1 and the image restoration generator G_2 are set to 1e-4. The learning rates for discriminators D_1 and D_2 are set to half of that of the generators.

The method of calculating perceptual loss and style loss is adopted from AOT-GAN[1].

4.3 SOTA Models and Evaluation Metrics

The most advanced image restoration model and a brief description of the model are as follows:

- AOT-GAN[1]: A generative adversarial network designed for high-resolution image restoration, leveraging aggregated contextual transformations for efficient and natural outcomes. It uniquely combines diverse contextual information and sophisticated feature transformations.
- MuralNet [3]:Tailored for artistic work restoration, this technique uses line drawings to guide the progressive reconstruction and repair of damaged murals, emphasizing the importance of outlines in restoration.
- EdgeConnect[16]: Targets missing regions in images, employing edge information to guide the restoration process, followed by refinement for final output. This model is notable for its focus on edge-guided restoration.
- DeepFill v2[33]: Enhances image restoration quality by integrating gated convolutions and contextual attention modules, offering significant improvements over its predecessors.
- RFR[34]: A deep learning approach to image restoration, focusing on recurrently inferring and filling missing regions to achieve image completeness.

The quantitative comparison employs established objective metrics, including PSNR, SSIM, and FID, as widely recognized indicators from prior research[1][3][16].

Beyond objective metrics, the performance of the restoration model is also gauged through subjective evaluation to ascertain the model's restoration efficacy. To this end, a comprehensive qualitative evaluation and user studies were undertaken for a thorough comparison.

4.4 Quantitative Comparison

Quantitative comparison of EAAOT-GAN using our created dataset. For each image in the test set, randomly select a mask from the free-form mask dataset. Several leading image inpainting models, including AOT-GAN[1], MuralNet[3], EdgeConnect[16], DeepFill v2[33], and RFR[34], are utilized to restore mural images post-masking and compare the outcomes with those from this model. For fairness of comparison, the same image-mask pairs are used for all methods.

The quantitative comparison results, as shown in Table 1, indicate that EAAOT-GAN exhibits superior performance. The results demonstrate that under conditions of higher mask rates, EAAOT-GAN shows the best performance in terms of PSNR, SSIM, and FID. Compared to our model, EdgeConnect achieves the best results under conditions of lower mask rates, but our model surpasses it in visual effects. As for AOT-GAN, it is completely outperformed by our model in every metric, but it achieves excellent results on the FID metric and outperforms other models at higher mask rates. As a variant of EdgeConnect, MuralNet shows similar outcomes in all metrics and is superior to our model concerning PSNR at lower mask rates. However, it is overall outclassed by our model. Similar to the above models, DeepFill v2 and the RFR model have the worst performance overall.

4.5 Qualitative Comparison

For fair qualitative comparison, we randomly selected restoration results from different models as shown in Fig. 4. Specifically, we compared EAAOT-GAN with state-of-the-art models, including AOT-GAN[1], MuralNet [3], EdgeConnect[4], DeepFill v2[33], and RFR[34].

Given that mural images possess a texture structure far surpassing that of datasets like CelebA[35], and considering the impracticality of restoring severely damaged murals with minimal known information, we opt to add moderate masks to mural images. As shown in Fig. 4, The repair results of DeepFill v2 exhibit significant structural blurring issues, to the point of blurring parts that were originally intact, such as the clear faces of figures. The effectiveness of the repair is worrisome. The restoration results of RFR have issues with texture distortion and incoherent edges, failing to achieve the desired effect. Compared to EdgeConnect, its variant MuralNet has issues with color distortion in the restoration results, but it shows some improvement in detail restoration compared to the two models mentioned above. EdgeConnect and AOT-GAN have issues with incomplete structures and blurriness, but their performance improves compared to the aforementioned models. With the incorporation of edge maps, enhanced generators, and histogram loss, our model surpasses the previously mentioned models in performance, demonstrating minimal blur in our restoration results, capable of reconstructing more plausible contextual structures, and generating sharper textures. In summary, our model's performance is superior in visual effects compared to the aforementioned models.

Table 1. Quantitative results over our dataset with models: AOT-GAN[1], MuralNet[3], EdgeConnect[16], DeepFill v2[33], RFR[34]. ↓Lower is better. ↑Higher is better. The best results are highlighted.

Mask		AOT-GAN	MuralNet	EdgeConnect	DeepFill v2	RFR	Ours
PSNR↑	1-10%	26.39	27.83	**28.37**	28.27	27.27	26.97
	10-20%	23.92	25.32	**25.57**	23.48	24.16	24.25
	20-30%	21.74	22.04	22.04	20.20	21.44	**22.05**
	30-40%	19.92	20.35	20.74	18.33	19.67	**20.82**
	40-50%	18.42	18.62	18.14	16.9	18.16	**18.64**
	50-60%	16.62	15.3	15.71	15.24	16.44	**16.84**
SSIM↑	1-10%	0.87	0.96	**0.97**	0.95	0.95	0.90
	10-20%	0.82	**0.91**	**0.91**	0.89	0.88	0.85
	20-30%	0.75	0.82	**0.84**	0.79	0.79	0.78
	30-40%	0.69	0.73	**0.75**	0.69	0.69	**0.75**
	40-50%	0.61	0.61	0.62	0.59	0.57	**0.63**
	50-60%	0.51	0.43	0.47	0.46	0.43	**0.54**
FID↓	1-10%	11.37	20.11	13.2	19.28	21.6	**9.94**
	10-20%	27.21	39.35	34.06	45.67	52.92	**24.008**
	20-30%	49.97	66.16	63.29	74.07	85.7	**45.67**
	30-40%	74.33	91.58	86.33	99.67	109.82	**68.16**
	40-50%	93.93	115.1	113.62	121.47	136.61	**85.97**
	50-60%	124.56	143.64	171.27	132.38	180.66	**112.02**

4.6 Ablation Experiment

Impact of EGN This section examines the influence of edge generation networks on the effectiveness of mural image restoration. Table 2 presents the comparative outcomes of mural image restoration with and without the incorporation of an edge generation network. The inclusion of this network significantly improves performance across several metrics, particularly in scenarios involving mural images with extensive missing areas.

Impact of edge threshold This section delves into the impact of varying thresholds on the performance of the edge generation network. Specifically, for thresholds set at $80, 96, \ldots, 176$, the DexiNed edge generator's output is binarized before training the EAAOT-GAN model. As illustrated in Fig. 5, optimal performance is observed at a threshold of 128, where metrics such as PSNR peak. This optimal setting is attributed to the balance it strikes: excessively high thresholds may only capture pronounced edges, leading to significant edge detail loss. In contrast, overly low thresholds could saturate the binarized edge map with extraneous details and noise.

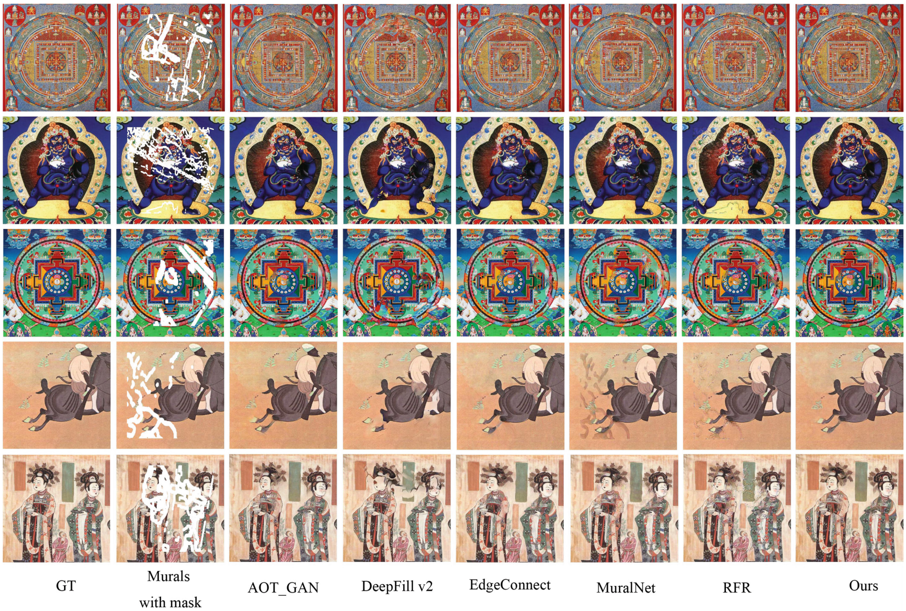


Fig. 4. Inpainting results of five mural images obtained by our method and five comparison ones - AOT-GAN[1], MuralNet[3], EdgeConnect[16], DeepFill v2[33], RFR[34]. As shown in these cases, our model surpasses the previously mentioned models in performance, demonstrating minimal blur in our restoration results, capable of reconstructing more plausible contextual structures, and generating sharper textures.

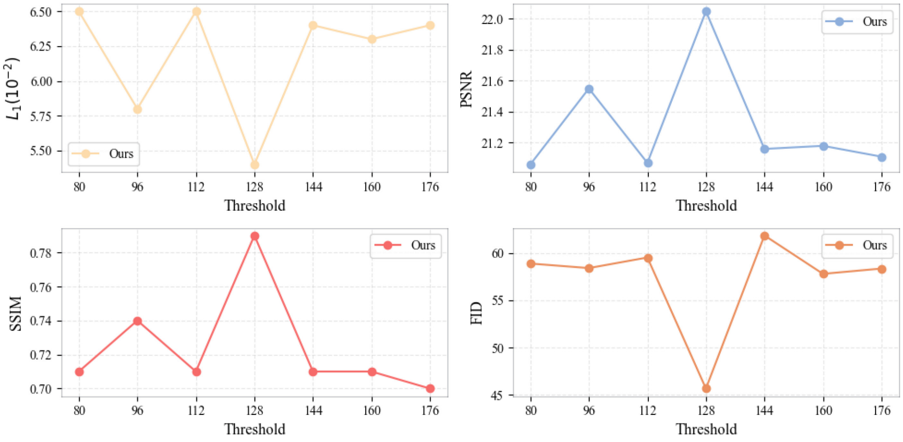


Fig. 5. The figure illustrates how each metric varies with the threshold, highlighting that the model attains optimal restoration outcomes at a threshold value of 128.

Table 2. Comparison of inpainting results with EGN and without EGN. Statistics are based on 100 random masks with size 20-30% of the entire image. ↓Lower is better. ↑Higher is better.

Edges	Our dataset	
	Yes	No
$L_1(10^{-2})\downarrow$	**5.40**	7.15
PSNR↑	**22.05**	21.08
SSIM↑	**0.78**	0.71
FID↓	**45.67**	61.51

Table 3. User study of EAAOT-GAN, EdgeConnect, and AOT-GAN. EAAOT-GAN's restoration results were more accepted by participants.

	Percentage
Ours>EdgeConnect	75.53%
Ours>AOT-GAN	85.53%
Ours>Real	27.18%

Impact of histogram loss It is well recognized that histogram loss effectively mitigates the issue of shape blurriness encountered in high-resolution mural image restoration. This section adopts a qualitative approach to assess the impact of histogram loss on mural restoration outcomes. The experimental findings, depicted in Fig. 6, demonstrate that integrating histogram loss significantly improves the model's restoration capabilities. By incorporating histogram loss, the model notably reduces shape blurriness, resulting in mural images of markedly higher realism.

Restoring real damaged murals The primary aim of investigating mural restoration algorithms is to effectively repair actual murals, thereby aiding in the preservation of cultural heritage. Studies have demonstrated that the EAAOT-GAN model exhibits exceptional structure and performance. This section delves into the model's restoration capabilities on real-world murals. The process begins by applying masks to the damaged regions of the mural, which are then restored using the EAAOT-GAN model. The restoration outcomes, illustrated in Fig. 7, yield highly realistic mural images, providing valuable insights for experts in mural restoration.

4.7 User Study

In our comparative analysis, EdgeConnect and AOT-GAN emerged as strong contenders for mural image restoration, showcasing notable performance. For further evaluation, we conducted a user study selecting 40 images from each of the restoration results of EAAOT-GAN, EdgeConnect, and AOT-GAN, alongside their corresponding original murals. Fifty participants were invited to select the most visually appealing image from each set. The outcomes, detailed in Table 3, highlight a marked preference for EAAOT-GAN's restoration results. Notably, EAAOT-GAN received a 27.18% preference rate, outperforming even the original murals in participant selections.

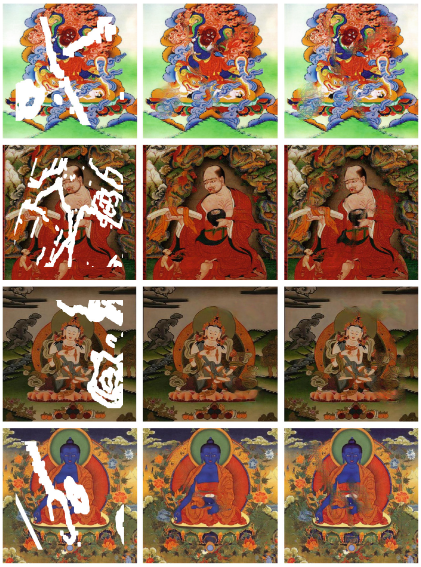


Fig. 6. On the left is the masked mural image, in the middle are the restoration results from the model trained using histogram loss, and on the right are the repair outcomes from the model not trained with histogram loss.

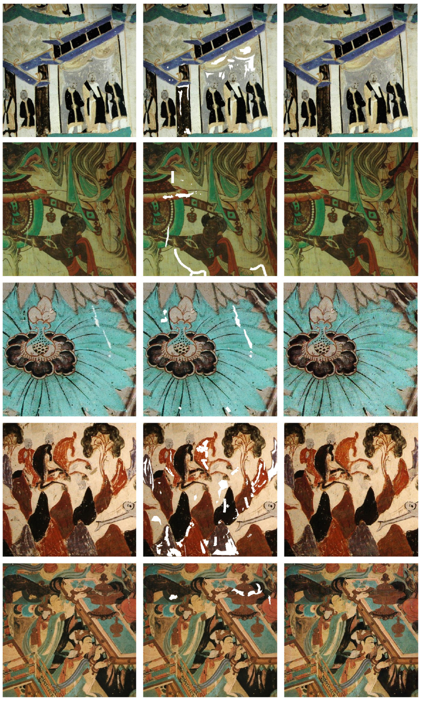


Fig. 7. The left side shows the original image of the damaged mural, the middle shows the mural after adding a mask, and the right side shows the EAAOT-GAN restoration result.

5 Conclusion

In this study, we introduce a mural image restoration model, EAAOT-GAN, which utilizes edge assistance and aggregated contextual transformations. The restoration process is bifurcated into two distinct stages: the EGR initially repairs the mural edges, and then the IIN completes the mural restoration, capitalizing on the reconstructed edge data. Furthermore, we have curated a dataset comprising 3,492 mural images, predominantly featuring Buddha statues. Trained on this dataset, EAAOT-GAN demonstrates exceptional performance in both quantitative metrics and qualitative assessments compared to leading models, positioning it at the forefront of mural image restoration technology.

However, due to the unparalleled complexity of mural images, restorations performed using our model may still exhibit a certain degree of blurriness and texture inconsistency.

In future work, we aim to develop an ensemble of Generative Adversarial Networks (GANs), each specialized in a specific aspect of mural restoration, including color correction, texture refinement, and edge enhancement. By leveraging the combined strengths of these specialized GANs, the ensemble is expected to achieve more comprehensive and nuanced restorations.

Acknowledgements. This research was funded by the national innovation and entrepreneurship training program for college students in China, project No. S202310497171.

References

1. Zeng, Y., Fu, J., Chao, H., Guo, B.: Aggregated contextual transformations for high-resolution image inpainting. IEEE Transactions on Visualization and Computer Graphics (2022)
2. Risser, E., Wilmot, P., Barnes, C.: Stable and controllable neural texture synthesis and style transfer using histogram losses. arXiv preprint arXiv:1701.08893 (2017)
3. Li, L., Zou, Q., Zhang, F., Yu, H., Chen, L., Song, C., Huang, X., Wang, X.: Line drawing guided progressive inpainting of mural damages. arXiv preprint arXiv:2211.06649 (2022)
4. Ma, Y., Liu, Y., Xie, Q., Xiong, S., Bai, L., Hu, A.: A tibetan thangka data set and relative tasks. Image Vis. Comput. **108**, 104125 (2021)
5. Poma, X.S., Riba, E., Sappa, A.: Dense extreme inception network: Towards a robust cnn model for edge detection. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 1923–1932 (2020)
6. Canny, J.: A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence **PAMI-8**(6), 679–698 (1986)
7. Dollár, P., Zitnick, C.L.: Fast edge detection using structured forests. IEEE Trans. Pattern Anal. Mach. Intell. **37**(8), 1558–1570 (2014)
8. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the IEEE international conference on computer vision. pp. 1395–1403 (2015)
9. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. pp. 234–241. Springer (2015)
10. Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous structure and texture image inpainting. IEEE Trans. Image Process. **12**(8), 882–889 (2003)
11. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. **13**(9), 1200–1212 (2004)
12. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Trans. Graph. **28**(3), 24 (2009)
13. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)
14. Nazeri, K., Ng, E., Joseph, T., Qureshi, F., Ebrahimi, M.: Edgeconnect: Structure guided image inpainting using edge prediction. In: Proceedings of the IEEE/CVF international conference on computer vision workshops. pp. 0–0 (2019)

15. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
16. Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019)
17. Ciortan, I.M., George, S., Hardeberg, J.Y.: Colour-balanced edge-guided digital inpainting: Applications on artworks. Sensors **21**(6), 2091 (2021)
18. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)
19. Chen, M., Zhao, X., Xu, D.: Image inpainting for digital dunhuang murals using partial convolutions and sliding window method. In: Journal of Physics: Conference Series. vol. 1302, p. 032040. IOP Publishing (2019)
20. Wang, N., Wang, W., Hu, W., Fenster, A., Li, S.: Thanka mural inpainting based on multi-scale adaptive partial convolution and stroke-like mask. IEEE Trans. Image Process. **30**, 3720–3733 (2021)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
22. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
23. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)
24. Deng, X., Yu, Y.: Ancient mural inpainting via structure information guided two-branch model. Heritage Science **11**(1), 131 (2023)
25. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018)
26. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
27. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
28. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)
29. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2794–2802 (2017)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
31. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5505–5514 (2018)
32. Kinga, D., Adam, J.B., et al.: A method for stochastic optimization. In: International conference on learning representations (ICLR). vol. 5, p. 6. San Diego, California; (2015)

33. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4471–4480 (2019)
34. Li, J., Wang, N., Zhang, L., Du, B., Tao, D.: Recurrent feature reasoning for image inpainting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7760–7768 (2020)
35. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision. pp. 3730–3738 (2015)

Video Analysis Engine for Predicting Effectiveness

Rushil Thareja[1(✉)], Deep Dwivedi[2,3], Ritik Garg[3], Shiva Baghel[3], Jainendra Shukla[2], and Mukesh Mohania[2]

[1] MBZUAI, Abu Dhabi, UAE
rushil.thareja@mbzuai.ac.ae
[2] IIIT Delhi, New Delhi, India
[3] Extramarks Education, Noida, India

Abstract. In the realm of digital education, the growing use of short-form online videos, coupled with innovative generative AI methods, has dramatically expanded the production of didactic academic videos. This shift, however, underscores a critical question - how to ascertain the ”effectiveness” of these videos for student learning? It is essential to devise a classification mechanism that filters videos for clarity, comprehensibility, and their capacity to meet student learning objectives. The automated evaluation of these learning videos holds substantial implications for student academic performance. Accordingly, this paper presents a novel supervised-learning-based approach, predicated on video feature analysis, to predict the effectiveness of K-12 science and mathematics videos. Our method integrates diverse features such as image, spoken text, and audio, among other hand-crafted elements, to accurately assess video effectiveness. We conduct an evaluation of our approach using a comprehensive dataset comprised of 3,134 short-form academic videos. The results demonstrate robust performance, with the system achieving an accuracy of 76.1% and an F1 score of 80.6%.

Keywords: AI for Education · E-learning · Video Processing · Computer Vision · Natural Language Processing · Deep Learning · Multimodal Frameworks

1 Introduction

The widespread use of social media has been associated with shortened student attention spans, a trend corroborated by recent studies [7]. In response to this shifting dynamic, educators are increasingly leveraging short-form content, mirroring a medium that students frequently engage with. Empirical research has demonstrated that this approach can yield learning outcomes that are comparable to, or even exceed, those derived from conventional pedagogical methods [6]. This underscores the significance of short-form content as a countermeasure to digital distraction and attention fragmentation [29,35]. Another appealing aspect of short-form content is its cost-effectiveness and reduced demand for

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 97–112, 2025.
https://doi.org/10.1007/978-3-031-78312-8_7

human resources, facilitating widespread production and dissemination. This is particularly applicable to academic video content with durations of less than two minutes. Additionally, the emergence of publicly accessible Large Language Models (LLMs) and Stable Diffusion-based image generation utilities has further streamlined the content creation process. These advancements, collectively, underscore the burgeoning potential of short-form educational content in today's digital learning landscape.

Academic videos are recognized as effective educational tools [6,7], surpassing text in enhancing learning and retention [5,33]. As online platforms increasingly utilize them, the challenge of quality disparity arises, where inferior content may harm learning outcomes. Given the challenge of maintaining students' focused attention during instructional videos, their time must be respected. Consequently, it's imprudent to overwhelm students with a deluge of generated learning videos as it risks impeding their learning process. This predicament brings into focus theories like *information overload* and *cognitive overload* [8], which occur when individuals confront more information or tasks than their processing capacity can handle, possibly leading to *decision paralysis* [3]. Such overload can also result in diminished interest or motivation, as the task can appear overwhelming or unmanageable. Thus, each instructional video must vie for students' engagement, and only the most pedagogically effective content should be presented to them.

Recent generative AI-based technologies enable content creators to generate videos in bulk, allowing them to experiment with various teaching styles and content designs, potentially transforming educational methods and enhancing content quality. This shift calls for AI-driven assessment of content effectiveness, enabling the identification of the most impactful content for student learning. This approach, utilizing an automated system for evaluating short-form academic video effectiveness, promises to refine educational content creation and selection, aiming to improve student engagement and learning outcomes.

In this paper, video effectiveness in education is defined by its ability to convey educational content effectively, enhance student comprehension, and support academic success. Effectiveness encompasses presentation clarity, content conciseness, logical structure, engagement, and alignment with learning objectives. It focuses on delivering clear, succinct information in an organized manner to captivate interest and meet educational goals. We adopt this holistic approach to define video effectiveness aimed at reflecting the nuanced interplay of various factors that contribute to the overall pedagogical effectiveness of academic videos.

Our approach leverages machine learning and deep learning to assess short-form educational videos, incorporating transfer learning for text, image, and audio feature extraction from best-performing models. It includes heuristic metrics advised by academic video creators, such as subject type, line, and word counts, dropped frames, and content quality via BERT score. Apart from these, we integrate readability metrics Flesch Reading Ease, Flesch-Kincaid Grade, Gunning Fog, and SMOG in our evaluation [13,16,21,25].

We evaluated our methodology using 3,134 AI-created K-12 science and math videos, hand-labeled as effective or not. Features from these videos were used to train various machine learning models (*after feature averaging*), alongside testing with LSTM and Bi-LSTM networks [18,30] with sigmoid attention [34] (*on time series data*). This study focuses on predicting video effectiveness, providing a biased and copyright-free dataset for assessment, and identifying key features that enhance video quality. Our findings guide content creators towards data-informed video production, advancing educational content quality.

The dataset, extracted features, and the code implemented in this paper can be accessed here: **DATA LINK** https://bit.ly/43qR7Df.

2 Related Work

Educational content effectiveness has been the subject of many studies, with various heuristics and metrics used for its evaluation such as those on content quality, social interaction, enjoyment, collaborative learning, and learner control level [14,32]. However, their practical implementation in e-learning is hampered by a critical challenge: the modality of the generated content, primarily videos, differs significantly from traditional learning environments. This difference highlights the necessity for video-based assessment techniques modified specifically for e-learning platforms. Additionally, recent studies have explored student performance as an indicative metric for content effectiveness such as the utilization of performance logs as predictive markers of content efficiency [31] or pre- test and post-test assessment methods [4,23]. The potency of instructional methods has been ascertained through retrospective pre-post evaluations as well [11], and such strategies have found application even in the realm of distance learning [26]. Student performance evaluated through Bayesian knowledge tracing has been used to find challenges in Massive Open Online Courses (MOOCs) and provide predictive evaluations of the respective modeling approach to each challenge [27]. Overall, these approaches posit a direct correlation between content quality and student performance on tests, thereby rooting their assessment in the hypothesis that high-quality content fosters knowledge augmentation, consequently driving test performance enhancement.

Recognizing standardized academic tests as the exclusive markers of content effectiveness might not capture the complexity of student performance entirely, as numerous factors like course interest, motivation, and discipline significantly impact it [20]. Additionally, pre-post research designs, although widely used in evaluation studies, are prone to statistical artifacts like regression to the mean, maturation, history, and test effects, potentially compromising the causal inference between an intervention (content exposure) and outcome (test performance improvement) [24]. Implementing these tests becomes increasingly challenging when scaling the evaluation process, requiring careful questionnaire selection, participant recruitment, ample time for test completion, and an evaluation process for discerning learning outcomes. In an era of large-scale e- learning platforms producing thousands of videos, there is an imperative need for more efficient, automatic methods for evaluating effectiveness. This situation calls for innovative approaches that can handle the vast, dynamic e-learning content while ensuring a reliable and scalable assessment of its effectiveness.

The challenge of automated video evaluation has been tackled within the context of the advertising industry, specifically for video commercials. This approach has successfully incorporated features such as text, color, and audio-visual components [4,23]. Consequently, we have extended these established features and have explored newer deep learning transfer learning techniques. Other studies have also employed multimodal techniques, incorporating physiological measures like EEG, ECG, PPG, and EDA, to understand user responses to content [11,35]. However, such methods, while accurate and insightful, are often deemed too invasive, costly, and time-consuming for large-scale automated deployment on learning platforms, making them less viable for this study.

3 Methodology

The proposed methodology comprises two key stages:

1. Feature extraction.
2. Subsequent binary classification into two distinct classes, denoted as 'ineffective' $\{0\}$ and 'effective' $\{1\}$.

For this paper, we define three subsets for each video v:

- The audio subset $A_v = \{a_{1_v}, a_{2_v}, \ldots, a_{n_v}\}$
- The text subset $T_v = \{t_{1_v}, t_{2_v}, \ldots, t_{n_v}\}$
- The image subset $I_v = \{i_{1_v}, i_{2_v}, \ldots, i_{n_v}\}$

where n denotes the total count of spoken lines within the video. Each element a, t, i corresponds to an ordered subset of frame-level audio, spoken text, and video image, respectively.

We establish a set $S = \{s_1, s_2, \ldots, s_n\}$, where each element s_i designates a distinct sentence in the video, and each sentence spans M_j frames, thus encapsulating the temporal continuity of the video content.

For the j-th sentence in video v, where $j \in \{1, 2, \ldots, n\}$, we define, $a_{j_v} \in A_v$, $t_{j_v} \in T_v$, and $i_{j_v} \in I_v$ represent frame-level audio, text, and image subsets, where:

- $a_{j_v} = \{a_k \mid k \in \{1, 2, \ldots, M_j\}\}$
- $t_{j_v} = \{t_k \mid k \in \{1, 2, \ldots, M_j\}\}$
- $i_{j_v} = \{i_k \mid k \in \{1, 2, \ldots, M_j\}\}$

Each element a_k, t_k, and i_k symbolizes the audio, text, and image data for the k-th frame of sentence j, where $k \in \{1, 2, \ldots, M_j\}$, M_j being the total frames for the j-th sentence. The cardinality of each set is dictated by M_j, capturing the continuity of sentence j across frames.

Let's consider a one-minute video recorded at a frame rate of 30 frames per second. If the video contains 10 sentences (i.e., $n = 10$), each sentence occupies an average of $60/10 = 6$ seconds. Given that the video's frame rate is 30 fps, this implies that each sentence spans over $30 \cdot 6 = 180$ frames (i.e., $M = 180$). Thus, considering all 10 sentences, we accumulate $180 \cdot 10 = 1800$ total data points $(M \cdot n)$, where each data point corresponds to a distinct frame associated with a specific sentence.

To reduce dimensionality and retain pertinent information, we introduce functions f_t, f_i, and f_a, which convert the subsets of frame-level features for each sentence into singular values, representing the overall audio, text, and image characteristics for a given sentence.

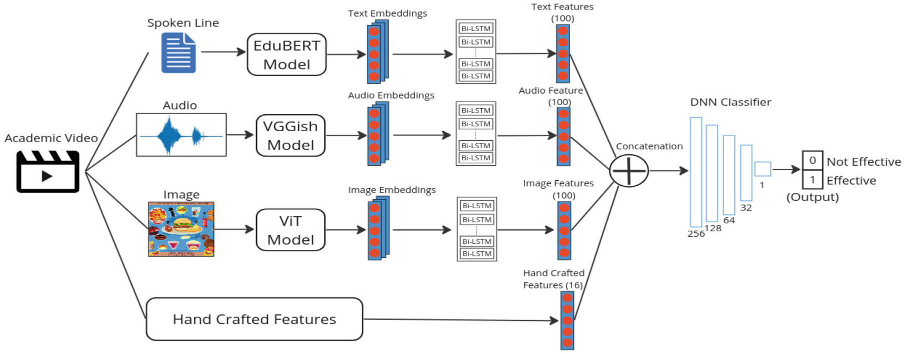


Fig. 1. Proposed deep learning network for predicting video effectiveness.

- *Audio features*: For each sentence j, we have a set of frame-level audio features, $a_{j_v} = \{a_k | k \in \{1, 2, ..., M_j\}\}$. The function $f_a : a_v \rightarrow a'_v$ takes as input this set a_{j_v} and maps it to a single feature vector a'_j. It does this for each sentence j in the video, resulting in a set of sentence-level audio features $F_{A_v} = \{a'_1, a'_2, ..., a'_n\}$.
- *Text features*: Similarly, for sentence j, frame-level text features $t_{j_v} = \{t_k\}$ are mapped by $f_t : t_v \rightarrow t'_v$ to t'_j, leading to $F_{T_v} = \{t'_1, ..., t'_n\}$.
- *Image features*: Finnaly, sentence j has frame-level image features $i_{j_v} = \{i_k\}$, transformed by $f_i : i_v \rightarrow i'_v$ into i'_j, forming $F_{I_v} = \{i'_1, ..., i'_n\}$.

Thus, each set in A_v, T_v, and I_v, which initially contained frame-level features, is now represented by a single feature vector per sentence, substantially reducing the dimensionality of the data. Following the transformation of individual frame-level subsets into respective feature sets $\{F_{I_v}, F_{T_v}, F_{A_v}\}$, we perform another level of abstraction by computing the mean of each set, resulting in singular vector representations associated with a specific sentence. Mathematically, we define three additional functions: $f_{\mu_i} : F_{I_v} \rightarrow F'_{I_v}, f_{\mu_i} : F_{T_v} \rightarrow F'_{T_v}$, and $f_{\mu_i} : F_{A_v} \rightarrow F'_{A_v}$, which map the initial feature sets to their respective mean feature vectors. Formally, we represent these as:

$$F'_{I_v} = f_{\mu_i}(F_{I_v}) = \frac{1}{n} \cdot \Sigma f_i(i), where\ i \in F_{I_v} \qquad (1)$$

$$F'_{T_v} = f_{\mu_i}(F_{T_v}) = \frac{1}{n} \cdot \Sigma f_t(a), where\ t \in F_{T_v} \tag{2}$$

$$F'_{A_v} = f_{\mu_a}(F_{A_v}) = \frac{1}{n} \cdot \Sigma f_a(a), where\ a \in F_{A_v} \tag{3}$$

Here, n denotes the total count of spoken lines in a given video, and Σ represents the summation operator. Consequently, each video v is reduced to a tuple of mean feature vectors $\{F'_{I_v}, F'_{T_v}, F'_{A_v}\}$, encapsulating a video's audio, text, and image modalities. Enriching the feature space, we append another feature vector F'_{M_v} to the tuple, where F'_{M_v} is derived from manually extracted heuristic-based features. Formally, we redefine f as follows:

$$f : \{F'_{I_v}, F'_{T_v}, F'_{A_v}, F'_{M_v}\} \rightarrow \{0, 1\}$$

Consequently, the final representation of each video v is an extended tuple of mean feature vectors $\{F'_{I_v}, F'_{T_v}, F'_{A_v}, F'_{M_v}\}$, which comprehensively encapsulates a video's audio, text, image, and manual heuristic modalities. This strategy yields a singular, vectorized, augmented representation for each video that serves as input to our supervised machine learning models, facilitating the learning of a binary classification function f which distinguishes between *effective* (1) and *non-effective* (0) videos. Nevertheless, this strategy has a shortcoming; the process of averaging the features for a compact representation can inadvertently omit crucial information, notably *temporal dynamics*, and intricate *high-dimensional patterns*. Despite this limitation, this step is indispensable, considering that classical machine learning models function optimally with unidimensional features, making averaging an essential operation.

To rectify this information loss, we utilize time-series models on the sentence-level feature sets, namely $F_{I_v}, F_{T_v}, F_{A_v}$. We then concatenate the learned embeddings from these time-series models with the manually extracted singular feature, F'_{M_v}. The resultant amalgamated feature vector is subsequently processed by a deep neural network, performing the final binary classification of the video's effectiveness. The overview of this architecture is shown in Figure 2.

To ensure the uniformity of the feature set inputs, we standardize the length based on the average number of sentences in the videos (n=15). Videos with sentences exceeding this average are pruned, whereas those with fewer sentences undergo padding with zero-valued features. This ensures a consistent input size across the video corpus, thus facilitating the robust training of our models.

3.1 Text based feature extraction

The function $f_t : t_v \rightarrow t'_v$ is designed to generate a representative feature vector for each sentence in K12 science and mathematics videos, addressing the challenge of domain-specific terminology. Rather than a standard text embedding strategy, f_t leverages a BERT model 1 [9] fine-tuned on Stanford's MOOCPosts dataset [1], supplemented with additional forum data from 18 courses from top UK and US universities. This extensive dataset of approximately 30,000 student forum posts across diverse courses equips each sentence with a contextually rich 768-dimensional embedding vector.

3.2 Image based feature extraction

The function $f_i : i_v \rightarrow i'_v$ employs the Vision Transformer (ViT) model [12] to extract image-based features. ViT adapts the Transformer architecture, typically used for natural language processing, to treat images as sequences of ”word” equivalents, i.e., grids of patches. This usage of self-attention mechanisms makes ViT highly effective for image classification. Given the nature of slide-based didactic videos, we select the median image, displayed at the sentence's midpoint, to extract a 768-dimensional feature embedding via ViT, generating the necessary image-based feature vector. This approach offers scalability, as it can handle more representative frames per sentence by averaging respective embeddings. Moreover, a dedicated model could be employed to select representative frames within a video image clip.

3.3 Audio based feature extraction

The function $f_a : a_v \rightarrow a'_v$ utilizes the VGGish model [17] for processing audio content from K12 science and mathematics videos, extracting representative embeddings that encapsulate aspects like pitch, frequencies, and silences. The VGGish model, a Convolutional Neural Network (CNN), excels at audio data feature extraction and has been trained on the comprehensive Audio Set Dataset [15]. The process converts raw audio WAV files from each sentence into mel spectrograms, visually representing audio frequency over time. The VGGish model, through its CNN architecture, identifies patterns in these spectrograms to create a 128-dimensional feature vector for each audio file, capturing essential audio characteristics.

3.4 Additional hand crafted features extracted for each video

We assessed additional factors affecting academic video effectiveness through a pilot trial with five expert academicians reviewing 20 randomly selected K12 science and math videos. This review identified the following key features contributing to video effectiveness: The *Quantity of Spoken Lines* indicates how much spoken content is in a video, affecting engagement and knowledge retention. *Word Count* measures information density and topic depth. *Dropped Frames Percentage* shows video quality, influencing viewer experience and learning. *Key Content Generation Metrics (BERT score)* assesses content relevance and coherence, essential for learning. *Readability Measures* like Flesch Reading Ease, Flesch-Kincaid Grade, Gunning Fog, and SMOG Index [13,16,21,25] determine transcript complexity, where simpler language enhances accessibility and learning outcomes. We also include the video's academic subject (e.g., physics, mathematics) in our feature set to include possible correlations between the content's domain and its effectiveness. These features are used to train supervised classification models to predict content efficacy.

4 Dataset

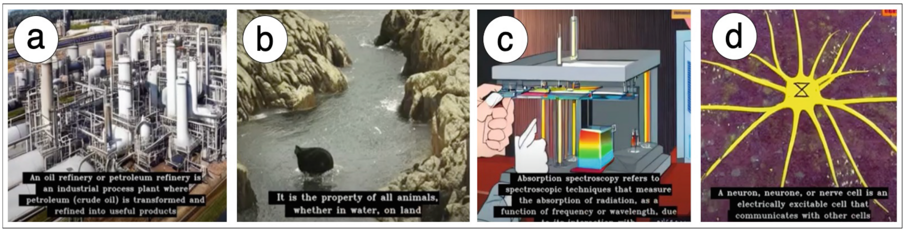

a
An oil refinery or petroleum refinery is an industrial process plant where petroleum (crude oil) is transformed and refined into useful products
b
It is the property of all animals, whether in water, on land
c
Absorption spectroscopy refers to spectroscopic techniques that measure the absorption of radiation, as a function of frequency or wavelength, due
d
A neuron, neurone, or nerve cell is an electrically excitable cell that communicates with other cells

Fig. 2. Some generated video frames for various topics: a. Petroleum Refining, b. Locomotion in Animals, c. Spectroscopy, d. Neuron.

We evaluated our method through a dataset of 3,134 academic videos (μ=65.4s), algorithmically produced. These videos were generated for specific academic topics from *mathematics, biology, physics, and chemistry* subjects, derived manually from online learning resources and K-12 textbooks by our annotators. To produce bias and copyright-free content in the traditional academic video format, we utilized a generative AI process. The video scripts were generated by a Large Language Model (LLM) fine-tuned with topic-video transcript pairs (10458 academic videos, 743938 lines of content) from *in-house* K-12 academic videos. This model was designed to create academic video transcripts based on input topics. Our tests involved three popular open-source LLMs with 7 billion parameters: llama2 [1], falcon [2], and wizard [3]. After a pilot study of 10 videos per subject, Falcon was selected due to its lowest factual error rate of 13.5%, compared to Wizard's 18.83% and Llama2's 20.5%, as determined by our annotators after academic fact extraction and assessment.

Corresponding images were created for each spoken line by the CompVis stable diffusion model [28], also fine-tuned on the same video dataset after extracting text-image pairs from videos by selecting the median frames for each spoken line. To supplement the text and images with spoken audio, we utilized the neural speech, Text2Speech model [22]. We then combined all the generated elements: audio, image, and text through an automated editing pipeline to form cohesive short-form academic videos to test our proposed system. Our methodology aligns with the evolving use of generative AI in the e-learning industry and guarantees copyright compliance, scalability, and comprehensive control over content. We chose this approach as it is expected that generative techniques are going to be crucial components of academic video generation in future. This approach not only fosters research reproducibility but also contributes to the future trajectory of generative AI advancements in education. The expansive dataset used ensures

[1] https://ai.meta.com/llama/

[2] https://falconllm.tii.ae/

[3] https://huggingface.co/WizardLM/WizardLM-7B-V1.0

diverse coverage of video types, subjects, and content. This diversity bolsters the generalizability of our methods for determining the effectiveness of various short-form academic videos. Therefore, the systematic approach proves crucial for our examination of academic video effectiveness. *The generated videos are available in the provided data and code folder that we have released alongside this paper.*

4.1 Annotations

To evaluate video effectiveness, each video was labeled as either 0 (non-effective) or 1 (effective). A panel of ten subject experts, each proficient in a K12 subject - precalculus, geometry, physics, biology, or chemistry, annotated the videos. For robustness, each subject's videos were evaluated by two specialists. This process was supervised by an experienced head teacher with a decade-long expertise in video content creation, enhancing the evaluation's credibility. Cohen's kappa scores, reflecting fair to good inter-annotator agreement, were recorded as 0.55, 0.61, 0.59, 0.58, and 0.64 for the respective subjects. The evaluators focused on content clarity, appropriateness of visual representation, audio quality, pace of delivery, and the relevance and coherence of the content.

5 Experiments and Discussion

5.1 Evaluating machine learning models

To infer the binary classification function $f : \{F'_{I_v}, F'_{T_v}, F'_{A_v}, F'_{M_v}\} \rightarrow \{0, 1\}$, we concatenate the feature vectors $F'_{I_v}, F'_{T_v}, F'_{A_v}$, and F'_{M_v}, each reflecting image-based, text-based, audio-based, and heuristic-based features, respectively, across all 3,134 videos. Following this, we deploy 18 distinct machine learning models using the unified ML platform Pycaret [2], training them on the integrated feature vector of size 1680 and evaluating their proficiency through 5-fold cross-validation. We contrast these results with two random baselines: the first allocating equal probabilities to both classes and the second distributing probabilities conforming to the class occurrence in the prior, P(C=0) = 0.415 and P(C=1) = 0.585. As depicted in Table 1, the top five models surpass the aforementioned random baselines. Of all models, the LightGBM model [19] proves to be the most efficacious in predicting the effectiveness of academic videos, delivering the highest accuracy and F1 scores. These metrics hold considerable importance in balanced binary classification tasks as they collectively portray the model's performance. While accuracy denotes the proportion of accurate predictions, the F1-score offers a harmonized measure of precision and recall. The outstanding performance of LightGBM can be attributed to its prowess in discerning complex feature interactions and non-linear relationships inherent in our multi-modal feature set, achieved through a gradient boosting framework leveraging tree-based learning algorithms.

Table 1. Comparison of top 5 performing Machine Learning models and random baselines using 5-fold cross validation.

Model	p	r	f1	a
LightGBM	**73.71***	84.73	**78.81***	**73.23***
Extra Trees	71.69	**85.66***	78.04	71.68
Gradient Boosting Classifier	72.70	82.79	77.39	71.54
Random Forest	71.43	84.50	77.40	71.00
Logistic Regression	70.41	74.34	72.29	66.48
Random Baseline 1	52.0	50.0	51.0	50.0
Random Baseline 2	50.0	50.0	50.0	50.0

5.2 Feature Analysis

In order to analyze our extracted features, we extract feature importance (*feature ranking coefficient*) from the trained LightGBM models, LightGBM calculates feature importance as ”split” importance, which is determined by the frequency and effectiveness of a feature used for data splitting across all model trees. We also showcase the Pearson correlation coefficient [10], which measures linear relationships between variables in Table 2. Our research sheds light on the complexities of determining academic video effectiveness. Feature 1, signifying the technical quality of the video, contributes significantly to the viewing experience. A lower value here leads to smoother, uninterrupted video playback, crucial for effective learning. Feature 2 measures the relevance of the content. Higher similarity to the prompt ensures that the content is in alignment with the intended topic, thereby increasing its educational value. Meanwhile, Feature 3 underlines the preference for concise content, aligning with the trend of short educational videos. Features 4 and 5 represent readability metrics, implying that a balance between content complexity and linguistic simplicity is beneficial for video effectiveness. Contrarily, lower-ranked features such as Feature 14 or Feature 11, show moderate to low correlations. Nonetheless, these features can influence the model's decisions due to their interaction with other features. This collective contribution of all features, despite their individual rankings or correlations, underscores the intricate nature of assessing video effectiveness.

5.3 Insights for educational content creators

The feature analysis in this research illuminates several key facets of effective educational video content that bear notable implications for educators and video creators. A critical insight lies in Feature 1, the percentage of dropped, corrupt or black frames, suggesting that high-quality video production goes beyond aesthetic considerations and significantly impacts the learning experience. Maintaining seamless video streams, free from interruptions, facilitates an immersive learning environment. *Content relevance*, as represented by Feature 2 (F1 of BERT score between video transcript and prompt), is found to be of considerable importance. With the contemporary educational landscape characterized

Table 2. Feature correlations and lightGBM extracted rankings for each feature.

F. no	Feature	Feature ranking coefficient	Feature Correlation
1	Percentage of dropped corrupt or black frames	11	0.165
2	F1 of BERT score b/w video transcript	11	0.216
3	Duration of the video in seconds	5	0.09
4	Recall of BERT score b/w video transcript	4	0.093
5	Flesch-Kincaid Grade of video transcript	4	0.157
6	Number of words in the transcript	2	0.139
7	Precision of BERT score b/w video transcript	2	0.231
8	SMOG index of video transcript	2	0.164
9	Average text embedding from EduBERT model	1.959	0.183
10	Average image embedding from ViT model	1.779	0.152
11	Flesch Reading Ease of video transcript	1	0.187
12	Average audio embedding VGGish model	0.671	0.248
13	One hot encoding of video academic subject	0	0.138
14	Number of spoken lines in the video	0	0.288
15	Gunning Fog index of video transcript	0	0.209

by information abundance, learners value content that exhibits close alignment with the intended academic objectives. Video creators should therefore accord high priority to the preparation of concise, focused content. *Language aspects*, as delineated by Features 5 (Flesch-Kincaid Grade of video transcript), 8 (SMOG index of video transcript), and 11 (Flesch Reading Ease of video transcript), emerge as pivotal. The research suggests a balance where intellectual stimulation is offered through accessible language, enhancing the effectiveness of the video content.

Subtler aspects, often overlooked, such as Feature 14 (Number of spoken lines in the video), are revealed to impact video effectiveness by affecting the cognitive load on students. It highlights the importance of comprehensive planning, where scriptwriting, pacing, and visual aids are all deliberately crafted. Overall, the analysis posits that an effective educational video, from the viewpoint of students, is a multifaceted construct. It involves the judicious interplay of technical quality, content precision, linguistic accessibility, and attention to minor, yet significant, details. These insights should guide content creators towards producing videos that are not merely informative, but engaging and enjoyable for students.

5.4 Evaluating deep learning models

We utilize the feature sets $F_{I_v}, F_{T_v}, F_{A_v}$, where F_{I_v} for the training of time-series deep learning models. These features represent the set of image features, F_{T_v} the set of text features, and F_{A_v} the set of audio features, each at the sentence level for each video v. Thus, each set is represented as $F_{X_v} = x_1, x_2, ..., x_n$, where x_i is the feature vector of the i^{th} sentence and n is the total number of sentences in the video. Given that the average number of sentences in a video is 15, we refine each feature set by either truncating or padding the set to ensure a consistent size. Formally, for each F_{X_v}, we apply the following operations:

$$Fnew_{X_v} = \begin{cases} \{x_1, x_2, ..., x_{15}\}, & \text{if } |F_{X_v}| > 15 \\ \{x_1, x_2, ..., x_n, 0, ..., 0\}, & \text{if } |F_{X_v}| < 15 \end{cases} \tag{4}$$

where 0 represents the zero-padding feature vector and $|F_{X_v}|$ represents the cardinality of set F_{X_v}.

Consequently, the processed feature subsets $Fnew_{I_v}$, $Fnew_{T_v}, Fnew_{A_v}$ each contain exactly 15 elements. We test both LSTM and Bi-LSTM networks, utilizing 15 cells for each video sentence. The architecture of one such model, termed as the *dual Bi-LSTM with attention*, consists of the input passing through a bidirectional LSTM layer of 15 units, followed by a sequence self-attention layer with sigmoid activation, another LSTM layer of 15 units, and ultimately a dense layer with 100 units, featuring ReLU activation.

Table 3. Comparative analysis of the performance of various Deep Learning architectural models.

Model	**Precision**	**Recall**	**F1 Score**	**Accuracy**
Dual Bi-LSTM with attention	76.4	**85.8***	**80.6***	**76.1***
Single Bi-LSTM with attention	**77.1***	84.5	80.1	75.9
Dual Bi-LSTM without attention	76.6	84.3	80.3	75.8
Single Bi-LSTM without attention	76.6	84.0	80.1	75
Random Baseline 1	52.0	50.0	51.0	50.0
Random Baseline 2	50.0	50.0	50.0	50.0

In a separate experimental setup, we develop an alternative architecture, the *single Bi-LSTM with attention*, which routes the input through a Bi-LSTM layer of 15 units, followed by a sequence self-attention layer with sigmoid activation. This output is then subjected to global max pooling and subsequently processed by a dense layer of 100 units with ReLU activation.

Moreover, we assess the impact of the attention layer by trialling both the *dual Bi-LSTM with attention* and *single Bi-LSTM with attention* architectures without the attention layer, resulting in the *dual Bi-LSTM without attention* and *single Bi-LSTM without attention* models, respectively.

We use these time series models separately to extract singular vectorized representations of size 100 for each of the features $Fnew_{I_v}, Fnew_{T_v}, Fnew_{A_v}$ as shown in Figure 2. These vectors have shape 15x768, 15x768 and 15x128 representing the image, text and audio embedding for each sentence (n=15) respectively. Then the manually extracted features F'_{M_v} of size 16 are concatenated with these 3 feature vectors of size 100 each to reach the final video embedding vector of size 316. The embedding is fed into a deep neural classification network that begins with a dense layer of 256 units with ReLU activation, followed by a dropout layer with a rate of 0.5. It then sequentially applies a dense layer of 128 units, a dropout layer (0.5 rate), a dense layer of 64 units, a dropout layer (0.2 rate), and a dense layer of 32 units-all with ReLU activation. The architecture concludes with a final output layer of 1 unit with sigmoid activation.

The sigmoid activation function is used in the final layer of a binary classification model because its output is a value between 0 and 1, which can be interpreted as the probability of the positive class. This allows the model to make a clear binary decision between the two effectiveness labels i.e. 0 not effective and 1 effective. Additionally, we add Dropout as a regularization technique used in the architecture to prevent overfitting by randomly setting a fraction of input neurons to 0 during training, which encourages a more robust, generalized model.

To validate our methodology, we employ 5-Fold cross-validation on our input dataset, comprising 3,134 videos. Table 3 presents our findings, demonstrating that the *dual Bi-LSTM with attention* model provides the most commendable performance, securing an accuracy rate of 76.1% and an F1 score of 80.6%. This outcome is evaluated against the same random baselines that we previously utilized for the assessment of the machine learning models. The first baseline allocates equivalent probabilities to both classifications, while the second distributes probabilities according to the frequency of class occurrence in the prior dataset.

Our empirical investigation vividly illustrates a clear performance demarcation between the traditional machine learning approaches and their deep learning counterparts. Machine learning models, largely dependent on averaged vector representations, typically fail to capture the intricate temporal dynamics inherent in sequence data. This approach inadvertently omits a wealth of valuable information, leading to sub-optimal performances. Contrastingly, deep learning architectures, particularly Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM), are proficient in extracting and capitalizing on such temporal dependencies, thereby manifesting in superior predictive performances.

Upon analysis of the empirical results delineated in Table 3, the dominance of deep learning models, specifically the dual Bi-LSTM model with attention, is evident. This model achieves an accuracy of 76.1% and an F1 score of 80.6%, significantly outperforming other assessed architectures. Closely following is the single Bi-LSTM with attention model, registering an accuracy of 75.9% and an F1 score of 80.1%. These results provide compelling evidence for two pivotal insights: firstly, the indispensability of the attention mechanism, and secondly, the effectiveness of the bidirectional architecture. Comparing the models with and without attention, it is apparent that both the dual and single Bi-LSTM models with attention supersede their non-attention counterparts in terms of both accuracy and F1 score. This showcases the efficacy of the attention mechanism, corroborating its ability to allocate focus on salient temporal features and thereby enhancing model performance. In addition, the dual Bi-LSTM model's superior performance highlights the benefits of utilizing multiple LSTM layers.

This dual-layer system operates hierarchically, with the first layer discerning low-level temporal dynamics and the second layer extrapolating these to complex, high-level temporal features. Such hierarchical temporal processing proves essential for effective pattern recognition in situations characterized by complex and non-linear temporal dependencies, which are prevalent in the current task of predicting academic video effectiveness.

6 Limitations & Future Work

For the purpose of this study, we utilize synthetically generated datasets. This approach may introduce certain limitations, such as inherent selection bias in the generated videos due to the nature of the training data and the possibility of hallucinations in text generated by large language models (LLMs). To address these concerns, we propose the procurement of commercially available real-world academic videos from various domains on which our proposed methodology can be implemented. It is important to note, however, that since the videos are generated from in-house animated academic videos, the models trained on this data are expected to generalize well to new forms of real-world data in similar domains. Additionally, as video embedding methods, particularly those based on transformers, continue to evolve, we will adopt more advanced spatio-temporal deep learning methods to encode and predict classes. This will also include new annotations for continuous values of academic video effectiveness, moving beyond binary ”*yes*” or ”*no*” prediction values. Additionally, as we look forward, our intention is to expand this methodology to a broader array of academic videos, potentially extending to those of greater length.

7 Conclusion

Our research presents a robust technique for predicting the effectiveness of short-form academic videos, utilizing a blend of image, text, audio, and other features. This methodology is evaluated using a dataset composed of 3,134 AI-generated videos. To underscore the practical applicability of our methodology, we have calculated the feature extraction time, which is less than 10ms per sample on A100 GPUs on average. Moreover, we have noted that the inference time for our machine learning models is exceptionally low (¡¡ 1 ms), highlighting the efficiency of our approach. Furthermore, we undertake a thorough examination of the features, determining their rankings and correlations with the aid of the best-performing machine learning model. This exploration substantiates the significance of factors such as content relevance, proficient language use, and meticulously crafted scripts in generating impactful academic videos. We envision our work as a catalyst, kindling the development of more nuanced content creation strategies and accelerating the emergence of sophisticated AI-based content evaluation techniques.

Acknowledgment. The primary author would like to extend thanks to the NLP department at MBZUAI and the department chair Professor Preslav Nakov, for their support.

References

1. Agrawal, A., Paepcke, A.: The stanford moocposts dataset. Accessed: Dec **15**, 2020 (2014)
2. Ali, M.: PyCaret: An open source, low-code machine learning library in Python (April 2020), https://www.pycaret.org, pyCaret version 1.0.0
3. Bawden, D., Robinson, L.: Information overload: An overview (2020)
4. Bhanji, F., Gottesman, R., de Grave, W., Steinert, Y., Winer, L.R.: The retrospective pre-post: A practical method to evaluate learning from an educational program. Acad. Emerg. Med. **19**(2), 189–194 (2012)
5. Boateng, R., Boateng, S.L., Awuah, R.B., Ansong, E., Anderson, A.B.: Videos in learning in higher education: assessing perceptions and attitudes of students at the university of ghana. Smart Learning Environments **3**, 1–13 (2016)
6. Brame, C.J., et al.: Effective educational videos (2015)
7. Chassiakos, Y., Radesky, J., Christakis, D., Moreno, M., Cross, C., Hill, D., et al.: Children and adolescents and digital media. pediatrics [internet]. 2016 nov 1 [cited 2021 jun 9]; 138 (5)
8. Chung, D., Chen, Y., Meng, Y.: Perceived information overload and intention to discontinue use of short-form video: The mediating roles of cognitive and psychological factors. Behav. Sci. **13**(1), 50 (2023)
9. Clavié, B., Gal, K.: Edubert: Pretrained deep language models for learning analytics. arXiv preprint arXiv:1912.00690 (2019)
10. Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. Noise reduction in speech processing pp. 1–4 (2009)
11. Davis, G.A.: Using a retrospective pre-post questionnaire to determine program impact. (2002)
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
13. Flesch, R.: A new readability yardstick. J. Appl. Psychol. **32**(3), 221 (1948)
14. Garzotto, F.: Investigating the educational effectiveness of multiplayer online games for children. In: Proceedings of the 6th international conference on Interaction design and children. pp. 29–36 (2007)
15. Gemmeke, J.F., Ellis, D.P., Freedman, D., Jansen, A., Lawrence, W., Moore, R.C., Plakal, M., Ritter, M.: Audio set: An ontology and human-labeled dataset for audio events. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 776–780. IEEE (2017)
16. Gunning, R.: The fog index after twenty years. J. Bus. Commun. **6**(2), 3–13 (1969)
17. Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., et al.: Cnn architectures for large-scale audio classification. In: 2017 ieee international conference on acoustics, speech and signal processing (icassp). pp. 131–135. IEEE (2017)
18. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)

19. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems **30** (2017)
20. Killen, R.: Differences between students' and lecturers' perceptions of factors influencing students' academic success at university. High. Educ. Res. Dev. **13**(2), 199–211 (1994)
21. Kincaid, J.P., Fishburne Jr, R.P., Rogers, R.L., Chissom, B.S.: Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel (1975)
22. Lee, S.g., Kim, H., Shin, C., Tan, X., Liu, C., Meng, Q., Qin, T., Chen, W., Yoon, S., Liu, T.Y.: Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. arXiv preprint arXiv:2106.06406 (2021)
23. Madeni, F., Horiuchi, S., Iida, M.: Evaluation of a reproductive health awareness program for adolescence in urban tanzania-a quasi-experimental pre-test post-test research. Reprod. Health **8**(1), 1–9 (2011)
24. Marsden, E., Torgerson, C.J.: Single group, pre-and post-test research designs: Some methodological concerns. Oxf. Rev. Educ. **38**(5), 583–616 (2012)
25. Mc Laughlin, G.H.: Smog grading-a new readability formula. J. Read. **12**(8), 639–646 (1969)
26. Michelazzo, M.B., Pastorino, R., Mazzucco, W., Boccia, S.: Distance learning training in genetics and genomics testing for italian health professionals: results of a pre and post-test evaluation. Epidemiology, Biostatistics and Public Health **12**(3) (2015)
27. Pardos, Z., Bergner, Y., Seaton, D., Pritchard, D.: Adapting bayesian knowledge tracing to a massive open online course in edx. In: Educational Data Mining 2013. Citeseer (2013)
28. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
29. Rosen, L., Samuel, A.: Conquering digital distraction. Harv. Bus. Rev. **93**(6), 110–113 (2015)
30. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
31. Shukor, N.A., Tasir, Z., Van der Meijden, H.: An examination of online learning effectiveness using data mining. Procedia. Soc. Behav. Sci. **172**, 555–562 (2015)
32. Ssemugabi, S., De Villiers, M.: Effectiveness of heuristic evaluation in usability evaluation of e-learning applications in higher education. South African computer journal **2010**(45), 26–39 (2010)
33. Stockwell, B.R., Stockwell, M.S., Cennamo, M., Jiang, E.: Blended learning improves science education. Cell **162**(5), 933–936 (2015)
34. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
35. Webster, J.G., Ksiazek, T.B.: The dynamics of audience fragmentation: Public attention in an age of digital media. J. Commun. **62**(1), 39–56 (2012)

A Stratified Pipeline for Vehicle Inpainting in Orthophotos

Benedikt Kottler(✉), Kevin Qiu, Gisela Häufel, and Dimitri Bulatov

Fraunhofer IOSB Ettlingen, Gutleuthaus Str. 1, 76275 Ettlingen, Germany
Benedikt.Kottler@iosb.fraunhofer.de

Abstract. The paper outlines a pipeline for the removal of transient objects from orthophotos to enhance the clarity and utility for orthophotos in military and civilian geo-databases generation as the main application. The presented deep-learning-based pipeline includes detecting the objects of interest, masking them out, and using the image and an enhanced inpainting mask to fill in these areas seamlessly. The approach combines semantic segmentation, utilizing an adapted DeepLabv3+ model, with shadow detection using Particle Swarm Optimization, and concludes with a generative inpainting process using a three-stage Generative Adversarial Network (3GAN) system for edge, segmentation, and texture inpainting. This method is applied to a well-known remote sensing dataset for detailed analysis, highlighting the integrated approach's effectiveness in creating realistic, cleaned-up orthophotos.

Keywords: 3GAN · DeepLabv3+ · Inpainting · Particle Swarm Optimization · Shadow Detection · Image Enhancement

1 Introduction

At last with the advent of OpenAI's ChatGPT and the global circulation of images featuring the Balenciaga Pope created by Midjourney[1], generative DL-based techniques have become widely recognized. The underlying task can be formulated in a surprisingly simple, non-technical manner. Given some image content, the objective is to refill the missing parts of the image as realistically and plausibly as possible. The process of retrieving missing information or cleaning images of disturbing foreground objects is called inpainting, a field in which generative techniques have made tremendous progress in recent years. Beyond their entertainment value, these techniques have several practical applications. Removing an ex-boyfriend from the photo album may provide more peace and harmony in certain families; realistic representations of building walls from publicly available images are important for the automatic computation of energy balances [10] as well as for virtual tourism [31]. But also in the field of remote sensing, cleaning geographic data from objects that frequently change in appearance can be used for automatic camouflage detection and preparation for rapid

[1] https://time.com/6266606/how-to-spot-deepfake-pope/

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 113–128, 2025.
https://doi.org/10.1007/978-3-031-78312-8_8

response missions [9] in the military, and for virtual tourism in civilian applications [17]. In this context, the images to be cleaned are large geo-data products, such as orthophotos, and the frequently occurring foreground objects to be removed are mostly vehicles and persons.

Strictly speaking, there are three problems to be solved: finding the objects of interest, masking them out, and using the image and the resulting mask for inpainting. The second step may appear trivial, because inpainting masks can be created by binarizing the classification results, but sometimes the undesired objects cast shadows that may not be included in the region an automatic method is supposed to overwrite. Consequently, these shadows can remain as disturbing artifacts in the final product, betraying its artificial nature. The first and last steps are usually performed in a stratified manner [24,39], not only because object detection or segmentation might be required at a later stage for a higher-level application [9] but also the classification result can be helpful for achieving realistic generated images, as has been shown for many state-of-the-art approaches [12,16,19,34]. Especially, this last approach exploits the potential of generative techniques by applying three GANs (Generative Adversarial Networks): the first for the inpainting of edge images, the second for the inpainting of (semantic) segmentation images and the third for the inpainting of texture images. This network has been successfully applied at the façade images, partially occluded by vegetation, road signs and other objects.

In this paper, with geo-databasis generation as the application in mind, we wish to apply the 3GAN method of [16] to the orthophoto with the aim to inpaint vehicles of all kinds. The three necessary inputs for the 3GAN method, namely, an RGB image, the land cover classification result, and the vehicle mask are extended by the shadow detection result, which is accomplished by means of Particle Swarm Optimization. Thus, aside from the application of the 3GAN approach to a different use-case, the main contribution is a stratified approach comprising a semantic segmentation and a generative inpainting module. For both modules, the relevant related work, methodology, and results will be discussed in Sections 2, 3, and 4, respectively. The conclusions are drawn in Section 5.

2 Previous work

Both subsections of this section will focus on deep-learning-based methods because they became state of the art in many tasks of object detection and semantic segmentation due to their universality.

2.1 Land cover classification and vehicle detection

Probably, the first approach developed on vehicle detection from remote sensing data using CNN techniques was that of [6] who extracted multi-scale features and combined it with a modified sliding window technique. Furthermore, [1] proposed extraction of deep features from segments and classification of these features using SVM. The authors have built on the progress in fully-convolutional networks and residual learning to perform accurate segmentation of object borders. In their semantic boundary-aware multitask learning network, detection and segmentation of vehicle instances were trained simultaneously. At the same time, it has become popular to detect small objects using a pyramid-based network with convolutional down-sampling as well as deconvolutional upsampling layers [36]. In [29], there were results for coarse object segmentation and fine delineation (that is, a pyramid with only one layer), which made indispensable a module for single vehicle extraction. Here, the authors used optical and elevation-based features within a pre-trained pseudo-Siamese network. The elevation data can also be considered as a post-processing routine [27]. Two contributions [35] and [22] rely on instance segmentation. The hyper-region proposal network [35] aims at predicting all the possible bounding boxes of vehicle-like objects with a high recall rate, followed by a cascade of boosted classifiers aiming at eliminating spurious detections by including them into the loss function (hard negative example mining) in explicit way. The authors of [22] focus on the preparation of training data reducing the category imbalance of different vehicle types, amplifying the features compensating pooling-based artifacts, and considering a center loss function to distinguish different types of vehicles. The authors of [15] applied a super-resolution convolutional neural network to train the detection of vehicles in an end-to-end manner. In the work of [3], a modification of DeepLabv3+ [5], aimed at recognizing fine-grained features, is proposed. A generalized Zero-shot learning framework is applied for recognition of previously unseen vehicles.

Overall, progress made on fully-convolutional networks, equipped either with encoder-decoder structures, with U-like skip connections, or atrous convolutions [5], nowadays help to overcome pooling artifacts within the state-of-the-art land cover classification pipelines, such as [21].

2.2 Inpainting

According to e.g., [8], where more details and sources can be found, inpainting methods based on deep learning can be roughly subdivided into two groups: based on pixel-filling predictions [18,20,25,26,38] and on GANs [14,23,32,37]. The authors of [37] propose to optimize the inpainting result by finding the best matching neural patches between the inpainting area and the given context, after which a multi-scale structure is applied to refine the texture in an iterative way to achieve the high-resolution performance. It could predict photo-realistic results, but the inference takes much more time than other methods. Another example of a method focusing on coarse-to-fine improvement of the synthesized texture is [33]. The context-encoder method of [25] is probably the earliest GAN-based method on inpainting and is based on an encoder-decoder architecture. Because of multiple pooling layers, fine details can hardly be reconstructed. To cope with high-resolution images, [13] adds dilation convolutional layers and applies Poison blending to smooth occasionally occurring artifacts. The EdgeConnect method was developed [23], which aims to reconstruct such good edges sketch automatically. There are two GANs, whereby the first GAN learns to complete the edge image of an RGB image. The edges serve as a-priori information for the second GAN, supposed to reconstruct the color image. Thus, the image structure in the edge image is captured with the first GAN, while the second GAN focuses on details of color image inpainting, such as the homogeneous color content of the regions enclosed by edges.

Finally, several works relying on semantic segmentation can be mentioned [12,16,19,34]. In the first two contributions, the semantic segmentation result is *undamaged* since it stems from an external source. Moreover, [30] accomplishes semantic image inpainting only. The work of [34] is a two-GANs-based network. The first GAN, called *Segmentation Prediction*, accomplishes inpainting of the segmentation image as an intermediate step. The second, called *Segmentation Guidance*, reconstructs the texture image. In the loss function, the discriminators take both original and down-sampled inputs, allowing the generator to capture both global structure and local texture. In this method, the segmentation result is a product of data processing and lacks the typical man-made features, such as rectangular structures, which can be observed in the results. In the work of [19], the corrupted image is initially completed in the feature space. Inpainting of segmentation and the texture image takes place alternately. The work of [16] modified the EdgeConnect algorithm by training an intermediate GAN for inpainting the classification image. The method has been proved successful for façade images with manually annotated foreground objects, i. e., practically under labor conditions. In this paper, we apply this method to a result of an automatic land cover classification procedure in an application rather belonging to the remote sensing field. In particular, such a result appears noisy or over-regularized, influenced by shadows or penumbras, and exhibiting misclassifications on building borders and ambiguous classes.

3 Methodology

As already mentioned, the main contribution of this paper is the workflow encompassing two main steps: land cover classification and inpainting. The entire pipeline is illustrated in Fig. 1. As input data, combined elevation and optical data are used. The elevation data is given as a normalized Digital Surface Model (NDSM). The first step employs the DeepLabv3+ model, described in Section 3.1, to create a land cover classification from the RGB image and the surface model. Vehicle masks V are extracted from the land cover classification and extended by the vehicle shadow masks. In the inpainting step, described in Section 3.2, the RGB image, the land cover classification, and the inpainting mask are used again.

3.1 Land cover classification and vehicle detection

The task of land cover classification is a semantic segmentation task, for which a DeepLabv3+ model of [5] was used. This network uses a ResNet101 encoder, and a decoder applying atrous and separable convolutions to increase the receptive field. The Atrous Spatial Pyramid Pooling Module (ASPP) aggregates the features at different scales.

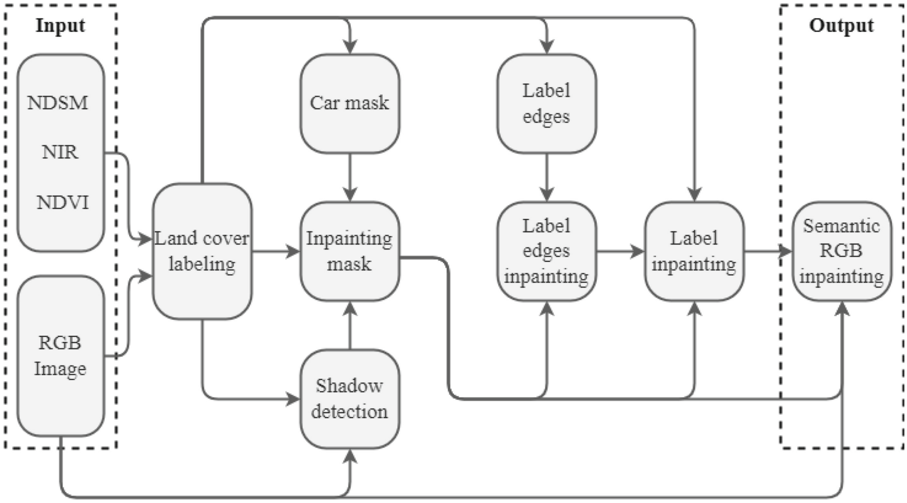


Fig. 1. Overview of the method's workflow: Inputs include RGB images and additional data (NDSM/NIR/NDVI). The process involves land cover classification, car and shadow masking, followed by inpainting steps, leading to the output of an inpainted RGB image.

In remote sensing, multi-modal data are usually captured and used down the line. Other than RGB imagery, there could be height/depth data from radar, LiDAR or photogrammetry, near-infrared to aid the detection of vegetation, as well as from other spectral bands. Additionally to RGB orthophotos, the Potsdam dataset [28], which we chose for this work, includes elevation information in the form of a digital surface model (DSM), derived photogrammetrically, and a near-infrared channel (NIR). From the near-infrared channel, we derive the NDVI, or Normalized Differential Vegetation Index. From the DSM, we derive the Normalized DSM using the method in [2]. The relative elevation is expected to be more valuable, since the ranges of the heights of the vehicles are better bounded, to enable their easier detection.

Since the input of DeepLabv3+ is constituted by RGB images, we extended the architecture by a second branch to make use of all data available. The input of the first branch are the RGB images, while the input of the second branch consists of the near-infrared channel, the NDVI channel, and the NDSM channel. The second branch is merged with the first branch after the first ResNet block by a convex combination with the factor $0 \leq \alpha \leq 1$. Setting $\alpha = \{0, 1\}$ means running the standard DeepLabv3+ algorithm with the images stored in the second resp. first branch while setting $\alpha = 0.5$ means averaging the features. For more details, see [27].

3.2 Inpainting

Preprocessing In remote sensing, orthophotos are composed of multiple images that may not be captured simultaneously or under uniform weather conditions, leading to variations in shadow direction and intensity. Additionally, objects such as cars can be obscured by shadows from larger structures like buildings. Thus, simple car mask dilation fails, making shadow detection essential for accurate image analysis. Shadows can significantly affect the process of object removal from images, particularly when aiming to eliminate visible artifacts around inpainted areas. This issue is especially prevalent when removing vehicles from RGB orthophotos, as the areas surrounding the vehicles tend to be darker due to the shadows cast by the vehicles themselves. To address this challenge, a shadow detection method based on Particle Swarm Optimization (PSO) is employed [11], utilizing the HSI (Hue, Saturation, and Intensity) color space. For better shadow region detection, an adaptive histogram equalization is applied to the RGB orthophotos leading to the effect of higher contrast between shadow and neighboring regions. After converting the RGB orthophoto into the HSI color space, applying a Gaussian filter to reduce noise is recommendable. For the PSO optimization, the features utilized include $(H+1)/(I+1)$ for highlighting shadows, the intensity, and the saturation S. This step also involves the removal of pixels associated with non-relevant classes such as buildings and trees to focus the analysis on relevant shadow regions.

The optimization begins with a randomly initialized *swarm* $X = \{X_i\}, i = 1, ..., i_{\max}, X_i \in \mathbb{R}^{3\times 2}$, where $i_{\max}$ denotes the swarm size. Each X_i represents the feature position of two cluster centers-one corresponding to shadow regions and the other to non-shadow regions-within the bounds of the feature data.

The PSO algorithm updates the positions and velocities of the swarm members iteratively using the following equations:

$$v_i^k = w \cdot v_i^{k-1} + c_1 \cdot r_1 \cdot \left(\hat{p}_i - X_i^{k-1}\right) + c_2 \cdot r_2 \cdot \left(\hat{g}^{k-1} - X_i^{k-1}\right) \tag{1}$$

$$w^k = w^{k-1} \cdot w_d \tag{2}$$

$$X_i^k = X_i^{k-1} + v_i^{k-1}. \tag{3}$$

In these equations, $k = 1, ..., k_{\max}$ indexes the iteration steps, with $k_{\max}$ set to 10. The coefficients c_1 and c_2 represent learning factors, each set to 2. The variables r_1 and r_2 are random values which can achieve values between 0 and 1. The inertia weight w is set to 1, and will be updated as in equation (2) using the factor w_d (here: 0.99), $\hat{p}_i^{k-1}$ and $\hat{g}^{k-1}$ denote the local and global cost, that means, the sum of the differences between the feature data and the nearest local cluster center of X_i^{k-1} and nearest global cluster center of $\cup_i X_i^{k-1}$, respectively. After completing the ten iterations, the cluster centers of the swarm member with the lowest local cost is used to generate a shadow map X, effectively distinguishing shadow from non-shadow regions in the orthophoto.

The inpainting mask M is created by enlarging the vehicle masks V, intersecting it with the shadow mask X, and then combining it with the original vehicle mask V. This ensures that only the vehicles and its shadows are inpainted.

Application of the 3GAN algorithms Generative Adversarial Networks (GANs) are a type of ML algorithm consisting of two neural networks, a generator and a discriminator, that play a zero-sum-game against each other. Resembling real data from the training data, the generator tries to fool the discriminator, while the discriminator attempts to distinguish between real and fake data. Through this process, GANs become proficient at generating data that closely mimic real data.

The 3GAN method involves a three-stage Generative Adversarial Network (GAN) approach designed to fill in holes masked by the binary map M in the RGB image J, by first completing the edge image E and then the semantic label image C. All three stages have similar loss function of the form:

$$L_G = L_a + \lambda_d L_d + \lambda_f L_f + \lambda_s L_s, \tag{4}$$

where L_a, L_d, L_f, and L_s denote adversarial, data-, feature- and style-based loss, respectively, and λ_d, λ_f, and λ_s are the corresponding weights, which, once defined empirically, vary from stage to stage. In what follows, we briefly emphasize the particularities of the losses mentioned in all three stages while for the technical description of these losses, we refer to [16],

1. **Label Edges Inpainting**: The first step involves inpainting E and is similar to the first stage of [23], however, adapted for label images C. This stage aims to understand and recreate the boundaries and shapes within E that are missing or occluded. We use the standard equation for the adversarial loss for L_a while the feature matching loss compares the activation maps in the middle layers of the discriminator to encourage the generator to produce results similar to real images. Since we only compare very sparse, binary images, in which only edge pixels are set to 255 and the rest to zero, there is no need to consider a data fidelity term. Thus, $\lambda_d = \lambda_s = 0$ and $\lambda_f = 10$.
2. **Label Inpainting**: After reconstructing E, the next step involves inpainting at the label level. The generator creates $\hat{C}$ using the edge image $\hat{E}$. Hereby, $\hat{C}$ is synthesized in the way that it is the completed image within the binary map M and the input image outside of it. The adversarial loss is then the standard one. As data fidelity loss, cross-entropy is considered, whereby labels are coded as one-hot representations. In this stage, $\lambda_f = \lambda_s = 0$ and $\lambda_d = 10$.
3. **RGB Inpainting**: Inpainting of J benefits significantly from the preceding stages. With the structural outlines and semantic context already established, the RGB inpainting process can focus on filling in the actual colors and textures with a higher level of accuracy and realism. Analogously to the previous step, the adversarial loss of the third GAN assesses J image generated using

synthesized $\hat{C}$ and incomplete J. We chose the standard ℓ_1-loss as our data fidelity term, which coincides with the choice made by [23] and [16], as it is in the case of L_f and L_s. Finally, $\lambda_d = 10, \lambda_f = 1$ and $\lambda_s = 2500$.

The input, output, and intermediate steps of 3GAN are shown in Fig. 2. Image 1 shows the original image, 2 the inpainting mask, 3 the semantic label image where orange is a car, dark blue is asphalt, light blue is a building, and dark red is clutter. Image 6 shows the result of the label edges, Image 5 shows the result of the label inpainting, and Image 4 shows the result of the RGB inpainting.

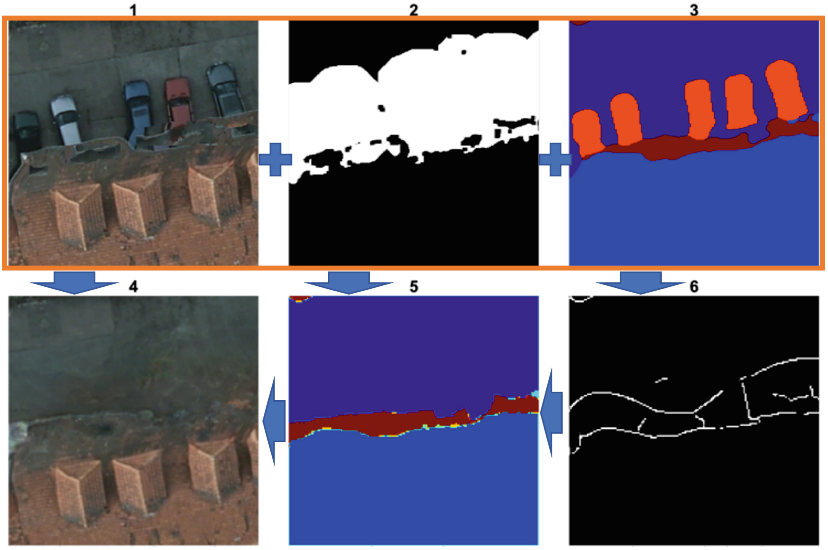


Fig. 2. Intermediate steps of 3GAN: First row: input (RGB image, inpainting mask, label image); second row: intermediate steps (inpainted RGB image, inpainted label image, inpainted label edge image)

4 Results

4.1 Dataset

The ISPRS Potsdam dataset includes 38 high-resolution (GSD of 5 cm per pixel) segments, of 6000 × 6000 pixels each, split into 24 for training, 8 for validation, and 6 for testing, featuring RGB, NIR, DSM channels, and ground truth. The first input of the DeepLab network is constituted by the RGB channels, and the second input includes NIR, NDVI, and normalized (NDSM) instead of absolute (DSM) elevation data, with NDVI rescaled and NDSM adjusted for height precision.

Training and validation data are maintained at original resolution but cropped into 512×512 pixel patches without data augmentation. During testing, the patches are larger and overlapping to ensure seamless transitions. Performance is gauged using overall accuracy, Cohen's kappa for class detectability, and F1-scores for individual class performance. For the inpainting procedure using 3GAN, merely the 6 testing segments are used, resizing the patches to 256×256 for efficiency. Internal experiments have shown that incorporating training or validation segments from the land cover classification stage does not improve the inpainting results.

4.2 Land cover classification and vehicle detection

Table 1. Results of DeepLabv3+ with pre-training for three characteristic values of α mentioned in the end of Section 3.1. The values are given in percent, and the best values are marked in bold. Here, "O. A." stands for Overall Accuracy, "κ" represents the Cohan's Kappa, "F1 C1" to "F1 C5" are the F1 scores for classes unpaved surface, building, low vegetation, tree, car (emphasized column), and clutter respectively, and "F1 av.*" and "F1 av." denote the average F1 scores, with and without including the clutter class.

α	O.A.	κ	F1_{C1}	F1_{C2}	F1_{C3}	F1_{C4}	F1_{C4}	F1_{C5}	$\text{F1}_{av.*}$	$\text{F1}_{av.}$
0	87.46	83.11	91.03	95.50	80.45	81.96	86.69	54.87	87.13	81.75
0.5	**88.99**	**85.15**	**92.17**	**95.17**	**83.23**	**83.63**	89.24	60.71	**88.83**	84.15
1	88.69	84.75	**92.17**	94.94	83.01	83.41	**89.62**	**62.04**	88.63	**84.20**

As shown in Table 1, the two-branch model with the additional channels as input performs mostly better than the original DeepLabv3+ on only RGB imagery. Most of the improvements, however, are in the building class. Due to temporal discrepancies, moving cars are often not visible in elevation data and do not have a striking infrared signature either. As a consequence, the combined model performs even slightly worse for the car class than the standard DeepLabv3+ method of [4]. For that reason, the pipeline can be simplified to RGB input only without a large sacrifice in performance. From Fig. 3 and also Fig. 4 to 6, displaying some qualitative results, it becomes obvious that the majority of cars are detected with a satisfactory overlap with the ground truth. In the upper image, one white car (red-circled) could not be detected well, probably due to an unusual background. The surroundings of cars, whether it be impervious surface or low vegetation, are classified well in all cases. In Fig. 4, one can see a car partly occluded by the leafless tree and, as a consequence, not entirely classified and reconstructed. In training data, such ambiguous regions were assigned to the tree class.

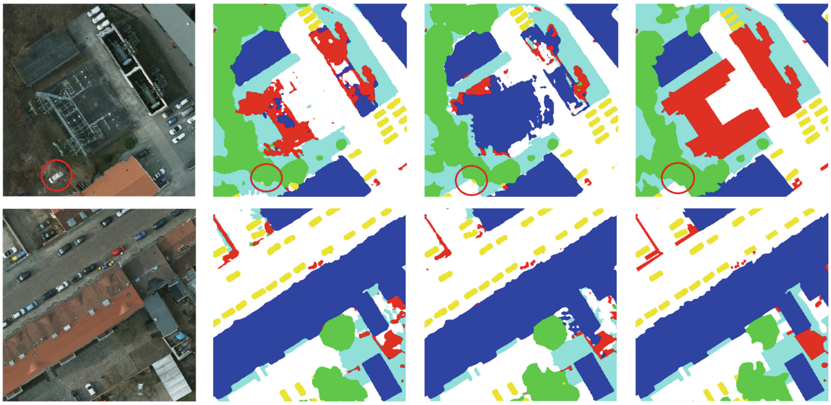


Fig. 3. Detection results of DeepLabv3+ on the Potsdam dataset. From left to right: Orthophoto, two-branch DeepLab, DeepLabv3+ RGB, ground truth.

4.3 Inpainting

We compared our method with EdgeConnect [23] as well as with a conventional method [7] on the test segments of the Potsdam dataset. The motivation of assessing a conventional method is that no training data and no label image are necessary. In our method, we are also interested to explore the influence of the shadow detection module on the performance of out method. The results without taking shadows into account are noted in the column 3GAN of Table 2 since they represent the conventional 3GAN method [16]. In case of EdgeConnect, shadow detection module is taken into account as well.

Table 2. Quantitative comparison of the inpainting via SSIM and PSNR, showcasing the allegedly superior quality of the conventional method of [7] and the performance of [16] in not considering shadows, emphasizing quantitative metrics' limitations.

	SSIM				PSNR			
	ours	[23]	[16]	[7]	ours	[23]	[16]	[7]
Segment0	0.910	0.910	0.917	0.978	27.893	27.911	28.188	29.325
Segment1	0.899	0.898	0.907	0.972	26.075	26.146	26.153	26.989
Segment2	0.934	0.934	0.937	0.987	29.271	29.256	29.343	30.919
Segment3	0.908	0.907	0.916	0.963	24.987	24.971	24.939	25.610
Segment4	0.896	0.895	0.904	0.964	24.878	24.940	24.811	25.463
Segment5	0.934	0.934	0.938	0.986	28.847	28.871	28.920	30.234

Table 2 shows a comparison between the inpainting methods via SSIM and PSNR Metrics, widely used to evaluate results of generative techniques. The method of [7] consistently achieves the highest SSIM and PSNR scores, allegedly indicating its superior image restoration quality. The method of [16] allegedly performs better without considering shadows. The quantitative metrics may not fully capture contextual inaccuracies because, for example, the artifacts of the conventional method [7] do not affect the scores despite having a relatively low degree of visual authenticity. Using SSIM and PSNR metrics, the conventional method consistently scores highest in SSIM and PSNR, suggesting superior image restoration quality. Notably, [16] performs better without shadow consideration. However, these quantitative metrics might overlook contextual inaccuracies, as artifacts from conventional method do not negatively impact the scores, despite potentially being contextually irrelevant. This discrepancy highlights the limitations of purely quantitative assessments in inpainting evaluations, especially where larger areas require inpainting due to shadow removal. In terms of performance, EdgeConnect tends to be outperformed by 3GAN.

Regarding qualitative analysis, we refer to Figs. 4 to 6. The shadow detection in the two parking lots of Fig. 4 brought a notable improvement because the bottom-right image exhibits almost no influence of shadows. Compared to [23], both methods perform similarly well in terms of shadow treatment or the minimization of artifacts. Contrastingly, [7] produces significant artifacts, presumably due to the inclusion of shadow elements within the reconstruction patches. These artifacts are particularly noticeable around a red and a light blue car. In Fig. 5, we observe that cars parked in the backyard do not cast distinct shadows due to extensive shading from the surrounding buildings. This condition led our shadow detection algorithm to classify most of the backyard as shaded. The results appear more natural without considering shadows, as the areas for inpainting were chosen to be larger. Regarding the comparative analysis between 3GAN and EdgeConnect, our findings indicate a clear advantage of 3GAN over EdgeConnect, especially in achieving clear distinctions between the classes of buildings and the ground. One issue in the dataset lies in adequately capturing moving objects in the DSM. In Fig. 6, the white car on the street is not visible in the DSM. The shadow detection tends to recognize too many shadows, which negatively affects the quality of the results. The application of the 3GAN method improves the situation by handling shadows more effectively.

5 Conclusion

Our paper presents a streamlined approach for digitally removing cars from orthophotos, leveraging a combination of semantic segmentation, shadow detection, and GAN-based inpainting. This integration effectively eliminates transient objects, ensuring structural and aesthetic integrity.

The application of semantic understanding ensures that the inpainted areas are contextually appropriate, allowing the algorithm to make informed decisions

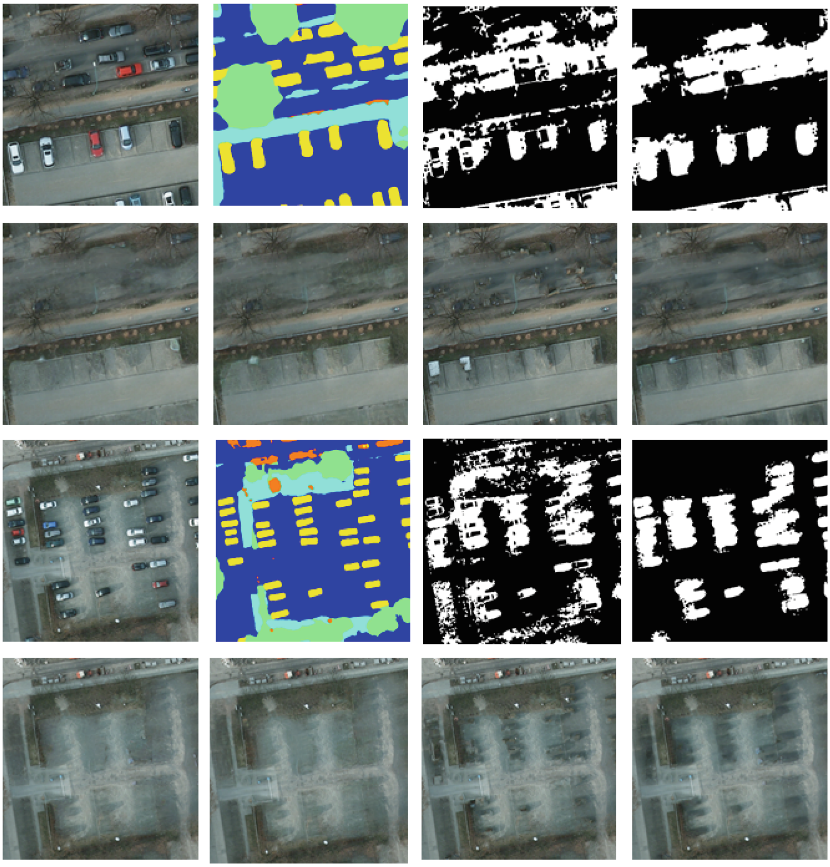


Fig. 4. Comparison of inpainting results for two parking lot areas, displayed over two sequences. Initially, we present the original image, shadow detection, inpainting mask, and land cover classification (first and third rows). The processed outcomes by 3GAN, [23], [7], and 3GAN excluding shadows are then shown (second and fourth rows).

that result in coherent reconstructions. This is particularly important for maintaining the architectural fidelity of the scene. Employing Particle Swarm Optimization, the shadow detection is essential for minimizing artifacts, improving object reconstruction quality. By sequentially addressing label edges and inpainting before RGB restoration, our method enhances visual consistency without guessing the inpainted area's structure or meaning.

The evaluation of the DeepLabv3+ model on the Potsdam dataset reveals that while adding extra input channels boosts classification accuracy for structures, its benefits vary across different objects like cars. Recognizing the limitations of quantitative analysis, we emphasize qualitative assessments. Here, the 3GAN method outperforms EdgeConnect in distinguishing between building and ground classes, especially in shadowed regions where cars are less discernible.

This advantage of 3GAN could be particularly crucial in urban and residential settings, where shadows and varying lighting conditions can significantly affect the visibility of objects and the overall accuracy of the inpainting process. In the future, we aim to explore the inpainting of multi-spectral data, potentially enhancing the utility and applicability of our methodology for a wider range of orthophoto analyses and applications.

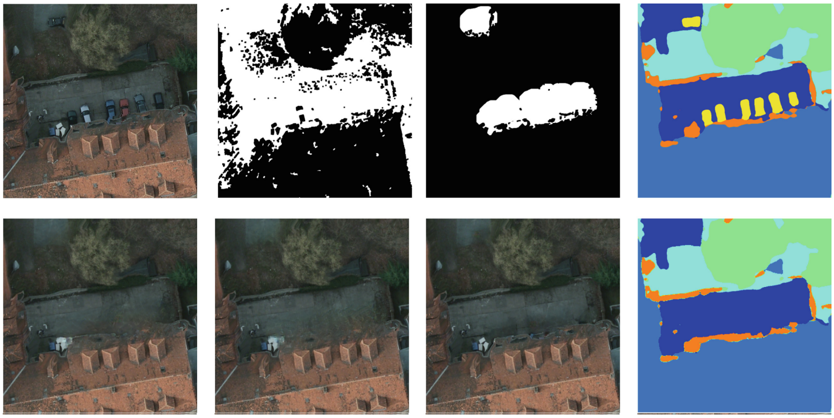


Fig. 5. Result of a backyard area: From upper left to bottom right: original image, shadow detection mask, inpainting mask, land cover classification, result of 3GAN, [23], 3GAN without shadows, inpainted land cover classification.

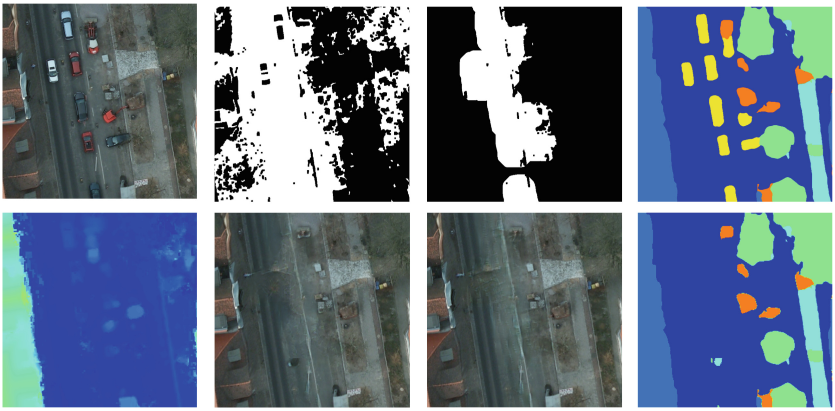


Fig. 6. Result of a street area: From upper left to bottom right: original image, shadow detection result, inpainting mask, land cover classification, DSM, result of 3GAN, [23], inpainted land cover classification.

References

1. Ammour, N., Alhichri, H., Bazi, Y., Benjdira, B., Alajlan, N., Zuair, M.: Deep learning approach for car detection in UAV imagery. Remote Sensing **9**(4/312), 1–15 (2017)
2. Bulatov, D., Häufel, G., Meidow, J., Pohl, M., Solbrig, P., Wernerus, P.: Context-based automatic reconstruction and texturing of 3D urban terrain for quick-response tasks. ISPRS J. Photogramm. Remote. Sens. **93**, 157–170 (2014)
3. Chen, H., Luo, Y., Cao, L., Zhang, B., Guo, G., Wang, C., Li, J., Ji, R.: Generalized zero-shot vehicle detection in remote sensing imagery via coarse-to-fine framework. In: International Joint Conference on Artificial Intelligence. pp. 687–693 (2019)
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2017)
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: European Conference on Computer Vision. pp. 801–818 (2018)
6. Chen, X., Xiang, S., Liu, C.L., Pan, C.H.: Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geosci. Remote Sens. Lett. **11**(10), 1797–1801 (2014)
7. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. **13**(9), 1200–1212 (2004)
8. Elharrouss, O., Almaadeed, N., Al-Maadeed, S., Akbari, Y.: Image inpainting: A review. Neural Process. Lett. **51**, 2007–2028 (2019)
9. Frommholz, D., Kuijper, F., Bulatov, D., Cheung, D.: Geospecific terrain databases for military simulation environments. In: Electro-Optical Remote Sensing XVI. vol. 12272, pp. 46–59. SPIE (2022)
10. Guo, S., Xiong, X., Liu, Z., Bai, X., Zhou, F.: Infrared simulation of large-scale urban scene through LOD. Opt. Express **26**(18), 23980–24002 (2018)
11. He, Z., Zhang, Z., Guo, M., Wu, L., Huang, Y.: Adaptive unsupervised-shadow-detection approach for remote-sensing image based on multichannel features. Remote Sensing **14**(12), 2756 (2022)
12. Huang, Z., Qin, C., Liu, R., Weng, Z., Zhu, Y.: Semantic-aware context aggregation for image inpainting. In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 2465–2469. IEEE (2021)
13. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Transactions on Graphics (TOG) **36**(4), 1–14 (2017)
14. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1125–1134 (2017)
15. Ji, H., Gao, Z., Mei, T., Ramesh, B.: Vehicle detection in remote sensing images leveraging on simultaneous super-resolution. IEEE Geosci. Remote Sens. Lett. **17**(4), 676–680 (2019)
16. Kottler, B., List, L., Bulatov, D., Weinmann, M.: 3GAN: A three-gan-based approach for image inpainting applied to the reconstruction of occluded parts of building walls. In: VISIGRAPP (4: VISAPP). pp. 427–435 (2022)
17. Leberl, F., Bischof, H., Grabner, H., Kluckner, S.: Recognizing cars in aerial imagery to improve orthophotos. In: Proc. ACM International Symposium on Advances in Geographic Information Systems. p. 2. ACM (2007)

18. Li, J., Wang, N., Zhang, L., Du, B., Tao, D.: Recurrent feature reasoning for image inpainting. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7760–7768 (2020)
19. Liao, L., Xiao, J., Wang, Z., Lin, C.W., Satoh, S.: Guidance and evaluation: Semantic-aware image inpainting for mixed scenes. In: Proc. 16th European Conference on Computer Vision, Part XXVII 16. pp. 683–700. Springer (2020)
20. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 85–100 (2018)
21. Liu, Y., Piramanayagam, S., Monteiro, S.T., Saber, E.: Dense semantic labeling of very-high-resolution aerial imagery and lidar with fully-convolutional neural networks and higher-order CRFs. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 76–85 (2017)
22. Mo, N., Yan, L.: Improved faster RCNN based on feature amplification and oversampling data augmentation for oriented vehicle detection in aerial images. Remote Sensing **12**(16/2558), 1–21 (2020)
23. Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: EdgeConnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019)
24. Park, J., Cho, Y.K., Kim, S.: Deep learning-based UAV image segmentation and inpainting for generating vehicle-free orthomosaic. Int. J. Appl. Earth Obs. Geoinf. **115**, 103111 (2022)
25. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 2536–2544 (2016)
26. Pyo, J., Rocha, Y.G., Ghosh, A., Lee, K., In, G., Kuc, T.: Object removal and inpainting from image using combined GANs. In: Proc. 20th International Conference on Control, Automation and Systems (ICCAS). pp. 1116–1119 (2020)
27. Qiu, K., Bulatov, D., Lucks, L.: Improving car detection from aerial footage with elevation information and markov random fields. In: Proceedings of the 19th International Conference on Signal Processing and Multimedia Applications, SIGMAP 2022. pp. 112–119. SCITEPRESS (2022)
28. Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J.D., Breitkopf, U., Jung, J.: Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. ISPRS J. Photogramm. Remote. Sens. **93**, 256–271 (2014)
29. Schilling, H., Bulatov, D., Niessner, R., Middelmann, W., Soergel, U.: Detection of vehicles in multisensor data via multibranch convolutional neural networks. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **11**(11), 4299–4316 (2018)
30. Schlagenhauf, T., Xia, Y., Fleischer, J.: Context-based image segment labeling (cbisl). arXiv preprint arXiv:2011.00784 (2020)
31. Shalunts, G., Haxhimusa, Y., Sablatnig, R.: Architectural style classification of building facade windows. In: International Symposium on Visual Computing. pp. 280–289. Springer (2011)
32. Shao, H., Wang, Y., Fu, Y., Yin, Z.: Generative image inpainting via edge structure and color aware fusion. Signal Processing: Image Communication **87–115929**, 1–9 (2020)
33. Song, Y., Yang, C., Lin, Z., Liu, X., Huang, Q., Li, H., Kuo, C.C.J.: Contextual-based image inpainting: Infer, match, and translate. In: Proc. IEEE European Conference on Computer Vision (ECCV). pp. 3–19 (2018)

34. Song, Y., Yang, C., Shen, Y., Wang, P., Huang, Q., Kuo, C.C.J.: Spg-net: Segmentation prediction and guidance network for image inpainting. In: Proc. British Machine Vision Conference. vol. 97, pp. 1–14 (2018)
35. Tang, T., Zhou, S., Deng, Z., Zou, H., Lei, L.: Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining. Sensors **17**(2), 336 (2017)
36. Tayara, H., Soo, K.G., Chong, K.T.: Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network. IEEE Access **6**, 2220–2230 (2017)
37. Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., Li, H.: High-resolution image inpainting using multi-scale neural patch synthesis. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 6721–6729 (2017)
38. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proc. IEEE/CVF International Conference on Computer Vision. pp. 4471–4480 (2019)
39. Zhang, J., Fukuda, T., Yabuki, N.: Automatic object removal with obstructed façades completion using semantic segmentation and generative adversarial inpainting. IEEE Access **9**, 117486–117495 (2021)

Deep Residual 1-Dimensional Convolutional Neural Networks in Vision

Nazmul Shahadat[1](✉) and Anthony S. Maida[2]

[1] Truman State University, Kirksville, MO, USA
nshahadat@truman.edu
[2] University of Louisiana at Lafayette, Lafayette, LA, USA

Abstract. While 2D convolutional neural networks (CNNs) demonstrate outstanding performance on computer vision tasks, their computational costs remain high. This paper reduces computational costs by introducing a novel architecture that replaces spatial 2D CNN operations with two consecutive 1D depthwise separable CNN (DSC) operations. Although vision inputs are two-dimensional, these 1D DSCs perform operations on 1D vision inputs. The 1D DSCs are predicated on the assumption that the dataset supports convolution operations with little or no loss of training accuracy. Deep 1D DSCs still suffer from gradient problems when training deep networks. We modify the construction of 1D CNNs with residual connections to improve the performance of deep 1D CNN architectures and introduce our final novel architecture, residual 1D convolutional networks (RCNs) for 1D vision inputs. Extensive benchmark evaluation shows that RCNs achieve at least 1% higher performance with about 77%, 86%, 75%, and 34% fewer parameters, and about 75%, 80%, 67%, and 26% fewer flops than ResNets, wide ResNets, MobileNets, and SqueezeNexts on CIFAR benchmarks, SVHN, and Tiny ImageNet image classification datasets. Moreover, our proposed RCNs improve deep recursive residual networks performance with 94% fewer parameters on the image super-resolution dataset.

Keywords: Deep CNN · 1D CNN · RCN · Parameter Efficient Network

1 Introduction

Convolutional neural networks (CNNs) have emerged as a core building block for computer vision tasks, including classification [7,8], object detection [19] and image super-resolution [14,15,25]. To solve major vision tasks, the CNN-based SOTA models, specifically ResNets [8], GoogleNets [13], AlexNets [17], and hypercomplex CNNs [22,23] have emerged in recent years. A common trend is to build deeper [8,9] or wider [13,29] networks to improve performance. However, increasing the depth or widening the network also increases its computational costs.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 129–144, 2025.
https://doi.org/10.1007/978-3-031-78312-8_9

A variety of CNNs were introduced to deal with these costs. Residual bottleneck blocks use 1 × 1 pointwise 2D convolutions to reduce and then increase the channel counts. As a result, the spatial 2D CNN processes fewer channels and reduces the model's computational costs. But these are not enough as they use standard 2D convolutions which consumes high costs. This cost reduction has not been analyzed for wider ResNets as the widening factor (α) multiplies the channel counts, raising the costs exponentially. The wide ResNets also use standard 2D convolutions.

As the standard 2D CNN is the core layer type of many computer vision models [8,17,22,23,29], and it consumes high costs, several modifications have been applied to reduce these costs. A depthwise separable convolution (DSC) convolves independently over each input channel to minimize the costs d_{in} (number of input channels) times than the standard 2D CNN operations. Although this DSC concept was introduced in 2014 for neural networks, it has been used more for CNN-based computer vision models, for example, Xception networks [3], and MobileNets [10]. Among these, MobileNet is an efficient, lightweight deep DSC for mobile-based vision tasks. It reduces the costs by a factor of 8 or 9 at only a small reduction in accuracy. But it requires two CNN layers ($k \times k$ DSC layer and pointwise CNN layer) to replace a standard 2D CNN layer. So, it increases the layer count. Moreover, the pointwise layer still uses standard 2D CNNs.

SqueezeNext [5] also reduces costs and is guided by SqueezeNet [12] and separable convolution (SC) (replace $k \times k$ 2D convolution using filters $k \times 1$ and $1 \times k$). This SC idea reduces the cost from k^2 to $2k$. They also squeeze the layer (like SqueezeNet [12]) before applying SC, reducing cost. These models still use standard 2D CNNs for 2D vision inputs.

Our work revisits the designs of the deep building blocks to boost their performance further, reduce computational costs, and improve the model's inference speed. To achieve these, we propose our novel architecture, RCNs, obtained by applying 1D DSC operations along the height and width axis instead of SCs in the InceptionV3 [24], and SqueezeNext [5] block. These height and width axis inputs are worked as 1D vision inputs, whereas the SCs and the other 2D CNNs are applied on 2D vision inputs. We split the 2D spatial CNN operation into two consecutive 1D DSC operations. These 1D DSC operations are mapped to the height and width axis. As 1D DSC operations propagate information along one axis at a time, this modification reduces cost at least $w \cdot d_{in} \cdot k$ times (explain below). Moreover, this RCN block does not increase layer counts as two 1D layers equal to one 2D layer.

A simple 1D CNN architecture reduces costs but does not improve performance. This is because forward information flowing across the 1D CNN blocks degrades (diminishing feature reuse [11]). We add residual connections to span the 1D CNN blocks to address this. By using both modifications, our novel and effective RCNs improve validation performance. The effectiveness of our proposed model is demonstrated experimentally on four image classifications and an image super-resolution dataset. Our assessments are based on parameter counts, FLOP counts (number of multiply-add operations), latency to process one image after training, and validation accuracy.

2 Background and Related Work

2.1 Convolutional Neural Networks

In a convolutional layer, the core building block is a convolution operation using a trainable weight W for 2D multichannel images applied to small neighborhoods to find input correlations. For an input image X with height h, width w, and channel count d_{in}, the convolution operation operates on region $(a, b) \in X$ centered at pixel (i, j) with spatial extent k. The output for this operation is,

$$C_{i,j,n} = \sum_{(a,b,m) \in \mathcal{N}_{k \times k(i,j)}} W_{a,b,m,n} X_{i+a-1,j+b-1,m} \tag{1}$$

where m, n, and $\mathcal{N}_{k \times k}$ are the index for input channel d_{in}, the index for output channel d_{out}, and the neighborhood of pixel (i, j) with spatial extent k of size $k \times k \times d_{in} \times d_{out}$, and W is the shared weights to calculate the output for all pixel positions (i, j). The computational cost of this convolutional operation is,

$$\text{Cost}_{\text{Conv2D}} = h \cdot w \cdot d_{in} \cdot d_{out} \cdot k \cdot k \tag{2}$$

where the computational cost depends multiplicatively on the kernel size $k \times k$, feature map size $h \times w$, d_{in}, and d_{out}.

2.2 Residual Networks

Residual networks (ResNets) are constructed using 2D CNN layers linked by additive identity connections [9] for vision tasks. They were introduced to address the problem of vanishing gradients found in standard deep CNNs.

The key architectural feature of ResNets is the residual block with identity mapping. Two kinds of residual blocks are used in residual networks, the basic block and the bottleneck block. We discuss the bottleneck block first. Figure 1a shows a bottleneck block for ResNets that is constructed using 1×1, $k \times k$, and 1×1 convolution layers with residual connection, where the 1×1 pointwise 2D CNN layers reduce and then increase the number of channels. The 3×3 2D CNN layer performs feature extraction. The computational cost of a 3×3 spatial 2D CNN layer is given in Equation 2 and a 1×1 pointwise 2D CNN layers is,

$$\text{Cost}_{\text{1x1Conv2D}} = h \cdot w \cdot d_{in} \cdot d_{out} \tag{3}$$

Hence, the computational cost of the bottleneck block is,

$$\text{Cost}_{\text{Bottle}} = h \cdot w \cdot d_{in} \cdot d_{out} \cdot k \cdot k + 2 \cdot h \cdot w \cdot d_{in} \cdot d_{out} \tag{4}$$

In contrast to the bottleneck block, the basic architecture of ResNet is constructed with two $k \times k$ convolution layers with residual connection where k is the size of the kernel and an identity shortcut connection is added to the end of these two layers. The computational cost of the residual basic block is,

$$\text{Cost}_{\text{Basic}} = 2 \cdot h \cdot w \cdot d_{in} \cdot d_{out} \cdot k \cdot k. \tag{5}$$

The performance of ResNets surpasses its learning speed, the number of learning parameters, the way of layer-wise representation, and memory mechanisms.

2.3 Wide Residual Networks

Wide ResNets [2,29] use fewer layers than standard ResNets but use higher channel counts (wide architectures) which compensate for their shallower architecture. Comparisons between shallow and deep networks have been studied in circuit complexity theory where shallow circuits require more components than deeper circuits. Inspired by this observation, He et al., proposed deeper networks with thinner architecture where a gradient goes through the layers [9]. But the problem such networks face is that the residual block weights do not flow through the network layers. Because of this, the network may be forced to avoid learning during training. To address this, Zagoruyko et al., proposed shallow but wide architectures and showed that widening the residual blocks improves the performance of residual networks compared to increasing their depth [29]. The computational cost of this 2D convolutional operation is,

$$\text{Cost}_{\text{Conv2D(WRNs)}} = h \cdot w \cdot d_{in} \cdot \alpha d_{out} \cdot k \cdot k, \tag{6}$$

where α is a widening factor.

2.4 MobileNet Architectures

Howard et al. developed a mobile-based shallower network for vision tasks depicted in Figure 1b. They used DSCs because it helps to build lightweight networks. A pointwise 1×1 convolution is used to combine the outputs of DSC [10]. These two steps are performed in standard convolution in a single step. This DSC performs convolution per input channel, and it can be defined as,

$$C_{i,j,n} = \sum_{(a,b) \in \mathcal{N}_{k \times k(i,j)}} W_{a,b,n} X_{i+a-1,j+b-1,n} \tag{7}$$

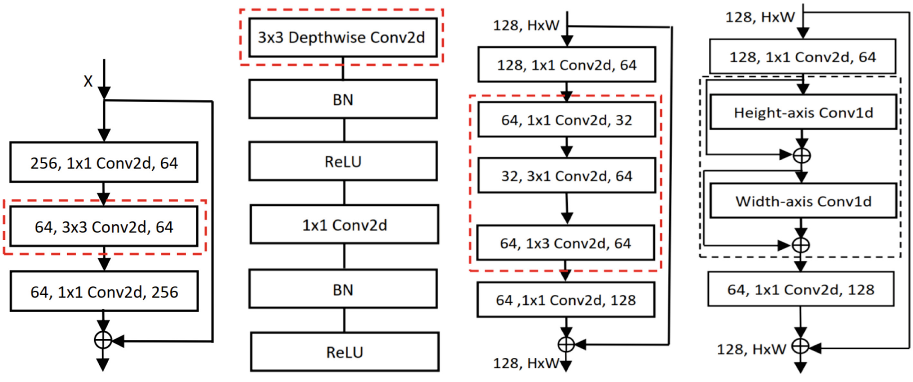
where the n_{th} channel of trainable weight W is applied to the n_{th} channel of input x to produce the n_{th} channel of the output feature map C. The computational cost of this 2-dimensional depthwise separable convolutional operation is,

$$\text{Cost}_{\text{DWConv2D}} = h \cdot w \cdot d_{out} \cdot k \cdot k. \tag{8}$$

And the pointwise 1×1 convolution has a computational cost which is explained in Equation 3. The computational cost of depthwise separable MobileNets is,

$$\text{Cost}_{\text{MobileNet}} = h \cdot w \cdot d_{out} \cdot k \cdot k + h \cdot w \cdot d_{in} \cdot d_{out} \tag{9}$$

which is the sum of the computational costs of depthwise (Equation 8) and pointwise (Equation 3) convolutions.



(a) ResNet (b) MobileNet (c) SqueezeNext (d) RCN block

Fig. 1. Block types. "bn" and "ReLU" stand for batch normalization and rectified linear unit, respectively. (a) Bottleneck modules found in [8], (b) MobileNet block found in [10], (c) SqueezeNext block found in [5], and (d) novel RCN block used in our model. The black dotted region in the RCN block replaces the red dotted region in blocks in Figures (a), (b), (c), and any 2D CNN layer.

2.5 Convolutions Meet Vision Transformers

Although transformers have demonstrated outstanding performance in vision tasks, their performances are not superior compared to the similar-si zed CNNs [6]. Guo et al. explained several reasons behind the transformer's inferior performance to CNNs and proposed a novel CMT (CNNs meet transformer) architecture by interacting with CNNs and transformers for visual recognitions [6]. They used the convolution stem for fine-grained feature extraction and then fed it into a stack of CMT blocks for representation learning. They also used depthwise separable convolutions to enhance local information. The CMT architecture is graphically described in [6].

2.6 SqueezeNext Architecture

A CNN with fewer input and output channels requires fewer trainable parameters, less cross-server communication for distributed training, lower bandwidth to export, and is easier to deploy on field-programmable gate arrays (FPGAs) with limited memory [12]. To achieve these advantages, Iandola et al. proposed SqueezeNet (SNet), where they squeeze the input channels to reduce the number of filters [12]. The computational cost of this SqueezeNet conv2d operation is,

$$\text{Cost}_{\text{Conv2D(SNet)}} = h \cdot w \cdot ds_{in} \cdot d_{out} \cdot k \cdot k \tag{10}$$

where ds_{in} is the squeezed input channels. Gholami et al. further reduce this cost by applying separable CNNs (3×1 and 1×3 Conv2d) instead of a spatial

CNN and they called it SqueezeNext [5] is depicted in Figure 1c. It reduces the cost compared to SqueezeNet and the new cost of the conv2d layer is,

$$\begin{aligned} \text{Cost}_{\text{SqNext}} &= h \cdot d_{in} \cdot d_{out} \cdot k \cdot k + w \cdot d_{in} \cdot d_{out} \cdot k \cdot k \\ &= 2 \cdot h \cdot d_{in} \cdot d_{out} \cdot k \cdot k. \end{aligned} \tag{11}$$

As height equals width for computer vision tasks. The SqueezeNext block performed better than the SNet [12] and MobileNet [10].

2.7 Recursive Residual Networks

Image super-resolution (SR) is the process of generating a high-resolution (HR) image from a low-resolution (LR) image. It is also known as single image super-resolution (SISR). Convolution-based recursive neural networks have been used on SISR [4,14,15,25], where recursive networks learn detailed and structured information about an image. Kim et al. introduce two deep CNNs for SR by stacking weight layers [14,15] where the chain structure recursive layer controls the model parameters and improves the performance. Deep SR models [14,15,20] demand large parameter counts and more storage.

To address these issues, deep recursive residual networks (DRRNs) were proposed, which achieves better performance with fewer parameters [25]. It includes both local (LRL) and global residual learning (GRL), where GRL might face degradation problems for deeper networks and LRL has been used to solve this problem. The DRRN also stacked several recursive blocks (B) of residual units to keep the model more compact, followed by a CNN layer, which reconstructs the residual between the LR and HR images. Each residual block decomposes into the number of residual units (U). The number of B and U is responsible for defining network depth d which is calculated as, $d = (1+2 \times U) \times B+1$. DRRN's recursive block definition, formulation, and the loss function are defined in [25]. The computational cost of each unit U will be the same as in equation 5.

3 Proposed Residual Convolutional Networks

The 2D CNN is highly performant with the help of several state-of-the-art architectures, like, ResNets [8], wide ResNets [27], scaling wide ResNets [29], MobileNets [10], SqueezeNets [12], SqueezeNexts [5], and deep recursive residual networks (DRRNs) [25] on image classification and image super-resolution datasets. The residual bottleneck block makes the networks thinner; still, the cost efficiency of these blocks can be improved. The cost of 2D convolution, residual bottleneck, and basic blocks is calculated in Equations 2, 4, and 5, respectively. The 2D convolution operation given in Equation 1 uses a $k \times k$ filter for the input $X \in h \times w \times d_{in}$. Equation 2 gives the costs of 2D convolution in the residual blocks. SC is used to reduce these costs in InceptionNetV3 [26], and SqueezeNext [5]. They decomposed this $k \times k$ convolution into two separable convolutions with $k \times 1$ and $1 \times k$ sized filters. This decomposition effectively reduces the number of parameters from $h \times w \times d_{in} \times d_{out} \times k \times k$ to $2 \times h \times w \times d_{in} \times d_{out} \times k$. Their decomposed convolution with spatial extent $k \times 1$ is defined as,

$$C_{i,j,n} = \sum_{(a,b,m) \in \mathcal{N}_{k \times 1(i,j)}} W_{a,b,m,n} X_{i+a-1,j+b-1,m} \tag{12}$$

where, m, and n are the indices for input channel d_{in}, and for output channel d_{out}. Also, $\mathcal{N}_k \in \mathbb{R}^{k \times 1 \times d_{in}}$ is the neighborhood of pixel (i, j) with spatial extent $k \times 1$ and $W \in \mathbb{R}^{k \times 1 \times d_{out} \times d_{in}}$ is the shared weights that are for calculating output for all pixel positions (i, j). For spatial extent $1 \times k$ is defined as,

$$C_{O(i,j,n)} = \sum_{(a,b,m) \in \mathcal{N}_{1 \times k(i,j)}} W_{a,b,m,n} C_{i+a-1,j+b-1,m} \tag{13}$$

where, $\mathcal{N}_k \in \mathbb{R}^{1 \times k \times d_{in}}$ is the neighborhood of pixel (i, j) with extent $1 \times k$ and $W \in \mathbb{R}^{1 \times k \times d_{out} \times d_{in}}$ is the shared weights all pixel positions (i, j).

Their cost efficiency can be improved as they still use 2D convolutions with spatial extent filters for 2D inputs (even though one dimension has a size equal to 1). To reduce these costs further, we propose a novel residual 1D convolutional network (RCN) to replace any 2D spatial convolutional layer in a network. We replace the 2D convolution operations (conv2D) of any block by using two 1D DSC operations with filters k. To apply 1D DSC in 2D input size of $h \times w$, we split the inputs into the height and width axes. Each 1D convolution layer applies to each input axis. The 1D DSC operation is defined as,

$$C_{O(i,n)} = \sum_{a \in \mathcal{N}_{k(i)}} W_{a,n} X_{i+a-1,n} \tag{14}$$

where $\mathcal{N}_k \in \mathbb{R}^{k \times d_{in}}$ is the neighborhood of pixel i with extent k and $W \in \mathbb{R}^{k \times d_{out} \times d_{in}}$ is the shared weights that calculate the output for all pixel positions i. In Equation 14, the n_{th} channel of trainable weight W is applied to the n_{th} channel of input X to produce the n_{th} channel of the output feature map C_O. The cost of this 1D DSC operation on axial vision inputs is calculated as,

$$\text{Cost}_{\text{Conv1D}} = h \cdot d_{out} \cdot k. \tag{15}$$

As our RCN block has two layers of 1D convolutions, this block costs $2 \cdot h \cdot d_{out} \cdot k$. Also, each 1D DSC operation has a residual connection to avoid vanishing gradients. Hence, our proposed novel architecture factorizes 2D convolution into two consecutive 1D DSCs along with residual connections depicted in Figure 1d.

We replace each 2D convolution layer from the residual basic and bottleneck blocks to construct residual blocks using our RCN block. The spatial 2D convolution in the residual bottleneck block (red marked in Figure 1a) is replaced by the RCN block to construct our proposed RCN-based residual bottleneck block. In the same way, the proposed RCN-based basic block architecture is constructed. To compare our proposed parameter-efficient RCN block with 2D DSC, the depthwise 2D spatial convolution (red marked in Figure 1b) of MobileNet and CMT are replaced by our proposed RCN block and constructed new RCN-based MobileNet and CMT. We also compare our proposed RCN block for SC by replacing the red-marked area in Figure 1c of SqueezeNext with the RCN block. The RCN block in Figure 1d is applied to other 2D convolution-based networks, for example, wide residual networks (to make our proposed wide RCNs) and deep recursive residual networks (to make RCRNs depicted in Figure 2), to check the effectiveness of our proposed method in all possible ways.

4 How RCNs are Cost Effective

This section compares the computational costs of different aspects of CNN layers to the networks. The costs of 2D convolutional operation, residual bottleneck and basic blocks, WRN operation, MobileNet block, SqueezeNet, and SqueezeNext block are calculated in Equations 2, 4, 5, 6, 9, 10, and 11. The computational costs of our proposed RCN block compared with the standard 2D convolutional operation, we get a reduction in the computation of:

$$\text{Cost}_R = \frac{\text{Cost of 2D Convolution}}{2 \cdot \text{Cost of 1D Convolution}} = \frac{h \cdot w \cdot d_{in} \cdot d_{out} \cdot k^2}{2 \cdot h \cdot d_{out} \cdot k} = \frac{w \cdot d_{in} \cdot k}{2} \tag{16}$$

where Cost_R is the cost reduction ratio where the RCN block reduces costs $(w \cdot d_{in} \cdot k) / 2$ times than the original standard 2D convolution. Hence, the RCN block can be used as a replacement for any networks where 2D convolutional layers are used. We apply this block as a replacement of 2D CNN in residual networks [8] specifically residual basic block (replacing the two 2D CNN layers), and bottleneck block (replacing the only spatial 2D CNN layer) and construct RCN-based ResNet blocks. These RCN-based ResNet basic blocks reduce costs $(w \cdot d_{in} \cdot k)$ times the original basic block costs as the original basic block used to standard CNN layers. For the ResNet bottleneck block, the RCN-based bottleneck block reduces costs similar to the cost reduction in Equation 16.

Now, we compare the cost-effectiveness of MobileNet and SqueezeNext architectures with our proposed RCN block-based MobileNet and SqueezeNext architectures. Our proposed RCN-based MobileNet block performs a reduction in the costs of:

$$\text{Cost}_R = \frac{\text{Cost}_{\text{DWConv2D}} \text{ in MobileNetV1}}{2 \cdot \text{Cost of 1D Convolution}} = \frac{h \cdot w \cdot d_{out} \cdot k \cdot k}{2 \cdot h \cdot d_{out} \cdot k} = \frac{w \cdot k}{2} \tag{17}$$

where d_{in} is 1 for the original and proposed spatial convolutions as both networks use depthwise separable convolutions. There is a huge $(w \cdot k)$ / 2 (75% reduction for CIFAR data in Table 1) reduction for this MobileNet architecture. For SqueezeNext architectures, our RCN-based SqueezeNext reduces the computational costs of:

$$\begin{aligned}\text{Cost}_R &= \frac{\text{PW1x1Conv2D} + 2 \cdot \text{kx1Conv2D}}{2 \cdot \text{Costof1DConvolution}} \\ &= \frac{h \cdot w \cdot d_{in} \cdot d_{out} + 2 \cdot h \cdot w \cdot d_{in} \cdot d_{out} \cdot k}{2 \cdot h \cdot d_{out} \cdot k} = \frac{w \cdot d_{in}}{2 \cdot k} + w \cdot d_{in}\end{aligned} \tag{18}$$

where Cost_R is the cost reduction ratio of the original SqueezeNext and our RCN-based SqueezeNext blocks. A pointwise 1×1, two separable 2D convolutions with filters, our proposed RCN block replaces $k \times 1$, and $1 \times k$ in SqueezeNext block. Our RCN-based SqueezeNext block takes $(w \cdot d_{in}) \,/\, (2 \cdot k) + (w \cdot d_{in})$ times fewer computing from separable convolutional operations in the original SqueezeNext block. However, our proposed RCNs are not similar to the separable 2D CNN operation. Compared with separable 2D CNN, our RCNs reduce the amount of the computational costs of:

$$\text{Cost}_R = \frac{2 \cdot \text{kx1Conv2D}}{2 \cdot \text{Cost of 1D Convolution}} = \frac{2 \cdot h \cdot w \cdot d_{in} \cdot d_{out} \cdot k}{2 \cdot h \cdot d_{out} \cdot k} = w \cdot d_{in} \qquad (19)$$

So, our proposed RCNs are cost-effective by a factor of $w \cdot d_{in}$ times than the SCs. These formulas show that our proposed RCN block is cost-effective in replacing any 2D convolution for computer vision tasks.

5 Experimental Analysis

We present experimental results on four image classification datasets and an image super-resolution dataset. Our experiments evaluate the original ResNets, wide ResNets, RCN-ResNets, wide RCNs, MobileNet architectures, SqueezeNext architectures, RCN-based MobileNet and SqueezeNext architectures, RCMTs, CMTs, DRRNs, and RCRNs. We compare our proposed RCN-based networks with the original ResNets, as these original networks used 2D CNN layers. Our comparisons use parameter counts, FLOPS, latency, and validation performance. The experiments were run on a workstation with an Intel(R) i9-9820X CPU @ 3.30GHz, 128 GB memory, and NVIDIA Titan RTX GPU (24GB).

5.1 Method: Convolutional Networks

To explore scalability, we compare our proposed RCNs and baseline models on four datasets: CIFAR-10 and CIFAR-100 benchmarks [16], Street View House Number (SVHN) [21], and Tiny ImageNet datasets [18]. The CIFAR bnchmarks have 10 and 100 distinct classes and 60,000 color images of size 32×32. We perform data normalization using per-channel mean and standard deviation. In preprocessing, we do the horizontal flips and randomly crop after padding with four pixels on each side of the image. The SVHN and Tiny ImageNet datasets contain 600,000 images of size 32×32 with ten classes and 110,000 images of 200 distinct classes downsized to 64×64 colored images, respectively. Our only preprocessing is mean/std normalization for both datasets. All models were run using the stochastic gradient descent optimizer and linearly warmed-up learning for ten epochs from zero to 0.1 and then used cosine learning scheduling from epochs 11 to 150. This experiment used batch normalization and 0.0001 weight decay.

Residual Networks ResNets and our proposed RCN-based ResNets were trained using similar designs (same hyperparameters and output channel counts). As our main concern was to reduce parameter counts of the residual bottleneck block, we implemented all baselines and the proposed architecture using only bottleneck blocks. The output channels of bottleneck groups are $120, 240, 480,$ and 960 for all networks. This experiment analyzes $26, 35, 50, 101,$ and 152-layer architectures with the bottleneck block multipliers "$[1, 2, 4, 1]$", "$[2, 3, 4, 2]$", "$[3, 4, 6, 3]$", "$[3, 4, 23, 3]$", and "$[3, 8, 36, 3]$", respectively. All models were trained using batch sizes of 128 for all datasets except the 101 and 152-layer architectures of the Tiny ImageNet dataset. The batch size was 64 for these architectures.

Wide Residual Networks Section 5.1 explained the method for deeper networks. This section describes methods for wide but shallow networks. To assess the widening factor on our proposed RCNs, we increase the width of our RCNs by factorizing the number of output channels for shallow networks like [29]. Like the original wide residual networks (WRNs) [29], we analyzed our proposed 26-layer bottleneck block of wide RCNs (WRCNs) with a widening factor, $k = 2, 4, 6, 8,$ and 10. We multiplied the number of output channels of RCNs with k to obtain WRCNs. We trained with the same optimizer and hyperparameters used in Section 5.1.

MobileNet Architectures MobileNet and RCN-based MobileNet architectures use the hyperparameters and number of output channels similar to Table 1 in [10]. For the MobileNet architectures, we also use a 0.045 initial learning rate decaying by 0.98 per epoch. Moreover, the standard RMSProp optimizer with decay and momentum is set to 0.9. Unlike original MobileNets [10], we trained the original MobileNet and our proposed RCN-based MobileNet architectures using a batch size of 128.

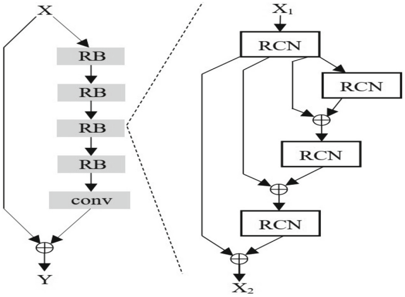


Fig. 2. Recursive 1D convolutional residual network (RCRN) architecture with $B = 4$ and $U_{RCN} = 3$. Here, the "RB" layer refers to a recursive block.

SqueezeNext Architectures For a fair comparison, we use similar hyperparameters to the original SqueezeNext [5]. The output channels of our RCN-based SqueezeNext groups are similar to the original SqueezeNext networks. This experiment analyzes 23-layer architectures with the block multipliers "[6, 6, 8, 1]". We analyze two 23-layer architectures, "SqNxt-23-1x", and "SqNxt-23-2x" where the channel widening factors are 1, and 2, respectively. All models were trained using batch sizes of 128 for CIFAR10, CIFAR100, and SVHN datasets.

Convolutions meet Transformers We build our model RCN-based CMT-ti and CMT-s similarly to CMT-ti and CMT-s [6]. All models are trained for 120 epochs using the SGD optimizer.

Recursive Networks This experiment compares the cost and performance of our novel RCRN with the DRRN on the super-resolution tasks. The RCRN is built by replacing the residual unit U with a RCN block described in Equation 14 and in Figure 1d. These modifications form a new network, a recursive 1D convolutional residual network whose depth d is given by

$$d = (1 + U_{RCN}) \times B + 1. \tag{20}$$

As two 1D layers are equivalent to one 2D layer and we replace each residual unit with a RCN unit (see Equation 14). Hence, we rewrite Equation ?? to Equation 20 by removing the multiplier for the residual unit. The proposed RCRN with four RB blocks is shown on the left, and an RB block is expanded on the right in Figure 2.

We trained our proposed RCRN using 291 images dataset [28] and tested using the Set5 dataset [1]. We also use different scales (×2, ×3, and ×4) in training and testing images. We used similar data augmentation, training hyperparameters, and implementation details like [25].

5.2 Results Analysis

Residual Networks Table 1 summarizes the classification results of the original ResNets and our proposed RCNs on the four datasets. We tested shallow and deeper networks by implementing $26, 35, 50, 101$, and 152-layer architectures. These architectures compare performance to check the effectiveness of our proposed methods for shallow and deep networks. Our proposed method is compared with original ResNets in terms of parameter count, FLOPS count, latency, and validation accuracy on the four datasets.

The $26, 35, 50, 101$, and 152-layer architectures reduce by 77%, 76.9%, 76.7%, 76.6%, and 76.5% trainable parameters respectively in comparison to the baseline networks. In addition to parameter reduction, our proposed method requires 15 to 36 percent fewer FLOPS for all analyzed architectures. Also, the validation performance improvement is significantly noticeable for all datasets in Tables 1 and 2. The latency of our proposed models is also lower than the original networks. Moreover, the deeper networks perform better than the shallow networks, demonstrating "the deeper, the better" in classification.

Table 1. Image classification performance on the CIFAR benchmarks and SVHN datasets for different architectures. Here, "Orig", "Ours", "SqNxt-1", and "SqNxt-2" are the original and our proposed network for corresponding models and 23 layers SqueezeNext with widening factors 1 and 2. Latency measures in ms.

Models	Params(M)		FLOPs		Latency		CIFAR10		CIFAR100		SVHN	
	Orig.	Ours	Orig.	Ours	Orig.	Ours	Orig.	Ours	Orig.	Ours	Orig.	Ours
ResNet-26	41	**9.4**	2.6G	**0.6G**	0.86	**0.52**	94.68	**96.08**	78.21	**79.66**	96.04	**97.83**
ResNet-35	58	**13.4**	3.3G	**0.8G**	0.96	**0.60**	94.95	**96.15**	78.72	**80.38**	95.74	**97.50**
ResNet-50	82	**19.2**	4.6G	**1.1G**	1.11	**0.73**	95.08	**96.25**	78.95	**81.29**	95.76	**97.32**
ResNet101	149	**34.8**	8.8G	**2.2G**	1.68	**1.28**	95.36	**96.27**	78.80	**80.88**	96.29	**97.29**
ResNet152	204	**47.9**	13G	**3.2G**	2.36	**1.80**	95.36	**96.37**	79.85	**80.94**	96.35	**97.38**
MobileNet	3.2	**0.8**	12M	**4M**	0.18	**0.18**	87.87	**93.34**	60.64	**61.1**	94.23	**94.53**
SqNxt-1	0.6	**0.4**	59M	**45M**	0.55	**0.37**	92.30	**93.34**	69.70	**70.14**	95.88	**97.13**
SqNxt-2	2.3	**1.7**	226M	**168M**	0.78	**0.47**	93.38	**94.91**	73.05	**74.94**	96.06	**97.40**
CMT-ti	8.1	**8.1**	25M	**24M**	0.48	**0.48**	98.28	**98.76**	81.03	**83.11**	98.45	**98.62**
CMT-s	24.8	**24.8**	80M	**77M**	0.95	**0.95**	98.62	**99.13**	83.62	**84.09**	98.48	**98.66**

Wide Residual Networks Table 1 shows "the deeper, the better" in vision classification for our proposed methods. To compare our proposed WRCNs with the original WRNs, we analyze our proposed method for different widening factors. Table 3 shows an overall comparison among the original WRN-28-10 (28-layers with a widening factor of 10) and our proposed 26-layer networks with different widening factors $(2, 4, 6, 8,$ and $10)$. Our proposed WRCNs show 4% better performance with 86% fewer parameters than the original WRN [29]. This table also demonstrates "the wider, the better" for our proposed WRCNs.

MobileNet Architectures RCN-based MobileNet, where the 2D convolution layers replace by the RCN block, and the original MobileNetV1 shows the direct effect of the RCN block in mobile-based shallower architectures. The RCN-based

Table 2. Image classification performance on the Tiny ImageNet datasets for $26, 35, 50, 101,$ and 152-layer architectures.

Models	Params	FLOPs	Latency	Accuracy
ResNet-26	41.6M	0.66G	2.31ms	57.21
RCNs-26	**21.3M**	**0.56G**	**2.58ms**	**62.28**
ResNet-35	58.5M	0.86G	2.85ms	57.80
RCNs-35	**31.3M**	**0.68G**	**3.0ms**	**59.31**
ResNet-50	82.6M	1.18G	3.75ms	59.06
RCNs-50	**45.8M**	**0.87G**	**4.02ms**	**62.40**
ResNet-101	149M	2.29G	6.86ms	60.62
RCNs-101	**85.1M**	**1.52G**	**7.19ms**	**64.18**
ResNet-152	204M	3.41G	9.29ms	61.57
RCNs-152	**117M**	**2.18G**	**9.72ms**	**66.16**

Table 3. Image classification performance on the CIFAR (CIF-10 and CIF-100) benchmarks for 26-layer architectures with different widening factors.

Model Name	Testing Accuracy	
	CIF-10	CIF-100
WRN-28-10 [29]	94.68	79.57
WRCN-26-2 (Ours)	96.32	83.54
WRCN-26-4 (Ours)	96.68	83.75
WRCN-26-6 (Ours)	96.77	83.78
WRCN-26-8 (Ours)	96.83	83.82
WRCN-26-10 (Ours)	96.87	83.92

Table 4. Benchmark testing PSNR results for scaling factors ×2, ×3, and ×4 on Set5 dataset.

Architectures	Scale		
	x2	x3	x4
SRCNN [4]	36.66	32.75	30.48
VDSR [14]	37.53	33.66	31.35
DRCN [15]	37.63	33.82	31.53
DRRN19 [25]	37.66	33.93	31.58
DRRN125 [25]	37.74	34.03	31.68
RCRN19	37.73	33.99	31.63
RCRN25	**37.84**	**34.11**	**31.84**

MobileNet performs more than 1% in validation accuracy with 75% fewer trainable parameters and almost 67% fewer FLOPs shown in Table 1. Our RCN-based MobileNet takes similar latency to the original MobileNet.

SqueezeNext Architectures We implement RCN-based SqueezeNext (RCN-SqueezeNext) to show the effectiveness of the RCN block compared with the SqueezeNext block. Table 1 compares the performance for 23-layer architecture with widening factors 1 and 2. Our RCN-SqueezeNexts outperform the original SqueezeNexts with 34%, 24%, and 33% fewer parameters, FLOPs, and latency, respectively.

Convolutions meet Transformers The performance analyses of CMT architectures are shown in Table 1 for CIFAR benchmarks and SVHN datasets. We implement RCN-based CMT-ti and CMT-s and compare them with the original CMT-ti and CMT-s models to show the effectiveness of RCN block in CMTs. Our RCMTs reduce parameters and FLOPs (just in thousands) and improve validation performance on image classification datasets.

Recursive Networks Table 4 shows the Peak Signal-to-Noise Ratio (PSNR) results of several CNN models, including DRRN, and our proposed RCRN on the Set5 dataset. The comparison between DRRN and RCRN is our main focus, as it directly indicates the effectiveness of using our proposed RCN block. DRRN19 and DRRN125 are constructed using $B = 1, U = 9$, and $B = 1, U = 25$, respectively. For fair comparison, we also construct similar architecture like RCRN19 ($B = 1, U_{RCN} = 9$) and RCRN125 ($B = 1, U_{RCN} = 25$). Our proposed models outperform all CNN models in Table 4 on the Set5 dataset and for all scaling factors. As we propose a parameter-efficient architecture, parameter comparison is essential along with the testing performance. Our proposed RCRN19 model takes 18182 parameters compared to $297,216$ parameters of DRRN19. RCRN, constructed using RCN blocks, reduces by 94% the trainable parameters compared to the DRRN.

6 Discussion and Conclusions

This work introduces a new block, the RCN block, which is constructed with two sequential 1D DSCs and residual connections. This RCN block can be used as a replacement for any variation of 2D convolution. These modifications help to reduce trainable parameters, FLOPs, and latency, as well as improve validation performance on image classification tasks. We also checked this proposed block for widened ResNets, MobileNet, SqueezeNext, and CMTs architectures and showed that the RCNs-based wide ResNets, MobileNet, SqueezeNext, and CMTs obtain better accuracy and take fewer parameters, FLOPs, and latency. As CMT uses fewer CNN layers, our RCMT can not reduce parameters, FLOPs, and latency significantly. We also checked the effectiveness of our RCN block on the SISR task to show how the RCN block performs for other areas than the classification. Our proposed recursive axial ResNets (RCRNs) improve image resolution and reduce around 94% trainable parameters compared to the other CNN-based super-resolution models. Extensive experiments and analysis show that RCNs can be shallow, deep, and wide. These are parameter-efficient and superior models for image classification and SISR. However, the limitation is that we cannot implement ImageNet or COCO object-detection-like datasets due to machine limitations. We have shown that our proposed model is a viable replacement for any 2D convolutional layer on the tested tasks. Further work is needed to determine the range of applications for which RCNs may offer advantages.

References

1. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding (2012)
2. Chen, L.C., Wang, H., Qiao, S.: Scaling wide residual networks for panoptic segmentation. arXiv preprint arXiv:2011.11675 (2020)
3. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1251–1258 (2017)
4. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 295–307 (2015)
5. Gholami, A., Kwon, K., Wu, B., Tai, Z., Yue, X., Jin, P., Zhao, S., Keutzer, K.: Squeezenext: Hardware-aware neural network design. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1638–1647 (2018)
6. Guo, J., Han, K., Wu, H., Tang, Y., Chen, X., Wang, Y., Xu, C.: Cmt: Convolutional neural networks meet vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12175–12185 (2022)

7. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
9. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European conference on computer vision. pp. 630–645. Springer (2016)
10. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
11. Huang, G., Sun, Yu., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep Networks with Stochastic Depth. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 646–661. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_39
12. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360 (2016)
13. Khan, R.U., Zhang, X., Kumar, R.: Analysis of resnet and googlenet models for malware detection. Journal of Computer Virology and Hacking Techniques **15**, 29–37 (2019)
14. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1646–1654 (2016)
15. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1637–1645 (2016)
16. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
18. Le, Y., Yang, X.S.: Tiny imagenet visual recognition challenge (2015)
19. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
20. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. Advances in neural information processing systems **29** (2016)
21. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
22. Shahadat, N., Maida, A.S.: Deep separable hypercomplex networks. In: The International FLAIRS Conference Proceedings. vol. 36 (2023)
23. Shahadat, N., Maida, A.S.: Enhancing resnet image classification performance by using parameterized hypercomplex multiplication. In: 2023 26th International Conference on Computer and Information Technology (ICCIT). pp. 1–6. IEEE (2023)
24. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
25. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3147–3155 (2017)

26. Wang, C., Chen, D., Hao, L., Liu, X., Zeng, Y., Chen, J., Zhang, G.: Pulmonary image classification based on inception-v3 transfer learning model. IEEE Access **7**, 146533–146541 (2019)
27. Wu, Z., Shen, C., Van Den Hengel, A.: Wider or deeper: Revisiting the resnet model for visual recognition. Pattern Recogn. **90**, 119–133 (2019)
28. Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution via sparse representation. IEEE Trans. Image Process. **19**(11), 2861–2873 (2010)
29. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)

Overcomplete U-Net Networks for Psoriasis Detection in Digital Color Images

Aruna Kumari Kovvuru[1], Narendra D. Londhe[1](✉), Ritesh Raj[2], and Rajendra S. Sonawane[3]

[1] Electrical Engineering Department, National Institute of Technology Raipur, Raipur, India
nlondhe.ele@nitrr.ac.in
[2] Amrita School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Bengaluru, India
r_ritesh@blr.amrita.edu
[3] Psoriasis Clinic and Research Center, Psoriatreat, Pune, India

Abstract. It is challenging to detect the presence of Psoriasis using subjective means. Its objective determination can help to understand the coverage and severity of the disease and consequently offer the appropriate treatment. Deep learning methods like U-Net are very popular methods for segmenting disease regions for objective analysis. U-Net is an encode-decode under-complete convolution network that focuses on learning high-level features and fails in detecting fine boundaries and smaller lesions. Hence in this paper, the overcomplete version of U-Net and its variants Residual U-Net, and Attention U-Net are studied for psoriasis lesions segmentation from the full body color images. The overcomplete versions are found sensitive focusing from larger to smaller regions providing more precision in identifying the impacted Psoriasis skin lesions. They showed significant performance measured using the Dice similarity index as 0.9280, 0.9780, and 0.9834 for Overcomplete U-Net, Overcomplete Residual U-Net, and Overcomplete Attention U-Net, respectively. Among them, Overcomplete Attention U-Net has demonstrated superior performance compared to others.

Keywords: U-Net · Image segmentation · Deep learning · Overcomplete convolutional layers

1 Introduction

Psoriasis is a noncontagious, long-term skin condition in which red, irritated, dry flaky, or wet plaques appear all over the human body [1]. Psoriasis is of various types like plaque, guttate, pustular, inverse and erythrodermic but the most common type is the plaque psoriasis [2]. Psoriasis patients generate their skin cells ten times faster than those in the healthy range, which results in thick epidermal layers. The unpleasant patches that result from this aberrant growth have a detrimental effect on the affected individuals' quality of life [3]. Psoriasis can cause physical signs and discomfort, but it can also have serious psychological and emotional impacts on people, affecting their general well-being. Good management and treatment plans are essential for improving

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 145–159, 2025.
https://doi.org/10.1007/978-3-031-78312-8_10

the mental and emotional well-being of the affected individual as well as for addressing the physical symptoms. Although there is currently no known treatment for psoriasis, the ailment can be effectively treated with consistent, organized approaches to disease management [4].

Most psoriasis assessments are currently subjective and depend on physician's physical and visual assessments. However, because individual perceptions vary, this subjective assessment process is laborious and prone to errors [5]. Furthermore, there are difficulties in accurately diagnosing psoriasis lesions due to significant problems in terms of severity grades and contrast to healthy skin. The creation of an automated, accurate, and effective psoriasis diagnostic method is badly needed to overcome these issues. The main challenge in putting into practice an automated diagnostic system is correctly segmenting lesions because they vary widely in size, shape, and color of Upper extremities, trunk, and back body regions, and lower extremities as shown in Fig. 1. Psoriasis is mainly assessed by the area, redness (color), scaliness (whitish appearance) and thickness (elevation). In addition to streamlining the diagnostic procedure, creating a strong automated system will improve the precision of lesion identification and severity grading, which would lead to more successful psoriasis therapy approaches.

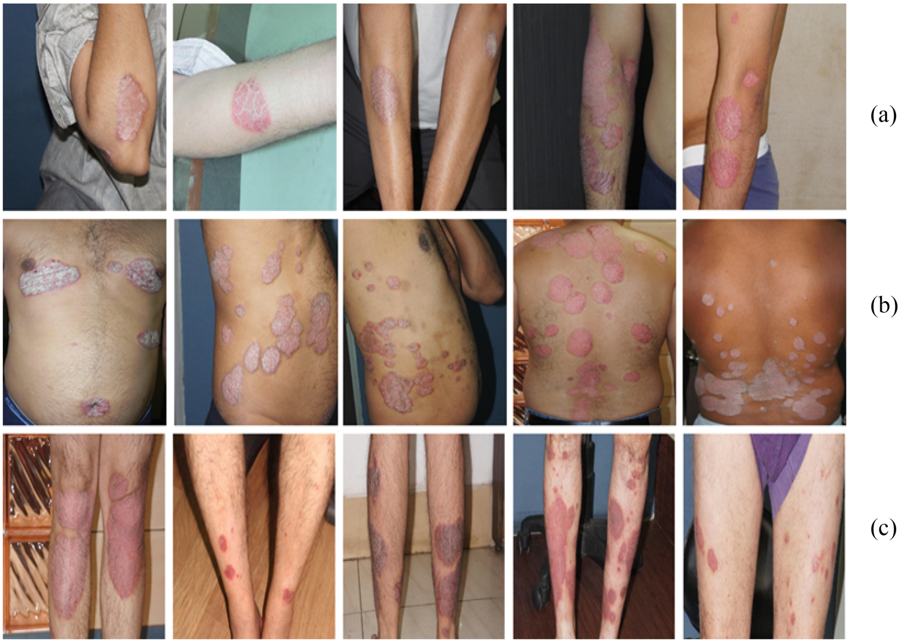


Fig. 1. Sample images of psoriasis patients of Indian origin (a) Upper extremities, (b) trunk and back body regions and (c) lower extremities (Note: The head area is omitted to protect the patient's identity through non-disclosure)

The most common method of automation adapted initially for this application was machine learning. However, these machine-learning strategies developed for the segmentation of psoriasis lesions rely largely on handmade characteristics and focus on clipped patches [6–10]. The introduction of deep learning models has nevertheless brought notable improvements in this application with no manual pre-processing and feature engineering [11]. Similarly, their contribution to identifying regions of interest in any kind of image has been significantly enhanced [12]. They follow pixel-wise classification followed by making borders around different pixel classes. Among them, one of the very popular models is U-Net inspired by the Fully Convolution Network (FCN). U-Net was a breakthrough revolution proposed by Ronneberger et al. [13] at the MICCAI conference in 2015 to address the critical problem of detecting small to large objects in medical images. This has given extensive focus to researchers on the application of deep learning technology in medical imaging. The U-shaped structure of U-Net consists of three modules i.e., encoder, bottleneck, and decoder. The acclimatization of U-Net to the smaller input data is faster due to context information through skip connections.

There are numerous applications of U-Net reported in the literature of medical image segmentation. The major challenges with medical image datasets are that they contain image elements with poor pixel contrast with the background, voluminous images, and tiny nature. Mostly medical imaging advancements have reached the generation of 3D images which are often processed using 2D image processing methods [14].This faces issues like computational complexity and poor performance. Hence to address these issues, a 3D U-Net was proposed and extensively used in volumetric CT and MR image segmentation studies [15–23] for single to multi-organ segmentation and for detecting the malignant cardiac objects, brain tumours, lung nodules, liver tumors, bone objects etc. U-Net is further equipped with the advantage of an attention gate [24] which helps to learn explicitly the regions of interest. Attention U-Net achieves this by making use of the attention gate that reduces the features not significant to the target region. It has been found effective in the diagnosis of cervical cancer, lung cancer, skin diseases like psoriasis, melanoma, brain tumours, fetus growth [25], and abdominal abnormalities [26]. In one other variant inception U-Net [27], filters of variable sizes have been used across each layer. They have been found to have less computational burden and effective performance. The challenge inherent in training neural networks with significant depth has been solved in residual U-Net by adding the input of the first layer to the output of the second layer via a skip connection. These residual skip connections help to counter the vanishing gradient problem and hence converge the deep networks faster. Their application in medical images to mine valuable information has been widely explored and found successful.

Furthermore, for the advancement of segmentation, Wang et al. [28] have proposed large and deeply supervised UNet + + having densely connected nested skip connections for seamless semantic learning. They have been applied for the successful segmentation of cell nuclei [29], cancer tissue [29], cardiac structures and vessels [30, 31], and pelvic organs.

For further accuracy improvement in medical image segmentation for organs of different sizes, a new version of U-Net named UNet 3 + was also devised by Huang et al. [32]. It exploits full-scale skip connections and deep supervisions by adding both low-level and high-level details at each level to learn hierarchical representation. The local inherent behaviour of convolution limits U-Net to learning dependencies. Hence transformer-based U-Net, TransUNet was proposed by Chen et al. [33] which has joined focus on global and local contexts from the input sequence of the CNN feature map.

To overcome the limitation of traditional U-Net, we present an Overcomplete U-Net, a novel segmentation architecture that departs from the conventional encoder-decoder paradigms. Unlike traditional techniques that map inputs to lower-dimensional embeddings, Overcomplete U-Net maps input data to higher dimensions spatially by increasing filters. By limiting the expansion of the receptive field size in deeper layers and enabling the extraction of small features, this strategic approach improves the network's ability to segment the small lesions on full body image. Overcomplete U-Net merges U-Net's ability to capture high-level features with the overcomplete capturing of high-quality low-level features. This is particularly beneficial for datasets containing diverse structural annotations.

2 Related Work

2.1 Traditional Encoder-Decoder Networks

In the past, researchers have commonly utilized traditional encoder-decoder networks, such as U-Net and its variants, which have demonstrated decent performance, as discussed in the introduction [13–18]. However, these methods frequently fail to accurately detect small lesions, which is crucial for precise diagnosis. Even when dealing with larger structures, U-Net may encounter challenges in accurately segmenting multiple lesions.

To demonstrate the effectiveness of U-Net, researchers evaluated it on the psoriasis dataset [34, 35]. The results indicate that U-Net struggles to detect multiple lesions in full-body images and may inaccurately segment them. Further analysis reveals that the architecture of traditional encoder-decoder networks, characterized by down-sampling in the encoder, results in an increased receptive field in deeper layers, prioritizing high-level features over the fine details necessary for accurate segmentation.

2.2 Overcomplete Representation

Overcomplete representations were initially demonstrated in denoising autoencoder models as effective feature detectors for image segmentation from noisy images. Later, a few researchers adopted overcomplete networks in various fields to demonstrate their superior ability to approximate various statistical distributions present in data [36]. Due to their increased resilience to noise compared to undercomplete representations, overcomplete representations are widely utilized for tasks such as source separation in signal mixes, signal reconstruction from noisy data, and biomedical image and volumetric segmentation [36–38]. According to research findings, overcomplete fully connected networks can more accurately detect features compared to traditional bottleneck topologies when utilized in denoising autoencoders. By leveraging the benefits of overcomplete across different network architectures, we aim to further enhance the segmentation performance for psoriasis lesion identification, ultimately improving diagnostic accuracy and patient care outcomes.

3 Methodology

In this work, we proposed Overcomplete U-Net architectures for identifying psoriasis lesions from digital color images. In this regard, we have implemented and tested an Overcomplete U-Net and its two variants namely Overcomplete Residual U-Net and Overcomplete Attention U-Net. In the proposed overcomplete architectures, the encoder part projects the input image into spatially higher dimensions by increasing filters at deeper layers which helps to increase feature mapping. The details of all three overcomplete networks are described in the sub-sections below.

3.1 Overcomplete U-Net

Overcomplete U-Net projects the input image (512 × 512) into a spatially higher dimension in the encoder path. Over-completing is achieved by incorporating a greater number of hidden layers in the encoder module thereby preserving finer details and enhancing the network's ability to capture subtle features. The proposed overcomplete network constrains the receptive field from expanding excessively as the network depth increases in two steps with constant dimension at every layer, as shown in Fig. 2. Every layer consists of two blocks with each block comprising conv2D(C), ReLU(R), and Batch Normalization (BN). The mathematical operations of the convolution layer with the kernel size of (3,3) and ReLU are represented by Eqs. (1) and (2), respectively.

$$O_{i,j} = \sum_{m} \sum_{n} I_{i+m,j+n} \times K_{m,n} \tag{1}$$

$$f(x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases} \tag{2}$$

where, $O_{i,j}$ is output feature map at position (i, j) and $K_{m,n}$ denote kernel/filter position at (m, n).

Furthermore, in the decoder module of Overcomplete U-Net architecture, each upsampling block consists of an Up-Sampled layer followed by a concatenation layer. The concatenate layer concatenates the up sample of each layer and the conv2D of the encoder layer.This design choice enhances the network's capability to capture detailed features during the segmentation process, facilitating the accurate delineation of psoriasis lesions even in challenging scenarios.

3.2 Overcomplete Residual U-Net

In this subsection, we present the outlines of the proposed Overcomplete Residual U-Net. This variation of U-Net incorporates residual network (i.e. ResNet) blocks into the conventional framework. By alleviating the vanishing gradient issue, ResNet blocks allow the network to learn residual mappings, which facilitates the training of deeper models. Each ResNet block is composed of two subblocks, each block consists of Conv2D, ReLU, and Batch Normalization, after which the output of the second convolutional layer is combined with a shortcut connection from an earlier layer using an element-wise addition operation based on dilation. This process improves the model's capacity to detect

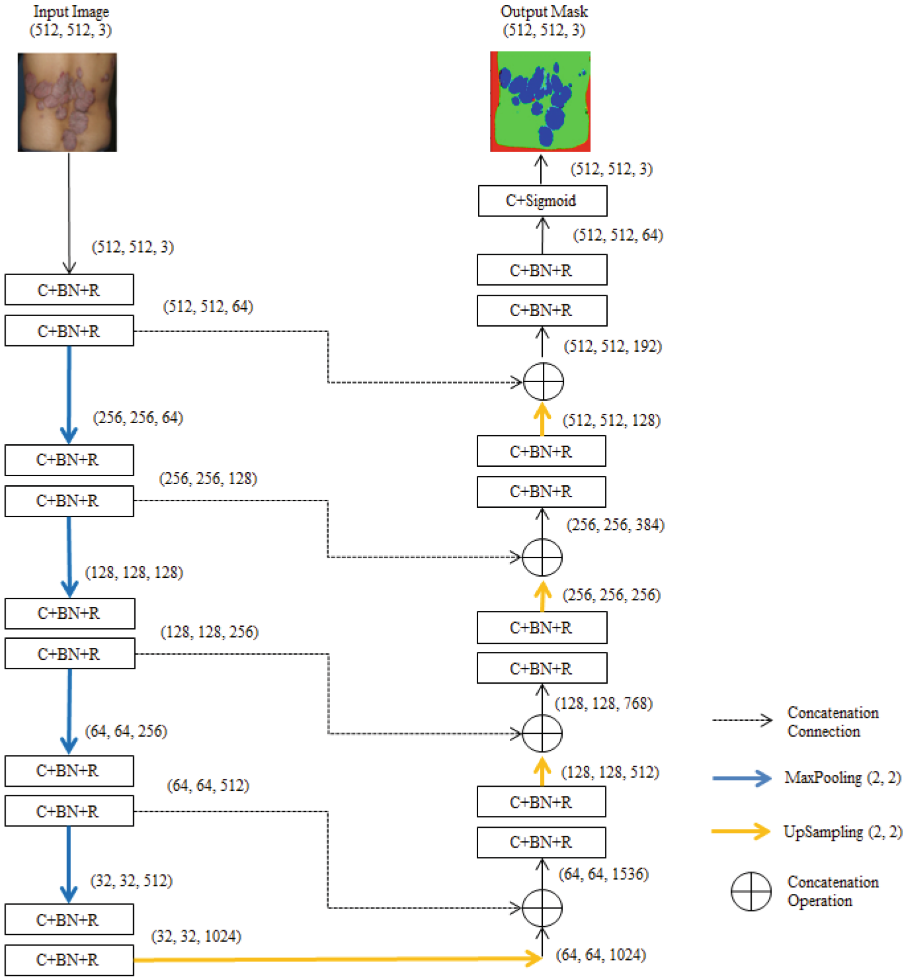


Fig. 2. Architecture details of Overcomplete U-Net Network. The output dimension of each layer is represented in the form of (*height*, *width*, *channel*)

small lesions accurately by facilitating the gradient flow during training. The encoded feature maps are processed in this manner until the lowest resolution is achieved.

Subsequently, in the decoder path by using the up-sampling layer, we up-sample the input signal which will be further input of the concatenate layer. The concatenate layer concatenates the output of the up-sample layer and conv2D layer for producing encoded feature maps. Furthermore, skip connections combine feature maps at matching resolutions from the encoder and decoder to provide accurate object boundary localization in the output.

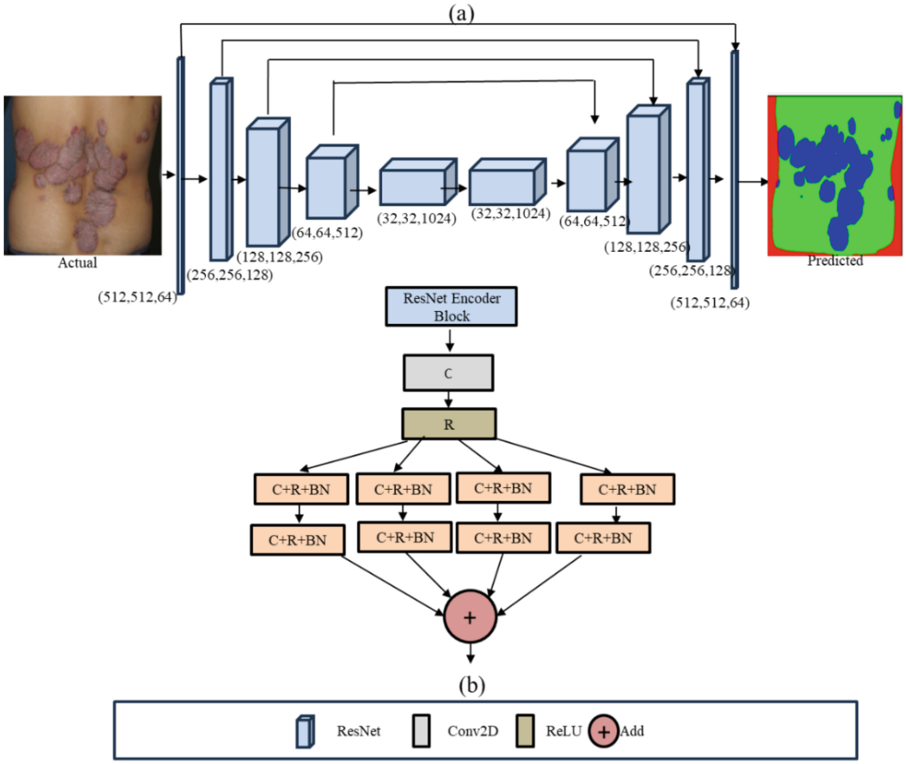


Fig. 3. Details of Residual U-Net architecture. The input image forwarded to the 3 × 3 Conv2D of an Overcomplete Residual U-Net Network (a) and the Internal connection of the Residual U-Net encoder has been visualized in (b).

3.3 Overcomplete Attention U-Net

Here, we present the outlines of the proposed Overcomplete Attention U-Net. The encoder path is the same as described for U-Net and Residual U-Net. Attention methods are added to the EfficientNet-B1 backbone for feature extraction in Overcomplete Attention U-Net architecture at each up-sampling stage in the decoder to improve semantic segmentation. The attention mechanism uses Softmax normalization and the dot product to calculate attention scores. It also comprises key, query, and value operations. Subsequently, the feature map is enhanced additively with the weighted sum of values, emphasizing pertinent areas. The decoder consists of convolutional processes with batch normalization and ReLU activation, concatenation with encoder features, and up-sampling layers as shown in Fig. 4. Where M is the multiply and A is the activation.

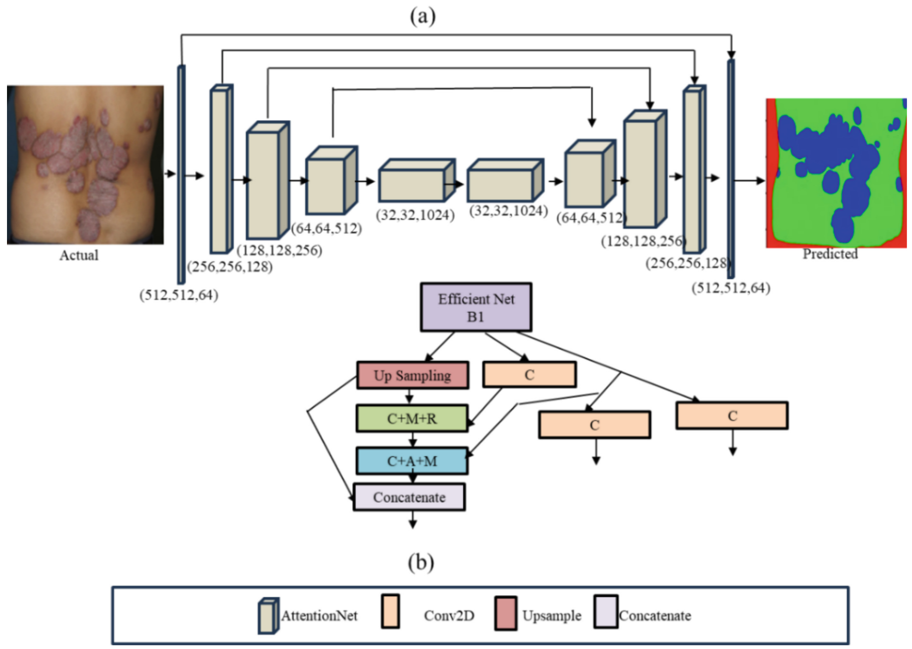


Fig. 4. Architecture details of Attention U-Net. The image forwarded the 3 × 3 Conv2D. (a) The architecture of the Overcomplete EfficientNet-B1 Attention U-Net Network and (b) Connection between the EfficientNet-B1 encoder and decoder path has been visualized.

4 Experimentation Details

4.1 Dataset

This research used psoriasis data from Psoriatreat, a Psoriasis Clinic and Research Centre in Pune, Maharashtra, India. The collection consists of 500 digital photos that were picked unbiasedly based on factors such as age, gender, race, or severity level from about 100 varied psoriasis patients. A measure to anonymize personal data was implemented, and ethical approval was secured for the creation of the dataset. Pictures of the head, upper limbs, trunk, and lower limbs were taken, among other body parts. For the sake of patient privacy, pictures taken in the head area were not included. A Sony NEX-5 camera with a 22 mm lens, utilizing uncontrolled settings, was used to take pictures at 350 dpi. Dermatologists and image tracing specialists worked together to manually produce segmentation annotations for psoriasis lesions using the Pixel Annotation tool. Preprocessing was done on the dataset to get it ready for the suggested model. First, using nearest-neighbour interpolation in OpenCV, all raw RGB images and their related ground truth labels were scaled to a fixed square dimension of 512 × 512. To improve data consistency and computational performance, pixel values were then standardized to the range [0, 1]. The dataset includes a wide range of unrestricted psoriasis photos with a variety of backgrounds and artefacts, including skin hair, shadows, clothes, variations in lighting, and differing perspectives. The production of a comprehensive dataset that includes a variety of noise sources and non-uniform backgrounds is made possible by the heterogeneity in imaging settings.

4.2 System Implementation

The Keras deep learning package, which is integrated with the TensorFlow framework, is used in this study, which is carried out using Python language. An Intel(R) Xeron(R) Gold 6248R CPU @3.00GHz, 3001MHz, 24 Core(s), 48 Logical, and an NVIDIA RTX A4000 GPU with 16 GB of memory are the components of the workstation used for the experiments.

4.3 Training and Testing

The models are trained and tested using the holdout validation technique, which ensures robustness and reduces bias. The dataset is divided into 70:30 at random for this technique. 70% is used for training and 30% is used for testing. The other training parameters used for experimental purposes are listed in Table 1.

During training, the losses used are described below:

- *Categorical cross-entropy loss (per pixel per channel)*

The categorical cross-entropy loss measures the dissimilarity between the true labels and the predicted probabilities for each pixel in each channel. For a single pixel (*i, j, c*), the loss is computed as the negative logarithm of the predicted probability of the pixel's true class label. Mathematically, this loss is represented by Eq. (3) below:

$$Loss_{ijc} = -y_{ijc}\log(\hat{y}_{ijc}) \tag{3}$$

where y_{ijc} is the true label of pixel i in channel c and $\hat{y}_{ijc}$ is the predicted probability of i belonging to class j in channel c.

- *Overall Categorical cross-entropy loss (per batch)*

The overall loss for the entire batch is computed by summing the losses for all pixels in all channels across all images and then averaging over the batch size and is mathematically represented by Eq. (4) shown below:

$$OverallLoss = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{H}\sum_{c=1}^{C} Loss_{ijc} \tag{4}$$

where N is the batch size, H is the total number of pixels per image, and C is the number of channels (for RGB images, $C = 3$).

4.4 Evaluation Metrics

By comparing the predicted segmented lesion with the ground truth lesion pixel-by-pixel, the suggested method's quantitative performance for the desired job is evaluated. Important assessment measures used include the Dice Similarity Index (DI), which is

Table 1. Training parameter

Parameter	Value
Mini batch size	8
Filters	[64,128,256,512,1024]
Initial learning	0.0010
Epochs	100
optimizer	Adam

well-known for being useful in evaluating segmentation success. The DI metric measures how similar the two are, indicating how effective the segmentation model is. Furthermore, other metrics that are computed include Accuracy (ACC), Precision (PR), Sensitivity/Recall (SE), and F1-score. Equations (5)-(9) define these performance metrics mathematically in terms of false positive (FP), true negative (TN), false negative (FN) and true positive (TP).

$$DI = \frac{2TP}{2TP + FP + FN} \tag{5}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$PR = \frac{TP}{TP + FP} \tag{7}$$

$$SE = \frac{TP}{TP + FN} \tag{8}$$

$$F1 - score = \frac{2 \times PR \times SE}{(PR + SE)} = \frac{TP}{TP + 0.5(FP + FN)} \tag{9}$$

5 Results and Discussion

The proposed overcomplete networks of U-Net, Residual U-Net, and Attention U-Net are evaluated using different performance indices, and the results are reported in Table 2. From Table 2, it can be observed that the DI of Attention U-Net is 0.9834, Residual U-Net is 0.9780, and U-Net is 0.9280. Attention U-Net outperforms the other two in terms of DI. Additionally in terms of other performance indices also Overcomplete Attention U-Net model performs better than all other compared networks with an ACC of 0.9909, PR of 0.9909, SE of 1.00 and F1-score of 0.9954.

Apart from reporting the segmentation results based on performance indices, we also present the visual comparison of segmentation results of actual RGB images with overcomplete networks of U-Net, Residual U-Net, and Attention U-Net in Fig. 5. It can be observed from Fig. 5 that Overcomplete Attention U-Net segments the small

Table 2. Performance Comparison of Various Overcomplete U-Net Models

Parameters	Overcomplete Attention U-Net	Overcomplete Residual U-Net	Overcomplete U-Net
DI	0.9834	0.9780	0.9280
ACC	0.9909	0.9744	0.9543
PR	0.9909	0.9748	0.8805
SE	1.0000	0.9760	0.9501
F1-Score	0.9954	0.9754	0.9140

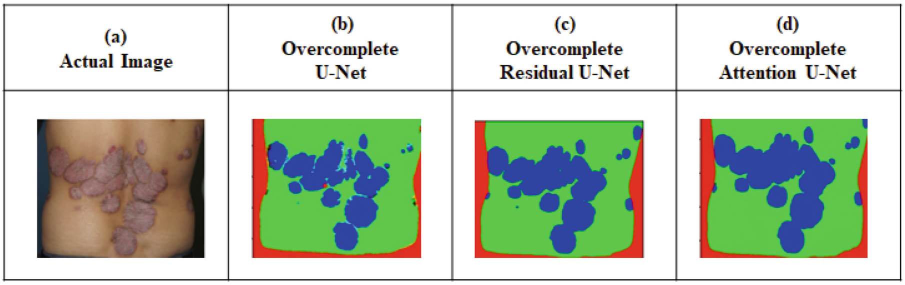


Fig. 5. (a) Actual Image, Segmentation prediction using (b) Overcomplete U-Net, (c) Overcomplete Residual U-Net and (d) Overcomplete Attention U-Net.

lesions more precisely than all other compared networks. This led to achieving better segmentation performance results with Overcomplete Attention U-Net compared to the other two models listed in Table 2.

To validate the effectiveness of overcomplete networks compared to the traditional approaches for identifying psoriasis lesions, we reported a comparison in Table 3. Based, on Table 3 it is clear that the overcomplete networks achieved a better result than the traditional networks. Also, the Overcomplete Attention U-Net model shows one of the most promising performances out of all the models.

Table 3. Comparison between the traditional approach and overcomplete networks

S.No	Networks	DI metric	
		Traditional Networks	Overcomplete Networks
1	U-Net [34]	0.8834	0.9280
2	Residual U-Net [35]	0.9481	0.9780
3	Attention U-Net [39]	0.9590	0.9834

Apart from the evaluation metrics we also present the training validation curve of Overcomplete Attention U-Net based on the performance. Figure 6. Shows the training and testing curves for this model. There are variations in the model during the first few training cycles, but eventually, at approximately the thirty-first epoch, adaptive changes in the learning rate cause the model to settle. A significant decline in the curve is what triggers this automatic adaptation. After stabilization, a well-performing model is indicated by the smallest gap between the training and validation curves.

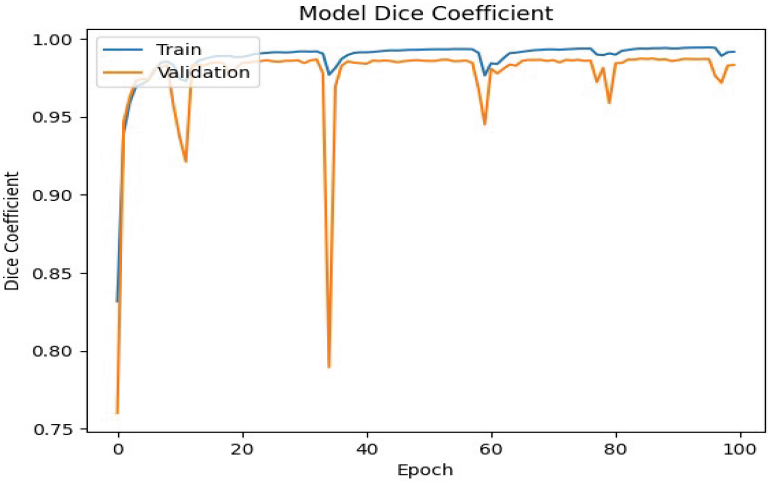


Fig. 6. Dice coefficient curve for Overcomplete Attention U-Net

On performing further analysis, it has been observed that the DI metric for the Overcomplete Attention U-Net model is less than 0.95 in some test images. This may be due to (a) huge irregular boundaries and the shape of multiple lesions with different severity levels, and (b) the high visual similarity between psoriasis and healthy skin region. These lead to over or under-segmentation of psoriasis lesions. These limitations may be overcome by using larger and more diverse datasets having images from multiple skin tones. However, a DI of more than 0.9 is obtained for all test images with the Overcomplete Attention U-Net model. This performance is satisfactory enough to obtain an objective and quantitative measurement of psoriasis area severity. This could help or assist dermatologists in reproducing objective psoriasis area severity easily with no change to perform prolonged diagnosis of psoriasis disease more efficiently.

6 Conclusion

We provide a series of fully automated deep learning-based methods in this research study, using U-Net and its variants (i.e. residual and attention networks) with an overcomplete approach. Without the use of feature engineering or preprocessing, these techniques seek to separate psoriasis lesions from digital photos obtained in a variety of environments. We verify these network's segmentation performance and obtain some impressive results: Overcomplete Attention U-Net achieves a DI of 0.9834 and an ACC of 0.9909, Overcomplete Residual U-Net yields a DI of 0.9780 and an ACC of 0.9744, and the simple U-Net with overcomplete approach yields a DI of 0.9280 and an ACC of 0.9543. These results show segmentation performance that is both efficient and promising. Besides, previous traditional deep learning-based segmentation models created for this problem are surpassed by our suggested model, among those suggested models Overcomplete Attention U-Net gave better results.

In summary, overcomplete networks offer a promising approach for addressing the limitations of traditional U-Net architectures in tasks such as medical image segmentation. By incorporating additional layers and carefully designing the network architecture, overcomplete networks enable more effective extraction of fine details and features, leading to improved segmentation performance and diagnostic accuracy.

Acknowledgement. Authors acknowledge the Science and Engineering Research Board (SERB), Government of India, for financial support vide Reference No. EEQ/2021/000129 to carry out this work.

References

1. Gudjonsson, Johann E., and James T. Elder. “Psoriasis: epidemiology.” Clinics in dermatology 25.6 (2007): 535-546.
2. https://www.psoriasis.org/about-psoriasis/
3. Sarac, Gulbahar, Tuba Tulay Koca, and Tolga Baglan. “A brief summary of clinical types of psoriasis.” Northern clinics of Istanbul 3.1 (2016): 79.
4. Kim, Whan B., Dana Jerome, and Jensen Yeung. “Diagnosis and management of psoriasis.” Canadian Family Physician 63.4 (2017): 278-285.
5. Feldman, S. R., and GG15708941 Krueger. “Psoriasis assessment tools in clinical trials.” Annals of the rheumatic diseases 64.suppl 2 (2005): ii65-ii68.
6. Dash, Manoranjan, et al. “Psoriasis Lesion Detection Using Hybrid Seeker Optimization-based Image Clustering.” Current Medical Imaging 17.11 (2021): 1330–1339.
7. Shrivastava, Vimal K., et al. “Computer-aided diagnosis of psoriasis skin images with HOS, texture and color features: a first comparative study of its kind.” Computer methods and programs in biomedicine 126 (2016): 98–109.
8. Shrivastava, Vimal K., et al. “Exploring the color feature power for psoriasis risk stratification and classification: A data mining paradigm.” Computers in biology and medicine 65 (2015): 54–68.
9. Shrivastava, Vimal Kumar, and Narendra D. Londhe. “Measurement of psoriasis area and severity index area score of Indian psoriasis patients.” Journal of Medical Imaging and Health Informatics 5.4 (2015): 675–682.
10. Shrivastava, Vimal K., et al. “A novel approach to multiclass psoriasis disease risk stratification: Machine learning paradigm.” Biomedical Signal Processing and Control 28 (2016): 27–40.
11. Razzak, Muhammad Imran, Saeeda Naz, and Ahmad Zaib. “Deep learning for medical image processing: Overview, challenges and the future.” Classification in BioApps: Automation of decision making (2018): 323–350.
12. Long, Jonathan, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

13. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18. Springer International Publishing, 2015.
14. Yin, Xiao-Xia, et al. "Anatomical landmark localization in breast dynamic contrast-enhanced MR imaging." Medical & biological engineering & computing 50 (2012): 91–101.
15. Wu, Jiong, Yue Zhang, and Xiaoying Tang. "Simultaneous tissue classification and lateral ventricle segmentation via a 2D U-net driven by a 3D fully convolutional neural network." 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2019.
16. Sánchez, José Carlos González, et al. "Segmentation of bones in medical dual-energy computed tomography volumes using the 3D U-Net." Physica medica 69 (2020): 241–247.
17. Bae, Hyun-Jin, et al. "Fully automated 3D segmentation and separation of multiple cervical vertebrae in CT images using a 2D convolutional neural network." Computer methods and programs in biomedicine 184 (2020): 105119.
18. Kolarik, Martin, et al. "Superresolution of MRI brain images using unbalanced 3D Dense-U-Net network." 2019 42nd International Conference on Telecommunications and Signal Processing (TSP). IEEE, 2019.
19. Owler, James, et al. "Comparison of multi-atlas segmentation and U-Net approaches for automated 3D liver delineation in MRI." Medical Image Understanding and Analysis: 23rd Conference, MIUA 2019, Liverpool, UK, July 24–26, 2019, Proceedings 23. Springer International Publishing, 2020.
20. Yu, Wei, et al. "Liver vessels segmentation based on 3d residual U-NET." 2019 IEEE international conference on image processing (ICIP). IEEE, 2019.
21. Zhao, Chen, et al. "Lung nodule detection via 3D U-Net and contextual convolutional neural network." 2018 International conference on networking and network applications (NaNA). IEEE, 2018.
22. He, Yu, et al. "A 3D dual path U-Net of cancer segmentation based on MRI." 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC). IEEE, 2018.
23. Heinrich, Mattias P., Ozan Oktay, and Nassim Bouteldja. "OBELISK-Net: Fewer layers to solve 3D multi-organ segmentation with sparse deformable convolutions." Medical image analysis 54 (2019): 1-9.
24. Oktay, Ozan, et al. "Attention u-net: Learning where to look for the pancreas." arXiv preprint arXiv:1804.03999 (2018).
25. Schlemper, Jo, et al. "Attention gated networks: Learning to leverage salient regions in medical images." Medical image analysis 53 (2019): 197–207.
26. Zhou, Zongwei, et al. "Unet++: A nested u-net architecture for medical image segmentation." Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4. Springer International Publishing, 2018.
27. Punn, Narinder Singh, and Sonali Agarwal. "Inception u-net architecture for semantic segmentation to identify nuclei in microscopy cell images." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 16.1 (2020): 1–15.
28. Wang, Zhou, Eero P. Simoncelli, and Alan C. Bovik. "Multiscale structural similarity for image quality assessment." The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. Vol. 2. Ieee, 2003.
29. Lin, Tsung-Yi, et al. "Focal loss for dense object detection." Proceedings of the IEEE international conference on computer vision. 2017.

30. Isensee, Fabian, et al. "nnU-Net for brain tumor segmentation." Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 6th International Workshop, BrainLes 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Revised Selected Papers, Part II 6. Springer International Publishing, 2021.
31. Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).
32. Huang, Huimin, et al. "Unet 3+: A full-scale connected unet for medical image segmentation." ICASSP 2020–2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2020.
33. Chen, Jieneng, et al. "Transunet: Transformers make strong encoders for medical image segmentation." arXiv preprint arXiv:2102.04306 (2021).
34. Raj, Ritesh, Narendra D. Londhe, and Rajendra Sonawane. "Automated psoriasis lesion segmentation from unconstrained environment using residual U-Net with transfer learning." Computer Methods and Programs in Biomedicine 206 (2021): 106123.
35. Raj, Ritesh, Narendra D. Londhe, and Rajendra S. Sonawane. "Automatic psoriasis lesion segmentation from raw color images using deep learning." 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2020.
36. Lewicki, Michael S., and Terrence J. Sejnowski. "Learning overcomplete representations." Neural computation 12.2 (2000): 337-365.
37. Valanarasu, Jeya Maria Jose, et al. "Kiu-net: Overcomplete convolutional architectures for biomedical image and volumetric segmentation." IEEE Transactions on Medical Imaging 41.4 (2021): 965–976.
38. Sadikine, Amine, et al. "Semi-overcomplete convolutional auto-encoder embedding as shape priors for deep vessel segmentation." 2022 IEEE International Conference on Image Processing (ICIP). IEEE, 2022.
39. Soni, Samiksha, Narendra D. Londhe, and Rajendra S. Sonawane. "Improving Performance of Psoriasis Lesion Segmentation Using Attention-UNet with EfficientNet Encoder." 2022 IEEE 1st International Conference on Data, Decision and Systems (ICDDS). IEEE, 2022.

Check for updates

Dual-Branch Task Residual Enhancement with Parameter-Free Attention for Zero-Shot Multi-label Image Recognition

Shizhou Zhang[1], Kairui Dang[1], De Cheng[2], Yinghui Xing[1(✉)], Qirui Wu[1], Dexuan Kong[1], and Yanning Zhang[1]

[1] Northwestern Polytechnical University, Xian, China
xyh_7491@nwpu.edu.cn
[2] Xidian University, Xian, China

Abstract. Zero-shot multi-label image recognition involves the task of recognizing multi-label images while “zero” visual information has been input into the model during training. Recently, with the emergence of large pre-trained vision-language model, the visual and semantic features can be well aligned after being trained with billions of image-text pairs collected from the internet. In this paper, by utilizing the pre-trained CLIP model, we propose a dual-branch task residual enhancement with parameter-free attention module that enhances interaction of inter-modal information to tackle the problem of multi-label image recognition. The method employs a dual-branch structure, including global and local branches. The local branch mitigates global feature dominance, improving image content understanding ability of local regions. Our method shows superiority in zero-shot multi-label learning on VOC2007, MS-COCO, and NUS-WIDE datasets, surpassing the state-of-the-art methods. Additionally, it also has excellent performance in partial label settings. Code is available in the supplementary materials.

Keywords: Multi-label · Task Residual · Zero-shot · Dual-branch

1 Introduction

Multi-label image recognition (MLR) aims to identify all object categories or concepts that appear in an input image. Due to the intrinsic multi-label nature of images, it has great potential value of developing specific algorithms to solve multi-label image recognition problems. It benefits a comprehensive understanding of the complex scenarios and can be helpful to other tasks like image retrieval

S. Zhang and K. Dang—The first two authors equally contributed to this work.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78312-8_11.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 160–171, 2025.
https://doi.org/10.1007/978-3-031-78312-8_11

etc.. In practice, some categories are not easy to collect for training an MLR model in many scenarios, which requires the model to be capable of recognizing novel classes in the 'zero-shot' setting.

With the emergence of large pre-trained vision-language models, like CLIP [1] and ALIGN [2], a well-aligned image-text feature space can be obtained after training with billions of image-text pairs in a contrastive learning way. Therefore, the pre-trained vision-language model can be empowered with a great capability of recognizing novel classes. To efficiently transfer the large pre-trained model to downstream tasks, there exists two types of popular and effective methods, namely prompt tuning [3–7] and adapter-style tuning [8–12] which aims to tune the model with a small portion of parameters. However, prompt tuning and adapter tuning can unavoidably damage the prior knowledge or can be excessively biased towards the prior knowledge [13]. To alleviate the problems, task residual learning is proposed to keep the original synthesized classifier weights frozen and introduce a set of prior-independent parameters that are added to the weights. However, it is not suitable for multi-label recognition.

On the other hand, prompt tuning methods require sufficient image data to achieve promising performances. Guo et al. [3] proposed TaI-DPT to learn the prompt with texts only given a large corpus of easily accessible image captions as alternatives form. In this paper, we improve the framework of TaI-DPT from two aspects. Firstly, we propose a dual-branch task residual to take consideration of both global and local concepts. With the help of the prior-independent dual-branch task residual, the final multi-label recognition results can be greatly boosted. Secondly, to make the image/text features concentrate more on the target classes, we employ a parameter-free attention module to enhance the interaction of intermodal information. To be specific, the dual-branch task residual is devised to preserve prior knowledge from both global and local perspectives, thereby enhancing flexibility, scalability, and the capability to learn task-specific knowledge. While the parameter-free attention mechanisms aim to identify relevant image parts with a prompt description. To summarize, the contributions of this work include:

- We propose a dual-branch task residual module to transfer the pre-trained models from both the global and local aspects for the multi-label image recognition task.
- Providing a bridge for the communication of inter-modal information, we additionally propose utilizing a parameter-free attention module to enhance the image/text features, focusing more on target classes.
- Our method has achieved excellent results on three zero-shot multi-label recognition datasets, namely VOC2007, MS-COCO, and NUS-WIDE, surpassing the state-of-the-arts methods. Furthermore, it performs excellently in partial label multi-label recognition tasks as well.

2 Related Work

2.1 Multi-Label Image Recognition

Multi-label recognition aims to identify all target objects in an image. To explore richer image information, various existing methods mine the relationship

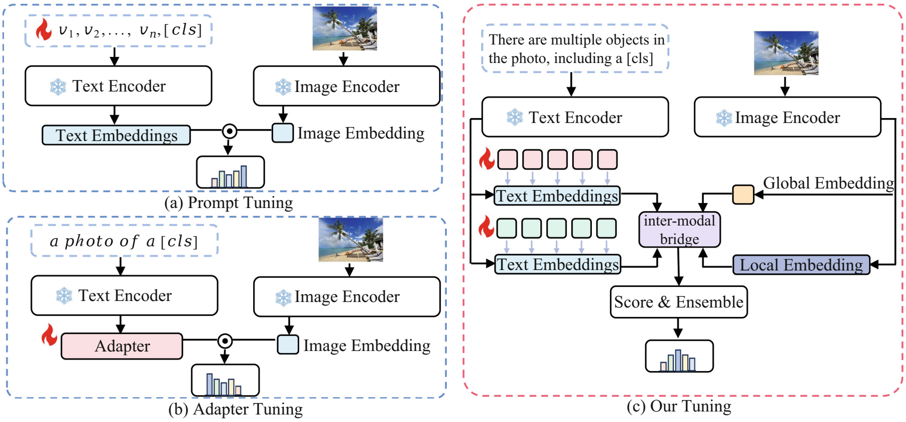


Fig. 1. Architecture comparison between existing tuning and our tuning for muti-label recognition. Prompt Tuning and Adapter Tuning have no communication between modalities before calculating similarity. We utilize a parameter-free attention module (inter-modal bridge) to enhance the image/text features, focusing more on target classes, thus providing a bridge for the communication of inter-modal information. And our tuning is devised to preserve prior knowledge from both global and local perspectives, while enhancing flexibility, and the capability to learn task-specific knowledge.

between labels through approaches such as graph convolution network (GCN), and enhance local information exploration to refine perception of small targets.

In order to explore the relationship between labels, methods such as SCP-Net [14] use GCN to learn semantic graph embeddings in multi-label classification. MlTr [15] effectively combines pixel attention and cross window attention to make local preliminary judgments on the target, and then globally matches relevant features to solve the problem of identifying small local targets. FL-Tran [16] utilizes a multi-scale fusion module to learn multi-scale features and accurately identify small-scale targets in an image. At the same time, it uses feature enhancement and suppression modules to mine various potential target object features in an image.

Although these methods are effective, they require a substantial amount of annotated images for training in order to achieve better performance. It remains a challenging issue for learning multi-label image recognition in image-limited or label-limited regimes. Based on vision-language pre-trained models (CLIP), we propose a dual-branch task residual with parameter-free attention approach for zero-shot multi-label recognition.

2.2 Efficient Transfer Learning for Vision-Language Models

Large vision-language pre-trained models (VLMs) have learned general visual representations and broad visual concepts. How to exploit these prior knowledge to multi-label recognition is a challenge that urgently needs to be addressed.

Efficient Transfer Learning (ETL) represents parameter efficient and data efficient transfer learning. Existing efficient transfer methods can be divided into the following two categories: prompt tuning and adapter style tuning. Prompt tuning [3,14,17] lacks prior knowledge preservation. Although the weights of pre-trained text encoder is frozen in prompt tuning, the original well learned classification boundaries are more or less damaged. It abandons the pre-trained text classifier and generates new one, which results in the loss of prior knowledge from VLMs. In addition, it also requires the pre-trained text encoder to participate in the training phase to generate new text embeddings after each parameter updates, which limits its scalability and increases computational overhead. Adapter style tuning [9,10] limits the flexibility of exploring new knowledge. Adapters fine tune text embeddings without learning independent knowledge to adapt to specific tasks, as its input is strictly limited by old pre-trained knowledge. Regardless of whether the pre-trained features are suitable for the task, the results of the adapter only depend on them, which limits the flexibility of adapter style tuning to learn new knowledge. Fig. 1 shows the architecture comparison between existing tuning for muti-label recognition and our tuning for muti-label recognition. To address these issues, we propose a dual-branch task residual with parameter-free attention method to solve the problem of efficiently transferring the prior knowledge from VLMs to multi-label recognition tasks.

3 METHODS

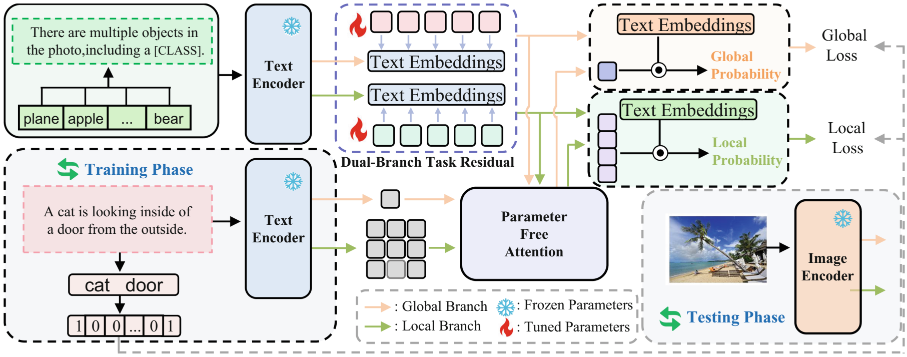


Fig. 2. An overall illustration of the proposed dual-branch task residual enhancement with parameter-free attention (DTRPA) framework for zero-shot multi-label image recognition. It consists of the dual-branch task residual (DBTR) and parameter-free attention (PFA) modules that aim to capture more fine-grained task-specific features and the interaction of inter-modal information.

The overview of our proposed method is illustrated in Fig. 2. We use two identical text encoders of pre-trained CLIP model to encode prompts and text descriptions. Dual branches are used when encoding handcraft template prompts. The text embeddings, acquired through the dual-branch task residual module, and image/text embeddings (treating text descriptions as images) are input into the parameter-free attention module to strengthen the interaction of inter-modal information and enhance the image/text features to concentrate on target classes. A noun filter is employed to produce classification pseudo-labels for every text description, thereby providing supervision for the classification outputs.

The traditional handcraft prompt 'A photo of a [class]' for single-label image recognition has limitations in multi-label recognition tasks, as it semantically restricts images to belong to only one category. We propose an improved template prompt for multi-label image recognition, 'There are multiple objects in the photo, including a [CLASS]'. This realistic prompt can handle label uncertainty, improve overall image understanding, and be applicable to complex scenarios, improving real-world multi-label recognition capabilities.

3.1 Dual-branch Architecture

A multi-label manual template prompt 'There are multiple objects in the photo, including a [CLASS]' is processed by text encoder. It generates two kinds of text embeddings: global and local text embeddings. The global and local text embeddings are processed through the dual-branch residual module to obtain the global classifier and the local classifier. In multi-label recognition, global features often prioritize the primary object, potentially overlooking other important objects. To address this, we introduce a local branch to mitigate global feature dominance. It enhances the exploration of fine-grained features and improves sensitivity to image details. The collaboration between the global and local branches allows us to comprehensively capture image information. This architecture can improve the multi-label recognition accuracy, especially in complex scenarios.

3.2 Dual-branch Task Residual Module

The dual-branch task residual module is a set of parameters that can be continuously optimized independently of the base classifier (based on text embeddings). The task residual can be represented as:

$$F_t^{'} = F_t + \alpha \cdot X, \tag{1}$$

where $X \in \mathbb{R}^{K\times D}$ denotes a set of learnable parameters, $F_t \in \mathbb{R}^{K\times D}$ represents the text embeddings obtained from the text encoder, which serves as the base classifier, K signifies the number of classes and D denotes the dimensionality. The hyperparameter α is used for scaling X, and we adopt an adaptive learning approach to learn a suitable coefficient α. We employ the hyperparameter α to scale the parameters X specifically learned for the given task. Subsequently, $\alpha \cdot X$ is incorporated into the base classifier to create a new classifier for the target task, denoted as $F_t^{'}$. In the global branch, the residual operation is as follows:

$$F_t^{g'} = F_t^g + \alpha \cdot X^g, \tag{2}$$

where F_t^g, $F_t^{g'}$, X^g, represents global base classifier, global new classifier, global learnable parameters, respectively. In the local branch, the residual operation is as:

$$F_t^{l'} = F_t^l + \alpha \cdot X^l, \tag{3}$$

where F_t^l, $F_t^{l'}$, X^l, represents local base classifier, local new classifier, local learnable parameters, respectively.

During the training phase, text descriptions are used to only optimize the prior-independent dual-branch task residual, while others remain frozen. The method allows us to reliably preserve prior knowledge and facilitate flexible exploration of new knowledge. During the testing phase, as shown in Fig. 2, we replace the text encoder with an image encoder to extract local and global features from the input test image.

3.3 Parameter-free Attention Module

To gain a deeper insight into the relationship between images and texts, we introduce an approach built on attention mechanisms. We utilize image/text embeddings as query vectors and text embeddings as key vectors to compute the attention matrix. This empowers us to identify the most relevant parts or features of an image concerning a given textual description, thereby enhancing the model's comprehension of the relationship between texts and images.

Given the global image embeddings F_{image}^g as query, and the global text embeddings processed by the dual-branch task residual moduleF_t^g as key and value, we first calculate the score matrix as follows:

$$Score = \frac{F_{image}^g \cdot F_t^g}{\sqrt{D}}, \tag{4}$$

where D represents the dimension of the key. Next, we use the softmax function to calculate attention weights:

$$A^g = Softmax(Score), \tag{5}$$

Finally, we use the attention weights A^g to sum the value vector by weight, to obtain the final output:

$$F_{image}^g = A^g \cdot F_t^g, \tag{6}$$

The same procedure is applied to the local branch, resulting in the final local image embeddings denoted as:

$$F_{image}^l = A^l \cdot F_t^l. \tag{7}$$

The global similarities p_i and aggregated local similarities $p_i^{'}$ are computed by:

$$p_i =< F_{image}^g, F_{t_i}^{g'} >, P_{ij} =< F_{image_j}^l, F_{t_i}^{l'} >, \tag{8}$$

where $< \cdot, \cdot >$ denotes the calculation of cosine similarity, i denotes the ith class, j denotes the jth patch of the image, P_{ij} denotes cosine similarities of each patch, $F^{g}_{image} \in \mathbb{R}^{1\times D}$ denotes either text features during training or visual features during testing of the global branch, and $F^{l}_{image} \in \mathbb{R}^{N\times D}$ denotes the corresponding part of the local branch. P_{ij} can be aggregated in a spatially weighted manner:

$$p_i^{'} = \sum_{j=1}^{N} \frac{\exp(P_{ij}/\tau_s)}{\sum_{j=1}^{N}\exp(P_{ij}/\tau_s)} \cdot P_{ij}, \tag{9}$$

where τ_s accommodates the extent of focusing on a specific location. p_i and $p_i^{'}$ are optimized by the loss terms $\mathcal{L}_g$ and $\mathcal{L}_l$, respectively. In the testing phase, p_i and $p_i^{'}$ are ensembled to obtain the final classification score.

We analyze the correlation between image and text by treating image as query and text as key. This approach enhances the model's understanding of the text-image relationship, highlighting relevant image elements. Multi-label recognition involves identifying multiple labels, potentially linked to different image regions. It allows the model to distinguish various labels in the image regions and assign accurate weights, improving multi-label classification accuracy.

3.4 Loss Function

The overall objective for the parameter optimization is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{global} + \mathcal{L}_{local}, \tag{10}$$

where $\mathcal{L}_{global}$ and $\mathcal{L}_{local}$ are obtained by calculating the classification probabilities from the global and local branches, respectively, and the pseudo labels are generated by the noun filter. We use the ranking loss [18] to measure the discrepancy between the predicted probabilities and the pseudo labels. $\mathcal{L}_{global}$ and $\mathcal{L}_{local}$ can be formulated as follows:

$$\begin{aligned}\mathcal{L}_{global} &= \sum_{i=1}^{c_+}\sum_{j=1}^{c_-} max(0, m - p_i + p_j),\\ \mathcal{L}_{local} &= \sum_{i=1}^{c_+}\sum_{j=1}^{c_-} max(0, m - p_i^{'} + p_j^{'}),\end{aligned} \tag{11}$$

where c_+, c_- denotes the positive labels and the negative labels, m represents the margin used to control the distance between positive and negative labels.

4 EXPERIMENTS

4.1 Experimental Settings

Datasets We evaluate our method on three widely adopted datasets, namely VOC2007 [21], MS-COCO [22], and NUS-WIDE [23]. VOC2007 includes 20 common categories, and we follow established references [17,24] to split it into a training set of 5,011 images and a test set of 4,952 images. MS-COCO contains 80 categories, with 82,081 training images and 40,504 validation images,

Table 1. Comparison with zero-shot methods and the current state-of-the-art (TaI-DPT) in zero-shot multi-label learning on VOC2007, MS-COCO, and NUS-WIDE datasets.

Method	Dual-branch	VOC2007	MS-COCO	NUS-WIDE	Avg.
ZSCLIP [1] (PMLR-21)	✗	76.2	47.3	36.4	53.3
	✓	77.3	49.7	37.4	54.8
TaI [3] (CVPR-23)	✗	86.0	61.1	44.9	64.0
	✓	$\underline{88.3}$	$\underline{65.1}$	46.5	66.6
DTRPA (Ours)	✗	88.1	65.0	**47.0**	$\underline{66.7}$
	✓	**89.7**	**68.6**	$\underline{46.7}$	**68.3**

conforming to official splits. NUS-WIDE, including 81 interconnected concepts, provides a comprehensive testing set of 107,859 images for the evaluation of our method. For zero-shot experiments in Sec. 4.2, following [3], we use captions from public MS-COCO and localized narratives from OpenImages as the language data source to learn the dual-branch task residual model.

Implementation Details We train our models with 100 epochs for zero-shot experiments on all datasets except for MS-COCO with 20 epochs. We employ Adam optimizer with an initial learning rate of 2e-3 on VOC2007 and NUS-WIDE, and SGD optimizer with 2e-2 on MS-COCO. The scaling factor α is set as a learnable parameter so as to adaptively adjust to the most suitable value. The experiment found that the optimal value of α is 4.7 on NUS-WIDE, 11.0 on MS-COCO and 5.0 on VOC2007. τ_s is set as 0.02 via validation. The batch size for the training is set to 256, while for the testing, it is set to 500. The input image is resized to (224, 224). Image transformations include operations such as random cropping and resizing, random flipping, and normalization, among others, which are used for data augmentation and pre-processing.

Evaluation Metric Following [3], we employ the mean average precision (mAP) as the evaluation metric for zero-shot setting. In partial label setting, we follow [3, 14, 17] to present the mAP for each proportion of labels available for optimization (from 10% to 90%) and calculate its overall average for all proportions.

4.2 Comparison with Zero-Shot Methods

In order to demonstrate the effectiveness of our proposed method, we compare the results with zero-shot CLIP [1] and the current state-of-the-art method in zero-shot multi-label learning, TaI-DPT [3]. Table 1 displays the results on VOC2007, MS-COCO, and NUS-WIDE. From Table 1, it can be observed that our method outperforms ZSCLIP by 13.5%, 21.3%, and 10.6% on the VOC2007, MS-COCO, and NUS-WIDE, respectively. Without using dual-branch architecture, our method exhibits an average improvement of +2.7% when compared to

Table 2. Comparison with the current state-of-the-art partial label multi-label recognition methods on VOC2007 and MS-COCO. * indicates the results of our reproduction.

Datasets	Method	10%	20%	30%	40%	50%	60%	70%	80%	90%	Avg.
MS-COCO	SST [19] (AAAI-22)	68.1	73.5	75.9	77.3	78.1	78.9	79.2	79.6	79.9	76.7
	SARB [20] (AAAI-22)	71.2	75.0	77.1	78.3	78.9	79.6	79.8	80.5	80.5	77.9
	DualCoOp [17] (NeurIPS-22)	78.7	80.9	81.7	82.0	82.5	82.7	82.8	83.0	83.1	81.9
	SCPNet [14] (CVPR-23)	80.3	**82.2**	**82.8**	**83.4**	**83.8**	**83.9**	**84.0**	**84.1**	**84.2**	**83.2**
	DualCoOp*	79.4	81.1	81.9	82.4	82.8	83.1	83.3	83.4	83.6	82.3
	+DTRPA (Ours)	**80.4**	81.8	82.5	82.9	83.2	83.5	83.7	83.8	83.9	82.9
VOC 2007	SST [19] (AAAI-22)	81.5	89.0	90.3	91.0	91.6	92.0	92.5	92.6	92.7	90.4
	SARB [20] (AAAI-22)	83.5	88.6	90.7	91.4	91.9	92.2	92.6	92.8	92.9	90.7
	DualCoOp [17] (NeurIPS-22)	90.3	92.2	92.8	93.3	93.6	93.9	94.0	94.1	94.2	93.2
	SCPNet [14] (CVPR-23)	91.1	92.8	93.5	93.6	93.8	94.0	94.1	94.2	94.3	93.5
	DualCoOp*	91.6	92.9	93.6	94.0	94.1	94.4	94.5	94.5	94.4	93.8
	+DTRPA (Ours)	**92.7**	**93.8**	**94.2**	**94.5**	**94.6**	**94.8**	**95.0**	**94.9**	**94.9**	**94.4**

the TaI-DPT method (+2.1% on VOC2007, +3.9% on MS-COCO, and +2.1% on NUS-WIDE). When incorporating dual-branch, our method demonstrates an average improvement of +1.7% across the three datasets compared to the TaI-DPT method (+1.4% on VOC2007, +3.5% on MS-COCO, and +0.2% on NUS-WIDE).

Our experiments indicate that the dual-branch task residual enhancement with parameter-free attention method can greatly improve the multi-label recognition results. With the dual-branch task residual module, the prior knowledge can be well preserved and greater flexibility is offered, whereas the parameter-free attention mechanisms enhance the accurate understanding of text descriptions, ultimately improving the recognition performance.

4.3 Comparison with Partially Labeled Methods

Following [14,17,19,20], our method can also perform multi-label recognition for partial labels. SCPNet [14] advocates addressing the partial label multi-label recognition (MLR) by deriving a structured semantic prior about the label-to-label correspondence via a semantic prior prompter. DualCoOp [17] encodes positive and negative contexts using prompts. With minimal additional learnable overhead on the VLMs, it swiftly adapts to partial label MLR tasks with limited annotations. We reproduce the partial labeled MLR of DualCoOp on the VOC2007 and MS-COCO datasets with the same experimental setting as reported. In Table 2, DualCoOp represents the original result, and DualCoOp* represents our reproduced result. The results indicate that our methods can enhance existing MLR methods. Our method exhibits an average mAP improvement of +0.9% on VOC2007 compared to the SOTA method. Especially when there are only few labels, our method exhibits an improvement of +1.6%. The average mAP surpasses DualCoOp by +0.6% on all datasets.

4.4 Ablation Study

Table 3 presents ablation experiments to show the effectiveness of the proposed task residual and parameter-free attention module. As shown in Table 2, when

Table 3. The ablation experiments about the dual-branch task residual (DBTR) and parameter-free attention (PFA) modules.

Method	Dual-branch	VOC2007	MS-COCO	NUS-WIDE	Avg.
TaI-DPT	✗	86.0	61.1	44.9	64.0
	✓	88.3	65.1	46.5	66.6
DBTR	✗	87.6 (+1.6)	64.0 (+2.9)	46.7 (+1.8)	66.1 (+2.1)
	✓	89.4 (+1.1)	67.5 (+2.4)	46.6 (+0.1)	67.8 (+1.2)
DBTR+PFA	✗	88.1 (+2.1)	65.0 (+3.9)	47.0 (+2.1)	66.7 (+2.7)
	✓	89.7 (+1.4)	68.6 (+3.5)	46.7 (+0.2)	68.3 (+1.7)

using only the task residual module, our method exhibits an average mAP improvement of +2.1% on three datasets in the single-branch setting and +1.2% in the dual-branch setting compared to the state-of-the-art methods. This suggests that our approach effectively preserves prior knowledge of VLMs to a considerable extent. Moreover, in Table 2, the inclusion of the parameter-free attention module alongside the task residual module brings an average mAP improvement of +0.6% in the single-branch scenario and +0.5% in the dual-branch scenario compared to the utilization of the task residual module alone. It's worth noting that the parameter-free attention module requires no additional training and performs exceptionally well on the MS-COCO dataset (+1.0% and +1.5%).

5 Conclusion

In this paper, we present a dual-branch task residual enhancement with parameter-free attention method to tackle the problem of zero-shot multi-label image recognition. In addition, it is also suitable for partial label MLR. Our method independently preserves prior knowledge from both global and local perspectives through a dual-branch task residual module, and utilizes a parameter-free attention module to enhance the interaction of inter-modal information and identify relevant image regions for prompt descriptions, thereby encouraging the image/text features to concentrate more on target classes. Experimental results on MS-COCO, VOC2007, and NUS-WIDE datasets validate the effectiveness of our approach.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (NFSC) under Grant 62101453 and 62201467; in part by the Project funded by China Postdoctoral Science Foundation under Grant 2022TQ0260 and Grant 2023M742842, in part by the Young Talent Fund of Xi'an Association for Science and Technology under Grant 959202313088, in part by Innovation Capability Support Program of Shaanxi (Program No. 2024ZC-KJXX-043), in part by the Fundamental Research Funds for the Central Universities No. HYGJZN202331 and in part by the Natural Science Basic Research Program of Shaanxi Province (No. 2022JC-DW-08).

References

1. Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al., *Learning transferable visual models from natural language supervision*, in *ICML*, pp. 8748–8763, 2021
2. Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, Tom Duerig, *Scaling up visual and vision-language representation learning with noisy text supervision*, in *ICML*, pp. 4904–4916, 2021
3. Zixian Guo, Bowen Dong, Zhilong Ji, Jinfeng Bai, Yiwen Guo, Wangmeng Zuo, *Texts as images in prompt tuning for multi-label image recognition*, in *CVPR*, pp. 2808–2817, 2023
4. Kaiyang Zhou, Jingkang Yang, Chen Change Loy, Ziwei Liu, *Learning to prompt for vision-language models*, *IJCV*, vol. 130, no. 9, pp. 2337–2348, 2022, Springer
5. Kaiyang Zhou, Jingkang Yang, Chen Change Loy, Ziwei Liu, *Conditional prompt learning for vision-language models*, in *CVPR*, pp. 16816–16825, 2022
6. Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, Ser-Nam Lim, *Visual prompt tuning*, in *ECCV*, pp. 709–727, 2022
7. Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, Kun Zhang, *Prompt learning with optimal transport for vision-language models*, arXiv preprint arXiv:2210.01253, 2022
8. Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, Yu Qiao, *Clip-adapter: Better vision-language models with feature adapters*, *IJCV*, pp. 1–15, 2023
9. Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, Hongsheng Li, *Tip-adapter: Training-free adaption of clip for few-shot classification*, in *ECCV*, pp. 493–510, 2022
10. Yi-Lin Sung, Jaemin Cho, Mohit Bansal, *Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks*, in *CVPR*, pp. 5227–5237, 2022
11. Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, Yu Qiao, *Vision transformer adapter for dense predictions*, arXiv preprint arXiv:2205.08534, 2022
12. Haoyu Lu, Mingyu Ding, Yuqi Huo, Guoxing Yang, Zhiwu Lu, Masayoshi Tomizuka, Wei Zhan, *UniAdapter: Unified Parameter-Efficient Transfer Learning for Cross-modal Modeling*, arXiv preprint arXiv:2302.06605, 2023
13. Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, Xinchao Wang, *Task residual for tuning vision-language models*, in *CVPR*, pp. 10899–10909, 2023
14. Zixuan Ding, Ao Wang, Hui Chen, Qiang Zhang, Pengzhang Liu, Yongjun Bao, Weipeng Yan, Jungong Han, *Exploring Structured Semantic Prior for Multi Label Recognition with Incomplete Labels*, in *CVPR*, pp. 3398–3407, 2023
15. Xing Cheng, Hezheng Lin, Xiangyu Wu, Dong Shen, Fan Yang, Honglin Liu, Nian Shi, *Mltr: Multi-label classification with transformer*, in *ICME*, pp. 1–6, 2022
16. Wei Zhou, Peng Dou, Tao Su, Haifeng Hu, Zhijie Zheng, *Feature learning network with transformer for multi-label image classification*, *PR*, vol. 136, pp. 109203, 2023, Elsevier
17. Sun, X., Ping, H., Saenko, K.: Dualcoop: Fast adaptation to multi-label recognition with limited annotations. NeurIPS **35**, 30569–30582 (2022)
18. Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev, Sergey Ioffe, *Deep convolutional ranking for multilabel image annotation*, arXiv preprint arXiv:1312.4894, 2013

19. Chen, T., Tao, P., Hefeng, W., Xie, Y., Lin, L.: Structured semantic transfer for multi-label recognition with partial labels. AAAI **36**(1), 339–346 (2022)
20. Tao, P., Chen, T., Hefeng, W., Lin, L.: Semantic-aware representation blending for multi-label image recognition with partial labels. AAAI **36**(2), 2091–2098 (2022)
21. Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, Andrew Zisserman, *The pascal visual object classes (voc) challenge*, *International journal of computer vision*, vol. 88, pp. 303–338, 2010, Springer
22. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
23. Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, Yantao Zheng, *Nus-wide: a real-world web image database from national university of singapore*, in *Proceedings of the ACM international conference on image and video retrieval*, pp. 1–9, 2009
24. Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, Yanwen Guo, *Multi-label image recognition with graph convolutional networks*, in *CVPR*, pp. 5177–5186, 2019

λ-Color: Amplifying Long-Range Dependencies for Image Colorization

Subhankar Ghosh[1](✉), Saumik Bhattacharya[2], Prasun Roy[1], Umapada Pal[3], and Michael Blumenstein[1]

[1] University of Technology Sydney, Sydney, NSW 2007, Australia
{subhankar.ghosh,prasun.roy}@student.uts.edu.au,
michael.blumenstein@uts.edu.au

[2] Indian Institute of Technology Kharagpur, Kharagpur 721302, WB, India
saumik@ece.iitkgp.ac.in

[3] Indian Statistical Institute Kolkata, Kolkata 700108, WB, India
umapada@isical.ac.in

Abstract. Colorization of images serves as a transformative tool, imbuing black and white pictures with vitality that mirrors the essence of the captured moment. Beyond merely transitioning aged images into modern color renditions, this process extends its reach to inferring colors for images where conventional color-capturing methods fail. In this paper, we introduce a novel algorithm designed to seamlessly convert grayscale images into perceptually consistent color compositions. We have also developed a novel layer by combining convolutional and lambda layers towards image colorization. Our proposed algorithm represents a significant advancement in the field of image colorization, offering a multifaceted solution to enhance visual storytelling and comprehension.

Keywords: Lambda Layers · Image Colorization · Transformer

1 Introduction

Image colorization holds immense significance as it has the power to inject vitality into monochrome images, revealing a plethora of emotions and intricacies hidden within grayscale limitations. Through the addition of color, historical visuals are revitalized, offering a richer understanding of the depicted scenarios and their surrounding ambiance. Moreover, colorization surpasses temporal barriers, allowing audiences to intimately engage with history. Colorization presents a significant challenge due to the diverse range of colors objects within a scene may possess, influenced by factors like lighting and texture. For instance, skin tones can vary under different lighting conditions, while landscapes may appear distinct based on time or season. To tackle this complexity, researchers have devised various colorization techniques, ranging from manual to automatic and semi-automatic methods. Manual colorization involves adding color by hand

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 172–187, 2025.
https://doi.org/10.1007/978-3-031-78312-8_12

using software like Photoshop, offering control and artistic freedom but demanding time and expertise. Automatic approaches leverage machine learning to predict colors based on patterns learned from extensive datasets. Semi-automatic methods, like scribbler-based techniques, allow user input for finer control. However, automatic colorization encounters challenges such as one-to-many associations, where a grayscale image can have multiple equally plausible colorizations. To address this, strategies like generative adversarial networks (GANs) and self-supervised learning have been proposed. Yet, many automatic methods struggle with color consistency and realism. To enhance results, we propose a novel Lamda net-based algorithm. These aids can guide the colorization process, ensuring more consistent and accurate outcomes. This holistic approach aims to improve colorization's realism, semantic understanding, and overall naturalness, bridging the gap between grayscale input and vibrant, lifelike color output. In this paper, we propose a lambda abstraction-based colorization model that takes grayscale image as input and produces the color components using local and global attentions computed using lambda module.

In this work, our key contribution is to include the long-range interactions without a transformer-based attention model for the colorization task. To the best of our knowledge, this is the first attempt to use lambda abstraction to invoke attention in the colorization process. Our extensive experiments show that our proposed method significantly outperforms the SOTA algorithms.

The remaining sections of this paper are structured as follows. In Section 2, we review existing literature pertaining to image colorization. Section 3 provides an overview of the pipeline and details the proposed methodology. Experimental results, encompassing dataset description, qualitative findings, comparisons with established methods, and various ablation studies, are presented in Section 4. Finally, Section 5 concludes the paper by summarizing key observations, addressing limitations, and outlining potential future avenues for enhancing the proposed algorithm.

2 Related Works

Over the past two decades, image colorization has emerged as a prominent focus in computer vision research, initially driven by conventional machine learning methods [4,13,22]. However, recent years have witnessed a shift towards deep learning (DL) techniques due to their remarkable success across various domains [2,10,24,26,27]. DL-based automatic image colorization systems have particularly demonstrated impressive performance [3,7–9,11,19,20,22,23,25,27,30].

The pioneering application of deep learning to image colorization was introduced by Cheng et al. in [8], employing a network architecture consisting of five fully connected layers with ReLU activation, and trained using the least-squares error loss function. Conversely, Carlucci et al. in [7] leveraged deep depth information from pre-trained ImageNet networks, utilizing them as feature extractors with frozen weights, and integrating this information into the colorization process.

In DDcolor [17], a dual decoder model is introduced for spatial resolution restoration. This model incorporates a query-based color decoder, which enhances features across multi-scale representations of color. By leveraging dual decoders, DDcolor effectively addresses spatial details while preserving color fidelity, contributing to the overall quality of the colorized images. Colorformer [16] presents a novel network architecture featuring a transformer-based encoder and a color memory decoder. By integrating global-local attention operations, Colorformer enhances global receptive field dependencies, thereby improving the overall coherence and realism of colorized images. This approach demonstrates the effectiveness of leveraging transformer architectures for image colorization tasks. Bigcolor [18] focuses on synthesizing vivid colors while alleviating the burden of synthesizing image structures. This is achieved by introducing a generative color prior learned by a BigGAN-inspired encoder-generator network. By decoupling color synthesis from structural synthesis, Bigcolor achieves superior results in terms of color fidelity and realism. CT2 [29] proposes an end-to-end transformer-based model designed to enhance color diversity in colorized images. Leveraging transformers' capability for long-range context extraction, CT2 adopts a holistic architecture that effectively captures intricate color variations and nuances. This approach showcases the potential of transformer architectures for advancing the state-of-the-art in image colorization.

3 Method

Long-range interactions without explicit attention mechanisms are a key aspect of many recent advancements in neural network architectures. These architectures are designed to capture dependencies between distant elements in the input data without relying on traditional attention mechanisms. Instead, they utilize various techniques to facilitate communication and information exchange across different parts of the network. One approach is to increase the receptive field of convolutional layers by stacking multiple layers or using dilated convolutions. This allows the network to capture information from a broader context without introducing additional parameters or computational overhead. Another technique incorporates recurrent connections between layers, enabling information to propagate across multiple time steps or processing stages. In colorization methods, the LAB color space is commonly employed. This method typically involves taking the "L" channel, which represents the grayscale image, and predicting the "AB" channel to add colorization. This process utilizes the LAB color space's separation of luminance (L) from chrominance (AB), allowing for efficient colorization algorithms.

3.1 Lambda Networks

In the Lambdanetwork [5] , we aim to construct a linear function $R^{|k|} \rightarrow R^{|v|}$, denoted by a matrix $\lambda_n \in \mathbb{R}^{|k|\times|v|}$. The lambda layer initially computes keys K and values V through linear projections of the context. Keys are then normalized across context positions using a softmax operation, resulting in normalized keys $\overline{K}$. The λ_n matrix is derived by aggregating the values V using the normalized keys $\overline{K}$ and position embeddings E_n, formulated as:

$$\lambda_n = (K^T \cdot V) + (E_n^T \cdot \text{ V})$$

Where:

- λ_n represents the content lambda and position lambda.
- The content lambda λ_c is shared across all query positions n and remains invariant to context permutation. It specifies how to transform the query q_n solely based on context content.
- The position lambda λ_{p_n} depends on the query position n through the position embedding E_n. It specifies how to transform the query q_n based on context elements c_m and their relative positions to the query (n, m).

Positional Embeddings Like traditional Transformer models, LambdaNetworks incorporate positional embeddings to encode the order or position of elements in the input sequence. However, unlike Transformers, which utilize these embeddings primarily for attention mechanisms, LambdaNetworks uses them as the basis for modeling interactions across distant elements in the sequence.

Pointwise Transformations LambdaNetworks applies pointwise transformations to the positional embeddings to generate context-aware representations for each element in the sequence. These transformations are applied independently to each element, allowing the model to capture long-range dependencies without the need for pairwise attention computations.

Local Interaction Window To limit the computational complexity of modelling long-range interactions, LambdaNetworks introduces a local interaction window. Instead of considering interactions between all pairs of elements in the sequence, the model focuses on a local neighbourhood around each element. This windowing strategy helps control the computational cost while still enabling the model to capture global context information.

Learnable Basis Functions LambdaNetworks use learnable basis functions to parameterize the pointwise transformations. These basis functions are shared across all elements in the sequence and are optimized during training to capture relevant interactions between positional embeddings.

Hierarchical Feature Representation LambdaNetworks are capable of learning hierarchical representations of features in the input sequence. By applying multiple layers of pointwise transformations, the model can progressively capture higher-level abstractions and dependencies in the data.

Down-Lambda layers We construct a specialized layer, combining a lambda layer with a convolutional operation. Initially, a convolutional layer with a 3x3 kernel is applied, facilitating downsampling by a factor of 2. Subsequently, a lambda layer is introduced, augmenting the feature count while concurrently diminishing the image dimensions.

Up-Lambda layers The up-lambda layer consists of a convolutional transpose layer followed by a lambda layer. The convolution transpose layer employs a 2x2 kernel with a stride of 2. This layer receives two inputs: one from the lower dimension and the other from the encoder side. Subsequently, the lambda layer is applied for further processing.

Pseudo-code for the Multi-query lambda layer Here is the pseudo-code for the Multi-query lambda layer. This lambda layer leverages tensor operations for efficient computation in deep learning models, specifically in the context of multi-query attention mechanisms.

```
def lambda_layer(queries, keys, embeddings, values):
    """Multi-query lambda layer."""
    # b: batch, n: input length, m: context length,
    # k: query/key depth, v: value depth,
    # h: number of heads, d: output dimension.

    content_lambda = einsum(softmax(keys), values, 'bmk,bmv->
                                   bkv')
    position_lambdas = einsum(embeddings, values, 'nmk,bmv->
                                   bnkv')

    content_output = einsum(queries, content_lambda, 'bhnk,
                                   bkv->bnhv')
    position_output = einsum(queries, position_lambdas, 'bhnk
                                   ,bnkv->bnhv')

    output = reshape(content_output + position_output, [b, n,
                                    d])
    return output
```

Here, The einsum operation represents generalized contractions between tensors of arbitrary dimensions. It is numerically equivalent to broadcasting its inputs to share the union of their dimensions, performing element-wise multiplication and summing across all dimensions not specified in the output.

3.2 Generator

The generator architecture is tailored to handle single-channel images sized 256x256. Initially, an input convolution layer with a 3x3 kernel and 64 channels is applied. Subsequently, four down-lambda layers are employed to gradually increase the dimensional representation of the image by reducing the matrix size. This downsampling process diminishes the matrix size to 16x16 by halving it iteratively, thereby enhancing the feature representation. Simultaneously, the number of channels or features is doubled in each down-lambda layer. This augmentation ensures richer feature extraction and better representation learning. Consequently, the generator transforms the input grayscale image into a higher-dimensional feature space, enabling it to capture intricate details and semantic information effectively. This design choice optimizes the generator's capability to generate high-quality colorized outputs while maintaining computational efficiency. Following the height representation of the feature metric, a decoder-like representation is crafted. This involves incorporating four up-lambda layers to generate an enhanced representation of the matrix with a size of 256x256. Additionally, features from the same down-lambda layers are utilized to compute the subsequent layer in the up-lambda sequence. Finally, an out convolutional layer is added, configuring the channel count to 2. Please see the detailed overview of the model in Fig 1.

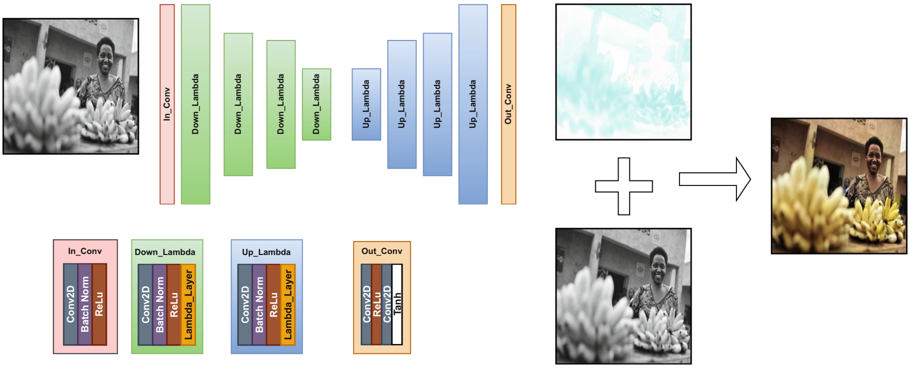


Fig. 1. Diagram of the proposed network.

3.3 Discriminator

To ensure effective local quality detection of colorized images, our colorization task employs a PatchGAN discriminator, denoted as D. This discriminator is pivotal in evaluating the quality of generated colorized images at a patch level, facilitating high-quality single-level generation. Grayscale images (L^i) are paired either with target images (T^i) or estimated images (E^i), where T^i and E^i represent the AB channels of the color image. The combination of (L^i,T^i) is labeled as real, while (L^i,E^i) is labeled as fake, thereby enforcing discrimination on image transitions rather than the images themselves.The Patch discriminator in our model processes a three-channel input dimension of 256×256. It consists of three convolution blocks, each containing 64, 128, and 256 filters, respectively, with a filter dimension of 4×4. Strides of 2 are employed for the first two convolution blocks, while a stride of 1x1 is used for the last two blocks. Batch normalization and leaky ReLU activation follow each convolution layer. Subsequently, a single filter of kernel size 4x4 is applied with a stride of 1 to compute the final response. The discriminator's output is the average of these final responses.

3.4 Losses

MAE loss The L_1 loss, also known as the mean absolute error (MAE) loss, measures the absolute differences between corresponding elements of two tensors. It is commonly used as a loss function in regression problems to penalize the magnitude of the errors between predicted and target values. The L_1 loss is calculated as follows:

$$\mathcal{L}_{L_1} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|$$

where:

- N is the number of samples or elements in the tensors.
- y_i is the true target value or ground truth.
- $\hat{y}_i$ is the predicted value.
- $|\cdot|$ denotes the absolute value.

GAN loss The GAN loss, used in Generative Adversarial Networks (GANs), is a key component in training the generator and discriminator networks. It comprises two main components: the generator loss ($\mathcal{L}_{GAN}^{G}$) and the discriminator loss ($\mathcal{L}_{GAN}^{D}$).

For the generator loss, $\mathcal{L}_{GAN}^{G}$, it is computed using binary cross-entropy loss ($\mathcal{L}_{BCE}$), which measures the difference between the discriminator's prediction on generated images and the target label (usually 1, indicating real images). The formulation is as follows:

$$\mathcal{L}_{GAN}^{G} = \mathcal{L}_{BCE}(D(L^i, G(L^i, S^i)),\ 1)$$

Similarly, for the discriminator loss, $\mathcal{L}_{GAN}^{D}$, it involves two binary cross-entropy terms. The first term computes the loss based on the discriminator's prediction on real images (T^i) compared to the real label (1), while the second term evaluates the discriminator's prediction on generated images ($G(L^i, S^i)$) against the fake label (usually 0, indicating fake images). The formulation is given by:

$$\begin{aligned}\mathcal{L}_{GAN}^{D} =& \mathcal{L}_{BCE}(D(L^i, T^i), 1) \\ &+ \mathcal{L}_{BCE}(D(L^i, G(L^i, S^i)), 0)\end{aligned} \tag{1}$$

In both equations, $\mathcal{L}_{BCE}$ represents the binary cross-entropy loss function, where y is the label and p is the predicted probability of the point.

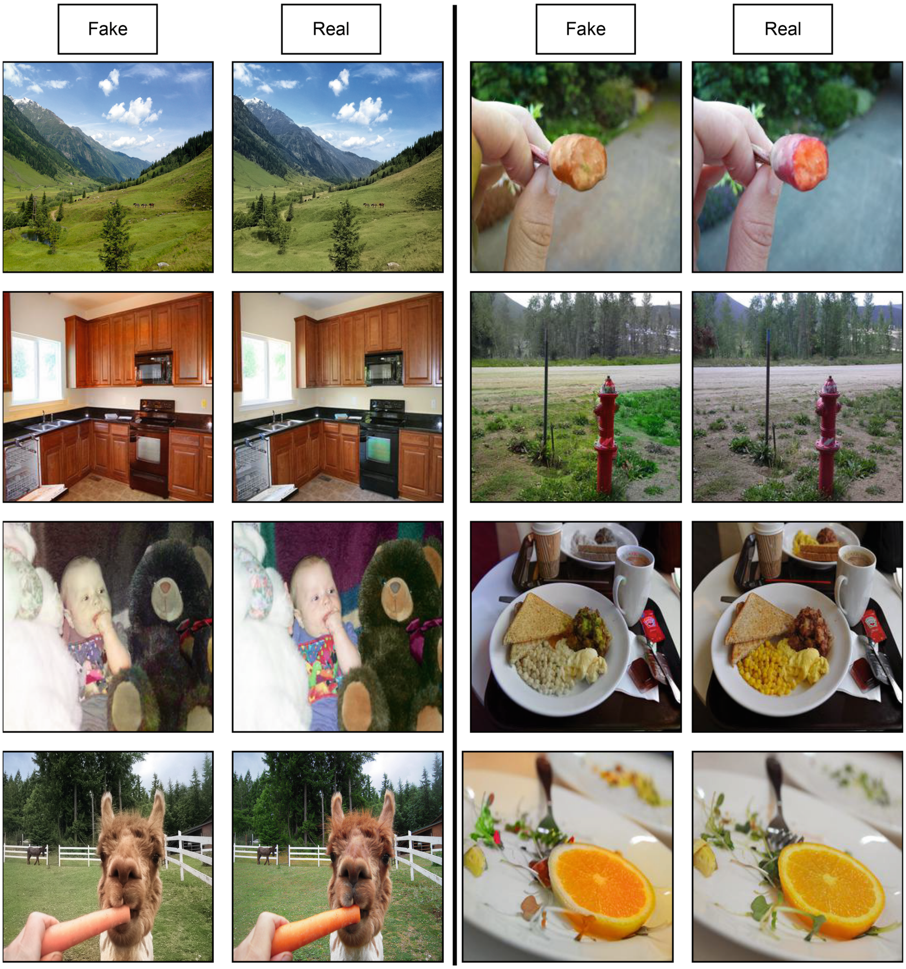


Fig. 2. Examples of some qualitative results generated from the COCO-Stuff dataset by the proposed framework. Here, "FAKE' means the Images generated by our method, and "REAL" means the Original images in the dataset. [Best visible in 300% zoom]

3.5 Training details

During training, we employed the Adam optimizer with a learning rate of 2e-4, utilizing beta1 = 0.5 and beta2 = 0.99. The training process utilized an NVIDIA GeForce RTX 3080 Ti with 12GB of memory. The model comprises 19.84 million parameters, and training was conducted efficiently with these specifications, ensuring optimal convergence and performance.

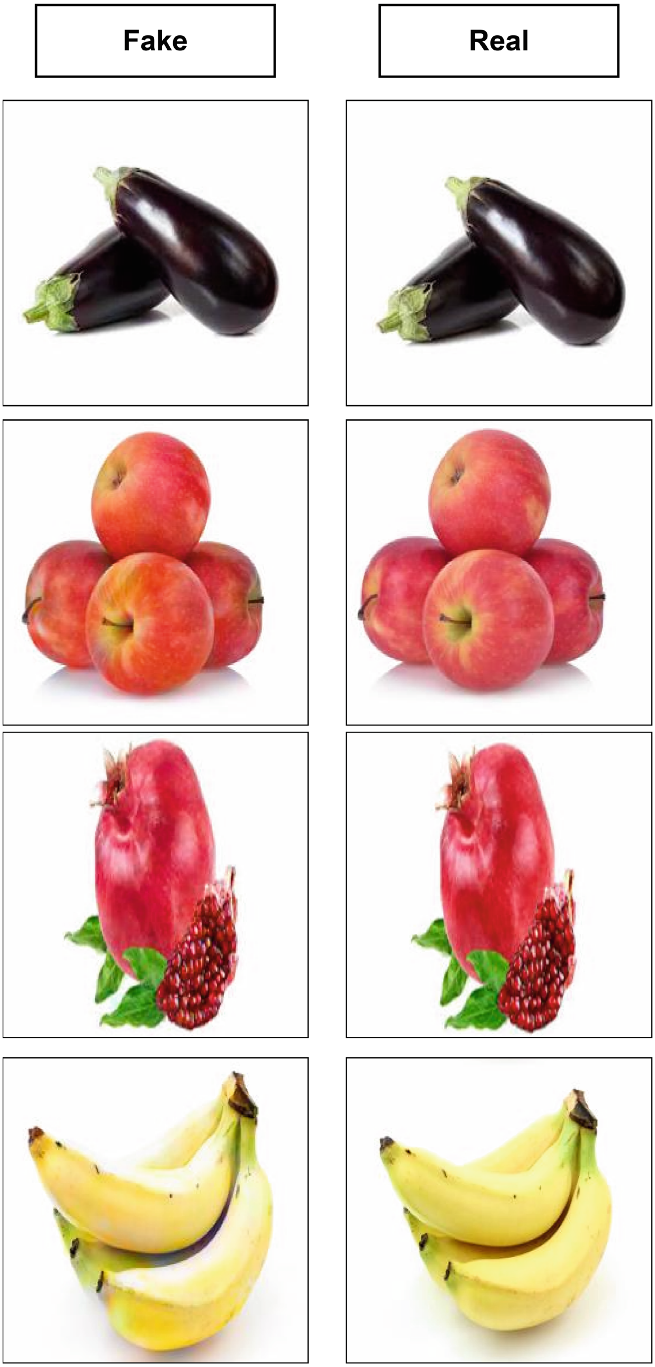


Fig. 3. Examples of some qualitative results generated from the NCD dataset by the proposed framework. Here, "FAKE' means the Images generated by our method, and "REAL" means the Original images in the dataset. [Best visible in 300% zoom]

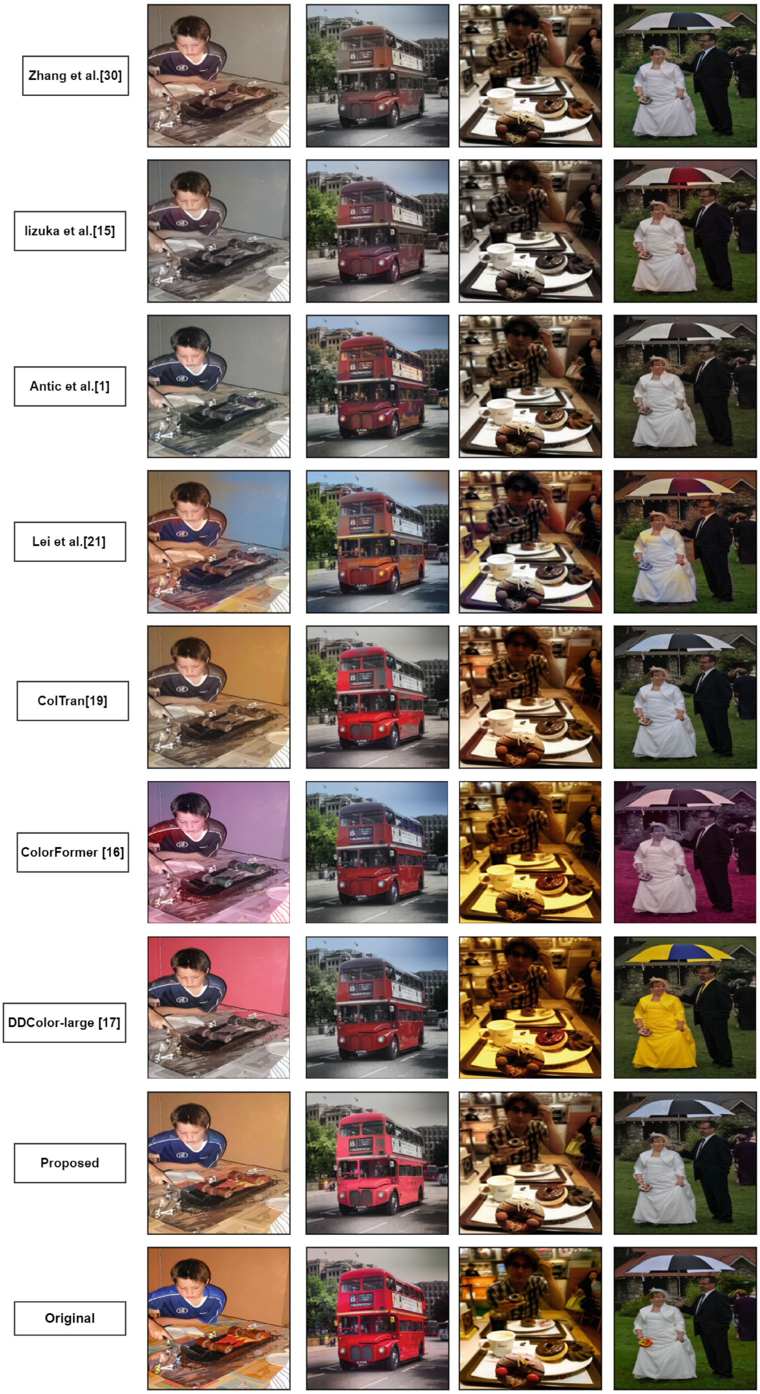


Fig. 4. Qualitative comparison results of the proposed algorithm with existing colourization algorithms in COCO-Stuff datasets. [Best visible in 300% zoom]

4 Results

4.1 Datasets

To generalize a method effectively, evaluating its performance on standard datasets is crucial. In our study, we utilized two prominent datasets for testing: the COCO dataset[6] and a natural color dataset. The COCO dataset comprises 118,000 images, serving as a benchmark for various computer vision tasks due to its diversity and scale. Additionally, we incorporated a natural color dataset [2] consisting of 723 images across 20 different categories, providing a more specific evaluation of color-related tasks. The utilization of these datasets allows for comprehensive assessment across different domains and scenarios. The COCO dataset offers a wide range of real-world scenes and objects, enabling robust evaluation under diverse conditions. Meanwhile, the natural color dataset provides insights into color-related tasks within specific categories, enhancing the method's applicability to real-world scenarios. By testing on these standard datasets, we ensure that the method's performance is validated across various contexts, facilitating its generalization and practical deployment in real-world applications.

4.2 Comparison of results

To ensure the effectiveness of our model, we conducted comparisons with state-of-the-art colorization models. While recent research has focused on diffusion models, which require additional descriptions or language to generate color images, our model offers fully automatic image colorization. Therefore, we specifically compared our results with models that do not rely on any supplementary information. Our comparisons included benchmarking against models such as Zhang et al.'s [30], Iizuka et al.'s [15], DeOldify [1], Lei et al.'s [21], Kumar et al.'s [19], ColorFormer [16], and DDColor-large [17] .Our method outperforms others across all metrics, including LPIPS[14], SSIM[28], PSNR, and FID[12]. Detailed values can be found in Table 1.,

Furthermore, we compared our visual results with the ground truth for additional insights. It's evident that our method surpasses others in terms of visual fidelity and accuracy. In the Figure 2, we observe accurate color generation, notably in the first image where the sky and forest hues are faithfully reproduced. In the subsequent row's first image, wooden furniture colors are accurately mimicked, while the fire hydrant's color is more pronounced in the second image. Similarly, in the first image of the fourth row, the camel's color is faithfully rendered, and the second image exhibits a more natural color palette. The Natural coloured Dataset results are in the Figure 3. Overall, our model demonstrates precise color reproduction across various elements, enhancing the fidelity and realism of the generated images.

Table 1. Quantitative comparison of results between the proposed algorithm and existing colorization algorithms on the COCO-Stuff datasets.

	Params.	LPIPS ↓	PSNR ↑	SSIM ↑	FID ↓
Zhang et al.[30]	32.2M	0.234	21.838	0.895	19.17
lizuka et al.[15]	25.6M	0.185	23.860	0.922	7.63
Antic et al.[1]	63.6M	0.180	23.692	0.920	3.87
Lei et al.[21]	21.6M	0.191	24.588	0.922	12.63
ColTran[19]	74.0M	0.184	23.696	0.922	6.14
ColorFormer [16]	44.8M	0.183	0.882	39.76	1.24
DDColor-large [17]	227.0M	0.190	23.74	0.927	**0.96**
Ours	9.84M	**0.180**	**24.982**	**0.929**	3.62

4.3 Ablation

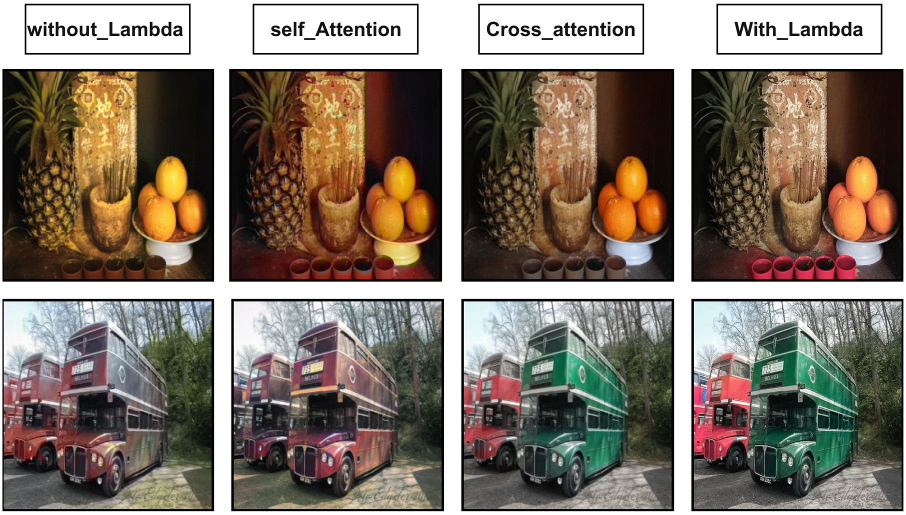


Fig. 5. Qualitative results of ablation studies on the COCO-Stuff datasets. [Best visible in 300% zoom]

In this study, we investigate the optimal integration of the Lambda network in the image colorization process through a series of ablation studies. We conduct experiments comparing different configurations: without the Lambda layer, with self-attention, with cross-attention, and with the Lambda layer. Additionally, we present qualitative results in Figure 5 to showcase the impact of these configurations on image quality. Our findings demonstrate that the inclusion of the Lambda layer consistently outperforms across all four metrics evaluated

Table 2. Quantitative ablation studies on the COCO-Stuff datasets.

	LPIPS ↓	PSNR ↑	SSIM ↑	FID ↓
Without Lamda	0.190	24.371	0.919	4.46
Self Attention	0.187	24.890	0.918	4.98
Cross Attention	0.183	24.879	0.922	4.08
With Lambda(ours)	**0.180**	**24.982**	**0.929**	**3.62**

Table 2. This suggests that the Lambda layer plays a crucial role in improving image quality in the context of image colorization tasks. We design a series of experiments to evaluate different configurations of the colorization model: Model without the Lambda layer, Model with self-attention mechanism, Model with cross-attention mechanism, and Model with the Lambda layer. Our experimental results indicate that the inclusion of the Lambda layer consistently leads to improved performance across all four metrics compared to the other configurations. Additionally, qualitative analysis of the colorized images in Figure 5 further supports this finding, demonstrating that the Lambda layer contributes to enhancing image quality in terms of color fidelity and realism.

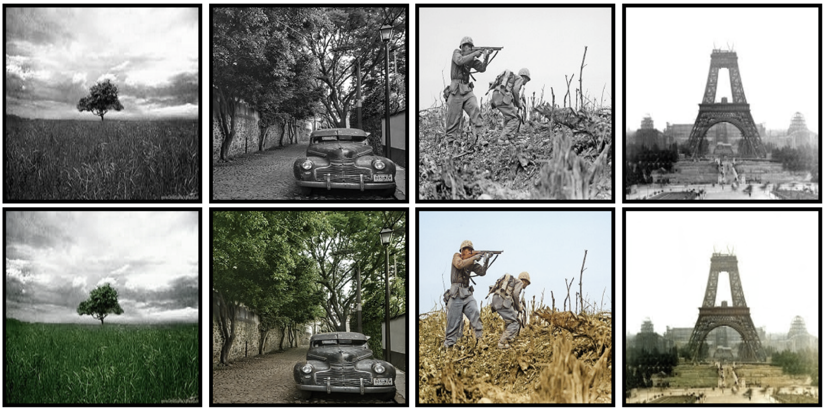


Fig. 6. Colorized old photos using the proposed method. The top Row is the original one, and the Bottom row is generated.

4.4 Result on old images

We have successfully generated color images from real historical black-and-white photos. Our method effectively colorizes such images, providing a natural and realistic color composition. We have added some colourized old photos in Fig 6 to have an idea about the results. Our method has certain limitations when it comes to the varying distribution of luminance channels. We also admit the luminance factor of the old images and it is described in the main manuscript.

4.5 User study

To evaluate the quality of our image generation, we performed a user study using 23 images((18 generated images and 5 real images) from the COCO-Stuff and NCD datasets. The observation test included a mix of generated and real photos, selected and shuffled randomly. Thirty-six people participated in the user study. Our method achieved a score of 68%, indicating that 68% of the generated images are marked wrongly, demonstrating the high realism of our generated images. Here is the link for the user study:

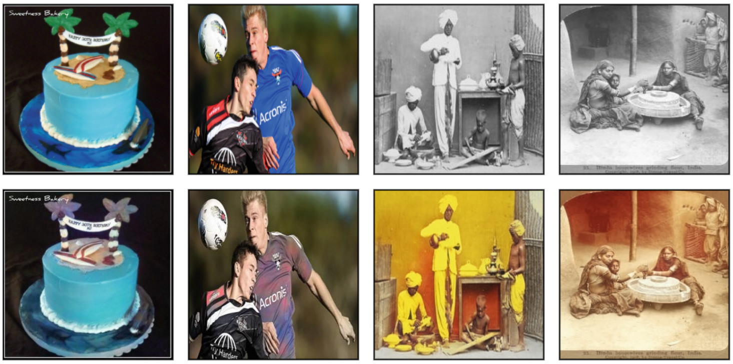


Fig. 7. Some examples of failure case. The top Row is the original one, and the Bottom row is generated.

4.6 Failure case

We performed some extensive studies on the channel distribution problem in old photos. We also find that the luminance channel mostly affects reduced contrast, noise, and blur problems. Our method performs poorly in the section, as in recent photos. We also included some failure cases caused by the different luminance channel distributions in Fig 7. We will make a more prominent effort to try this in the future.

5 Conclusions

In summary, our investigation delved into various aspects of image colorization methods. We discussed the utilization of standard datasets such as COCO and natural color datasets for evaluation. Additionally, we compared our model against state-of-the-art techniques, highlighting its superiority across metrics such as LPIPS, SSIM, PSNR, and FID. Visual comparisons against the ground truth further underscored the effectiveness of our approach, particularly in accurately reproducing colors for diverse elements. Furthermore, ablation studies emphasized the significance of incorporating the Lambda layer, showcasing its efficacy in enhancing long-range dependencies crucial for image colorization tasks. Overall, these findings underscore the robustness and efficacy of our proposed method in the realm of image colorization. Further research and development in this direction holds the potential to enhance various applications, including medical imaging, satellite imagery analysis, and artistic rendering, ushering in a new era of image processing capabilities.

References

1. Antic., J.: A deep learning based project for colorizing and restoring old images (and video!). https://github.com/jantic/deoldify, (2019)
2. Anwar, S., Tahir, M., Li, C., Mian, A., Khan, F.S., Muzaffar, A.W.: Image colorization: A survey and dataset. arXiv preprint arXiv:2008.10774 (2020)
3. Bahng, H., Yoo, S., Cho, W., Park, D.K., Wu, Z., Ma, X., Choo, J.: Coloring with words: Guiding image colorization through text-based palette generation. In: ECCV (2018)
4. Bastos, R., Wynn, W.C., Lastra, A.: Run-time glossy surface self-transfer processing (2013)
5. Bello, I.: Lambdanetworks: Modeling long-range interactions without attention. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=xTJEN-ggl1b
6. Caesar, H., Uijlings, J.R.R., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 1209–1218 (2018)
7. Carlucci, F.M., Russo, P., Caputo, B.: $(de)^2co$: Deep depth colorization. IEEE Robotics and Automation Letters (2018)
8. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. 2015 IEEE International Conference on Computer Vision (ICCV) pp. 415–423 (2015)
9. Hanyuan Liu and Jinbo Xing and Minshan Xie and Chengze Li and Tien-Tsin Wong: Improved Diffusion-based Image Colorization via Piggybacked Models. ArXiv **abs/2304.11105** (2023)
10. Subhankar Ghosh and Prasun Roy and Saumik Bhattacharya and Umapada Pal and Michael Blumenstein: TIC: text-guided image colorization using conditional generative model. Multimedia Tools and Applications (2023)
11. Güçlütürk, Y., Güçlü, U., van Lier, R., van Gerven, M.: Convolutional sketch inversion. In: ECCV Workshops (2016)
12. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Neural Information Processing Systems (2017), https://api.semanticscholar.org/CorpusID:326772
13. Huang, Y.C., Tung, Y.S., Chen, J.C., Wang, S.W., Wu, J.L.: An adaptive edge detection based colorization algorithm and its applications. In: MULTIMEDIA '05 (2005)

14. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5mb model size. arXiv preprint arXiv:1602.07360 (2016)
15. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be color! ACM Transactions on Graphics (TOG) **35**, 1–11 (2016)
16. Ji, X., Jiang, B., Luo, D., Tao, G., Chu, W., Xie, Z., Wang, C., Tai, Y.: Colorformer: Image colorization via color memory assisted hybrid-attention transformer. In: European Conference on Computer Vision (2022), https://api.semanticscholar.org/CorpusID:253120584
17. Kang, X., Yang, T., Ouyang, W., Ren, P., Li, L., Xie, X.: Ddcolor: Towards photo-realistic image colorization via dual decoders (2022), https://api.semanticscholar.org/CorpusID:254974200
18. Kim, G.Y., Kang, K., Kim, S.H., Lee, H., Kim, S., Kim, J., Baek, S.H., Cho, S.: Bigcolor: Colorization using a generative color prior for natural images. In: European Conference on Computer Vision (2022), https://api.semanticscholar.org/CorpusID:250699343
19. Kumar, M., Weissenborn, D., Kalchbrenner, N.: Colorization transformer. ArXiv **abs/2102.04432** (2021)
20. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 840–849 (2017)
21. Lei, C., Chen, Q.: Fully automatic video colorization with self-regularization and diversity. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3748–3756 (2019)
22. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: SIGGRAPH 2004 (2004)
23. Noda, H., Niimi, M.: Colorization in ycbcr color space and its application to jpeg images. Pattern Recognit. **40**, 3714–3720 (2007)
24. Perazzi, F., Pont-Tuset, J., McWilliams, B., Gool, L.V., Gross, M.H., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 724–732 (2016)
25. Su, J.W., kuo Chu, H., Huang, J.B.: Instance-aware image colorization. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 7965–7974 (2020)
26. Tola, E., Lepetit, V., Fua, P.V.: A fast local descriptor for dense matching. 2008 IEEE Conference on Computer Vision and Pattern Recognition pp. 1–8 (2008)
27. Wang, P., Patel, V.M.: Generating high quality visible images from sar images using cnns. 2018 IEEE Radar Conference (RadarConf18) pp. 0570–0575 (2018)
28. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing (TIP) (2004)
29. Weng, S., Sun, J., Li, Y., Li, S., Shi, B.: Ct2: Colorization transformer via color tokens. In: European Conference on Computer Vision (2022), https://api.semanticscholar.org/CorpusID:253512662
30. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016)

Memory Matching is Not Enough: Jointly Improving Memory Matching and Decoding for Video Object Segmentation

Jintu Zheng[1], Yun Liang[2](✉), Yuqing Zhang[3], and Wanchao Su[4]

[1] University of Chinese Academy of Sciences, Beijing, China
zhengjintu22@mails.ucas.ac.cn

[2] South China Agricultural University, Guangzhou, Guangdong, China
yliang@scau.edu.cn

[3] Beijing University of Technology, Beijing, China
yuqingz@emails.bjut.edu.cn

[4] Department of Human Centered Computing, Faculty of Information Technology, Monash University, Melbourne, Australia
wanchao.su@monash.edu

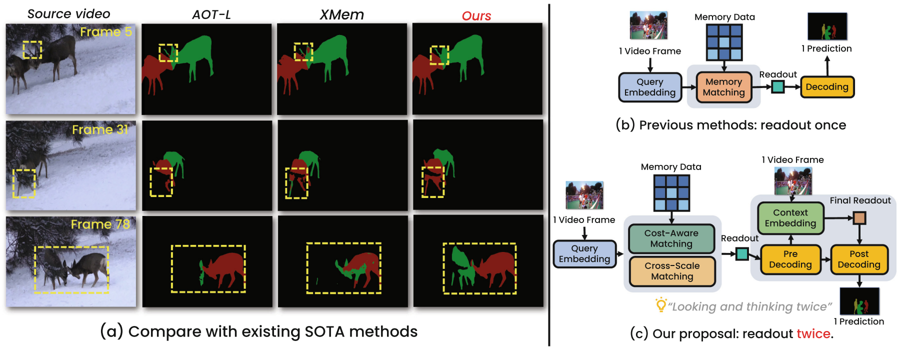
Abstract. Memory-based video object segmentation methods model multiple objects over long temporal-spatial spans by establishing memory bank, which achieve the remarkable performance. However, they struggle to overcome the false matching and are prone to lose critical information, resulting in confusion among different objects. In this paper, we propose an effective approach which jointly improving the matching and decoding stages to alleviate the false matching issue. For the memory matching stage, we present a cost aware mechanism that suppresses the slight errors for short-term memory and a shunted cross-scale matching for long-term memory which establish a wide filed matching spaces for various object scales. For the readout decoding stage, we implement a compensatory mechanism aims at recovering the essential information where missing at the matching stage. Our approach achieves the outstanding performance in several popular benchmarks (i.e., DAVIS 2016&2017 Val (92.4%&88.1%), and DAVIS 2017 Test (83.9%)), and achieves 84.8%&84.6% on YouTubeVOS 2018&2019 Val.

Keywords: Video Object Segmentation · False Matching Alleviation · Compensatory Decoding

1 Introduction

Video object segmentation (VOS) is a fundamental procedure for many multimedia applications, such as special effects editing in movies, robot interaction, and smart camera surveillance, which requires instance segmentation of objects of interest in videos. The work in this paper focuses on semi-supervised VOS, which completes instance segmentation of the remaining frames based on multiple instances given in the first frame. Recently, matching-based approaches have

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 188–203, 2025.
https://doi.org/10.1007/978-3-031-78312-8_13



(a) Compare with existing SOTA methods

(b) Previous methods: readout once

(c) Our proposal: readout twice.

Fig. 1. (a) Comparisons of a representative video clip in DAVIS 2017 Test. AOT [27] and XMem [1] (two state-of-the-art matching-based VOS models) present false matching errors and our method can produce more accurate masks. (b-c) A simplified comparison on pipeline between ours and previous matching-based methods. Previous matching-based lacks the consideration of combining two stages to improve.

gained popularity, wherein the basic idea is to establishing and maintaining a memory to store previous frames and their corresponding masks. These stored memories are then matched with the query frame to generate a memory readout, and finally produce masks of the target objects from the memory readout.

Memory matching essentially relates to the accuracy in generating the target object masks, which becomes a crucial component in improving the accuracy of VOS tasks. Early matching-based methods such as STM [13] and its variants [2,16,18] employ attention mechanisms to achieve matching between query frames and the memory. Such methods treat all the memory units with equal importance, without special design regarding the individual memory unit in the process. Inspired by the human cognitive process where there is a distinctive difference between short-term and long-term memories, current memory matching methods treat short-term and long-term memory differently in VOS. Long-term memory stores multiple historical frames, and records the change across frames in a coarse-grained manner. The objects in long-term memory may have different scales, so the long-term matching mechanism should not be limited to a single-scale. Short-term memory focuses on the adjacent frames, which are similar with each other, meaning the variations are fine-grained. Thus, the short-term matching mechanism must capture the variants sufficiently.

Some state-of-the-art methods (i.e., AOT [27], variants of AOT [25,29], and XMem [1]) divide long-term and short-term memory, these methods still have limitations: As shown in Fig. 1(a), the AOT [27] and XMem [1] produce results with slight errors in the impact of the short-term memory insufficiency (Frame 5 in Fig. 1(a)), the later frames (Frame 31 & 78 in Fig. 1(a)) present object confusion and crucial information loss. On the one hand, they employ single-scale attention in long-term memory, which makes them exhibit rapid performance degradation in handling multiple objects, especially undergoing different morphological changes simultaneously. On the other hand, these methods also need to improve in matching short-term memory. For example, AOT [27] and its variants [25,29] implement the local correlation attention, which may lose critical information when the morphological changes occur outside the local memory unit's perceptive field. In addition, slight errors in short-term memory matching accumulate in long-term memory, which can be fatal for VOS methods that rely on long-term memory.

In this paper, we propose an improved memory matching mechanism, including cost-aware matching for short-term memory and cross-scale matching for long-term memory. Cost-aware matching mechanism focuses on a stronger relationship between corresponding pixels of adjacent frames. Inspired by optical flow prediction methods [7,19,20], we construct the cost volume for the query frame and the previous frame. Cost volume is a vector that stores the matching degree of corresponding pixels between two frames [4]. After patch embedding the cost volume, we introduce a group of learnable query tokens for collecting coarse-grained spatial variations. Furthermore, to explore fine-grained details, we construct a spatial readout head via SS-attention [3]. Note that cost volume in our cost-aware matching is a global relationship of pixels in adjacent frames which is different from the neighborhood correlation of short-term transformer in AOT [27] and its variants [25,29]. We implement multiple scales for long-term memory and more effective models objects of various scales simultaneously within a matching block. Our improved memory mechanism reduces the accumulation of slight errors in short-term memory and adequately adapts the objects of various spatial morphology in the previous frames.

Matching-based VOS methods inevitably produce false matches [1,2,13], which may lead to object confusion (see XMem in Fig. 1(a)) or missing objects (see AOT in Fig. 1(a)). However, matching-based methods usually focus solely on improving memory matching, and they implement a naive FPN [8] for decoding (e.g., STCN [2], HMMN [18] and RMNet [23]), lacking consideration of modifying the decoding process. AOT [27] and XMem [1] make extensive modifications for memory matching, the problem of false matches still exists. Unlike fully supervised segmentation that understands rich semantics, semi-supervised VOS requires more the low-level semantic feature prompt of target objects. We argue that suppressing false matches requires improving memory matching and improving decoding process. There is significant potential for improving the decoding process. For instance, AOML [5] achieves excellent performance by designing bi-decoders for online learning VOS. CFBI [26] and CFBI+ [28] emphasizes separating the foreground and background in decodng process to improve matching. They have an explicit foreground-background embedding feature and low-level feature incorporation, which are beneficial for distinguishing the foreground from the background. Such explicit foreground-background distinction still struggle to overcome the false matching problem. Our paper aims to give a more suitable and comprehensive answer, which jointly improves both stages and rethinks all details toward reducing the false matching instead of the simple foreground-background distinction.

Therefore, we propose a compensatory decoding mechanism (as shown in Fig. 1(c)), which consists of three steps, 1) pre-decoding, 2) context embedding, and 3) post-decoding. The initial memory readout inevitably lose some critical information and looking twice at the original image effectively compensates such losses; thus, we embed a context embedding process in the decoding stage to force the encoder to look at the query frame one more time, which supplements the critical information lost in the memory matching stage. Pre-decoding provides a guiding prompt for context embedding, and the post-decoding generates the final segmentation masks. The compensatory decoding mechanism not only sufficiently embeds the critical information of the target objects but also suppresses the false matches in the initial memory readout to some extent.

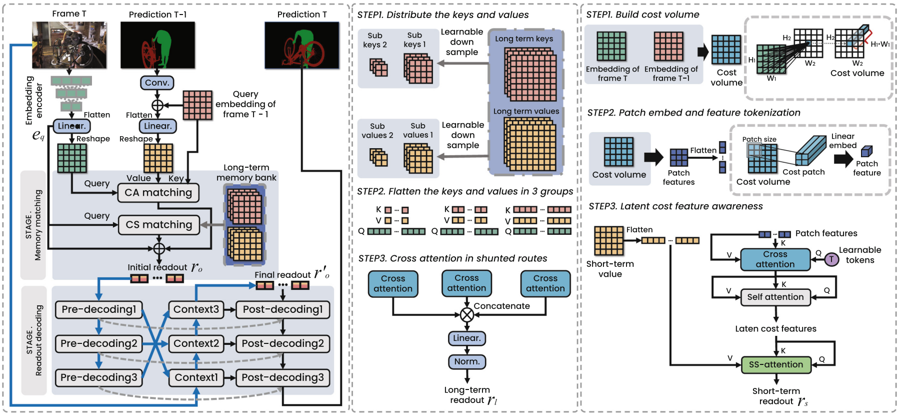
The mainly contributions of this paper are summarized as:

- Different existing methods, we improve the matching and decoding stages in a jointly paradigm, which give a more suitable and comprehensive answer to alleviate false matching issue.
- We propose a improved mechanisms for the memory matching stage. Cost-aware matching in short-term memory prompts the network to perceive changes between two frames more adequately. Cross-scale matching in long-term memory prompts the network to explore the variations in different scaled objects.
- We propose a novel compensatory decoding mechanism that can suppress false matches and supplement the crucial information loss of target objects for the readout decoding stage.
- Our approach achieves state-of-the-art performance in several popular benchmarks (i.e., DAVIS 2016&2017 Val (92.4%&88.1%), and DAVIS 2017 Test (83.9%)), and achieves 84.8%&84.6% on YouTubeVOS 2018&2019 Val with the specific training strategy.

2 Related Work

Semi-supervised Video Object Segmentation In semi-supervised VOS, according to the object masks given in the first frame of the video, the corresponding object is segmented in the remaining video frames. The mainstream of semi-supervised VOS methods can be roughly divided into online fine-tuning and methods without fine-tuning. In the methods without fine-tuning, matching based is the popular study branch.

Matching based Methods Matching-based is an VOS method without fine-tuning that has achieved notable success, and the proposal in this paper is focus on the matching based methods. STM [13] introduces a spatio-temporal network to establish the matching relationship between the current frame (query frame) and all historical frames (memory), which can be roughly divided into three stages: query embedding, memory matching, and readout decoding. The following matching-based methods almost focus on improving the memory matching stage. Some methods [6,9,17] focus on designing novel memory structures, such as HMMN [18], which designs a hierarchical memory structure. In addition, some methods [2,16,23] proposed novel matching mechanisms, such as STCN [2] proposed utilizing negative squared euclidean distance to calculate the affinity of matching, and KMN [16] introduced gaussian kernel to reduce the non-locality of matching. All the above methods treat all memories fairly, which makes the performance of the memory model have a bottleneck. Recently, some state-of-the-art methods such as AOT [27] and its variants (AOST [25], DeAOT [29]), XMem [1] distinguish memory between long and short term to improve the inference performance. However, there are still significant improvements possibilities for these methods. On the one hand, the design of this long-short term matching mechanism does not fully adapt to multi-scale objects and ignores part of the crucial inter-frame changes. In this paper, we propose an improved long short-term memory matching mechanism that outperforms these methods in performance. On the other hand, we argue that *"memory matching is not enough"*, and previous methods have neglected the effect of improvements on the readout decoding for semi-supervised VOS.



(a) Pipeline of JIMD (b) Cross-scale matching (c) Cost aware matching

Fig. 2. (a) Pipeline of our proposal, which improves the memory matching stage by cost-aware and cross-scale matching, and improves the decoding stage by compensatory decoding. (b-c) Illustration of cross-scale and cost-aware matching.

3 Methodology

3.1 Proposal Overview

We propose an effective approach to alleviate the false matching issue from the memory matching and readout decoding stages. We name this proposal as

Jointly **I**mprove **M**atching and **D**ecoding (JIMD) in the following content. The pipeline of JIMD is illustrated in 2. JIMD sequentially processes each frame for a video clip. For the current frame T, we extract the backbone features from the embedding encoder and then input into a linear layer to obtain the embedding query feature e_q. Then we implement the cost volume matching for short-term memory, and employ the cross scale matching for the long-term memory to obtain two matching readout results (i.e., r_s and r_l). Initial readout feature r_o can be formulated as: $r_o = r_l + r_s + e_q$. We firstly decode the r_o for guiding the context block to extract the information form source frame. We obtain the final readout feature $r_o^{'}$ after embedding the source context into the initial readout r_o. Finally, we decode the readout feature $r_o^{'}$ and upsample into the object masks. During the entire procedure, JIMD maintains two sets of memory data, which are long-term and short-term memories. The short-term memory is stacked into the long-term memory at intervals, and the memory data is stored as *key* & *value*. Memory value is generated by fusing the previous embedding feature and the mask feature from the ID module, which is borrowed from AOT [27]. Here, we implement a convolution with kernel size of 17×17 and stride of 16 to encode the masks of frame $T - 1$ as the identify encoder.

3.2 Cross-Scale in Long-Term Matching

Long-term memory records the change across frames in a coarse-grained manner (e.g., an object may have multiple morphologies across frames or multiple scale objects in the same frame). Introducing cross-scale in long-term memory is beneficial to match targets with variable scales. Therefore, we shunt the keys and values, downsampling the long-term memory keys and values at different scales. Let the $K_{\{0,1,2,\ldots,T-1\}}$ and $V_{\{0,1,2,\ldots,T-1\}}$ denote all previous keys and values in long-term memory. As shown in 2, we employ three spatial rates for non-local matching. The downsample is a convolution layer with the decreasing kernel size as illustrated in 2. Here, we denote d_i as the spatial rate for downsampling:

$$K^{di}_{\{0,1,2,\ldots,T-1\}} = downsample(K_{\{0,1,2,\ldots,T-1\}}, d_i) \tag{1}$$

$$V^{di}_{\{0,1,2,\ldots,T-1\}} = downsample(V_{\{0,1,2,\ldots,T-1\}}, d_i) \tag{2}$$

We perform cross-attention with e_q after obtaining keys and values of different sizes:

$$\sigma_i = Atten(e_q, K^{d_i}_{\{0,1,2,\ldots,T-1\}}, V^{d_i}_{\{0,1,2,\ldots,T-1\}}). \tag{3}$$

Three attention results are concatenated, then projected as the readout feature r_l of long-term memory:

$$r_l = LN(concat(\sigma_0, \sigma_1, \sigma_2)), \tag{4}$$

where LN is the linear norm layer. This shunting matching benefits the long-term memory and performs well in handling multi-scale cases.

3.3 Cost-Aware Matching

The purpose of the short-term memory matching mechanism is to learn the changes in adjacent frames, which is crucial for VOS. If the short-term memory matching produces inaccurate masks that are used in generating values for the next round, it would lead to a vicious circle process. Representing the changes in adjacent frames is essential for effective short-term memory matching, it requires more larger local receptive field but with a low computational cost increasement. We construct a cost volume to represent the interframe variations and tokenize the cost volume with patch embedding. Then we implement the multi-head attention mechanism to produce the latent cost features that involve critical variations. Finally, the SS-attention [3] generates the readout feature with the latent cost features and the previous frame's value. We construct cost-aware matching as a transformer architecture, as shown in 2.

Cost Volume Generation. The first step of short-term memory matching requires the generation of initial information representing the variations between adjacent frames. Instead of the local correlation matching in AOT [27] and its variants [25,29] with high computational cost, we build a global 4D cost volume $c_m \in \mathbb{R}^{H_1 \times W_1 \times H_2 \times W_2}$ for e_q and e_k, the volume is constructed through dot product operations. Here, H_1 and W_1 donate the e_k's height and width, H_2 and W_2 donate the e_q's height and width. As shown in the bottom right corner of 2, each 2D sub-map in cost volume c_m can be regarded as the visual similarity between the source pixel and all target pixels, and e_k is derived from the e_q of the previous frame. We patch embed the cost volume, which divides the cost volume into multiple patches according to the patch size, and perform feature embedding on each patch, as shown in the upper right corner of 2. We define the number of the patches as S, embedding dimension as C', the patch embedding features $P_s \in \mathbb{R}^{(H_1 * W_1) \times S \times C'}$ as:

$$P_s = PatchEmbed(e_q \cdot e_k). \tag{5}$$

Patch Features Tokenization. For complicated variations between adjacent frames, we need to pay more attention to the patches relating to the target objects. To achieve this goal, we introduce a set of learnable tokens $L_q \in \mathbb{R}^{N_l \times C_l}$ to extract the patch features $P'_s \in \mathbb{R}^{(H_1 * W_1) \times N_l \times C_l}$ that contains latent critical variations, where N_l is the token number and C_l is the tokens' embedding dimension. We implement a cross-attention, which applies the learnable tokens L_q as query, patch embedding features P_s as the key and value:

$$P'_s = Atten(L_q, P_s, P_s). \tag{6}$$

Then we apply a multi-head self-attention to output the latent cost features $C_l \in \mathbb{R}^{(H_1 * W_1) \times N_l \times C_l}$. The latent cost features C_l implicitly represent critical variants between adjacent frames.

Producing Readout Feature. We implement the SS-attention [3], which focus more on capturing spatial features to produce short-term readout result r_s:

$$r_s = SSAtten(C_l, C_l, V_{T-1}) + C_l, \tag{7}$$

where V_{T-1} is the frame $T-1$ value, and there is a final fusion for attention output and C_l. After the cost-aware matching, the short-term readout feature r_s sufficiently capture the fine-grained variations between adjacent frames.

3.4 Compensatory Decoding

We observe that only improving the memory matching is insufficient in solving the object confusion and missing; thus, we propose a new compensatory decoding mechanism to further resolve the issues. As shown in 3, the compensatory decoding process consists of three steps, 1) pre-decoding, 2) context embedding, and 3) post-decoding. Pre-decoding aims to obtain a set of upsampled intermediate results as the guide spatial prompt in context embedding. Context embedding gradually recovers the lost critical features in the initial memory readout r_o by embedding the frame T and spatial prompt feature, resulting a final readout feature $r_o^{'}$. In post-decoding, we generate the mask prediction base on the final readout feature $r_o^{'}$ and the skip-connections from context embedding.

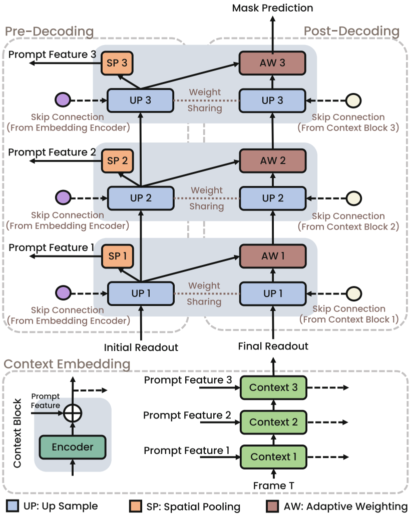


Fig. 3. Illustration of compensatory decoding which compensates the low-level information for the initial readout.

Context Embedding. Context block (CB) applies the same residual network layer as the encoder and three context blocks form a cascaded structure. Prompt features g_i and the residual encoder output $Res(f_{i-1})$ fuse as f_i in each cascade:

$$f_i = g_i + Res(f_{i-1}), \tag{8}$$

where Res is the residual layer. f_i is the input to the next CB as well as the skip connection for post-decoding, which contains richer semantic information than the query feature e_q from embedding encoder. Critical information of the target objects is sufficiently embedded into the final readout feature $r_o^{'}$, and the false matches in the initial memory readout are suppressed.

Recursive Decoding Process. Inspired by [15], we introduce the *"looking and thinking twice"* idea into memory readout decoding, namely recursive decoding, which consists of pre-decoding and post-decoding. Recursive decoding process shares weights in upsample blocks (**UP**, as shown in 3), which is essential for improving readout decoding due to seeing both the pre-decoded features and the context compensated features. Let D_i^b and D_i^p denote pre-decoding and post-decoding outputs, i is the cascading level. The implementation of **UP** can be formally defined as follows:

$$D_i^p = upsample(D_{i-1}^p, u) + f_i, \tag{9}$$

where u is the upsample rate, here u is equal to 2. Spatial pooling block (**SP**, as shown in 3) produce the spatial prompt feature in pre-decoding step, which is implemented via atrous spatial pyramid pooling block (ASPP). In post-decoding step, we implement an adaptive weighting block (**AW**, as shown in 3) in each cascaded **UP** block to fuse context features and pre-decoding features. We apply a convolution with a kernel size equal to 1 as the **AW** block. The adaptive weighting process formulates as:

$$w = sigmoid(\mathbf{AW}(D_i^p)), \tag{10}$$

$$D_i = w * D_i^p + D_i^b * (1 - w), \tag{11}$$

where w is the weight of the post-decoding output. Then D_i is the input of the next cascaded **UP** block.

Table 1. Comparison with state-of-the-art methods on DAVIS (i.e., DAVIS 2017 test-dev set, DAVIS 2017 validation set and DAVIS 2016 validation set). †: Specific strategy without BL30K pre-training. AOT-L [27] and its variants [25,29] use the ResNet50 as backbone. The best score in each column is on red bold-faced.

	DAVIS 2017 test-dev			**DAVIS 2017 val**				**DAVIS 2016 val**		
Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	FPS	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
(CVPR′2018) **RGMP** [12]	52.8	51.3	54.4	66.7	64.8	68.6	-	68.8	68.6	68.9
(CVPR′2019) **FEELVOS** [21]	57.8	55.1	60.4	71.6	69.1	74.0	2.0	81.7	81.1	82.2
(ICCV′2019) **STM** [13]	72.2	69.3	75.2	81.7	79.2	84.3	-	89.4	88.7	90.1
(ECCV′2020) **CFBI** [26]	74.8	71.1	78.5	81.9	79.1	84.6	5.9	89.4	88.3	90.5
(CVPR′2021) **RMNet** [23]	75.0	71.9	78.1	83.5	81.0	86.0	-	88.8	88.9	88.7
(ECCV′2020) **KMN** [16]	77.2	74.1	80.3	82.8	80.0	85.6	4.2	90.5	89.5	91.5
CFBI+ [28]	78.0	74.4	81.6	82.9	80.1	85.7	5.6	89.9	88.7	91.1
(ICCV′2021) **HMMN** [18]	78.6	74.7	82.5	84.7	81.9	87.5	10.0	90.8	89.6	92.0
(NeurIPS′2021) **AOT-L** [27]	79.6	75.9	83.3	84.9	82.3	87.5	18.0	91.1	90.1	92.1
(NeurIPS′2021) **STCN** [2]	79.9	76.3	83.5	85.3	82.0	88.6	20.2	91.7	90.4	93.0
AOST-L [25]	79.9	76.2	83.6	85.6	82.6	88.5	17.5	92.1	90.6	93.6
(NeurIPS′2022) **DeAOT-L** [29]	80.7	76.9	84.5	85.2	82.2	88.2	19.8	92.3	90.5	94.0
(ECCV′2022) **XMem** †[1]	81.0	77.4	84.5	86.2	82.9	89.5	**22.6**	91.5	90.4	92.7
(CVPR′2023) **ISVOS** †[22]	82.8	79.3	86.2	87.1	83.7	90.5	-	**92.6**	**91.5**	93.7
JIMD (ours)	**83.9**	**80.3**	**87.4**	**88.1**	**85.2**	**91.0**	13.2	92.4	90.6	**94.2**

4 Experiments

4.1 Implementation Details

Training. We deploy the ResNet50 as the backbone for embedding encoder. Following popular matching-based VOS methods [2,13,26], we employ the two-stage training strategy. In the first training stage, the model is pre-trained on static datasets from AOT [27], where objects with masks are augmented (e.g., flip, shift, crop) and randomly synthesized onto the backgrounds. In the second training stage, we perform training on real videos. For the evaluation of DAVIS [14], we use the training sets of DAVIS and YouTube, while for the evaluation of YouTubeVOS [24], we only use the YouTube training set. The embedding encoder is frozen in the second training stage to avoid overfitting to the seen object categories. All modules apply the learning rates from initial 2e-4 then the learning rates gradually decay to 2e-5 in a polynomial manner. We employ the cross entropy loss and soft jaccard loss [11] to train the model. During training, we use an input size of 384 × 384 and a batch size of 8, which is distributed on 4 RTX3090 GPUs. Note that our method **do not** adopt the BL30K for training in the following reports.

Inference. Our method uses a resolution of 480p by inference. Following common matching-based methods, we set the update frequency of memory to 5 (i.e., every five frames stack the short-term memory in long-term memory). For a fair comparison, we **do not** employ the multi-scale inference trick on val/test datasets.

4.2 Datasets and Metrics

We evaluate our method on the five most popular VOS task benchmark datasets, consisting of a single-object dataset (DAVIS2016-val) and four multi-object datasets (DAVIS2017-val, DAVIS2017-test, YouTube2018-val, YouTube2019-val). A total of 971 real videos incorporated in the evaluation. We evaluate our method by region similarity (i.e., J) and contour accuracy (i.e., F). In YouTubeVOS, there are additional 26 unseen categories; thus we separately report the J score and the F score for *"seen classes"* in training set and *"unseen classes"* that are not. G is the global average score of all metrics. We submit the val/test results on official online evaluation servers for a fair comparison.

4.3 Compare with the State-of-the-Art Methods

Quantitative Comparison. As shown in Table 1, we compare the performance of our method with up-to-date methods on a series of DAVIS datasets. Compared with the state-of-the-art memory matching-based methods (i.e., XMem [1]), we achieve a 2.9% and 1.9% J&F improvement on DAVIS2017 Test and DAVIS2017 Val, respectively. Compared with the latest understanding-based method (i.e., ISVOS [22]), we improve the J&F by 1.1% and 1% on DAVIS2017 Test&Val, respectively. Our method achieves the top ranking of F performance on DAVIS 2016 single object performance, DAVIS2017 Test&Val multiple objects performance. On the Youtube dataset validation, for a fair comparison, we only adopt YouTubeVOS's train-set for training and compare it with XMem [1] schemes using the same training data. As shown in Table 2, even without any fancy training and inference tricks, our method still achieves excellent performances in YouTube2018&2019.

Unseen Categories. Youtube dataset needs to evaluate the performance of unseen categories in training, as shown in Table 2; our method outperforms

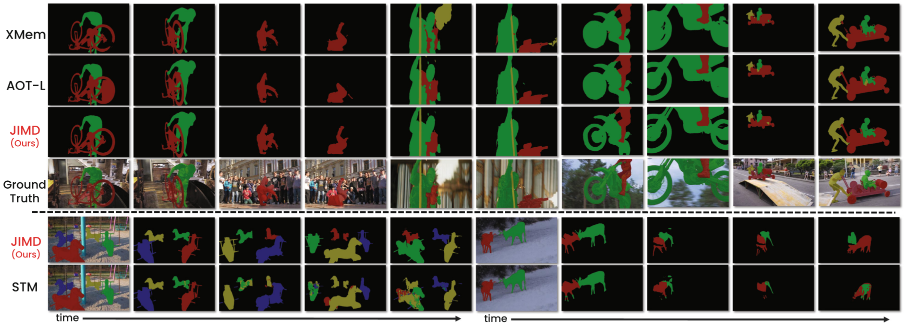


Fig. 4. Representative challenge cases of qualitative comparison with XMem [1], AOT [27], and STM [13].

other methods under the metrics of unseen categories. Our proposed compensatory decoding stage generates guide information in pre-decoding that can more discriminatively help segment unseen category objects.

Qualitative Results. We visualize some video segmentation results in evaluation, including some common VOS challenge cases (i.e., tremendous motion, object reappearance, similar objects confusion), and compare them with two state-of-the-art matching-based methods AOT-L [27] and XMem [1], as shown in Fig. 4. Our method is superior to the other two methods in obtaining object details. We can see that AOT-L [27] and XMem [1] make obvious errors when the human body appears in drastic motions. XMem [1] is more prone to errors when dealing with objects of different scales (e.g., the rope and the human body in the Col 5 and 6 of Fig. 4). Furthermore, we compare the classical method STM [13] based on memory matching for a long span. Our method does not have object confusion after multiple similar objects are occluded, indicating that the proposed jointly improving method is helpful in overcoming the challenge of similar objects.

4.4 Ablation Studies

We improve the memory matching stage and the decoding stage separately, achieving cost-aware matching (CA) and cross-scale matching (CS) in the memory matching stage and compensatory decoding (CD) in the decoding stage. In order to evaluate the effectiveness of the three improvements, we conduct separate experiments on JIMD for each modification and explore the improvement performance of their combination. All ablation studies are evaluated on DAVIS2017 Val split. The corresponding modules used by the baseline in Table 3 (i.e., long short-term matching modules, decoding process) are replaced by AOST-L [25] (a evolution method of AOT-L [27]), as shown in the first row of Table 3.

Impact of Memory Matching Mechanism. As shown from Row 2 to Row 4 in Table 3, both our cost-aware and cross-scale in the memory matching stage

Table 2. Results of YouTube2018&2019 validation. ⋆: Stage2 only training by YouTube dataset (without extra data).

Methods	YTB2018				
	$\mathcal{G}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$
CFBI	81.4	81.1	85.8	75.3	83.4
CFBI+	82.8	81.8	86.6	77.1	85.6
AOT-L⋆	84.1	**83.7**	88.5	78.1	86.1
STCN	84.3	83.2	87.9	79.0	87.3
XMem⋆	84.4	**83.7**	88.5	78.2	87.2
JIMD (ours)⋆	**84.8**	**83.7**	**88.7**	**79.1**	**87.6**
Methods	YTB2019				
	$\mathcal{G}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$
CFBI	81.0	80.6	85.1	75.2	83.0
CFBI+	82.6	81.7	86.2	77.1	85.2
AOT-L⋆	84.1	83.5	**88.1**	78.4	86.3
STCN	84.2	82.6	87.0	79.4	87.7
XMem⋆	84.3	**83.6**	88.0	78.5	87.1
JIMD (ours)⋆	**84.6**	82.9	87.8	**79.7**	**87.9**

Table 3. Ablation performance on DAVIS 2017 Val of proposed improvements. CA: cost-aware matching. CS: cross-scale matching. CD: compensatory decoding.

CD	CS	CA	$\mathcal{J}$ & $\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	FPS
✗	✗	✗	85.6	82.6	88.5	17.5
✗	✓	✗	86.1	83.5	88.6	15.7
✗	✗	✓	87.3	84.3	90.3	**17.8**
✗	✓	✓	87.7	84.8	90.5	14.3
✓	✗	✗	87.6	84.9	90.2	17.1
✓	✓	✗	87.7	85.0	90.3	13.6
✓	✗	✓	87.9	85.0	90.8	15.0
✓	✓	✓	**88.1**	**85.2**	**91.0**	13.2

Table 4. Internal functional ablation for compensatory decoding.

Method	$\mathcal{J}$ & $\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
CD-Basline	87.6	84.9	90.2
CD w/o Context	86.4(↓**1.2**)	83.5 (↓**1.4**)	89.3 (↓**0.9**)
CD w/o AW	87.4	84.6	90.1
CD w/o SP	87.1	84.6	89.6

Table 5. Compensatory decoding migrating to existing matching-based VOS methods.

Method	$\mathcal{J}$ & $\mathcal{F}$	
	DAVIS 2017	DAVIS 2016
STM [13]	81.7	89.4
STM w/ CD	83.1 (↑ **1.4**)	90.0 (↑ **1.2**)
TransVOS [10]	83.9	90.5
TransVOS w/ CD	85.1 (↑ **1.2**)	90.7 (↑ **0.2**)

play positive roles when compensatory decoding is removed. The improvement of cross-scale matching is more conducive to improving regional similarity (i.e., J), and cost-aware matching improves both metrics and edge accuracy (i.e., F). Cross-scale matching has more powerful matching for objects of different scales in the video and thus is beneficial to improve region similarity. Cost-aware constructs the cost volume for learning, explores the changes between two frames, and therefore is more effective for preserving object details and edge features. Cost-aware matching also improves the inference speed (FPS) in Col 7 of Table 3, which abandons the calculation of neighbourhood cross-correlation in short-term memory and constructs the pixel relationship by dot product that is more efficient. We can observe that the combined improvement strategy of cost-aware and cross-scale improves J by 2.6% and F by 2% compared to the baseline.

Impact of Decoding Mechanism. Compensatory decoding provides an essential information supplement for matching readout results and suppresses false matching to a certain extent by implementing context embedding compensation in the decoding stage. Row 5 of Table 3 demonstrates the improvement of compensatory decoding over the baseline. To study the binding function effect in compensatory decoding, as shown in Table 4, we remove three implements or modules (i.e., context compensation (Context), adaptive weighting (AW), and spatial block (SP)) and evaluate the gains separately. CD-Baseline in Table 4 for the experimental setup in the 5th row of Table 3. On the context removal setting, we simultaneously remove the share-weighted recursive decoder and replace it with two decoders in the cascade that do not share weights. As shown in Table 4, after removing context comparison, we can observe the most significant drop in performance (i.e., 1.4% drop in J and 0.9% drop in F). Therefore, context compensation is vital in improving the decoding stage. Furthermore, we migrate our decoding improved mechanism to other existing matching-based methods, as shown in Table 5, which indicates the feasibility and the plug-and-play potential of our compensatory decoding.

Impact of Matching and Decoding Improved Jointly. The contribution of this paper is to explore the role of joint improvement of memory matching and decoding. Rows 6 to 8 from Table 3 show the gain of the matching and decoding jointly improving mechanisms. We can see that the combination of either CD & CA or CD & CS has a more substantial positive effect than the memory matching improved alone. Therefore, we believe the joint improvement of memory matching and decoding proposed in this paper is crucial to the matching-based VOS approach.

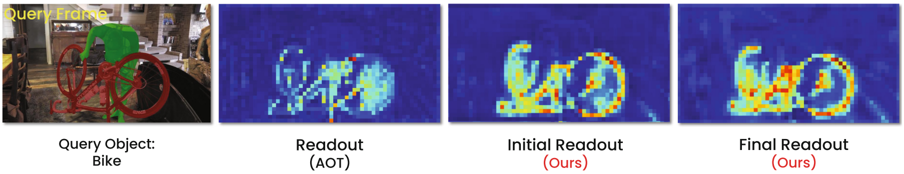


Fig. 5. Visualization of the memory readout features of AOT, our method's initial readout and the final readout (i.e., feature after context embedding).

Visualize the Readout Features. We visually compared the readout features with AOT [27] as shown in Fig. 5. After cost-aware and cross-scale matching, our initial readout results are significantly better than AOT [27]. We can see that our initial readout still has false matching area at the right wheel of the bike. However, after context embedding, the final readout results suppress false matches and increase some high-response features. This suggests that our approach of jointly improving the matching and decoding stages could facilitate producing more accurate and precise masks.

5 Conclusion

This paper proposes a network JIMD that jointly improves the memory matching and decoding stages to address the issue of false matching. We design an improved mechanism for the memory matching stage consisting of cost-aware matching and cross-scale matching for short-term and long-term memory. Cost-aware matching in short-term memory prompts the network to perceive changes between two frames more adequately. Cross-scale matching in long-term memory prompts the network to explore the variations in different scaled objects. For the readout decoding stage, we propose a novel compensatory decoding mechanism that can suppress false matches and supplement the crucial information loss of target objects. We conduct extensive experiments on the effectiveness of joint improvement, and results on popular benchmarks demonstrate that JIMD outperforms existing matching-based methods. Therefore, JIMD has considerable potential to be applied to multimedia applications in the future.

Acknowledgments. This project was supported by the key R&D project of Guangzhou (202206010091) and the fund of Southern Marine Science and Engineering Guangdong Laboratory (Zhanjiang)(ZJW-2023-04).

References

1. Cheng, H.K., Schwing, A.G.: XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In: ECCV (2022)
2. Cheng, H.K., Tai, Y.W., Tang, C.K.: Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In: Advances in Neural Information Processing Systems. vol. 34, pp. 11781–11794 (2021)
3. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C.: Twins: Revisiting the design of spatial attention in vision transformers. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 9355–9366. Curran Associates, Inc. (2021)
4. Gelautz, M.: Short papers_. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE **35**(2) (2013)
5. Guo, P., Zhang, W., Li, X., Zhang, W.: Adaptive online mutual learning bi-decoders for video object segmentation. IEEE Trans. Image Process. **31**, 7063–7077 (2022)
6. Hu, L., Zhang, P., Zhang, B., Pan, P., Xu, Y., Jin, R.: Learning position and target consistency for memory-based video object segmentation pp. 4144–4154 (2021)
7. Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K.C., Qin, H., Dai, J., Li, H.: Flowformer: A transformer architecture for optical flow. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII. pp. 668–685. Springer (2022)
8. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
9. Lu, X., Wang, W., Danelljan, M., Zhou, T., Shen, J., Van Gool, L.: Video object segmentation with episodic graph memory networks pp. 661–679 (2020)

10. Mei, J., Wang, M., Lin, Y., Liu, Y.: Transvos: Video object segmentation with transformers. arXiv preprint arXiv:2106.00588 (2021)
11. Nowozin, S.: Optimal decisions from probabilistic models: the intersection-over-union case. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 548–555 (2014)
12. Oh, S.W., Lee, J.Y., Sunkavalli, K., Kim, S.J.: Fast video object segmentation by reference-guided mask propagation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7376–7385 (2018)
13. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9225–9234 (2019)
14. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017)
15. Qiao, S., Chen, L.C., Yuille, A.: Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10213–10224 (2021)
16. Seong, H., Hyun, J., Kim, E.: Kernelized Memory Network for Video Object Segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12367, pp. 629–645. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58542-6_38
17. Seong, H., Hyun, J., Kim, E.: Video object segmentation using kernelized memory network with multiple kernels. IEEE transactions on pattern analysis and machine intelligence (2022)
18. Seong, H., Oh, S.W., Lee, J.Y., Lee, S., Lee, S., Kim, E.: Hierarchical memory matching network for video object segmentation. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12869–12878 (2021)
19. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8934–8943 (2018)
20. Teed, Z., Deng, J.: RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12347, pp. 402–419. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58536-5_24
21. Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., Chen, L.C.: Feelvos: Fast end-to-end embedding learning for video object segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9473–9482 (2019)
22. Wang, J., Chen, D., Wu, Z., Luo, C., Tang, C., Dai, X., Zhao, Y., Xie, Y., Yuan, L., Jiang, Y.G.: Look before you match: Instance understanding matters in video object segmentation. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 0–0 (2023)
23. Xie, H., Yao, H., Zhou, S., Zhang, S., Sun, W.: Efficient regional memory network for video object segmentation. In: CVPR (2021)
24. Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., Huang, T.: YouTube-VOS: Sequence-to-Sequence Video Object Segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11209, pp. 603–619. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01228-1_36

25. Yang, Z., Miao, J., Wang, X., Wei, Y., Yang, Y.: Associating objects with scalable transformers for video object segmentation. arXiv preprint arXiv:2203.11442 (2022)
26. Yang, Z., Wei, Y., Yang, Y.: Collaborative Video Object Segmentation by Foreground-Background Integration. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12350, pp. 332–348. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58558-7_20
27. Yang, Z., Wei, Y., Yang, Y.: Associating objects with transformers for video object segmentation. In: Advances in Neural Information Processing Systems. vol. 34, pp. 2491–2502 (2021)
28. Yang, Z., Wei, Y., Yang, Y.: Collaborative video object segmentation by multiscale foreground-background integration. IEEE Trans. Pattern Anal. Mach. Intell. **44**(9), 4701–4712 (2022)
29. Yang, Z., Yang, Y.: Decoupling features in hierarchical propagation for video object segmentation. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 36324–36336. Curran Associates, Inc. (2022)

Unsupervised Low-Light Image Enhancement via Spectral Consistency

Bing Li, Wei Yu, Naishan Zheng, Jie Huang, and Feng Zhao(✉)

University of Science and Technology of China, Hefei 230027, China
{bing0123,patrick914y,nszheng,hj0117}@mail.ustc.edu.cn,
fzhao956@ustc.edu.cn

Abstract. Retinex-based unsupervised low-light enhancement methods have demonstrated notable performance without paired data. However, existing Retinex-based unsupervised methods implicitly relax the constraints of Retinex theory and cannot predict the illumination and reflectance exactly, resulting in unstable outcomes. In order to alleviate this issue, we propose a novel framework with stringent consistent constraints for robust Retinex decomposition. Our work is inspired by the spectral characteristics of the low-light images and primarily utilizes the spectral perturbations to establish the training constraints. Specifically, we first investigate the invariant and equivariant components for low-light enhancement under spectral perturbations. Based on these consistency attributes, we design an illumination invariance constraint and a reflectance equivariance constraint for robust decomposition. Furthermore, motivated by the noise distribution under spectral perturbations, we introduce a cross multi-scale noise regularization technique to tackle the severe noise on the reflectance maps. Extensive experiments conducted on diverse datasets have demonstrated the superior performance over state-of-the-art approaches, highlighting its effectiveness and potential for various applications.

Keywords: Image enhancement · Unsupervised learning · Consistency loss

1 Introduction

Images captured under low-light conditions often suffer from reduced visibility, acute noise, and inaccurate color. These degradations not only affect the visual quality but also burden the performance in downstream recognition tasks such as classification and image detection [25]. To tackle these challenges, various low-light image enhancement (LLIE) algorithms have been proposed.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78312-8_14.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 204–220, 2025.
https://doi.org/10.1007/978-3-031-78312-8_14

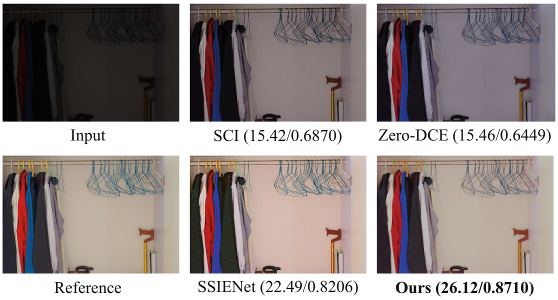

Input
SCI (15.42/0.6870)
Zero-DCE (15.46/0.6449)
Reference
SSIENet (22.49/0.8206)
Ours (26.12/0.8710)

Fig. 1. Qualitative and quantitative (PSNR/SSIM) comparisons of state-of-the-art unsupervised methods, including SCI [33], Zero-DCE [9], SSIENet [43], and our approach. Our method preserves more consistent information than the others.

Traditional LLIE methods resort to enhancing the image contrast by sketching the dynamic range of its histogram or adjusting the image non-linearly. However, these methods often exhibit limited adaptability and result in unnatural appearances. Recently, the advancements in deep learning have led to remarkable progress in data-driven LLIE approaches [2,5,28,30,44]. The most widely studied is the supervised approach. One line of research involves utilizing illumination transformation techniques, where the enhancement is applied to the decomposed illumination map, or a transformation curve is directly employed on the original input image. Another line of work focuses on learning a straightforward mapping from low-light images to normal-light images via an end-to-end training strategy. Despite the impressive results achieved, these methods rely on low-light and normal-light image pairs. Unfortunately, it is challenging to acquire high-quality paired images in real-world scenarios.

More recently, semi-supervised and unsupervised methods have emerged to alleviate the reliance on paired data during the training process. These approaches enable the model to be adapted to enhancement across diverse scenarios by incorporating general physical priors and assumptions. However, due to the complexity of different low-light scenarios and the utilization of lenient priors to constrain network training, achieving high-quality enhancement results remains challenging for unsupervised methods.

To alleviate this issue, we propose leveraging a more stringent prior based on the Retinex model [20] to constrain unsupervised training. While several notable approaches incorporating the Retinex model have been proposed for low-light image enhancement [6,37,38,42], they are mainly based on supervised learning. On the other hand, there are unsupervised methods available that are based on the Retinex theory [29,33,43]. However, it is essential to note that these approaches may introduce inappropriate assumptions or relax the decomposition constraints implicitly, leading to inconsistent (i.e., color shift) results.

Consistency regularizations are common self-supervised learning methods, which impose constraints on the model to produce invariant or equivariant output when input is perturbed. The core insight of consistency regularization is to enforce models to learn the inherent properties of images under certain perturbations. In this paper, we develop a novel framework for deep Retinex decomposition based on consistency regularization. By exploring the intrinsic nature of low-light images, we propose two consistent attributes within spectral perturbation for the decomposition. Firstly, we regard the illumination map as an invariant attribute of the low-light images under spectral shuffle perturbation. This hypothesis comes from the observation that numerous Retinex-based approaches [8,10,29,38] leverage the Max-RGB value as the initial illumination estimation and the Max-RGB value remains unchanged regardless of the channel shuffle. Following the above hypothesis, we argue that the reflectance map and the noise distribution obey the equivariance under the spectral shuffle perturbation. Leveraging these properties, we introduce the illumination invariance constraint, reflectance equivariance constraint, and cross multi-scale noise regularization for robust decomposition. It is worth noting that these constraints are applied between low-light input and perturbed low-light input, rather than between low-light and normal-light input. This allows us to train the entire decomposition network without paired data. Fig. 1 illustrates the qualitative and quantitative comparisons of state-of-the-art unsupervised methods. As we do not relax the constraints of Retinex theory, our method preserves more consistent information than previous methods.

To summarize, our contributions are the following:

- We point out the invariant and equivariant attributes of low-light images for deep Retinex decomposition under spectral perturbation.
- Based on these consistent attributes, we propose a novel self-supervised low-light image enhancement framework, which is trained with constraints between low-light input and perturbed input for robust decomposition.
- We conduct extensive experiments on representative datasets and the results demonstrate the superiority of our approach compared to state-of-the-art methods.

The structure of this paper is as follows: The first section provides an introduction to the topic and the research questions we aim to address. The second section contains a review of related work. The third section then details our effective designs. Subsequently, we present experiments and ablation study in the forth section. Finally, we conclude our method and discuss limitations, while suggesting avenues for future work.

2 Related work

2.1 Low-light Image Enhancement

Conventional Methods. The earlier approaches for low-light image enhancement (LLIE) relied heavily on manually devised operations or filters to correct the image exposure with poor illumination. These approaches can be divided into two categories, one based on histogram [1,16,22,34], which performs light enhancement by expanding the dynamic range of an underexposure image; and the other based on Retinex theory [20], which decomposes an image into the reflectance component and illumination component. The Retinex-based methods primarily focus on enhancing the illumination component to enhance overall lightness, while regularizing the reflectance component to suppress the noises [7,10,18,26,36].

Learning-based Methods. Recently, deep learning-based low-light image enhancement approaches have attracted extensive attention [31,35,38,40]. As representative work, RetinexNet [37] and KinD [42] follow the Retinex theory and decompose an image into reflectance and illumination components with low-light and normal-light pairs. More recently, Fu *et al.* [6] utilized contrastive learning and self-knowledge distillation for Retinex decomposition. PairLIE [8] achieved the decomposition based on paired low-light instances. By leveraging aligned data pairs, these methods achieved remarkable results.

However, in practice, it is difficult to obtain the paired data of the same scene simultaneously [4,19,27]. Therefore, unsupervised/semi-supervised methods have been proposed [9,17,41]. For example, SSIENet [43] employs maximum entropy to constraint Retinex decomposition. However, this assumption ignores the color consistency, resulting in color-shifted results. RUAS [29] builds a Retinex-inspired framework leveraging the unrolling technique and architecture search strategies. SCI [33] develops a learning framework based on a self-calibrated module for illumination enhancement. While these approaches accomplish LLIE tasks without paired images, they lack the exploration of the reflectance property, leading to unstable outcomes.

2.2 Invariance and Equivariance Regularization

Invariance and equivariance regularization are common self-supervised learning methods. The invariance ensures that the output of a model remains consistent despite perturbations of input, while equivariance implies that the output should exhibit a corresponding transformation to the input. The invariance property finds extensive application in representation learning [3,14], where distinctive and invariant features are extracted to enable robustness. In terms of equivariance, CNNs are shown to have approximate translation equivariance due to the nature of convolution [24]. Additionally, the exploration of rotation and scale equivariance [12,23] represents notable areas of research in the field. By imposing these constraints, the generalizability and robustness of the network can be improved.

Although these regularization constraints have demonstrated remarkable performance in fields such as classification, detection, and segmentation, there has been limited exploration in the context of LLIE. Motivated by previous research [45,46] and leveraging the specific characteristics of low-light images, we introduce spectral-based invariance and equivariance constraints to establish an unsupervised low-light image enhancement framework.

3 Methodology

Initially, we investigate the spectral characteristics of low-light images and identify the spectral consistency (*i.e.*, invariance and equivariance) under spectral perturbation. Next, we present the framework of our method. Lastly, we detail the consistency constraints.

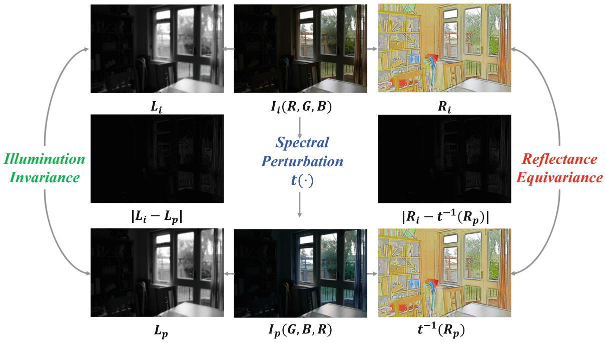


Fig. 2. The decomposition results with RetinexNet [37]. I_i and I_p refer to the low-light image and perturbed low-light image. L_* and R_* represent the estimated illumination and reflectance of I_*. To enhance visual comparison, we reorder the spectrum of R_p. The decomposition of I_i and I_p is presented in the first and third rows, respectively. The first column represents the proposed illumination invariance, while the third column shows the reflectance equivariance. The absolute errors displayed in the second row verify the proposed consistency of low-light images.

3.1 Invariance and Equivariance

Definition 1 (Invariance). A function $f : X \rightarrow Y$ is invariant to a symmetry group G if for all transformation $g \in G$ on the input $x \in X$ the result remains unchanged, *i.e.*, for any transformation g :

$$f \circ g(x) = f(x). \tag{1}$$

Definition 2 (Equivariance). A function $f : X \rightarrow Y$ is equivariant to symmetry group G on X and G' on Y if for any transformation $g \in G$ there exists $g' \in G'$ such that:

$$f \circ g(x) = g' \circ f(x). \tag{2}$$

In particular, invariance can be regarded as a special case of equivariance where g' is the identity transformation. In addition, if transformations on both X and Y domains are the same, *i.e.*, $G = G'$, the equivariance can be denoted as:

$$f \circ g(x) = g \circ f(x). \tag{3}$$

This case is common in image processing, where we consider both g and g' as the transformations applied to the image. Both invariance and equivariance are commonly employed forms of consistency in deep learning.

3.2 Spectral Consistency

The Retinex model assumes an observed image I can be decomposed into illumination L and reflectance R, and represented as their element-wise product, denoted as:

$$I = L \cdot R, \tag{4}$$

where $\cdot$ denotes element-wise multiplication. The paired-based methods [8,37] mainly utilize the assumptions that images captured in different light conditions maintain the same reflectance and the illumination map is expected to be smooth to achieve decomposition. However, when only single light condition images are available, it is necessary to explore alternative assumptions.

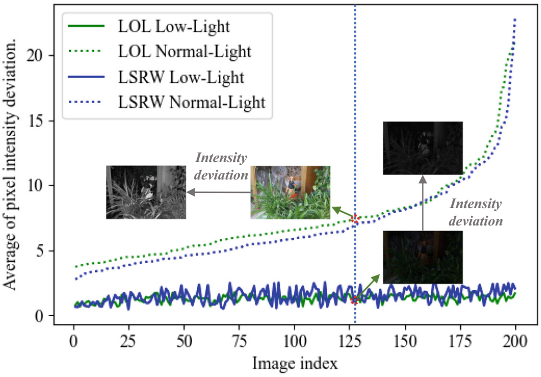


Fig. 3. The intensity deviation across channels of LOL [37] and LSRW [11] datasets. We randomly select 200 paired images from each dataset and linearly rescale the deviation to enhance visualization. The observation reveals that low deviation is a common statistical characteristic among low-light images.

Illumination Invariance Notably, numerous Retinex-based approaches [8,10,29,38] have leveraged the Max-RGB value as the initial illumination estimation and Retinex theory assumes the different color spectrum have the same illumination. These observations inspire us to regard the illumination map as an invariant attribute of the low-light images under spectral shuffle perturbation since the Max-RGB value remains unchanged regardless of the spectral shuffle. We refer to this property as the *illumination invariance* of the low-light image, which can be denoted as:

$$l \circ t(I) = l(I), \tag{5}$$

where I denotes the low-light image, $l(\cdot)$ denotes the illumination estimation operator, and $t(\cdot)$ denotes the spectral shuffle perturbation.

To validate our hypothesis, we conducted a pilot study to investigate the illumination component of low-light images within low-light image decomposition [37] and results are shown in Fig. 2. It can be observed that illumination maps have consistent predictions with different spectral order inputs, which indicates that illumination is an invariant attribute. Consequently, illumination invariance can serve as a consistency constraint for low-light decomposition.

Another insight regarding the illumination invariance is related to the reduced color information and decreased contrast in low-light conditions. These factors contribute to a limited range of pixel values in low-light images. As a consequence, low-light images exhibit less intensity deviation across channels compared to corresponding normal-light images. To illustrate this, we present the intensity deviation across channels of low-light images from various low-light datasets [11,37] in Fig. 3. It can be seen that the low deviation is a general statistical property of low-light images. This statistical property suggests that if we were to rearrange the spectral order of low-light images, the rearranged low-light images would still appear similar to the initial images. Therefore, the estimation of low-light image illumination is minimally affected by the spectral shuffle perturbation as the estimation primarily relies on intensity characteristics.

Reflectance Equivariance Following the Retinex theory (*i.e.*, Eq. 4) and illumination invariance (*i.e.*, Eq. 5), we can further derive that reflectance obeys an equivariance within spectral shuffle perturbation, which is called *reflectance equivariance* and can be expressed as:

$$r \circ t(I) = t \circ r(I), \tag{6}$$

where $r(\cdot)$ denotes the reflectance estimation operator. The derivation can be found in Supplementary Materials[1].

Intuitively, the reflectance component of an image captures the intrinsic properties of the scene, including shape, texture, and color. The shape and texture are color-independent properties, and they can be correctly predicted regardless of spectral order. The main concern of reflectance equivariance is whether the color information can be well maintained as the order of spectral is highly related to the color information. To alleviate this concern, we conducted two investigations. First, we employ a trained decomposition network [37] to examine whether the reflectance output maintains a consistent transformation to the input under spectral shuffle perturbations. As shown in Fig. 2, the results demonstrate a strong alignment between the reflectance outputs. Additionally, we directly apply perturbed input to the pre-trained enhancement network to assess the potential color shift in the output. Interestingly, our finding reveals a negligible error between the direct input-output mapping and the perturbed input-inverse output mapping. This compelling evidence suggests that spectral shuffle acts as a

[1] https://github.com/lbu19/SCLLIE/blob/main/supplementary.pdf

mild augmentation for low-light image enhancement, exerting minimal influence on color preservation. More details are shown in Supplementary Materials.

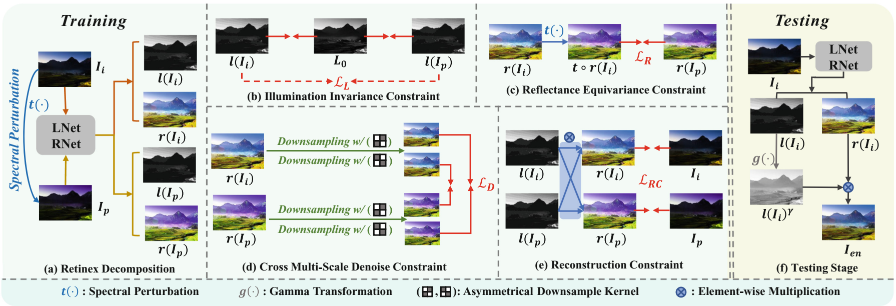


Fig. 4. Overview of the proposed framework. (a) During the training stage, both the low-light image and perturbed image are fed into LNet and RNet to estimate the illumination and reflectance. The training stage is guided by four well-designed loss functions, including (b) illumination invariance constraint $\mathcal{L}_L$, (c) reflectance equivariance constraint $\mathcal{L}_R$, (d) cross multi-scale denoise constraint $\mathcal{L}_D$, and (e) reconstruction constraint $\mathcal{L}_{RC}$. (f) In the testing stage, LNet and RNet are employed to decompose the low-light image, and the enhanced output is obtained by adjusting the illumination and recomposing it with the reflectance.

3.3 Framework

We illustrate our framework in Fig. 4. Our method primarily utilizes spectral perturbations to establish self-supervised training. We employ two simple networks, referred to as LNet and RNet, to estimate illumination and reflectance components, respectively. The details of network architecture can be found in Supplementary Materials. During the training phase, the low-light image and perturbed low-light image are fed into LNet and RNet to estimate the illumination and reflectance with four well-designed loss functions. To guide the illumination estimation, we employ an illumination invariance constraint as the first loss function. As for the reflectance estimation, we enforce the RNet to explore reflectance properties by incorporating a reflectance equivariance constraint as the second loss function. Additionally, we incorporate a third loss function, known as the cross multi-scale denoise constraint, to effectively suppress severe noise in the reflectance component. Finally, we introduce the reconstruction loss that ensures the decomposed illumination and reflectance satisfy the Retinex theory. In the testing stage, we directly employ LNet and RNet to decompose the input image and correct the illumination with gamma transformation. The enhanced result can be achieved by:

$$I_{en} = g(L) \cdot R = l(I)^\gamma \cdot r(I), \tag{7}$$

where I and I_{en} represent the input low-light image and enhanced image, $g(\cdot)$ denotes gamma transformation, and γ is the correction factor.

3.4 Self-supervised Constraints

In this section, we provide detailed expressions of our constraints based on spectral consistency and Retinex theory. Given the input low-light image I_i, we first get perturbed low-light image I_p via random spectral shuffle transformation $t(\cdot)$, which can be denoted as $I_p = t(I_i)$. We then construct the following loss functions with these two images.

Illumination Invariance Constraint. As discussed in Section 3.2, I_i and I_p possess the same illumination component (*i.e.*, $l(I_i) = l(I_p)$). However, directly adopting this property as a loss function tends to lead to training collapse in practice. Therefore, we constrain an alternative upper bound by introducing the initialized illumination L_0 as the bridge. Specifically, the L_0 is calculated via the maximum of the R, G, and B spectrum [8,10,29]: $L_0 = \max_{c\in\{R,G,B\}} I^c(x)$. Noticed that I_i and I_p shared a same L_0, we have:

$$\frac{1}{2}||l(I_i)-l(I_p)||_2^2 = \frac{1}{2}||l(I_i)-L_0+L_0-l(I_p)||_2^2 \leq ||l(I_i)-L_0||_2^2+||l(I_p)-L_0||_2^2. \quad (8)$$

We also impose a smooth term on estimated illumination. The whole illumination loss can be formulated as:

$$\mathcal{L}_L = \sum_{j\in\{i,p\}} ||l(I_j) - L_0||_1 + \lambda_s||\nabla l(I_j) \cdot exp(-\lambda_r \nabla r(I_j))||_1, \quad (9)$$

where λ_s and λ_r denotes the smooth weight, ∇ represents the horizontal and vertical gradients.

Reflectance Equivariance Constraint. Reflectance Equivariance indicates that the reflectance map of I_i and I_p differ from each other by a transformation $t(\cdot)$. Thus, we have the loss function based on Eq. 6:

$$\mathcal{L}_R = ||t \circ r(I_i) - r(I_p)||_2^2. \quad (10)$$

By incorporating this constraint, we enable the network to focus on exploring the intrinsic properties of low-light images, contributing to consistent structure preservation.

Cross Multi-Scale Denoise Constraint. To cope with the severe noise on the reflectance, we formulate the denoising constraint based on Neighbor2Neighbor [15]. Different from Neighbor2Neighbor, we apply the asymmetrical sub-samplers on $r(I_i)$ and $r(I_p)$ instead of a single noise image. By employing asymmetrical sub-samplers, we not only effectively remove noise but also maintain the consistency of the reflectance component across the images. Nevertheless, the constraint between sub-sampled images results in the block artifact. To mitigate these block artifacts, we introduce a smoothness constraint on reflectance. The whole denoise loss can be expressed as:

$$\mathcal{L}_D = \sum_{s\in\{2,4\}} ||t \circ r(I_i) \downarrow^s - r(I_p) \downarrow^s ||_2^2 + \lambda_d \sum_{j\in\{i,p\}} ||\nabla r(I_j)||_1, \quad (11)$$

where $\downarrow^s$ represents downsample with scale s.

Reconstruction Constraint. Based on the illumination invariance assumption that I_l and I_p share the same illumination map, the reconstruction loss is formulated as:

$$\mathcal{L}_{RC} = \sum_{j \in \{i,p\}} \sum_{k \in \{i,p\}} ||l(I_j) \cdot r(I_k) - I_k||_1. \tag{12}$$

Overall Constraint. The overall loss function is a linear combination of the above constraints:

$$\mathcal{L}_{total} = \mathcal{L}_L + \mathcal{L}_R + \lambda_D \mathcal{L}_D + \mathcal{L}_{RC}, \tag{13}$$

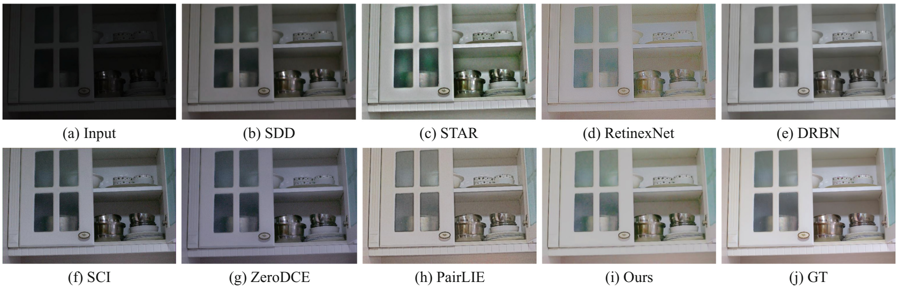
where λ_D is the weight.

Table 1. Quantitative comparison of the state-of-the-art LLIE methods on the LOL and LSRW datasets. The top three results are marked in bold. T, S, and U represent traditional methods, supervised methods, and unsupervised methods, respectively.

Dataset		LOL				LSRW			
Metrics		PSNR↑	SSIM↑	LPIPS↓	DeltaE↓	PSNR↑	SSIM↑	LPIPS↓	DeltaE↓
T	SDD [13]	13.34	0.6368	0.2623	21.83	14.71	0.4998	0.4137	15.52
	STAR [39]	16.47	0.6972	0.2943	23.46	14.61	0.5039	0.4749	15.01
S	RetinexNet [37]	17.61	0.6481	0.3858	12.69	15.58	0.4312	0.4017	14.12
	KinD [42]	17.65	**0.7750**	**0.1713**	12.49	16.15	**0.5422**	0.3504	12.46
	DRBN [41]	16.29	0.5515	0.2597	13.44	15.97	0.5393	0.3442	12.63
	URetinexNet [38]	**19.84**	**0.8260**	**0.1281**	**10.65**	**18.27**	**0.5368**	**0.2994**	**10.04**
U	ZeroDCE [9]	14.86	0.5588	0.3352	18.81	15.83	0.4664	**0.3250**	14.85
	EnGAN [17]	17.48	0.6645	0.3159	14.50	16.31	0.4697	**0.3136**	**11.66**
	SSIENet [43]	19.50	0.7003	0.2898	12.73	16.74	0.4873	0.3732	15.02
	SCI [33]	14.78	0.5254	0.3393	19.52	15.02	0.4846	0.3260	15.13
	PairLIE [8]	**19.51**	0.7364	0.2477	**10.80**	**16.92**	0.5015	0.3370	12.31
	Ours	**19.72**	**0.7776**	**0.2241**	**11.70**	**17.82**	**0.5574**	0.3950	**11.31**

4 Experiments

In this section, we first provide details of our experiment implementation, evaluation datasets, and performance criteria. Then, we present the quantitative and qualitative comparisons between our proposed method and state-of-the-art LLIE methods. Finally, we conduct ablation experiments to validate the efficiency of our designs.



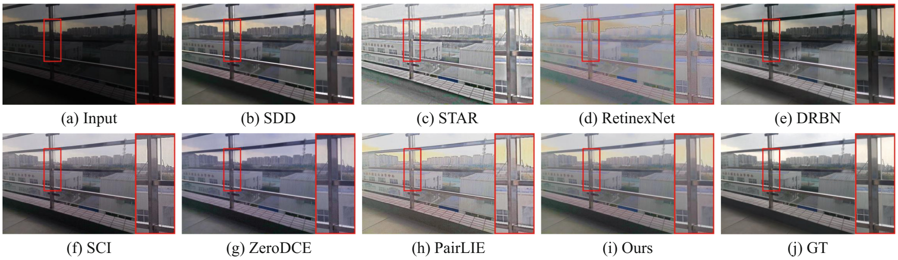
(a) Input (b) SDD (c) STAR (d) RetinexNet (e) DRBN

(f) SCI (g) ZeroDCE (h) PairLIE (i) Ours (j) GT

Fig. 5. Visual comparison of the state-of-the-art LLIE methods on the LOL dataset. More comparison results can be found in the Supplementary Materials.

4.1 Implementation Details

We conduct experiments by the PyTorch platform with one NVIDIA GeForce RTX 3070 GPU. The training images are cropped into 256 × 256 pixels. Adam optimizer is adopted with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ for a total of 200 epochs. For the loss function, we set coefficients λ_r and λ_d to 10 and 0.2, respectively. To cope with different noise levels, λ_D is set to 0.1 and 0.02, experimentally. The initial learning rate is 1×10^{-4} and we apply a half-decay per 50 epochs to adaptively adjust the learning rate during training.



(a) Input (b) SDD (c) STAR (d) RetinexNet (e) DRBN

(f) SCI (g) ZeroDCE (h) PairLIE (i) Ours (j) GT

Fig. 6. Visual comparison of the state-of-the-art LLIE methods on the LSRW dataset. More comparison results can be found in the Supplementary Materials.

4.2 Datasets and Metrics

We collect low-light images from the LOL dataset [37] and LSRW dataset [11] to train the model. For performance evaluation, we utilize the official evaluation set of the LOL dataset (15 images) and the LSRW dataset (50 images). We adopt three full-reference metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), Learned Perceptual Image Patch Similarity (LPIPS) and DeltaE with CIE2000 standard as numerical evaluation metrics. Higher PSNR or SSIM

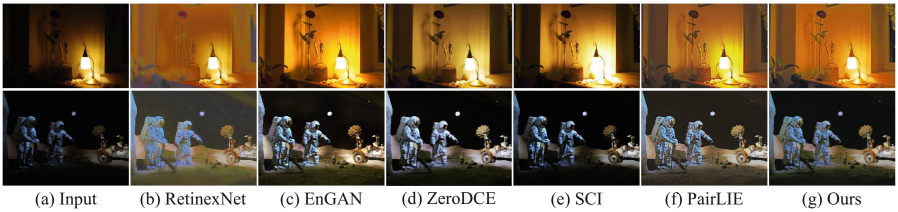


Fig. 7. Visual results on unpaired datasets, where the first row shows the results on the MEF dataset, and the second row displays the results on the DICM dataset. Please zoom in for details.

means a closer resemblance to the reference image, while lower LPIPS or DeltaE value indicates better visual quality. We also extend evaluation to LIME [10], MEF [32], NPE [36], and DICM [21] datasets for a more convincing comparison. Since these datasets do not provide reference images for evaluation, we employ the Naturalness Image Quality Evaluator (NIQE) as an alternative metric. In general, a lower NIQE means better visual quality. The results are presented in Supplementary Materials due to the limited space.

4.3 Comparison with State-of-the-art Methods

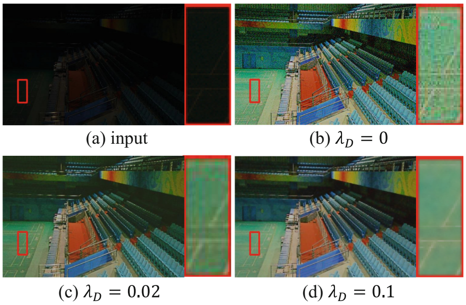
We compare our method with two traditional methods, including SDD [13], STAR [39]. Furthermore, to verify the efficiency of our methods, we compare with state-of-the-art learning-based methods including four supervised methods (RetinexNet [37], KinD [42], DRBN [41], URetinexNet [38]) and five unsupervised methods (ZeroDCE [9], EnGAN [17], SSIENet [43], SCI [33], PairLIE [8]). It is worth noting that EnGAN [17] is trained with unaligned low-normal-light pairs and PairLIE [8] is trained with paired low-light images. In contrast, our method is trained with single low-light images. This difference in training data highlights the flexibility of our approach.

Quantitative Comparisons. We present the quantitative comparisons of the LOL and LSRW datasets in Table 1. Since the light conditions vary across different datasets, we experimentally set the correction factor γ to 0.15 and 0.2 for LOL and LSRW, respectively. As can be observed, existing unsupervised methods tend to result in poor quantitative results, especially in terms of SSIM. This is because they employ the sub-optimal assumption or lenient priors to constrain training, which may destroy the structure or color information of the low-light images. As illustrated in Table 1, our method outperforms other unsupervised methods and achieves comparable results with the supervised algorithm. Notably, our method encourages superior structural preservation by introducing consistency constraints over reflectance and exhibits significant superiority to other methods by large margins in terms of SSIM.

Qualitative Comparisons. In Figs. 5, 6, and 7, we present visual quality comparisons for the LOL, LSRW, MEF, and DICM datasets. Upon observation, it is evident that SDD [13], SCI [33], and ZeroDCE [9] struggle to effectively enhance the overall brightness of the images. Furthermore, STAR [39], RetinexNet [37], DRBN [41], and PairLIE [8] introduce noticeable artifacts on the structure details of the images, which can result in unappealing visual outcomes. Remarkably, our method can successfully suppress noise and produce clear and natural results. More visual results can be found in Supplementary Materials.

Table 2. Results of ablation studies on the LOL dataset. The baseline is our full framework. The best results are marked in bold.

Setting	PSNR↑	SSIM↑	LPIPS↓
A	10.87	0.3454	0.8763
B	19.09	0.7556	0.2447
C	17.80	0.6299	0.4538
Baseline	**19.72**	**0.7776**	**0.2241**


(a) input

(b) $\lambda_D = 0$

(c) $\lambda_D = 0.02$

(d) $\lambda_D = 0.1$

Fig. 8. The enhancement results with different denoise constraint weights λ_D.

4.4 Ablation Study

To demonstrate the rationality of our designs, we conduct ablation studies on the LOL dataset. The ablation results are shown in Table 2. Firstly, (**A**) we utilize the direct constraint between illumination to replace the upper bound in Eq. 8. Due to the significant loss of prior knowledge about the illumination component, the training process becomes unstable and leads to a large performance decline. Secondly, (**B**) we discard the $\mathcal{L}_R$ (Eq. 10) to explore the effectiveness of reflectance equivariance. Without the reflectance equivariance constraint, there is a certain performance degradation, which suggests that reflectance equivariance can assist in learning structure details by forcing the network to focus on the intrinsic characteristics of the image itself. Lastly, (**C**) we set λ_D to 0 in Eq. 13 to validate denoise constraint. The absence of the denoise constraint contributes to a large performance decline. This is primarily because the low-light images are polluted by noise severely and noise removal is an essential aspect of the low-light enhancement task. By neglecting the denoising constraint, the model fails to effectively address the noise present in the images, leading to a noticeable deterioration in performance. We further provide visualizations of the enhancement results obtained with different λ_D values in Fig. 8. As observed, the enhanced result is highly influenced by noise artifacts when the denoise constraint is discarded while remarkably appealing results can be achieved with our denoise strategy.

5 Conclusions and Limitations

In this paper, we first investigate the spectral consistency of low-light images. Then, we point out that the illumination and reflectance of low-light images should maintain the invariance and equivariance under the spectral shuffle perturbations, respectively. Building upon the above observation, we propose a novel framework for unsupervised LLIE by constraining spectral consistency between images before and after perturbations. These constraints contribute to a more robust decomposition process and enable the enhanced image to retain more consistent information throughout. Extensive experimental results have demonstrated the effectiveness and flexibility of our method. However, the proposed approach is not optimized for capturing the distortion information caused by extreme dark scenes, leading to less detailed results compared to the ground truth or supervised methods due to the inherent loss of details in low-light images. A robust semantic feature may provide the guidance for content recovery. For future work, it is worth exploring the integration of a large pre-trained model in normal light scenes that can introduce stable semantic feature to assist the detail restoration.

Acknowledgments. This work was supported by the Anhui Provincial Natural Science Foundation under Grant 2108085UD12. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

1. Arici, T., Dikbas, S., Altunbasak, Y.: A histogram modification framework and its application for image contrast enhancement. IEEE Trans. Image Process. **18**(9), 1921–1935 (2009)
2. Cai, Y., Bian, H., Lin, J., Wang, H., Timofte, R., Zhang, Y.: Retinexformer: One-stage retinex-based transformer for low-light image enhancement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12504–12513 (2023)

3. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15750–15758 (2021)
4. Du, W., Chen, H., Yang, H.: Learning invariant representation for unsupervised image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14483–14492 (2020)
5. Fei, B., Lyu, Z., Pan, L., Zhang, J., Yang, W., Luo, T., Zhang, B., Dai, B.: Generative diffusion prior for unified image restoration and enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9935–9946 (2023)
6. Fu, H., Zheng, W., Meng, X., Wang, X., Wang, C., Ma, H.: You do not need additional priors or regularizers in retinex-based low-light image enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18125–18134 (2023)
7. Fu, X., Zeng, D., Huang, Y., Zhang, X.P., Ding, X.: A weighted variational model for simultaneous reflectance and illumination estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2782–2790 (2016)
8. Fu, Z., Yang, Y., Tu, X., Huang, Y., Ding, X., Ma, K.K.: Learning a simple low-light image enhancer from paired low-light instances. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22252–22261 (2023)
9. Guo, C., Li, C., Guo, J., Loy, C.C., Hou, J., Kwong, S., Cong, R.: Zero-reference deep curve estimation for low-light image enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1780–1789 (2020)
10. Guo, X., Li, Y., Ling, H.: Lime: Low-light image enhancement via illumination map estimation. IEEE Trans. Image Process. **26**(2), 982–993 (2016)
11. Hai, J., Xuan, Z., Yang, R., Hao, Y., Zou, F., Lin, F., Han, S.: R2RNet: Low-light image enhancement via real-low to real-normal network. Journal of Visual Communication and Image Representation **90**, 103712.1–12 (2023)
12. Han, J., Ding, J., Xue, N., Xia, G.S.: Redet: A rotation-equivariant detector for aerial object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2786–2795 (2021)
13. Hao, S., Han, X., Guo, Y., Xu, X., Wang, M.: Low-light image enhancement with semi-decoupled decomposition. IEEE Trans. Multimedia **22**(12), 3025–3038 (2020)
14. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738 (2020)
15. Huang, T., Li, S., Jia, X., Lu, H., Liu, J.: Neighbor2neighbor: Self-supervised denoising from single noisy images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14781–14790 (2021)
16. Ibrahim, H., Kong, N.S.P.: Brightness preserving dynamic histogram equalization for image contrast enhancement. IEEE Trans. Consum. Electron. **53**(4), 1752–1758 (2007)
17. Jiang, Y., Gong, X., Liu, D., Cheng, Y., Fang, C., Shen, X., Yang, J., Zhou, P., Wang, Z.: Enlightengan: Deep light enhancement without paired supervision. IEEE Trans. Image Process. **30**, 2340–2349 (2021)
18. Jobson, D.J., Rahman, Z.u., Woodell, G.A.: A multiscale retinex for bridging the gap between color images and the human observation of scenes. IEEE Transactions on Image Processing **6**(7), 965–976 (1997)

19. Ke, R., Schönlieb, C.B.: Unsupervised image restoration using partially linear denoisers. IEEE Trans. Pattern Anal. Mach. Intell. **44**(9), 5796–5812 (2021)
20. Land, E.H.: The retinex theory of color vision. Sci. Am. **237**(6), 108–129 (1977)
21. Lee, C., Lee, C., Kim, C.S.: Contrast enhancement based on layered difference representation. In: Proceedings of the 19th IEEE International Conference on Image Processing. pp. 965–968 (2012)
22. Lee, C., Lee, C., Kim, C.S.: Contrast enhancement based on layered difference representation of 2D histograms. IEEE Trans. Image Process. **22**(12), 5372–5384 (2013)
23. Lee, J., Kim, B., Kim, S., Cho, M.: Learning rotation-equivariant features for visual correspondence. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21887–21897 (2023)
24. Lenc, K., Vedaldi, A.: Understanding image representations by measuring their equivariance and equivalence. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 991–999 (2015)
25. Li, C., Guo, C., Han, L., Jiang, J., Cheng, M.M., Gu, J., Loy, C.C.: Low-light image and video enhancement using deep learning: A survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(12), 9396–9416 (2021)
26. Li, M., Liu, J., Yang, W., Sun, X., Guo, Z.: Structure-revealing low-light image enhancement via robust retinex model. IEEE Trans. Image Process. **27**(6), 2828–2841 (2018)
27. Lin, X., Ren, C., Liu, X., Huang, J., Lei, Y.: Unsupervised image denoising in real-world scenarios via self-collaboration parallel generative adversarial branches. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12642–12652 (2023)
28. Lin, X., Yue, J., Ding, S., Ren, C., Guo, C.L., Li, C.: Unlocking low-light-rainy image restoration by pairwise degradation feature vector guidance. arXiv preprint arXiv:2305.03997 (2023)
29. Liu, R., Ma, L., Zhang, J., Fan, X., Luo, Z.: Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10561–10570 (2021)
30. Liu, S., Zhang, Y.: Detail-preserving underexposed image enhancement via optimal weighted multi-exposure fusion. IEEE Trans. Consum. Electron. **65**(3), 303–311 (2019)
31. Lore, K.G., Akintayo, A., Sarkar, S.: LLNet: A deep autoencoder approach to natural low-light image enhancement. Pattern Recogn. **61**, 650–662 (2017)
32. Ma, K., Zeng, K., Wang, Z.: Perceptual quality assessment for multi-exposure image fusion. IEEE Trans. Image Process. **24**(11), 3345–3356 (2015)
33. Ma, L., Ma, T., Liu, R., Fan, X., Luo, Z.: Toward fast, flexible, and robust low-light image enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5637–5646 (2022)
34. Park, G.H., Cho, H.H., Choi, M.R.: A contrast enhancement method using dynamic range separate histogram equalization. IEEE Trans. Consum. Electron. **54**(4), 1981–1987 (2008)
35. Wang, R., Zhang, Q., Fu, C.W., Shen, X., Zheng, W.S., Jia, J.: Underexposed photo enhancement using deep illumination estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6849–6857 (2019)

36. Wang, S., Zheng, J., Hu, H.M., Li, B.: Naturalness preserved enhancement algorithm for non-uniform illumination images. IEEE Trans. Image Process. **22**(9), 3538–3548 (2013)
37. Wei, C., Wang, W., Yang, W., Liu, J.: Deep retinex decomposition for low-light enhancement. arXiv preprint arXiv:1808.04560 (2018)
38. Wu, W., Weng, J., Zhang, P., Wang, X., Yang, W., Jiang, J.: Uretinex-net: Retinex-based deep unfolding network for low-light image enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5901–5910 (2022)
39. Xu, J., Hou, Y., Ren, D., Liu, L., Zhu, F., Yu, M., Wang, H., Shao, L.: Star: A structure and texture aware retinex model. IEEE Trans. Image Process. **29**, 5022–5037 (2020)
40. Xu, X., Wang, R., Fu, C.W., Jia, J.: SNR-aware low-light image enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17714–17724 (2022)
41. Yang, W., Wang, S., Fang, Y., Wang, Y., Liu, J.: From fidelity to perceptual quality: A semi-supervised approach for low-light image enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3063–3072 (2020)
42. Zhang, Y., Zhang, J., Guo, X.: Kindling the darkness: A practical low-light image enhancer. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 1632–1640 (2019)
43. Zhang, Y., Di, X., Zhang, B., Wang, C.: Self-supervised image enhancement network: Training with low light images only. arXiv preprint arXiv:2002.11300 (2020)
44. Zhao, L., Lu, S.P., Chen, T., Yang, Z., Shamir, A.: Deep symmetric network for underexposed image enhancement with recurrent attentional learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12075–12084 (2021)
45. Zhong, Y., Bhattad, A., Wang, Y.X., Forsyth, D.: Improving equivariance in state-of-the-art supervised depth and normal predictors. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 21775–21785 (2023)
46. Zhu, A., Zhang, L., Shen, Y., Ma, Y., Zhao, S., Zhou, Y.: Zero-shot restoration of underexposed images via robust retinex decomposition. In: Proceedings of the IEEE International Conference on Multimedia and Expo. pp. 1–6 (2020)

MMSISP: A Satellite Image Sequence Prediction Network with Multi-factor Decoupling and Multi-modal Fusion

Fanbin Mo, Yixiang Huang, Ming Wu, Xun Zhu, and Chuang Zhang(✉)

School of Artificial Intelligence, Beijing University of Posts and Telecommunications,
Beijing, China
{mofanbin,huangyixiang,wuming,zhuxun,zhangchuang}@bupt.edu.cn

Abstract. Satellite image sequence prediction is a branch of spatio-temporal prediction, which holds considerable potential for practical applications. However, the complex and diverse changes of satellite images over time hinder existing spatio-temporal prediction models from achieving high-accuracy long-term predictions. In this paper, we propose a method called **MMSISP** (**M**ulti-Factor **M**ulti-Modal **S**atellite **I**mage **S**equence **P**redictor). This method decomposes satellite image changes into multiple factors and models them using two branches. The motion branch is utilized for predicting cloud movement, while the appearance branch is employed for forecasting cloud variations (*e.g.*, formation and dissipation), as well as brightness change. Additionally, we introduce two modalities: capture time and meteorological data, enabling the model to have more clues for predicting future frames. For the capture time, we design a time embedding module that enables the model to infer brightness and learn seasonal patterns of cloud formation and dissipation. Regarding meteorological data, which contains information about cloud movement and cloud variations, we devise different spatio-temporal multi-modal fusion mechanisms for the two branches. Based on experiments conducted on the Himawari-8 satellite images, our method demonstrates a significant improvement in accuracy compared to other methods.

Keywords: Satellite image sequence prediction · Spatio-temporal prediction · Multi-modal fusion

1 Introduction

Satellite image sequence prediction, as a branch of spatio-temporal prediction, holds broad application prospects, such as photovoltaic power generation prediction [12], floods prediction [5] and weather nowcasting [11]. Inferring accurate satellite image sequences is beneficial to the normal conduct of social activities, thus work in this area has gradually increased in recent years. Since general

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 221–236, 2025.
https://doi.org/10.1007/978-3-031-78312-8_15

spatio-temporal prediction methods have shown limited effectiveness on satellite image sequences, many models considering the characteristics of satellite images have been proposed. Nonetheless, they still exhibit certain limitations. Some works [7,21] have limited practical value due to short lead times. Other works [1,2,12] use GANs to avoid the blurriness of long-term predictions, but there is limited discussion on improving the accuracy by considering the diversity of satellite image changes. In contrast, our primary goal is to maximize the practical utility of satellite image sequence prediction by focusing on improving the accuracy of long-term predictions.

Additionally, previous works typically avoid using visible channels due to factors such as nighttime unavailability and significant brightness change. However, visible channels also hold unique value, often possessing higher resolution and enabling more intuitive differentiation of land cover types, which is crucial for agricultural and environmental monitoring. Therefore, our method primarily focuses on experiments using visible channels.

We argue that there are two main reasons for the low accuracy of long-term predictions. One is that multiple changing factors in satellite images are not adequately separated, and the other is the lack of information to infer image changes. In this work, we propose MMSISP, a method that decouples multiple changing factors in satellite images and uses multi-modal data to complement information.

The changes in satellite images over time can be generally categorized into cloud movement, cloud variations and brightness change, as shown in Fig.1. Cloud variations include the formation and dissipation of the cloud, while brightness change comes from the change of solar illumination. Therefore, we decouple the network accordingly, using two branches to handle different factors. The motion branch is responsible for predicting cloud movement, while the appearance branch handles cloud variations and brightness change. After obtaining the outputs of the two branches, we design an Adaptive Fusion Module (AFM) to perform adaptive fusion between them.

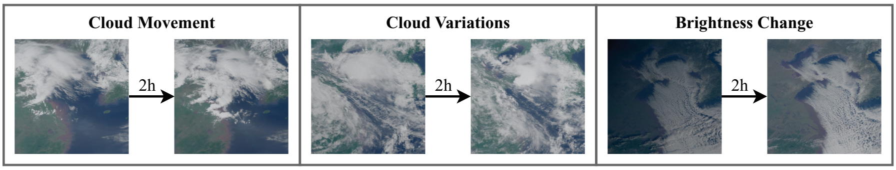


Fig. 1. Satellite image changes over time come in various forms, such as cloud movement, cloud variations (*e.g.*, formation and dissipation) and brightness change.

In addition, we utilize two extra modalities: capture time and meteorological data, to aid the model in inferring image changes. For the capture time, we design a lightweight plug-in time embedding module for ConvRNNs, such as ConvLSTM [10] and PredRNN [18]. Based on the capture time, the model can determine whether it is currently morning or afternoon, thereby inferring the change of brightness in the future. Furthermore, by considering the date of capture, the model can learn seasonal patterns of cloud formation and dissipation.

Meteorological data, such as temperature, humidity, wind direction and wind speed, contain information about cloud movement, formation and dissipation. Like satellite image sequences, meteorological data sequences belong to spatio-temporal data. However, fusion between two spatio-temporal modalities has rarely been explored. Therefore, based on spatial multi-modal fusion methods such as [15], we propose different spatio-temporal multi-modal fusion mechanisms for two distinct branches. Specifically, for the motion branch, we devise an alignment module. For the appearance branch, we introduce an additional ConvRNN, a module called Spatial Channel Attention (SCA), and a dedicated loss function.

Our main contributions are summarized as:

1) We propose MMSISP, a novel method for satellite image sequence prediction, which exhibits superior performance across various metrics during evaluation on Himawari-8 satellite images.
2) We disentangle satellite image changes over time into different factors and design two branches: the motion branch and the appearance branch, to model them separately.
3) For the first time, we incorporate multi-modal information such as capture time and meteorological data into the task. Additionally, we devise a lightweight time embedding module and spatio-temporal multi-modal fusion mechanisms tailored for different branches.

2 Related Work

2.1 Spatio-Temporal Prediction

Spatio-temporal prediction refers to the task of predicting future frames using given past frames. Recurrent-based spatio-temporal prediction models [4,10,14,17–20], renowned for their exceptional ability to capture spatio-temporal correlations, have been widely applied across various domains. In recent years, the potential of recurrent-free models [3,13,23] has been gradually uncovered, surpassing recurrent-based models in some domains. Furthermore, due to the ability of generative models such as GANs and diffusion models to generate high-resolution images, they have garnered increasing attention in spatio-temporal prediction as well. However, generic spatio-temporal prediction models often struggle to achieve optimal results across all domains. Therefore, some studies have designed specific models for different domains, such as precipitation prediction [8,9,22].

2.2 Satellite Image Sequence Prediction

As a branch of spatio-temporal prediction, satellite image sequence prediction also has models designed for it. Lee *et al.* [7] design a network that can fully utilize the characteristics of multichannel satellite images. Xu *et al.* [21] propose a network combining the generating ability of the GAN with the forecasting ability of the LSTM network. Dai *et al.* [2] propose a network with a multiscale generator with axial attention and a temporal discriminator to address the blurry issue and maintain the motion consistency. Dai *et al.* [1] design a network that can maintain spatial-temporal consistency with a multilevel motion memory-based predictor and a time-variant frame discriminator.

Previous works often neglect the diversity of satellite image changes and rarely integrate multi-modal data to improve prediction accuracy. While [7] integrate meteorological data to support critic networks in discerning network-generated images during training, none of the existing methods, including [7], utilize meteorological data directly to enhance prediction during inference. Additionally, none of these methods consider the capture time.

3 Method

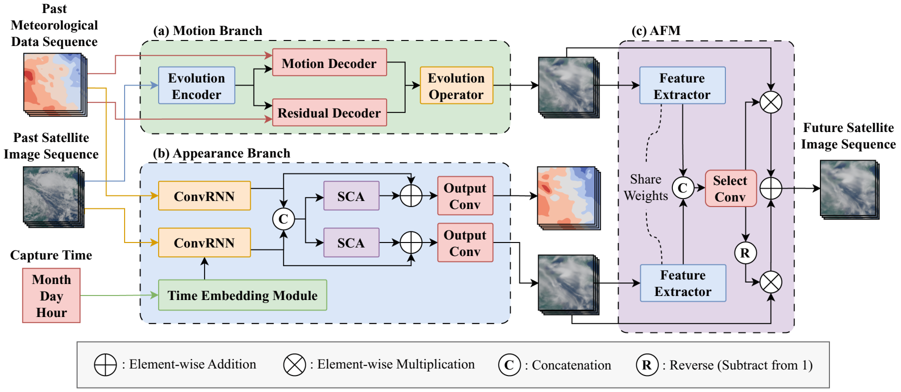


Fig. 2. The overall architecture of MMSISP. We first train the first stage of the network, including the motion branch predicting cloud movement, and the appearance branch predicting cloud variations and brightness change. Then, we freeze both branches and train the AFM in the second stage to adaptively fuse the predictions.

The formal definition of the satellite image sequence prediction problem is as follows: given the past frames $\mathbf{X}_P = \{X_0, X_1, ..., X_{T_P-1}\} \in \mathbb{R}^{T_P \times C \times H \times W}$, the model need to predict the future frames $\mathbf{X}_F = \{X_{T_P}, X_{T_P+1}, ..., X_{T_P+T_F-1}\} \in \mathbb{R}^{T_F \times C \times H \times W}$.

Based on the characteristic that satellite image changes have multiple factors, we design a two-stage network as shown in Fig.2. The first stage is divided into two branches: a motion branch and an appearance branch. The motion branch takes images and meteorological data as inputs and is responsible for predicting cloud movement. The appearance branch, on the other hand, takes images, capture time, and meteorological data as inputs and is tasked with predicting cloud variations and brightness change. In the second stage, we use an AFM with gating mechanism to adaptively fuse the predictions from two branches and obtain the final prediction. In order to reduce the difficulty of network optimization, two stages are trained separately.

3.1 Motion Branch

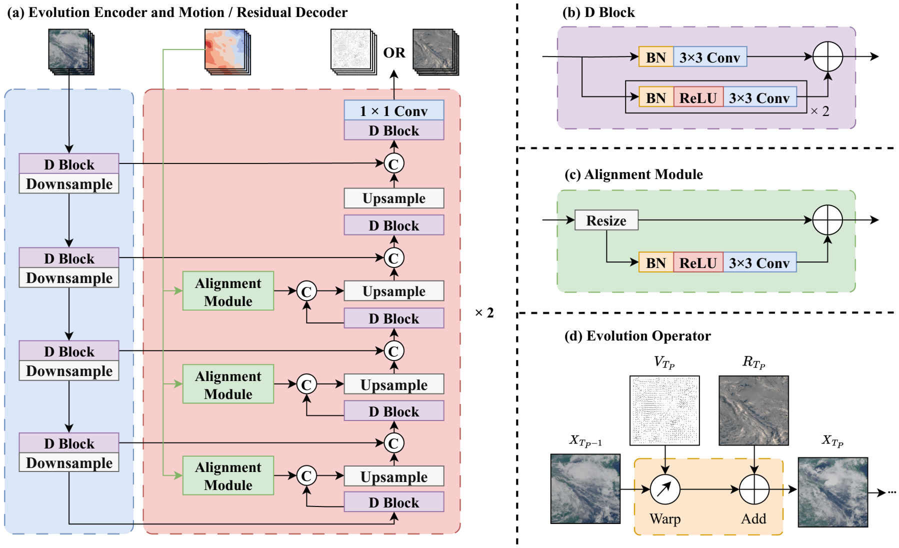


Fig. 3. The components of the motion branch: (a) evolution encoder, motion encoder and residual decoder; (b) D block; (c) our designed alignment module; (d) evolution operator.

Evolution network [22] is a network that produces mesoscale precipitation forecasts. Due to the similarity between the motion patterns of clouds and precipitation fields, we apply the evolution network to model the cloud movement and design an alignment module to incorporate meteorological data into it.

As shown in Fig.3, our motion branch use an evolution encoder to extract features, a motion decoder to predict cloud movement, and a residual decoder to recover the background. The evolution encoder consists of four D blocks, each followed by a max-pooling layer for downsampling by a factor of two. The motion decoder and the residual decoder share the same architecture, consisting of five D Blocks, with bilinear interpolation for upsampling by a factor of two between every two blocks. Skip connections in the form of concatenation exist between the encoder and decoder.

We incorporate three alignment modules in each decoder to align the resolution of meteorological data with the image feature maps at each layer. Specifically, after resizing the meteorological data to the same size using bilinear interpolation, we apply a simple transformation to the meteorological data using a convolutional layer with batch normalization and ReLU, followed by residual connection.

The evolution encoder takes the past satellite images $\mathbf{X_P} \in \mathbb{R}^{(T_P \times C) \times H \times W}$ as input, while the motion decoder and residual decoder respectively output the motion fields $\mathbf{V} \in \mathbb{R}^{(T_F \times 2) \times H \times W}$ and the residual fields $\mathbf{R} \in \mathbb{R}^{(T_F \times C) \times H \times W}$. On the basis of the last past frame, the evolution operator recursively predicts the next frame. Specifically, it uses the predicted motion field for each time step to perform a warp operation [6] to simulate cloud movement. Then, it adds the residual field for each time step to recover the background obscured by clouds.

The evolution network is only suitable for predicting the cloud movement. Without constraining the motion field, cloud variations will make the model training unstable. Therefore, a motion regularization term is used to exclude the influence of cloud variations, making the motion field smoother:

$$\mathbf{L}_{reg} = \frac{\sum_{t=T_P}^{T_P+T_F-1} (\parallel \nabla V_t^1 \parallel_2^2 + \parallel \nabla V_t^2 \parallel_2^2)}{T_F} \tag{1}$$

in which V_t^1 and V_t^2 are the two components of each motion field V_t. The gradients of the motion field components ∇V_t^1 and ∇V_t^2 are computed approximately using the sobel filter.

The complete loss function of the motion branch is:

$$\mathbf{L}_m = MAE(\mathbf{X}_F, \hat{\mathbf{X}}_F^m) + \lambda_{reg} \cdot \mathbf{L}_{reg} \tag{2}$$

where $\mathbf{X}_F$ is the ground truth of future frames, $\hat{\mathbf{X}}_F^m$ is the prediction of the motion branch, and λ_{reg} is a hyperparameter used to adjust the influence of the motion regularization term.

3.2 Appearance Branch

Due to the inherent limitations of evolution network in predicting cloud variations and brightness change, we design an appearance branch. This branch consists of two internal branches dedicated to modeling satellite images and meteorological data, respectively. Each internal branch comprises a ConvRNN, an SCA module, and a 1×1 convolution for generating outputs. The ConvRNN used for satellite images also includes a time embedding module. In this section, we will provide a detailed introduction to each component of the appearance branch.

ConvRNN To fully capture the spatio-temporal dependencies in satellite images and meteorological data, we use ConvRNNs to model them separately. We validate the effectiveness of our architecture by implementing two variants, each utilizing a different ConvRNN. The variant using ConvLSTM is denoted as MMSISP-C, while the variant using PredRNN is denoted as MMSISP-P.

Time Embedding Module We design a lightweight time embedding module, as shown in Fig.4, which can realize the introduction of image capture time using only a few parameters.

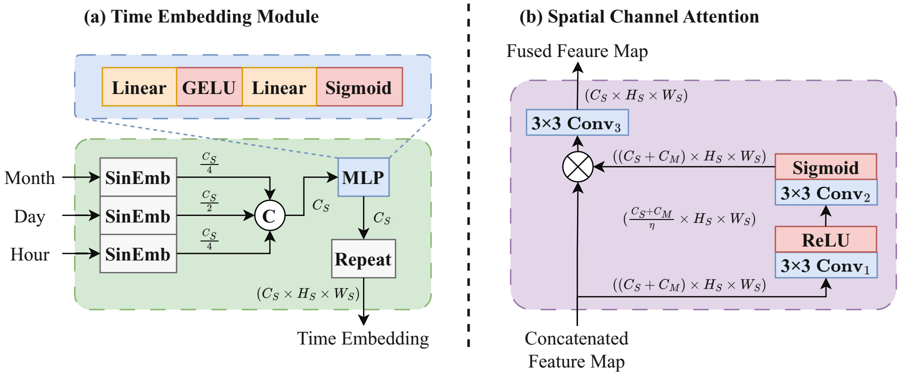

(a) Time Embedding Module
Linear
GELU
Linear
Sigmoid
Month
Day
Hour
SinEmb
SinEmb
SinEmb
$\frac{C_S}{4}$
$\frac{C_S}{2}$
$\frac{C_S}{4}$
C
C_S
MLP
C_S
Repeat
$(C_S \times H_S \times W_S)$
Time Embedding
(b) Spatial Channel Attention
Fused Feaure Map
$(C_S \times H_S \times W_S)$
3×3 Conv$_3$
$((C_S + C_M) \times H_S \times W_S)$
Sigmoid
3×3 Conv$_2$
$(\frac{C_S+C_M}{\eta} \times H_S \times W_S)$
ReLU
3×3 Conv$_1$
$((C_S + C_M) \times H_S \times W_S)$
Concatenated
Feature Map

Fig. 4. Some components of the appearance branch: (a) time embedding module; (b) SCA for ConvRNN applied to satellite images. C_S and C_M represent the number of hidden state channels for ConvRNN applied to satellite images and meteorological data, respectively. H_S and W_S denote the height and width of the hidden state for ConvRNN applied to satellite images, respectively.

Firstly, we employ the sinusoidal embedding [16] to transform the numerical values of month, day, and hour into vectors:

$$\begin{aligned} \mathbf{SE}(t, 2i) &= \sin(t/10000^{2i/D}) \\ \mathbf{SE}(t, 2i+1) &= \cos(t/10000^{2i/D}) \end{aligned} \tag{3}$$

where t represents the month, day, or hour when the first past frame was captured, i denotes the index of an element in the vector, and D represents the length of the vector.

Since we only use daytime images, with 12 possible values for months, 31 for days, and 8 for hours, we set the length of the day vector $D_d = C_S/2$ and the lengths of the month and hour vectors $D_m = D_h = C_S/4$, where C_S represents the number of hidden state channels of the ConvRNN used for satellite images.

Next, we concatenate the three vectors into a single vector of length C_S and use an MLP to obtain a joint time embedding:

$$\mathbf{TE} = Sigmoid(\mathbf{W}_2(GELU(\mathbf{W}_1([\mathbf{SE}_m, \mathbf{SE}_d, \mathbf{SE}_h])))) \tag{4}$$

Finally, we repeat the joint time embedding of length C_S, transforming it into the shape of (C_S, H_S, W_S), where H_S and W_S denote the height and width of the hidden state, respectively. The final time embedding will serve as the initial hidden state for the ConvRNN used for satellite images, which was originally initialized as all zeros.

Spatial Channel Attention After extracting features from satellite images and meteorological data using ConvRNN separately, it is crucial to perform effective feature fusion. Concatenation followed by convolution is a straightforward fusion method, but it overlooks the varying importance of different channels in the feature map and different pixels within each channel. We use the SCA module to address this issue, emphasizing important elements in the feature map and thereby achieving better predictions.

Taking the SCA for ConvRNN used for satellite images as an example, assuming C_M represents the number of hidden state channels of the ConvRNN used for meteorological data, and η is the reduction ratio, we first concatenate the feature maps of the images and meteorological data at each time step into a single feature map $\mathbf{Z}_{concat} \in \mathbb{R}^{(C_S+C_M)\times H_S \times W_S}$. The number of channels in the concatenated feature map is subsequently reduced to $(C_S + C_M)/\eta$ by $\mathbf{Conv}_1$, and then expanded back to $C_S + C_M$ by $\mathbf{Conv}_2$. Subsequently, the importance of each element in the feature map $\mathbf{A} \in \mathbb{R}^{(C_S+C_M)\times H_S \times W_S}$ is obtained using the sigmoid function:

$$\mathbf{A} = \sigma(\mathbf{Conv}_2(ReLU(\mathbf{Conv}_1(\mathbf{Z}_{concat})))) \tag{5}$$

Finally, the concatenated feature map $\mathbf{Z}_{concat}$ is element-wise multiplied with $\mathbf{A}$ and passed through $\mathbf{Conv}_3$ to obtain the final feature map $\mathbf{Z}_{SCA} \in \mathbb{R}^{C_S \times H_S \times W_S}$:

$$\mathbf{Z}_{SCA} = \mathbf{Conv}_3(\mathbf{A} \otimes \mathbf{Z}_{concat}) \tag{6}$$

Loss Fuction In addition to using the prediction loss of satellite images, adding a prediction loss of meteorological data might provide constraints on meteorological aspects, enhancing the robustness of the model. However, due to the noise contained in the interpolated meteorological data, excessive constraints may affect the performance of the model. Therefore, we use a cosine decay factor α from 1 to 0 as the weight of the prediction loss of the meteorological data.

The complete loss function of the appearance branch is:

$$\mathbf{L}_a = MSE(\mathbf{X}_F, \hat{\mathbf{X}}_F^a) + \alpha \cdot MSE(\mathbf{M}_F, \hat{\mathbf{M}}_F) \tag{7}$$

$$\alpha = 0.5 \times (1 + \cos(\frac{\pi \times current\_iter}{total\_iters})) \tag{8}$$

where $\mathbf{X}_F$ is the ground truth of future satellite images, and $\hat{\mathbf{X}}_F^a$ is the prediction of the appearance branch. $\mathbf{M}_F$ is the ground truth of future meteorological data, and $\hat{\mathbf{M}}_F$ is the predicted values of future meteorological data.

3.3 Adaptive Fusion Module

After separately training the motion branch and appearance branch in the first stage, we freeze the parameters of both branches. Subsequently, we use MAE loss to train the AFM, enabling adaptive fusion of predictions from both branches.

In the AFM, the predictions from different branches are respectively passed through the same feature extractor. It consists of three parallel convolutional layers with batch normalization and ReLU. The convolutional layers have kernel sizes of 3×3, 5×5, and 7×7 respectively. Then, the feature maps of the two branches are concatenated, and the select gate is obtained using a select convolution operation with a sigmoid function. Ultimately, the final prediction is obtained using the select gate.

The above process can be represented by the following formula:

$$\hat{\mathbf{X}}_F = \mathbf{G} \otimes \hat{\mathbf{X}}_F^m + (1 - \mathbf{G}) \otimes \hat{\mathbf{X}}_F^a \tag{9}$$

$$\mathbf{G} = Sigmoid(\mathbf{Conv}([\mathbf{Extractor}(\hat{\mathbf{X}}_F^m), \mathbf{Extractor}(\hat{\mathbf{X}}_F^a)])) \tag{10}$$

where $\otimes$ is the element-wise multiplication and $\mathbf{G}$ is the select gate.

4 Experiments

4.1 Datasets

Satellite Images The satellite images used in this work are three-channel true-color images from the Himawari-8 satellite, synthesized from channels 1, 2, 3 and 4. Afterward, a specific geographical region (28.7°N - 41.5°N, 116.2°E - 129.0°E) was cropped and resized to a size of 3×256×256 pixels. Due to the absence of visible channel during nighttime, only daytime images within the interval of UTC 00:00 - 08:00 were included in the dataset, with any instances of missing data being excluded. The prediction task is to predict 8 future frames based on 4 past frames, with a time interval of 30 minutes between each frame. To facilitate model training and validation, data spanning from 2017 to 2020 were partitioned into training and validation sets in a randomized manner, adhering to a ratio of 4:1. Consequently, the resulting sets consisted of 5208 samples for training and 1308 samples for validation. Subsequently, data from the year 2021 were designated as the test set, comprising 1236 samples.

Meteorological Data The meteorological data used in this work are extracted from ERA5 hourly data on pressure levels. This encompasses the u-component of wind, v-component of wind, temperature, and relative humidity at 250hPa, 500hPa, 850hPa, and 1000hPa, covering the same spatial and temporal domain as the satellite images. The original spatial resolution of the meteorological data is 0.25 degrees, with a temporal resolution of 1 hour. Later, the data were interpolated to a size of 16×64×64 pixels, resulting in a spatial resolution of 0.2 degrees, to match the images with a patch size of 4. Additionally, the time intervals were also interpolated to 30 minutes.

4.2 Implementation Details

In the motion branch, the base hidden layer dimension of the evolution network is set to 32, and λ_{reg} is set to 0.01. In the appearance branch, the ConvRNN used for satellite images has a patch size of 4, a convolutional kernel size of 5, a hidden layer dimension of 128, and 4 layers. The ConvRNN used for meteorological data has a patch size of 1, a convolutional kernel size of 5, a hidden layer dimension of 32, and 4 layers. The reduction ratio η of SCA is set to 16. In the AFM, the hidden layer dimension is set to 8.

All experiments were conducted using PyTorch on a single NVIDIA RTX 3090. Both the motion branch and the appearance branch were trained for 50 epochs, and the AFM was trained for 5 epochs, all with a batch size of 8. Throughout the training processes, the Adam optimizer was employed with an initial learning rate of 0.0001, complemented by a cosine learning rate scheduler.

Table 1. Quantitative comparison of state-of-the-art methods and our method. MMSISP-C is based on ConvLSTM and MMSISP-P is based on PredRNN. AB means the appearance branch.

Model	MAE ↓	MSE ↓	SSIM ↑	PSNR ↑	LPIPS ↓	Param	FLOPs
Recurrent-based							
ConvLSTM [10]	9691.29	1067.43	0.7796	23.77	0.6267	15.497M	0.545T
PredRNN [18]	8873.17	946.11	0.7927	24.36	0.5207	24.56M	1.107T
PredRNN++ [17]	8823.37	958.33	0.7919	24.33	0.5298	39.305M	1.623T
MIM [19]	8752.85	926.45	**0.7954**	24.40	0.4750	47.612M	1.675T
PhyDNet [4]	11934.64	1405.71	0.7541	22.28	0.6456	3.093M	0.142T
MotionRNN [20]	9191.81	996.42	0.7937	24.20	0.5122	7.215M	0.319T
SwinLSTM-D [14]	9581.99	1106.13	0.7706	23.59	0.5399	20.208M	0.162T
Recurrent-free							
SimVP [3]	9812.64	1099.58	0.7806	23.58	0.5846	39.429M	0.19T
TAU [13]	9139.03	1032.73	0.7857	23.91	0.5408	37.647M	0.182T
MMVP [23]	10837.91	1287.74	0.7414	22.71	0.7369	10.781M	0.375T
Generative							
DGMR [9]	14441.74	2136.83	0.6585	20.48	0.4726	53.577M	0.118T
LDCast [8]	15114.14	2416.43	0.6857	20.13	**0.4059**	670.8M	44.532T
Ours							
MMSISP-C (AB)	8937.79	968.23	0.7850	24.17	0.5725	16.595M	0.583T
MMSISP-C	8703.77	956.78	0.7891	24.16	0.4899	27.215M	0.617T
MMSISP-P (AB)	8661.20	921.32	0.7926	24.46	0.5236	26.234M	1.181T
MMSISP-P	**8549.25**	**918.82**	0.7937	**24.47**	0.4854	36.854M	1.214T

4.3 Comparisons to State-of-the-Art Methods

We compare our method with existing spatio-temporal prediction methods, including recurrent-based methods, recurrent-free methods and generative methods. Table 1 shows that our method is superior to other methods in terms of most metrics.

MMSISP-P surpasses all methods in MAE, MSE, and PSNR, with a notable 203.6 reduction in MAE compared to MIM, indicating a significant improvement in accuracy. Despite a slightly lower SSIM compared to MIM, it boasts a 23% reduction in parameters. The two generative models exhibit the lowest LPIPS, suggesting their predictions closely resemble ground truth in human visual perception. However, visualization in section 4.5 reveals that while LDCast's predictions are clear, noticeable differences in cloud positions from ground truth lead to high MAE.

Additionally, even when using only the appearance branch of MMSISP, there is a significant improvement. Compared to ConvLSTM and PredRNN, the appearance branches of MMSISP-C and MMSISP-P respectively increase the parameter count by 1.098M and 1.674M, yet reduce MAE by 753.5 and 211.97, demonstrating the accuracy boost from incorporating capture time and meteorological data.

4.4 Ablation Study

Motion Branch As shown in Table 2, for the motion branch, both the residual field and the motion regularization term are important. The introduction of meteorological data also improves accuracy.

Without the residual field, the background obscured by clouds in the last frame cannot be recovered and the model tends to copy the content of the last frame, resulting in a 5289.01 increase in MAE. Without the motion regularization term, the model struggles with optimization due to unsmooth motion fields caused by cloud variations, leading to a 522.87 increase in MAE. Introducing meteorological data through the alignment module assists the model in inferring cloud movement, reducing MAE by 21.05, demonstrating the efficacy of using meteorological data.

Table 2. Ablation study for the motion branch of MMSISP-C.

	MAE ↓	SSIM ↑	Param
w/o residual field	14395.23	0.6970	6.979M
w/o motion regularization	9629.09	0.7447	10.508M
w/o alignment module	9127.27	0.7828	8.864M
Motion Branch	**9106.22**	**0.7830**	10.508M

Appearance Branch As shown in Table 3, we conduct an ablation study on the appearance branch of MMSISP-C. Specifically, we remove the time embedding module and various components of the spatio-temporal multi-modal fusion mechanism individually to validate their effectiveness.

The time embedding module only needs 0.033M more parameters to reduce MAE by 576.84. This demonstrates that incorporating the capture time enhances the model's ability to predict cloud variations and brightness change. The effectiveness of all components in the spatio-temporal multimodal fusion mechanism has been validated, including the ConvRNN for meteorological data, SCA, and the cosine decay factor in the loss function, all of which contribute to improved accuracy. Together, they add 1.065M parameters but reduce the MAE by 427.3.

Table 3. Ablation study for each component in the appearance branch of MMSISP-C, including time embedding module (TEM), ConvRNN for meteorological data (CMD), spatial channel attention (SCA), and cosine decay factor (CDF).

TEM	CMD	SCA	CDF	MAE ↓	SSIM ↑	Param
				9691.29	0.7796	15.497M
✓				9114.45	0.7828	15.53M
	✓			9437.25	0.7823	16.505M
	✓	✓		9308.07	0.7805	16.562M
	✓		✓	9433.59	0.7829	16.505M
	✓	✓	✓	9263.99	0.7812	16.562M
✓	✓	✓	✓	**8937.79**	**0.7850**	16.595M

Adaptive Fusion Module The fusion methods for the predictions of the two branches can be divided into image-level fusion and pixel-level fusion. In image-level fusion, which involves techniques like averaging two images or using an AFM with global average pooling, the same weight is assigned to different pixels within the same image. Conversely, in pixel-level fusion, such as the AFM, different weights are assigned to different pixels of the same image.

As shown in Table 4, compared to the image-level fusion methods, the AFM has lower MAE and higher SSIM. Compared to averaging two images and using an AFM with global average pooling, the MAE decreased by 27.62 and 35.87, respectively.

4.5 Visualization

We select samples from two scenarios to demonstrate the accuracy and robustness of our method. Fig.5 illustrates that in a scenario with rapid cloud movement, our method can generate the most accurate long-term predictions. In contrast, other methods have misjudged the final position of the cloud. Fig.6

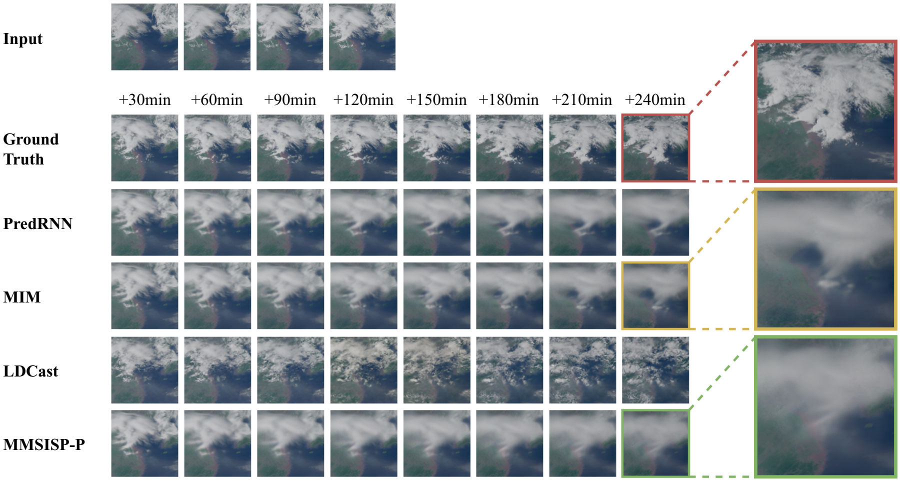


Fig. 5. Visualization of predictions in a scenario with rapid cloud movement.

Table 4. Ablation study for the AFM of MMSISP-C. GAP represents global average pooling.

	MAE ↓	SSIM ↑	Param
Image-Level (1:1)	8731.39	0.7886	0
Image-Level (AFM + GAP)	8739.64	0.7886	2.1K
Pixel-Level (AFM)	**8703.77**	**0.7891**	2.1K

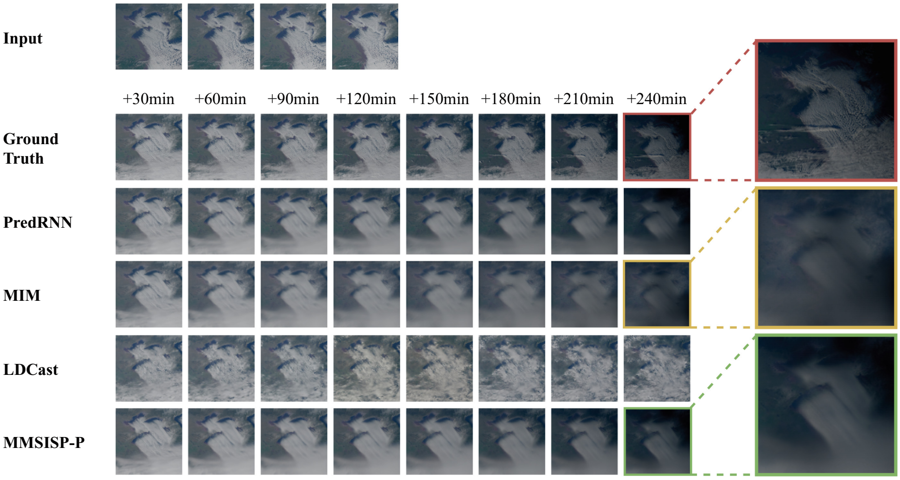


Fig. 6. Visualization of predictions in a scenario with significant brightness change.

demonstrates that in a scenario with significant change in brightness, our method exhibits strong robustness. Due to the lack of capture time information, the prediction result of MIM tends to be brighter. In contrast, our method, which incorporates capture time, accurately predicts the brightness of the final frame.

Additionally, from the two figures, it can also be observed that generative models like LDCast can generate the clearest predictions, but they fail to make accurate predictions for cloud movement, cloud variations and brightness change.

5 Conclusion

To address the issue of low accuracy in long-term predictions in satellite image sequence prediction, we decouple various changing factors and propose a dual-branch method. Additionally, we design a lightweight time embedding module to utilize the capture time and different spatio-temporal multi-modal fusion mechanisms for the two branches to leverage meteorological data, thereby assisting in the inference of satellite image changes. Through experiments, our method demonstrate superior accuracy compared to state-of-the-art methods.

However, the proposed method still has certain limitations. Firstly, although it shows an improvement in accuracy compared to other methods, there is no significant enhancement in the clarity of the generated results. Secondly, the meteorological data used is reanalysis data, which, despite having a more uniform spatial distribution than observational data, cannot be obtained in real-time. In future work, we plan to introduce multi-factor decoupling and multi-modal fusion into the generative model to improve clarity, use real-time observational meteorological data to enhance the method's practicality, and conduct experiments on satellite images from more regions to validate the model's generalization capability.

References

1. Dai, K., Li, X., Ma, C., Lu, S., Ye, Y., Xian, D., Tian, L., Qin, D.: Learning spatial-temporal consistency for satellite image sequence prediction. IEEE Transactions on Geoscience and Remote Sensing (2023)
2. Dai, K., Li, X., Ye, Y., Feng, S., Qin, D., Ye, R.: Mstcgan: Multiscale time conditional generative adversarial network for long-term satellite image sequence prediction. IEEE Trans. Geosci. Remote Sens. **60**, 1–16 (2022)
3. Gao, Z., Tan, C., Wu, L., Li, S.Z.: Simvp: Simpler yet better video prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3170–3180 (2022)
4. Guen, V.L., Thome, N.: Disentangling physical dynamics from unknown factors for unsupervised video prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11474–11484 (2020)

5. Hirpa, F.A., Hopson, T.M., De Groeve, T., Brakenridge, G.R., Gebremichael, M., Restrepo, P.J.: Upstream satellite remote sensing for river discharge forecasting: Application to major rivers in south asia. Remote Sens. Environ. **131**, 140–151 (2013)
6. Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial intelligence **17**(1–3), 185–203 (1981)
7. Lee, J.H., Lee, S.S., Kim, H.G., Song, S.K., Kim, S., Ro, Y.M.: Mcsip net: Multichannel satellite image prediction via deep neural network. IEEE Trans. Geosci. Remote Sens. **58**(3), 2212–2224 (2019)
8. Leinonen, J., Hamann, U., Nerini, D., Germann, U., Franch, G.: Latent diffusion models for generative precipitation nowcasting with accurate uncertainty quantification. arXiv preprint arXiv:2304.12891 (2023)
9. Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., Fitzsimons, M., Athanassiadou, M., Kashem, S., Madge, S., et al.: Skilful precipitation nowcasting using deep generative models of radar. Nature **597**(7878), 672–677 (2021)
10. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems **28** (2015)
11. Shukla, B.P., Kishtawal, C.M., Pal, P.K.: Prediction of satellite image sequence for weather nowcasting using cluster-based spatiotemporal regression. IEEE Trans. Geosci. Remote Sens. **52**(7), 4155–4160 (2013)
12. Son, Y., Zhang, X., Yoon, Y., Cho, J., Choi, S.: Lstm-gan based cloud movement prediction in satellite images for pv forecast. J. Ambient. Intell. Humaniz. Comput. **14**(9), 12373–12386 (2023)
13. Tan, C., Gao, Z., Wu, L., Xu, Y., Xia, J., Li, S., Li, S.Z.: Temporal attention unit: Towards efficient spatiotemporal predictive learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18770–18782 (2023)
14. Tang, S., Li, C., Zhang, P., Tang, R.: Swinlstm: Improving spatiotemporal prediction accuracy using swin transformer and lstm. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13470–13479 (2023)
15. Valada, A., Mohan, R., Burgard, W.: Self-supervised model adaptation for multimodal semantic segmentation. Int. J. Comput. Vision **128**(5), 1239–1285 (2020)
16. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
17. Wang, Y., Gao, Z., Long, M., Wang, J., Philip, S.Y.: Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In: International Conference on Machine Learning. pp. 5123–5132. PMLR (2018)
18. Wang, Y., Long, M., Wang, J., Gao, Z., Yu, P.S.: Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. Advances in neural information processing systems **30** (2017)
19. Wang, Y., Zhang, J., Zhu, H., Long, M., Wang, J., Yu, P.S.: Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9154–9162 (2019)
20. Wu, H., Yao, Z., Wang, J., Long, M.: Motionrnn: A flexible model for video prediction with spacetime-varying motions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15435–15444 (2021)

21. Xu, Z., Du, J., Wang, J., Jiang, C., Ren, Y.: Satellite image prediction relying on gan and lstm neural networks. In: ICC 2019-2019 IEEE international conference on communications (ICC). pp. 1–6. IEEE (2019)
22. Zhang, Y., Long, M., Chen, K., Xing, L., Jin, R., Jordan, M.I., Wang, J.: Skilful nowcasting of extreme precipitation with nowcastnet. Nature **619**(7970), 526–532 (2023)
23. Zhong, Y., Liang, L., Zharkov, I., Neumann, U.: Mmvp: Motion-matrix-based video prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4273–4283 (2023)

Leveraging Dual Encoders with Feature Disentanglement for Anomaly Detection in Thermal Videos

Divya Bhardwaj, Anishka Singh, Sparsh Goenka, and Poonam Goyal(✉)

Birla Institute of Technology and Science, Pilani Campus,
Vidya Vihar, Pilani 333031, Rajasthan, India
{p20180013,f20200816,f20212413,poonam}@pilani.bits-pilani.ac.in
https://web.bits-pilani.ac.in/

Abstract. Anomaly detection is critical for real-time applications, e.g., monitoring elderly people or kids from a remote place; gas leakage detection, night vision surveillance, etc. Detecting anomalous behavior becomes even more challenging when the device used for capturing scenes is the thermal camera. The thermal videos have the ability to preserve the identity of the subjects involved in the scenes. The info-deficit nature of thermal imagery, i.e., lack of texture, contours, and colors, makes it difficult to fetch the salient details required to differentiate between normal and abnormal events. Most approaches for anomaly detection in videos explicitly model regions of interest (ROIs). However, this modeling poses limitations of accurate RoI detection more in thermal videos when the size of ROIs is smaller than the size of the frame. Moreover, the techniques, that take advantage of corresponding visible videos to detect anomalies in thermal videos, have a limitation of requiring twin videos. To address these limitations, we present a frame-level unsupervised approach that learns two sets of features from two different encoders in a disentangled fashion. The learning objectives of the proposed approach is aggregation of reconstruction error of the middle frame and disentanglement error between two encodings. We perform extensive experiments on two benchmark thermal video datasets, Thermal Rare Event and TSF. The proposed approach outperforms state-of-the-art models for anomaly detection from visible and thermal spectrum.

Keywords: Convolution Neural Networks · Anomaly Detection · Thermal Imagery · Visible Imagery

1 Introduction

Anomaly detection aims to identify abnormal events that are defined by situations. In many cases, it becomes highly challenging, yet subjective in nature, to decide that an event is normal or anomalous. For example, vehicles on a walkway or people walking on a runway. To counter these ambiguities, most existing learning methods [4,20,39] leverage the finer details in the visible spectrum with

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 237–253, 2025.
https://doi.org/10.1007/978-3-031-78312-8_16

details like fine-grained texture and color, etc. However, working with the visible spectrum always carries privacy concerns for the subjects. Also, the need for illumination sources for capturing scenes limits its applicability for applications requiring operability for 24 × 7 and in dark places. Thermal imagery addresses privacy concerns and dependency on illumination sources. Moreover, thermal imaging applications for anomaly detection may arise where the ROI is based on the emission of heat energy. For example, crack detection [30], railway inspection [2], and industrial applications [1]. Other applications include healthcare applications [18], night vision surveillance [10], etc.

In the literature, anomaly detection has been attempted by both supervised learning and unsupervised learning methods. Supervised learning methods need annotated data, which requires manual effort and time [3,17,18,21,25,28,38]. Moreover, abnormal activity detection is a rare event, and thus capturing enough instances of anomalies becomes a tedious task. Annihilating the need for labeled datasets, several methods have been proposed for anomaly detection [11,20,33,36] that utilize unsupervised paradigm. Particularly, these methods use data with normal events for training and evaluate on data having both normal and abnormal events. This eradicates the need for labeled datasets resulting the unsupervised approach a natural choice. Therefore, we also follow the unsupervised approach for **A**nomaly **D**etection using **T**hermal **I**magery (ADTI).

The earlier unsupervised approaches for ADTI [1,2] are based on the image processing methods that require preprocessing the thermal images *e.g.* computing 3D matrix, geometry of objects in the thermal images for anomaly detection. Some recent researches [23,33] use thermal images in pair with visible images; which assists in better anomaly detection in thermal images by perceiving the details from visible images. Few unsupervised approaches [8,10,11,27] explicitly model ROIs/objects based on their appearance and motion characteristics using object and flow detectors. However, explicit modeling of objects limits anomaly detection in cases where ROIs are of very small regions compared to the frame. Apart from approaches designed for thermal images, various researchers have worked on **A**nomaly **D**etection using **V**isible **I**magery (ADVI) [12,13,20,24,31,36] which can be applied on thermal images to evaluate their performance on ADTI.

We propose an approach having `Dual Encoders` with `Feature Disentanglement Network` (**DEFD-Net**) which inherently learns the salient features in thermal videos by disentangling two different set of features from two encoders. Our approach also works well in the aforementioned limiting situations.

We use prediction of the middle frame from a set of consecutive input frames in an encoder-decoder setup and utilize 2D Convolution Neural Network (CNN) and 3D CNN encoders for feature extraction. However, the simple ensemble of features of two encoders may learn common features, affecting the perceptual quality of the predicted frame. Our feature distinctiveness approach allows the encoders to learn exclusive features and thus represent better information for precisely predicting the middle frame. We also exploit the intermediate representations of pose estimation network [5], Part Affinity Fields (PAFs) and Heatmaps

(HMs) to enhance the salient areas of the thermal images. The proposed DEFD-Net outperforms the existing best-published works [12,13,20,24,31,36] of ADVI significantly on TSF [35] and Thermal Rare Event (TRE) [26] thermal datasets. The salient features of our approach are as follows:

1. We propose a simple yet effective approach, called DEFD-Net, that exploits distinct features using a feature disentanglement approach. Our novel idea of using feature disentanglement technique for ADTI helps in understanding details from thermal images precisely at a negligible cost.
2. We performed extensive experiments on two benchmarked thermal datasets namely, TSF and TRE, and compared the performance of our proposed approach with the state-of-the-art approaches for ADVI and ADTI and achieved a new state-of-the-art.

2 Related Work

The unsupervised learning approaches for anomaly detection can be categorized into: 1) object-level and 2) frame-level approaches. The object-level anomaly detection approaches [12,13,27] enables the exact localization of anomalies in an image. But these approaches are limited by the performance of object detection algorithms [12,27]. This becomes even more challenging with thermal images, where recognizing the object is hard because of the fewer details. The frame-level approaches [24,26,31,36] consider a frame as anomalous even if only one pixel is abnormal. Georgescu et al. [12] has also presented their results using the fusion of object and frame level anomaly scores based on late-fusion strategy.

Anomaly Detection in Thermal Imagery. Benmoussat et al. [1] identify the anomaly as the deviation from the background, from a statistical point of view. They claimed that the calibrated thermal images after denoising and dimensionality reduction show a better performance. The need for calibration of three thermal devices and use of specialized detectors limits its applicability. A method in [2] collected thermal video from the moving train and detected anomalous objects around rails by thresholding geometric properties. A binary mask with the thermal image is used for the correction to improve anomaly detection. However, this method works only for the application considered. Gasparini et al. proposed a neural network to inspect the railway at night using thermal videos. The method required objects to be annotated for training involving manual annotations [10]. Fall detection as the anomaly detection task is addressed by [8]. They built a spatio-temporal residual autoencoder that takes ROI from a temporal sliding window method as the input. The explicit extraction of temporal information through LSTM along with spatial information using ConvLSTM boosts the model's performance. While, Mehta et al. address fall detection using an adversarial approach that reconstructs a thermal frame and optical flow. They computed the ROI of a person using R-FCN in conjunction with contour box localization using Ostu thresholding and tracking using Kalman filtering. The fusion of flow-ROI and the thermal image-ROI yielded a better AUC [27]. All the above-mentioned models are based on the assumption that foreground objects are brighter than the background. These methods require the use of object detectors or trackers or flow detectors [9] for extracting salient features from thermal videos. However, in thermal modality, detecting objects becomes challenging in adverse weather and lightning conditions e.g., detecting a person under bright sun.

Few approaches use visible and thermal images in pair to improve anomaly detection. Lile et al. used VGG-16 to predict the thermal frame from its corresponding visible frame. The predicted image was majorly blurred, and anomalous regions were deducted as hotspots, confirming the low quality of the generated image [23]. The methods [33,34] detect anomalies using paired images for monitoring District Heating Systems. Sledz et al. fused the information of salient maps of both images using Dempster-Shafer evidence theory [33]. Another study detects the anomalies as high-temperature areas using Laplacian of Gaussian blob detector. The method then performs segmentation of ROIs and classification and localization of anomalies [34]. The use of handcrafted features lead to degraded generalization capabilities of the method. These fusion models require multimodal paired dataset which is an overhead of capturing scenes from both cameras with the same viewpoint.

SSMCTB [26] is a transformer block which can be integrated with any DL model to generate powerful features using 3D masked convolution and channel attention. This block is applied on TRE dataset with MNAD [31] and showed enhanced performance [26].

Anomaly Detection in Visible Imagery. Recently, He et al. proposed a low cost model which works on compressed videos for anomaly detection. This allows the use of the model directly on edge devices [14]. A two-stream framework is proposed which utilizes the motion information for understanding the local context of abnormal events along with the consistency between the testing event and the knowledge accquired from normal events of the training data. The spatio-temporal U-Net proposed by this method leverages the temporal information for predicting the future frame without using explicit optical flow estimation network [4]. Zhou et al. developed a multiscale flow-based method (MSFlow) to detect varying-sized anomalous objects. MSFlow fuses the flow information at different scales with the input frames and computes the anomaly score using the log-likelihood [39]. Hong et al. mention that powerful deep networks can also generate anamalous frame because of their generalization capabilities. Therefore, distinguishing between normal and abnormal frames based on the reconstruction/prediction error is difficult [15].

Many approaches for ADVI in the unsupervised setting use object-level approaches [12,13] and frame-level approaches [20,24,31,36]. Georgescu et al. utilized multiple proxy tasks by an object-centric approach [12] using YOLOV3. Authors in [24,31,36] exploit the memory module for learning the normalities from training data. Particularly, [31] uses two separate approaches: reconstructing the input frame and predicting the future frame with reconstruction, compactness and, seperatedness loss between feature maps and memory items. Liu et al. proposed a multi-level memory-augmented mechanism to reconstruct optical flow. These reconstructed optical flows with the input video frames predict a future frame using conditional variational autoencoder [24]. Whereas Wang et al. address the limitation which occurs because of ignoring the relationship between normal and abnormal events. They proposed an autoencoder that learn motion and appearance features by building a bridge between normal and abnormal events using memory module [36].

In another work, the authors construct the pseudo-anomalous to overcome the deficit of anomalous data during training. These examples are used for training appearance and motion convolutional autoencoders [13]. ASTNet uses WiderResNet to extract the spatio-temporal features with a spatial and temporal branch. These features are concatenated and then passed to the decoder to predict the future frame [20]. Though these methods show a boost in performance but their computation time is high.

In contrast to thermal images, visible images are information rich which include color, texture, contours, etc. and lot of information in the background. According to Nikolov et al., the object detectors pre-trained on visible images do not work well on thermal images [29]. Also, the object-centric approach attained poor results in [26] for the thermal domain. Moreover, the flow-based methods do not generalize well with varying sizes of objects [39]. In view of above points we propose a frame prediction framework with a disentanglement approach capable of performing better for thermal images.

3 The Proposed Approach

3.1 Problem Statement.

Our proposed method, DEFD-Net, denoted by $\mathcal{F}$, takes a sequence of frames as input and predicts the middle frame as output. We denote the sequence of input frames as $S = \{f_{-i}, ...f_{-2}, f_{-1}, f_1, f_2...., f_i\}$, where i^{th} frame in input sequence is represented as f_i. We denote the middle frame as f_0 and its predicted frame as $\hat{f}_0$. The task of predicting the middle frame is defined as follows:

Given S, our proposed method $\mathcal{F}$ aims to predict the middle frame $\hat{f}_0$ as

$$\hat{f}_0 = \mathcal{F}(S). \tag{1}$$

3.2 Model Network.

DEFD-Net ($\mathcal{F}$) consists of two CNN encoders, a feature disentanglement module, the attention block, and a decoder. An overview of our proposed architecture is shown in Fig. 1. We present the detail of each sub-module as follows:

Dual Encoder. DEFD-Net consists of two different encoders namely 2D CNN (E_{2D}) and 3D CNN (E_{3D}). Since 2D CNN works on spatial dimension, *i.e.* width and height taking the time dimension (all frames) with the channel dimension of the image. On the other hand, 3DCNN learns features by keeping all frames separately on time dimension. We give input to E_{2D} as (B, (C × T), H, W) representing batch size, channel, temporal offset (number of input frames), height, and width of the frame, respectively. It is built using U-Net [32]. E_{3D} takes (B, C, T, H, W) as the input and comprises the deep wide network of [12]. We have modified the skip connections used in [32] as per our requirements. We concatenated the skip connections at three (384×384, 192×192, 96×96) resolutions of two encoders at channel dimension. These concatenated skip connections from three stages are sent to the decoder to preserve the high-frequency details in features while propagating through the network. This avoids the vanishing gradient problem, which helps more for thermal images as these have less information.

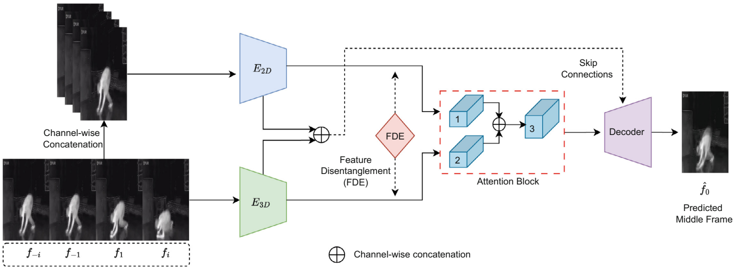


Fig. 1. The DEFD-Net Framework.

Part Affinity Fields and Heatmaps for Salient Information. We use a 2D pose estimation network [5], which estimates the pose of multiple people in a frame. We exploit the heatmaps to extract salient information and part affinity fields to fetch orientation characteristics from thermal images. The heatmaps are the confidence maps representing the probability of a body part at that pixel. PAFs are a set of 2D vector fields that encode the association between the body part locations. Particularly, PAF represents the orientation and location of limbs within a frame. We concatenate the PAF and HM with the input S at the channel dimension. In this case, the input has the same configuration as previously mentioned for S except for the channel dimension, which becomes thrice the original channel dimension.

Attention Block. Incorporating the attention mechanism in the network architecture helps the model learn the relevant details of the input image. We use [37] to compute the attention map from the feature maps of E_{2D} and E_{3D} separately. These attention maps capture the global and long-range dependencies of the input feature maps. The attended feature maps from the E_{2D} and E_{3D} are then concatenated at the channel dimension

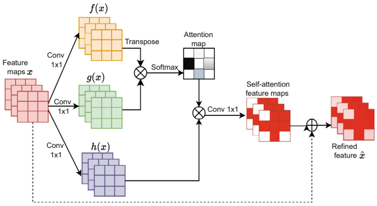


Fig. 2. Attention mechanism used in the attention block of Figure-1. ⊗ represents the matrix multiplication and ⊕ element-wise summation.

and refined with a separate attention layer to obtain final enriched feature maps which are passed to the decoder for the middle frame prediction task.

Each attention map is obtained by dividing the feature map x into three feature spaces f, g, h where $f(x) = w_f(x)$, $g(x) = w_g(x)$, $h(x) = w_h(x)$. Each x, f, g, h has a configuration of $C \times H \times W$, indicating the channel, height, and width of the feature map. $w_f(x), w_g(x), w_h(x)$ are the learned weight matrices. The final output after using the attention mechanism is represented by $\hat{x}$, which is obtained by using the following eqs:

$$z = f^T * g, \tag{2}$$

$$A_{j,i} = \frac{exp(z_{ij})}{\sum_{i,j} exp(z_{ij})} \tag{3}$$

where T represents the transpose of the feature map, $z_{ij} = f(x_i)^T g(x_j)$, and $A_{j,i}$ represents the degree by which the model attends to the i^{th} location of f for generating the j^{th} location of g.

$$v = h * A_{j,i} L \tag{4}$$

$$\hat{x} = x + v \tag{5}$$

We also show the mechanism for obtaining the attention map in Fig. 2.

Feature Disentanglement Module (FDE). The objective of this module is to explicitly exploit the distinct information from the features obtained from E_{2D} and E_{3D}. The feature disentanglement technique leverages encoders to learn distinct features. We compute the cosine similarity between the maxpooled vectors of feature maps obtained from dual encoders [19]. The feature disentanglement approach is used as the learning objective, given in eq. (7), which minimizes the similarity between features from different encoders.

Decoder. The final refined features from the attention module are given to the decoder to predict the middle frame. We use 2D CNN in the decoder as we are predicting a single frame. The decoder receives the concatenated skip connections from E_{2D} and E_{3D} at various stages (384×384, 192×192, and 96×96 resolutions). The feature maps after each stage in the decoder are concatenated with the corresponding skip connection after upsampling at the channel dimension. The output of the decoder is the predicted middle frame.

3.3 Learning Objectives.

We use two learning objectives to train $\mathcal{F}$. The first objective is the Mean Squared Error (MSE), which computes the average squared difference between the predicted and the target frame. It is defined as:

$$\mathcal{L}_{MSE} = ||\hat{f}_0 - f_0||_2. \tag{6}$$

We also use the feature disentangled loss as our second learning objective. It is formulated as below:

$$\mathcal{L}_{FDE} = \max\left(\frac{m_{2D} \cdot m_{3D}}{||m_{2D}||_2||m_{3D}||_2}, 0\right) \tag{7}$$

where m_{2D} and m_{3D} are the max pooled feature vectors obtained from the feature maps of the E_{2D} and E_{3D}, respectively. A margin of 0 is given with the loss function so that the dissimilarity between the feature maps will not affect the representation of actual feature maps obtained from both encoders.

The goal of the total loss function is to minimize both L_{MSE} and L_{FDE}. The total loss function for $\mathcal{F}$ is defined as.

$$\mathcal{L}_{total} = \mathcal{L}_{MSE} + \mathcal{L}_{FDE} \tag{8}$$

3.4 Anomaly Detection.

We follow the frame-level anomaly detection for calculating the abnormality score during inference rather than object-level. We compute the abnormality score using the Peak-Signal-to-Noise Ratio (PSNR) ratio and L_{MSE}. The PSNR is the measure for estimating the quality of the image. We compute the PSNR between the predicted and the ground-truth middle frame defined as:

$$\mathcal{PSNR}(f_0, \hat{f}_0) = 10 \log 10 \frac{(Max_{\hat{i}})^2}{(1/N)\Sigma_{i=1}^{N}(f_0 - \hat{f}_0)^2} \tag{9}$$

N denotes the number of elements in a frame, which is $\#rows \times \#columns$, and Max_i is the maximum intensity of the predicted frame.

Another measure, L_{MSE}, computes the absolute difference between the predicted and the ground truth middle frame, as in eq. (6). The PSNR and L_{MSE} values are used separately as the abnormality score for DEFD-Net.

4 Experiments and Results.

4.1 Datasets.

We evaluate the proposed framework DEFD-Net on two publicly available thermal datasets namely TSF [35] and TRE [26].

TSF Dataset. The dataset contains 9 training videos with normal activities and 35 testing videos with normal and abnormal activities. The frames are 640 × 480 in size. The normal training samples are the daily activities of living e.g., a person entering a room or lying on the bed, an empty room, etc. Abnormal activities include a person falling from a bed or from a chair, etc. This dataset contains only one subject in a frame.

Thermal Rare Event Dataset. The Seasons in Drift dataset [29] is manually annotated at frame level by [26] to make its applicability for anomaly detection problem. The dataset contains the activities near a harbor during the day and night. It has 3,480 frames for training and 36,120 for testing, with a size of 384 × 288. The usual activities are people walking and sitting, vehicles moving, etc. The abnormality constituted events are people embarking or debarking from a boat, reverse driving, stalled car, group jumping, person near the pier, etc.

4.2 Methods for Comparison

We compare the proposed work DEFD-Net, against the seven best performing models for ADVI and two methods for ADTI. We tested the performance of these models on TSF and TRE datasets.

MNAD [31] learns the normality of training data and stores it in the memory module. During testing, the memory module is used resulting in a high prediction error for anomalous frames. It computes the abnormality score by the weighted sum of PSNR between the input and predicted frame and the L2 distance between the query and the nearest memory item.

SSMTL [12] integrated multiple tasks: arrow of time, motion irregularity, middle bounding box prediction, and knowledge distillation into a single architecture for designing a self-supervised framework for anomaly detection. The average sum of scores by four proxy tasks is the abnormality score.

HF2VAD [24] used multi-level memory modules for flow reconstruction. These reconstructed flows with the input video frames are used to predict the future frame in a unified way. They computed abnormality score as the weighted sum of flow reconstruction and future frame prediction error, using L_2 distance.

Background-Ag [13] framework has motion and appearance autoencoders, each with binary classifiers. It uses an adversarial learning strategy to overcome the limitation of anomalous data during training. The abnormality score is the average of class probabilities using cross entropy loss of binary classifiers.

ASTNet [20] uses a residual autoencoder, in which the encoder exploits spatial and temporal features, and the decoder uses channel attention from various stages. The PSNR between the predicted and target frames is the anomaly score.

MAAM-Net [36] exploits an encoder with a memory module to reconstruct the frame using an appearance decoder and predicts the flow using the motion decoder. It uses L_p distance as the anomaly score, computed by the weighted sum of motion and appearance error maps.

F2LM gen & dest [15] predicts future frame using a generator-destroyer architecture. The anomaly score is the weighted sum of triplet and MSE loss.

Motion and Region [27] built a unified framework for thermal frame and flow reconstruction using the sliding window method. And mean and standard deviation as the anomaly score. In Table 1, we have mentioned the abnormality score used by each method for computing the frame-level AUC.

SSMCTB [26] block is added at the penultimate layer of MNAD [31] network with its reconstruction loss added to the loss function of MNAD. The anomaly score used is the same as that of MNAD.

Video frame interpolation. Anomaly detection can be visualized as a frame interpolation task where an intermediate frame is interpolated at any time step t using the neighboring frames. Therefore, we compare our model with two video interpolation methods CAIN [6] and UPR-Net [16]. In CAIN, the need for an optical flow estimation network is eliminated and replaced it with a deep neural network that attends to the motion information from the multiple channels required for frame synthesis. In UPR-Net, a pyramid network was designed to synthesize an intermediate frame from a pair of consecutive frames using a bidirectional flow module. We present the comparison in Table 5.

4.3 Experimental Setup

Architecture details. Our E_{2D} consists of four blocks. The first three blocks have a similar configuration, comprising two sequential sets of 2D convolution (Conv2D), batch normalization (BN), and ReLU layers. The kernel size of each Conv2D is 3×3, and the stride is set to 1. The last block consists of a set of Conv2D, BN, ReLU, and Conv2D layers with 3×3 filter size and stride 1. After every first three blocks, a max-pool layer is used with the filter size 2×2 and stride=2. The E_{3D} comprises of six blocks. Each block contains a sequential set of 3D convolution (Conv3D), BN, and ReLU layers. The kernel size of each Conv3D is $3 \times 3 \times 3$, with a stride of 1. After every two such blocks, a max-pool layer is used with a filter of $1 \times 2 \times 2$ and a stride of 1, 2 at the channel and spatial dimensions, respectively. After the last block, the max-pool layer uses a filter of size $N \times 2 \times 2$, where N denotes the input sequence length.

The last component of DEFD-Net is the decoder, which uses 2D CNN. The first three blocks of the decoder resemble the structure of the first three blocks of the 2D encoder with the same filter size and stride. After each of these blocks, an upsampling operation is performed, and skip connections from the dual encoders are concatenated with the corresponding intermediate features of the decoder at the channel dimension. The last block consists of two sets of Conv2D, BN, ReLU layers, and then a Conv2D and ReLU layer with 3×3 filter size and stride=1.

Parameter tuning. We use two settings for DEFD-Net, a sequence of fifteen frames by skipping two from a consecutive set of frames, and another with SL=5, SF=0 where SL and SF are abbreviations for sequence length and skipped frames, respectively. Each frame is resized to 384×384, and the intensity of each frame is normalized to [0,1]. We use the Pytorch framework (version 2.1.2+cu121) on Ubuntu (version 20.04.4) operating system to implement our approach. We used NVIDIA A100 80GB PCIe to perform all experiments with a CUDA version 12.0. We train our model for 35 epochs using the Adam optimizer and learning rate of $2e^{-4}$. We also decay the learning rate using the cosine annealing method. All the settings are same for both datasets, TSF and TRE.

We performed experiments for best performing models [12,13,15,24,31,36] of ADVI, using their default configuration. And we compare DEFD-Net against MNAD + SSMCTB [26] as a comparison for ADTI using the default configuration of MNAD. For dataset-specific parameters, we use the same setting as UCSD Ped2 [22] for TSF and TRE because thermal data is more similar to grayscale data than visible, e.g., we use the detection threshold=0.5 same as given in [12,13].

4.4 Evaluation Metric

Taking inspiration from the prior works in anomaly detection [12,31,36] etc., we evaluate the performance of framework DEFD-Net using the Area Under the ROC Curve (AUC) at frame level. Specifically, as suggested in [26], we are using micro AUC, which means the frames of all videos are concatenated into a single video, and then AUC is computed. We used PSNR and L_{MSE} as the measure to compute the abnormality score.

4.5 Results.

Table 1. Comparison of DEFD-Net with ADVI (†) and ADTI (*) methods on TSF and TRE thermal datasets using AUC (in %). '-' represents results could not obtained.

Year	Methods	TSF	TRE	AUC Criteria
2020	MNAD [31]†	78.04	57.60	Weighted sum of L_2 distance and PSNR
2021	SSMTL [12]†	43.0	46.7	Averaged sum of proxy tasks
2021	HF2VAD [24]†	66.50	52.58	L_2 distance
2021	Background-Ag [13]†	40.8	46.3	Averaging the Cross entropy loss
2023	ASTNet [20]†	83.0	57.2	PSNR
2023	MAAM-Net [36]†	65.33	52.35	L_p distance
2024	F2LM gen & dest [15]†	86.67	59.7	Weighted sum of triplet and MSE loss
2020	Motion & Region [27] *	88.0	-	Mean
		90.0	-	Standard deviation
2023	MNAD + SSMCTB [26]*	81.40	58.9	Weighted sum of L_2 distance and PSNR
2024	DEFD-Net (ours)*	95.52	60.28	PSNR
		94.98	61.05	MSE loss

Table 2. Comparison of DEFD-Net with existing methods on TSF and TRE thermal datasets using AUC (in %). '-' represents results could not obtained. SL and SF represent sequence length and skipped frames, respectively. P and L represent PSNR and L_{MSE}, respectively.

(SL, SF)	Method	TSF	TRE
(5, 0)	MNAD	84.50	56.63
	ASTNet	-	-
	DEFD-Net	**86.07** (P) 84.68 (L)	57.78 (P) **59.04** (L)
(5, 2)	MNAD	94.16	**60.20**
	ASTNet	76.7	59.3
	DEFD-Net	**94.95** (P) 93.62 (L)	57.96 (P) 58.38 (L)
(15, 2)	MNAD	93.69	57.89
	ASTNet	83.3	55.7
	DEFD-Net	**95.52** (P) 94.98 (L)	60.28 (P) **61.05** (L)

Table 3. AUC scores (in %) for DEFD-Net using PAF and HM with thermal images of TSF and TRE. P and L represent PSNR and L_{MSE}, respectively.

Input	TSF	TRE
Thermal frames	86.07 (P) 84.68 (L)	57.78 (P) 59.04 (L)
Thermal frames + PAF + HM	86.46 (P) 85.15 (L)	58.51 (P) 60.0 (L)

Table 4. Comparison of DEFD-Net against ADVI and ADTI methods for average running time during inference.

Method	Time (ms)
MNAD	8.1353
HF2VAD	36.5724
ASTNet	44.4359
F2LM gen & dest	33.8534
MNAD + SSMCTB	8.2656
DEFD-Net	20.7816

Quantitative Analysis. We perform experiments to compare DEFD-Net using top performing competitors, MNAD [31], ASTNet [20] (as per Table 1). We did not use F2LM gen & dest [15] for comparison as it uses pretrained object detector on visible images. We compare these three models on each others' default settings as MNAD: SL=5, SF=0; ASTNet: SL=5, SF=2 and DEFD-Net: SL=15 and SF=2 and input size to 384 × 384. It can be seen from Table 2 that our model surpasses MNAD and ASTNet in all cases except one i.e., for SL=5, SF=2. We could not run ASTNet without skipping any frame; therefore, the results are not available for SL=5, SF=0 settings.

In our next setup, we present the performance of MNAD, ASTNet, and DEFD-Net with settings SL=5, 7, 15 and SF=1, 2, 3. in Fig. 3 on TSF and TRE datasets. DEFD-Net surpasses both the models on all the settings except one. This shows the our model's robustness against setting parameters.

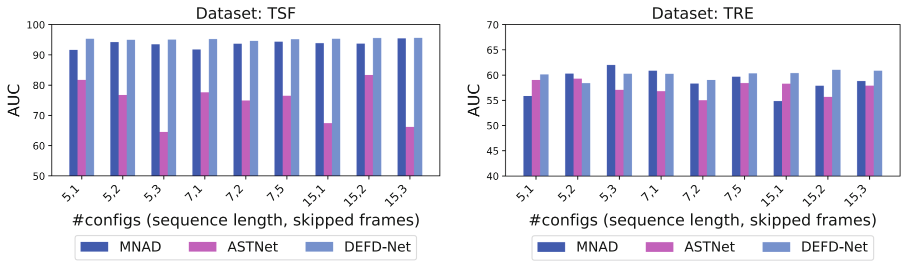


Fig. 3. Comparison of MNAD [31], ASTNet [20] and DEFD-Net using various sequence lengths and skipped frames for TSF and TRE datasets.

Ablation Study. We also present an ablation analysis of DEFD-Net. We analyze four cases: 1) removing all attention layers; 2) removing attention layer 3; 3) removing the attention layers 1 and 2; and 4) removing 2D encoder, attention layers 1 and 3 (refer Fig. 1). The column 1 in the Table 6 represents the above four cases. The performance of the TRE dataset reduced by 1.9% and 2.44% while using PSNR and MSE, respectively in case 1. The performance of other cases are between DEFD-NET and case 1. This is because the objects are of low spatial resolution in compared to the frame. On the other hand, the performance of the TSF dataset is not affected much because there is only one object in the frame with a high spatial resolution.

Running Time. Table 4 reports the analysis of analysis of per frame inference time. Our proposed model is faster than various competitive methods, including top-performing methods except MNAD [31].

Table 5. Comparison of DEFD-Net with VFI methods on TSF and TRE thermal datasets using AUC (in %). SL and SF represent sequence length and skipped frames, respectively. P and L represent PSNR and L_{MSE}, respectively.

Year	VFI Methods	TSF	TRE
2020	CAIN	85.39 (P) 83.99 (L)	52.46 (P) 52.5 (L)
2023	UPR-Net	82.54 (P) 83.23 (L)	55.55 (P) 56.83 (L)
2024	DEFD-Net (SL=3, SF=0)	84.43 (P) 83.24 (L)	56.7 (P) 57.77 (L)
	DEFD-Net (SL=15, SF=2)	95.52 (P) 94.98 (L)	60.28 (P) 61.05 (L)

Table 6. Ablation study for DEFD-Net.

DEFD-Net	TSF	TRE
-$\hat{x}$ 1,2,3	95.33 (P) 94.60 (L)	58.38 (P) 58.61 (L)
-$\hat{x}$ 3	95.47 (P) 94.79 (L)	60.17 (P) 60.54 (L)
-$\hat{x}$ 1,2	95.44 (P) 94.88 (L)	59.63 (P) 60.41 (L)

-E_{2D} - $\hat{x}$ 1,3	95.02(P)
	94.78(L)
		58.71 (P)
		59.31(L)

-	95.52(P) 94.98 (L)	60.28 (P) 61.05 (L)

To our best efforts, we did not find any work on VFI for anomaly detection in thermal videos using unsupervised setup except one [7] that too in visible imagery. But, the code is not available. However, we compared DEFD-Net with two popular methods of VFI from visible domain i.e., CAIN [6], UPR-Net [16] which take two frames as input and output an intermediate frame. The results are reported in Table 5. We present AUC in two settings for DEFD-Net - 1) our default setting (SL=15, SF=2) and, 2) same as that of competitive models (SL=3, SF=2).

Multitasking. We add a classification head with the prediction. For this, input frame sequence is provided in original or reverse order randomly to the model and classification head has to predict the input sequence class label i.e, 0 for original sequence and 1 for reverse sequence. We used cross entropy loss with our objective function defined for the task. Order classification is intended to make the proposed model learn temporal dependencies in a better way. The resultant performance gets improved on both the datasets, i.e. an AUC of 95.71 (PSNR) & 95.501 (L_{MSE}) on TSF; 60.76 (PSNR) & 61.203 (L_{MSE}) on TRE in comparison to 95.52 (PSNR) & 94.98 (L_{MSE}) on TSF; 60.28 (PSNR) & 61.05 (L_{MSE}) on TRE. Similarly, AUC may improve after adding more proxy tasks.

Finally, we exploit PAF and HM by using them as input to E_{2D} and E_{3D} with thermal frames. Table 3 shows that with modified input DEFD-Net yielded a better AUC. We have shown the results for SL=5 and SF=0.

Qualitative Analysis. We visualize the performance of DEFD-Net, using the normal and abnormal samples from TSF and TRE datasets. Fig. 4 shows the ground truth frames and the feature maps obtained after the final attention layer with first and third rows having anomalous frames. A small size of object, girl, is standing alone in a boat and person falling in first and third frames, respectively are anomalies. It can be seen from the figure that clear visible features are obtained by our model. These features improve after adding PAF and HM channels, concluding the potency of using PAF and HM with thermal frames. Though, the size of the object is small, feature maps obtained by the model has white spot came in left bottom corner which gets brighter with the use of HM and PAF channels. similar behavior is observed in third row.

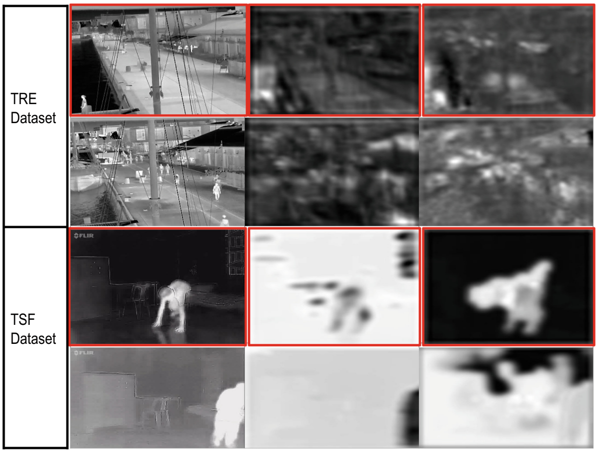


Fig. 4. Visualizing ground truth image, feature maps using thermal frames, and feature maps using thermal frames, HM, and PAF from left to right columns. Red bold-lined samples are the anomalous ones.

5 Conclusion.

We presented an anomaly detection framework for thermal imagery. We have chosen thermal imagery as it preserves the privacy of subjects and handles illumination challenges. We have proposed DEFD-Net which uses a dual encoder with disentanglement to learn distinctive features to identify anomalies in the data. It has been shown through our extensive empirical study that our method on global features not only outperforms object-centric and flow detection methods but also performs better when object size is comparatively very small than that of the frame. Also, our qualitative results show that addition of PAF and HM helps in enhancing the features of anomalous regions.

We take enhancing features for very small objects in images as future direction of the work.

References

1. Benmoussat, M.S., Guillaume, M., Caulier, Y., Spinnler, K.: Automatic metal parts inspection: Use of thermographic images and anomaly detection algorithms. vol. 61, pp. 68–80. Infrared Physics & Technology (2013)
2. Berg, A., Öfjäll, K., Ahlberg, J., Felsberg, M.: Detecting Rails and Obstacles Using a Train-Mounted Thermal Camera. In: Paulsen, R.R., Pedersen, K.S. (eds.) SCIA 2015. LNCS, vol. 9127, pp. 492–503. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19665-7_42
3. Bhatia, Y., Rai, R., Gupta, V., Aggarwal, N., Akula, A.: Convolutional neural networks based potholes detection using thermal imaging. vol. 34, pp. 578–588. Journal of King Saud University-Computer and Information Sciences (2022)
4. Cao, C., Lu, Y., et al.: Context recovery and knowledge retrieval: A novel two-stream framework for video anomaly detection. IEEE Trans. on Image Processing (2024)
5. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition. pp. 7291–7299 (2017)
6. Choi, M., Kim, H., Han, B., Xu, N., Lee, K.M.: Channel attention is all you need for video frame interpolation. In: Proceedings of the AAAI Conf. on Artificial Intelligence. vol. 34, pp. 10663–10671 (2020)
7. Deng, H., Zhang, Z., Zou, S., Li, X.: Bi-directional frame interpolation for unsupervised video anomaly detection. In: Proceedings of the IEEE/CVF Winter Conf. on App. of Computer Vision. pp. 2634–2643 (2023)
8. Elshwemy, F.A., Elbasiony, R., Saidahmed, M.T.: A new approach for thermal vision based fall detection using residual autoencoder. vol. 13. Int Journal of Intelligent Engineering & Systems (2020)
9. Farnebäck, G.: Two-Frame Motion Estimation Based on Polynomial Expansion. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 363–370. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45103-X_50
10. Gasparini, R., D'Eusanio, A., Borghi, G., Pini, S., Scaglione, G., Calderara, et al.: Anomaly detection, localization and classification for railway inspection. In: Int. Conf. on Pattern Recognition. pp. 3419–3426 (2021)
11. Gaus, Y.F.A., Bhowmik, N., Isaac-Medina, B.K., Shum, H.P., Atapour-Abarghouei, A., Breckon, T.P.: Region-based appearance and flow characteristics for anomaly detection in infrared surveillance imagery. In: Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition. pp. 2994–3004 (2023)

12. Georgescu, M.I., Barbalau, A., Ionescu, R.T., Khan, F.S., Popescu, M., Shah, M.: Anomaly detection in video via self-supervised and multi-task learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12742–12752 (June 2021)
13. Georgescu, M.I., Ionescu, R.T., Khan, F.S., Popescu, M., Shah, M.: A background-agnostic framework with adversarial training for abnormal event detection in video. vol. 44, pp. 4505–4523. IEEE Trans. on Pattern Analysis and Machine Intelligence (2021)
14. He, L., Zhang, M., Liu, H., Wang, L., Li, F.: Compressed video anomaly detection of human behavior based on abnormal region determination. IEEE Trans. on Cognitive and Developmental Systems (2024)
15. Hong, S., Ahn, S., Jo, et al.: Making anomalies more anomalous: Video anomaly detection using a novel generator and destroyer. IEEE Access (2024)
16. Jin, X., Wu, L., Chen, J., Chen, Y., Koo, J., Hahm, C.h.: A unified pyramid recurrent network for video frame interpolation. In: Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition. pp. 1578–1587 (2023)
17. Kim, C., et al.: Automatic detection of linear thermal bridges from infrared thermal images using neural network. vol. 11, p. 931. Applied Sciences (2021)
18. Kim, D., Hwang, H., Kim, H.: Cecvt: Initial diagnosis of anomalies in thermal images. IEEE Access (2023)
19. Kim, M., Cho, M., Lee, S.: Feature disentanglement learning with switching and aggregation for video-based person re-identification. In: Proceedings of the IEEE/CVF Winter Conf. on App. of Computer Vision. pp. 1603–1612 (2023)
20. Le, V.T., et al.: Attention-based residual autoencoder for video anomaly detection. vol. 53, pp. 3240–3254. Applied Intelligence (2023)
21. Lee, E.K., Viswanathan, H., Pompili, D.: Model-based thermal anomaly detection in cloud datacenters using thermal imaging. vol. 6, pp. 330–343. IEEE Trans. on Cloud Computing (2015)
22. Li, Weixin, a.o.: Anomaly detection and localization in crowded scenes. vol. 36, pp. 18–32. IEEE Trans. on Pattern Analysis and Machine Intelligence (2013)
23. Lile, C., Yiqun, L.: Anomaly detection in thermal images using deep neural networks. In: Int. Conf. on Image Processing. pp. 2299–2303 (2017)
24. Liu, Z., Nie, Y., Long, C., Zhang, Q., Li, G.: A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In: Proceedings of the IEEE/CVF Int. Conf. on Computer Vision. pp. 13588–13597 (2021)
25. Ma, H., Zhang, L.: Attention-based framework for weakly supervised video anomaly detection. J. Supercomput. **78**(6), 8409–8429 (2022)
26. Madan, N., Ristea, N.C., Ionescu, R.T., Nasrollahi, K., Khan, F.S., Moeslund, T.B., Shah, M.: Self-supervised masked convolutional transformer block for anomaly detection. IEEE Trans. on Pattern Analysis and Machine Intelligence (2023)
27. Mehta, V., Dhall, A., Pal, S., Khan, S.S.: Motion and region aware adversarial learning for fall detection with thermal imaging. In: Int. Conf. on Pattern Recognition. pp. 6321–6328 (2021)
28. Mishra, C., Bagyammal, T., Parameswaran, L.: An algorithm design for anomaly detection in thermal images. In: Innovations in Electrical and Electronic Engineering: Proceedings of ICEEE 2020. pp. 633–650. Springer (2021)
29. Nikolov, I.A., Philipsen, M.P., Liu, J., Dueholm, J.V., Johansen, A.S., Nasrollahi, K., Moeslund, T.B.: Seasons in drift: A long-term thermal imaging dataset for

studying concept drift. In: Thirty-fifth Conf. on Neural Information Processing Systems. Neural Information Processing Systems Foundation (2021)

30. Park, G., Lee, M., Jang, H., et al.: Thermal anomaly detection in walls via cnn-based segmentation. vol. 125, p. 103627. Automation in Construction (2021)
31. Park, H., Noh, J., Ham, B.: Learning memory-guided normality for anomaly detection. In: Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition. pp. 14372–14381 (2020)
32. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer
33. Sledz, A., Heipke, C.: Thermal anomaly detection based on saliency analysis from multimodal imaging sources. vol. 1, pp. 55–64. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences (2021)
34. Sledz, A., Unger, J., Heipke, C.: Uav-based thermal anomaly detection for distributed heating networks. vol. 43, pp. 499–505. The Int. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (2020)
35. Vadivelu, S., Ganesan, S., Murthy, O.R., Dhall, A.: Thermal imaging based elderly fall detection. In: Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part III 13. pp. 541–553. Springer (2017)
36. Wang, L., Tian, J., Zhou, S., Shi, H., Hua, G.: Memory-augmented appearance-motion network for video anomaly detection. vol. 138, p. 109335. Pattern Recognition (2023)
37. Zhang, H., Goodfellow, I., Metaxas, D., et al.: Self-attention generative adversarial networks. In: Int. Conf. on Machine Learning. pp. 7354–63. PMLR (2019)
38. Zhong, J.X., Li, N., Kong, W., Liu, S., Li, T.H., Li, G.: Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In: Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition. pp. 1237–1246 (2019)
39. Zhou, Y., Xu, X., Song, J., Shen, F., Shen, H.T.: Msflow: Multiscale flow-based framework for unsupervised anomaly detection. IEEE Trans. on Neural Networks and Learning Systems (2024)

Adaptive Data Association for Enhanced Multi-object Tracking and Segmentation with Pre-matching and Selective Association

Longtao Chen[1](✉), Guoxing Liao[1], Gaofeng Zhu[1], and Huanqiang Zeng[1,2](✉)

[1] Engineering College, Huaqiao University, Quanzhou, China
{longtaochen,zeng0043}@hqu.edu.cn
[2] Quanzhou Digital Institute, Quanzhou, China

Abstract. Multi-Object Tracking and Segmentation (MOTS) is a critical task in autonomous driving, robotic perception, and video analysis. The challenge lies in accurately identifying and associating objects within video sequences, especially when the number of objects is uncertain, motion patterns vary, and frequent object overlaps occur. In this paper, we propose a method with pre-matching and selective association (PS-Track) to adaptively combine motion and appearance cues to cope with continuously changing scenes. Unlike solely relying on the single cue or predetermined combination schemes, our method facilitates data association by discerning similarities in appearance among tracks across different scenarios through the dynamic selection of suitable schemes. Through experimentation, we also found that our method exhibits advantages in tracking efficiency compared to complex models. Our method achieved outstanding results on the MOTS dataset, with scores of 61.0 for sMOTA, 56.4 for IDF1, 76.0 for MOTSA, and 76.5 for FPS.

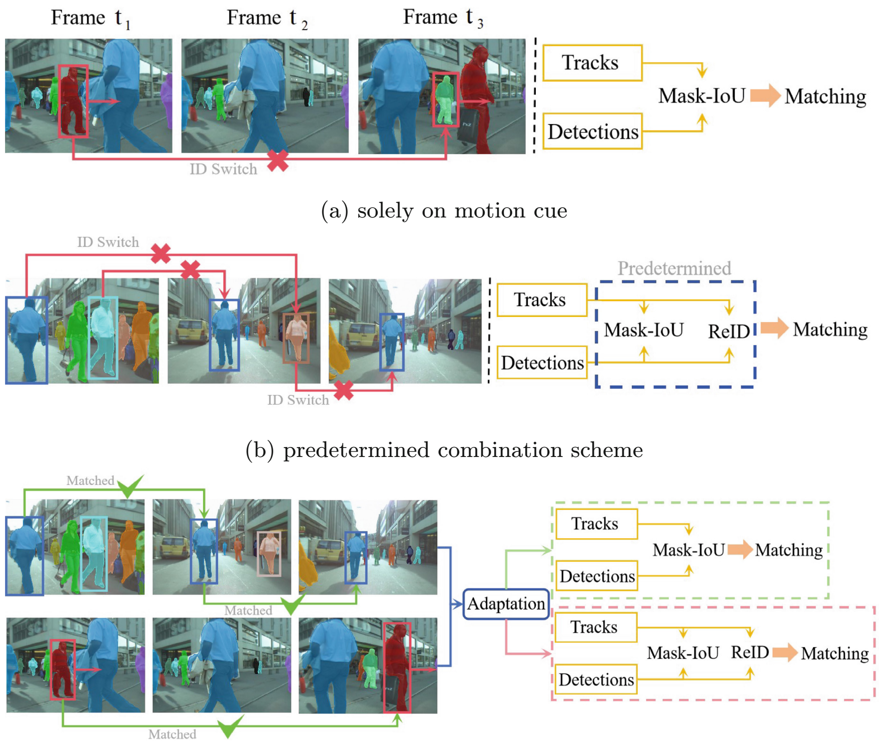
Keywords: Data Association · Multi-object Tracking and Segmentation · Online Tracking · Video Processing

1 Introduction

Multi-object Tracking [1,4] and Segmentation (MOTS) [28,34] is a critical computer vision task [2,20,25,32] with various real-world applications, such as autonomous driving [10,27], robotic perception [19,33], and video analysis [22,23,30]. However, MOTS remains challenging in realistic scenarios due to uncertainties in object counts, varied motion patterns and frequent occlusions. Data association, a pivotal step in MOTS, plays a vital role in addressing these complexities. Existing data association methods can be categorized into two main groups: those relying solely on the single cue, and those using a predetermined combination scheme with multiple cues. The single cue methods [2,5,31], like

L. Chen and G. Liao—Co-first authors.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 254–268, 2025.
https://doi.org/10.1007/978-3-031-78312-8_17



(a) solely on motion cue

(b) predetermined combination scheme

(c) adaptive combination scheme

Fig. 1. The flowcharts of different data association schemes.

motion-only cues, can be effective in certain situations but often struggle with complex backgrounds and occlusions. As illustrated in Figure 1(a), when objects are obscured for extended periods, relying solely on motion cues is insufficient for effective object tracking. The predetermined schemes [6,7,9,16,36,37] aim to improve association reliability by incorporating appearance and motion cues, but the varying reliability of appearance features across different lighting and viewpoint conditions can limit their performance. As illustrated in Figure 1(b), when multiple objects share similar appearance features, depending on a predetermined scheme can result in Identity Switch (ID SW.) errors, due to unreliable appearance features dismissing the combination.

To overcome the limitations of the two main groups above, our research proposes a flexible and adaptive association method, as illustrated in Figure 1(c), which dynamically selects the most suitable scheme for each scene in a video sequence. Unlike these above methods, our method enables the dynamic adjustment of the association scheme based on specific circumstances, thus selecting the most appropriate schemes depending on the context. For instance, we believe that combining motion cues with appearance features is effective in scenarios involving prolonged occlusions or co-directional movements. Conversely, we consider motion-only schemes effective in situations characterized by changes in lighting conditions or camera motion. To select the most suitable schemes, we propose a Selective Association based on Pre-matching. This process involves extracting appearance features, calculating appearance similarity cost, and selecting matching pairs of tracks and detections with varying appearance similarity for Pre-matching. During Selective Association, we associate matching pairs with strong appearance similarity from pre-matching results using the fusion of appearance features and Mask-IoU, while for other matching pairs, only Mask-IoU is utilized. This method ensures that different scenes have the most suitable scheme for data association.

As Figure 1(c) shows, similar-looking objects like pedestrians in white shirts can confuse associations based on appearance features. Using Mask-IoU to metric motion cues is more appropriate here. When there's a resemblance of objects' appearance features before and after occlusion like a pedestrian in a red shirt re-emerging, relying solely on motion cues is insufficient due to the temporary disappearance. Combining appearance features and motion cues at this stage aids tracking effectively.

Recent advancements [1,4,8,11,13,15,21,24,26,29] in deep learning-based tracking have significantly improved accuracy. However, the intricate network architectures of these methods can impede tracking speed and increase computational expenses. Through experimentation, we unexpectedly discovered that our method outperforms deep learning methods in terms of tracking efficiency. This advantage arises from the incorporation of a versatile data association mechanism, which combines efficient frame-by-frame segmentation with a lightweight re-identification model. Our proposed method, named PS-Track, was evaluated on the MOTS dataset in the MOT challenge. Using the same detection input as the baseline TrackRCNN [28] method, PS-Track achieved a sMOTA score of 61.3, which is about 34% higher, and a MOTSA score of 76.3, which is 27% higher. Furthermore, when using the same re-identification model and similarity measure as TRR [2], PS-Track outperformed TRR by around 10% in sMOTA.

2 Related Work

Multi-object tracking and segmentation are extensions of multi-object tracking at the pixel level. Data association is a core component of these methods [6,7,9,16,37,38]. Data association for object tracking has evolved through different schemes.

Initially, methods [2,5,16] that rely solely on object motion cues from video sequences were employed, analyzing position changes between frames and associating objects with similar motion patterns. While effective in constant motion states, these methods struggle in complex environments with diverse backgrounds and object occlusions, leading to limited tracking capability. TRR [2] is an extension of SORT [5] that incorporates Kalman filtering to predict the mask positions of individual objects in the current frame. It computes Mask-IoU

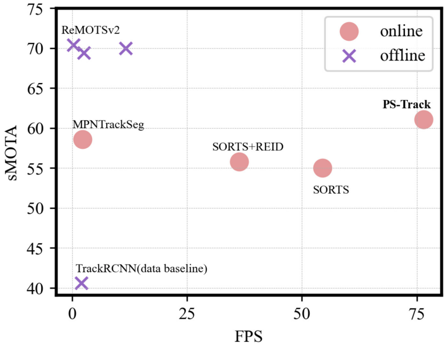


Fig. 2. We evaluated the sMOTSA and FPS of all methods on the MOTS dataset. Some methods from the MOT Challenge benchmark were named.

to quantify motion cues and employs the Hungarian algorithm [18] for association. If objects are occluded for an extended period or move at the same speed between two objects, the motion cues between the objects become similar or are lost, resulting in ID switches in the final matching results.

A predetermined combination scheme of object motion cues with appearance features such as color and texture was proposed to address these limitations. In DeepSORT [31], the first association is based on appearance features. If the tracks remain unmatched after this initial round, a second association is conducted, utilizing similarity in motion cues. TRR+ReID [2] builds upon TRR by employing an additional ReID model to address occlusion issues. FWT [14] achieves a more comprehensive tracking system by integrating multiple detectors with different characteristics, such as full-body and body-parts detectors. JCC [16] integrates motion segmentation with multiple object tracking. MHT-DAM [25] utilizes the Multiple Hypothesis Tracking (MHT) algorithm with online appearance model training.

Deep learning-based methods [1,4,14,16,17] have emerged to tackle data association challenges in multi-object tracking and segmentation in recent years. These methods utilize deep neural networks to learn intricate feature representations of objects, enabling more robust data association through end-to-end training. MPNTrackSeg [1] utilizes message-passing networks within a structured graph framework to leverage contextual cues for accurate tracking. PointTrack [35] follows a tracking-by-points paradigm, improving tracking by learning instance embeddings from selected points and integrating diverse data modalities. Additionally, MOTSNet [25] and MOTD [8]; MOTD utilizes deep learning representations to enhance tracking accuracy. Despite their effectiveness, complex network architectures may result in slower tracking speeds and increased computational demands.

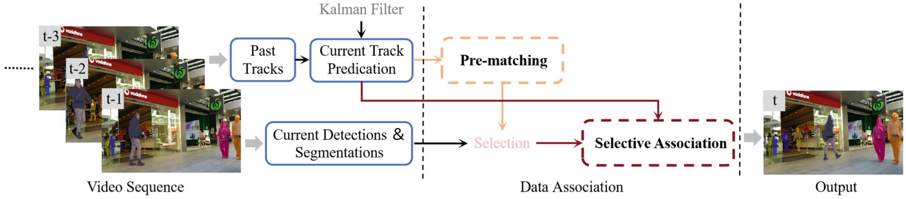


Fig. 3. The workflow of PS-Track. PS-Track optimizes data association through selective association based on pre-matching. The system categorizes tracks into strong and weak appearance similarity groups and conducts selective association based on pre-matched inputs.

Algorithm 1 Pseudo-code of PS-Track

```
Input: A video sequence $\mathbf{V}$ and object detector $\mathbf{Det}$
Output: Tracks $T$ of the video
Initialization: $T \leftarrow \emptyset$
for $frame\ f_k$ in $V$ do
    $D_k \leftarrow \mathbf{Det}(f_k)$
    for $t$ in $T$ do
        $t \leftarrow$ KalmanFilter($t$)
    end for
    /* Pre-matching */
    $P_{strong} \leftarrow \emptyset$; $P_{weak} \leftarrow \emptyset$
    Matched Pairs, Unmatched Pairs $\leftarrow$ Associate $D_k$ and $T$ using cost $C_a$
    $P_{strong} \leftarrow$ Matched Pairs
    $P_{weak} \leftarrow$ Unmatched Pairs
    /* Selective Association */
    Associate $P_{strong}$ using cost $C_s$ from Equation 1
    Associate $P_{weak}$ using cost $C_w$ from Equation 2
    $T_{remain} \leftarrow$ remaining matched tracks from $T$
    /* Delete Unconfirmed tracks */
    $T \leftarrow T \setminus T_{remain}$
    /* Initialize New tracks */
    for $d$ in $\mathbf{D}_k$ do
        $T \leftarrow T \cup \{d\}$
    end for
end for
return $T$
```

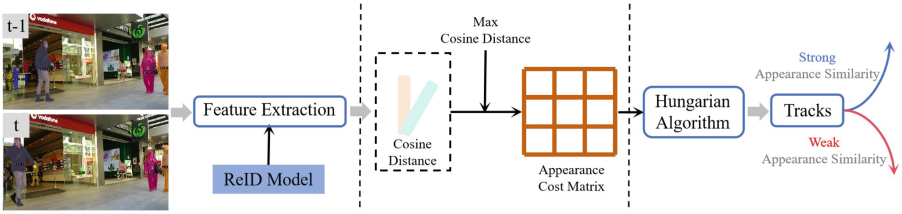


Fig. 4. The flowchart of pre-matching

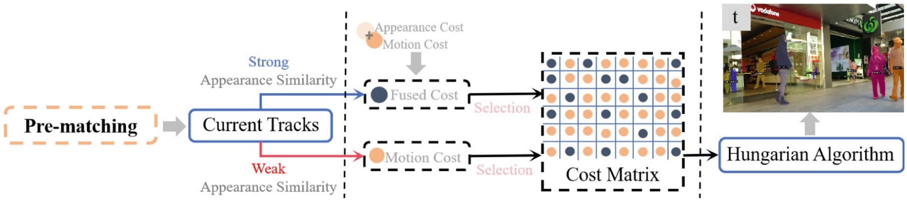


Fig. 5. The flowchart of selective association mechanism

3 Methodology

3.1 PS-Track

We introduce a versatile and robust data association method called PS-Track. As illustrated in Figure 3, PS-Track builds upon conventional detection and tracking frameworks while incorporating a selective association based on pre-matching to adapt to various tracking scenarios. The pseudocode of PS-Track is outlined in Algorithm 1.

PS-Track takes a video sequence (V) and an object detector (Det) as input. PS-Track outputs tracks (T) of the video, predicting new positions for each track using the Kalman filter. PS-Track involves two key steps: pre-matching and selective association.

During the Pre-matching stage, the method initially employs a simple ReID model to extract features from objects, yielding feature vectors. Subsequently, it calculates the cosine distance between feature vectors of detected tracks to assess the appearance similarity across all tracks and detections. The Hungarian algorithm is then applied to categorize matching pairs of tracks and detections into groups based on the level of appearance similarity. Tracks matched with detections by the Hungarian algorithm are classified as having strong appearance similarity and denoted as P_{strong}; conversely, those without such matches are considered to possess weak appearance similarity and labeled as P_{weak}, as delineated in lines 7 to 10 of Algorithm 1 and Figure 4.

In the Selective Association stage, when matching pairs of tracks and detections in strong appearance P_{strong} are identified through pre-matching, the method adopts a fusion method using ReID and Mask-IoU to consider appearance feature and occlusion comprehensively. Here, Mask-IoU is used to evaluate motion similarity between objects, ensuring accurate identification and tracking even in crowded or occluded scenarios. ReID utilizes a simplified re-identification model based on the ResNet architecture to extract feature vectors and compute cosine distances between feature vectors to assist in handling occlusions. This balanced fusion method is crucial for PS-Track to handle matching pairs with strong appearance similarity, allowing the system to effectively combine motion cues of different objects in highly similar appearance situations. For matching pairs identified with weak appearance similarity during the pre-matching process, PS-Track directly calculates their Mask-IoU similarity, as shown in lines 11 to 12 of Algorithm 1 and Figure 5.

After association, unmatched tracks are removed from the tracklist. To preserve track identity for long-term association, unmatched tracks T_{remain} are placed into T_{lost}. Tracks in T_{lost} are removed from the track set T if they exist for more than 70 frames; otherwise, they are retained in T. Additionally, new tracks are initialized from unmatched detections D_{remain} after Selective Association. The output of each frame consists of masks and identities of tracks T in the current frame, excluding those in T_{lost}.

To enhance the tracking performance of Multiple Object Tracking and Segmentation (MOTS), we integrate PS-Track with the TrackRCNN benchmark method. This integration harnesses the potential of data association methods for improved tracking accuracy.

3.2 Pre-matching

As illustrated in Figure 5, PS-Track initially evaluates appearance features among objects in consecutive video frames through pre-matching. This process generates feature vectors, which are used to calculate cosine distances, thereby evaluating appearance similarity across all tracks and detections. Matching pairs of tracks and detections are then categorized based on their appearance similarity to detections, as processed by the Hungarian algorithm. Those matching pairs with strong similarity are labeled p_{strong}, whereas others with weak similarity are labeled p_{weak}.

Building upon this, for tracks and detections with weak appearance similarities, we posit that targets in dense scenes either share similar appearance features or are undergoing rapid changes in occluded appearance cues. For tracks strongly associated with appearance, we infer that objects in dense scenes possess intact appearance features, facilitating the recovery of tracking for occluded objects.

3.3 Selective Association

In PS-Track, Selective Association is accomplished by integrating appearance and motion cues through pre-matching, ensuring the adoption of the most appropriate scheme for diverse scenes.

During the pre-matching stage, feature vectors are extracted from each detection d_j and track t_i using a simplified ReID model based on the ResNet architecture. The cosine distance between these feature vectors yields the appearance cost $C_a(t_i, d_j)$. The matching pairs of tracks and detections with weak appearance similarity p_w and strong appearance similarity p_s are selected using the Hungarian algorithm.

In the selective association stage, for matching pairs with strong appearance similarity, the motion cost $C_m(t_i, d_j)$ is computed based on the Mask-IoU between detection d_j and track t_i. Additionally, the appearance cost $C_a(t_i, d_j)$ is computed as the cosine distance between their appearance feature vectors.

These costs are then fused to obtain the total cost $C_t(t_i, d_j)$, where a weight parameter λ balances the importance of appearance and motion costs.

$$C_s(t_i, d_j) = \lambda \times C_m(t_i, d_j) + (1 - \lambda) \times C_a(t_i, d_j) \tag{1}$$

Where $C_s(t_i, d_j)$ represents the fusion cost of associating detection d_j with track t_i in the case where that matching pairs p have strong appearance similarity. $C_a(t_i, d_j)$ denotes the appearance cost, indicating the dissimilarity of appearance features between the detection and the track, while $C_m(t_i, d_j)$ represents the motion cost, measuring dissimilarity in their motion patterns. Setting the parameter λ balances the importance of appearance and motion costs.

In the case of matching pairs p of tracks and detections exhibiting weak appearance similarity, only the motion cost $C_m(t_i, d_j)$ is directly calculated.

$$C_w(t_i, d_j) = C_m(t_i, d_j) \tag{2}$$

This equation represents the similarity cost C_w when associating a detection d_j with a track t_i in the case where the matching pairs have weak appearance similarity. In this scenario, only the motion cost $C_m(t_i, d_j)$ is considered.

The final association cost is determined based on whether the matching pairs belong to the strong P_s or weak P_w appearance similarity sets.

$$C_t(t_i, d_j) = \begin{cases} C_s(t_i, d_j), & \text{if } p_{i,j} \in P_s \\ C_w(t_i, d_j), & \text{if } p_{i,j} \in P_w \end{cases} \tag{3}$$

This equation defines the final association cost $C_t(t_i, d_j)$ based on whether the matching pairs $p_{i,j}$ belong to the set of tracks and detections with strong appearance similarity P_s or weak appearance similarity P_w. If $p_{i,j}$ is in P_s, the cost is determined by $C_s(t_i, d_j)$; if $p_{i,j}$ is in P_w, the cost is determined by $C_w(t_i, d_j)$. Equations 1 and 2 are normalized before association. Consistency between the fused costs is maintained, where a smaller value indicates a higher similarity. All similarity of costs are between 0 and 1.

Through the fusion of Mask-IoU and Re-ID cues based on track visual similarity and employing the Hungarian algorithm for further refinement, PS-Track determines the optimal association scheme, thereby enhancing multi-object tracking and segmentation performance.

4 Experiments

4.1 Setting

Datasets. The current study employs the MOTS20 dataset [28] within the MOT-Challenge MOTS benchmark, with a specific focus on pedestrian tracking. It's noteworthy that the MOTS benchmark, within this challenge, places particular emphasis on handling crowded scenes, thus adding complexity to multi-object tracking, especially in scenarios requiring precise multi-object segmentation. This benchmark expands upon traditional multi-object tracking benchmarks by introducing detailed segmentation masks at the pixel level. All tracking, segmentation, and evaluation tasks are performed in image coordinates to ensure standardized and rigorous assessment. It's crucial to highlight that each sequence in this dataset is meticulously annotated at the pixel level with a high degree of precision, following a well-defined protocol.

Metrics. Various metrics [3] are commonly used to evaluate the effectiveness of proposed methods on public datasets, including Multi-Object Tracking and Segmentation Accuracy (MOTSA), Multi-Object Tracking and Segmentation Precision (MOTSP), sMOTSA (Multi-Object Tracking and Segmentation Accuracy), and IDF1 (Intersection over Union of Detection and tracking F1 score). MOTSA assesses performance regarding object detection and track maintenance, with higher scores indicating greater accuracy. MOTSP measures the precision of object positioning, primarily focusing on detector performance rather than the tracker effect. The higher the score, the better the effect. sMOTSA is a crucial statistic for evaluating the quality of detection, segmentation, and tracking. IDF1 represents the ratio of correctly identified detections over the average number of ground-truth and computed detections. Other metrics include IDS (number of identity switches), Frag (total number of times a track is fragmented), MT (mostly tracked objects), ML (mostly lost objects), FP (total number of false positives), FN (total number of false negatives), and FPS (processing speed in frames per second, excluding the detector, on the benchmark).

Implementation Details. In PS-Track, we utilize the Mask R-CNN detector [12,13] from the baseline method TrackRCNN for instance segmentation. The default detection threshold is set to 0.5 unless specified otherwise. For MOTS benchmark evaluation, we use Mask-IoU to measure motion cues similarity and cosine distance to assess appearance similarity. In the linear assignment phase, we perform two linear assignments. The first assignment categorizes tracks into groups based on strong and weak appearance similarities, preparing data for fusion during selective association. The second assignment processes the final cost matrix to obtain the ultimate matching results. Regarding lost tracks, we retain them for 70 frames in case of reappearance. For MOTS, we employ the same Re-ID model as TRR. Our method was experimented on a machine equipped with a GTX3060TI GPU, an i7 CPU, and 12GB of VRAM. The results of all comparative methods were provided by their respective papers.

Results on MOTS20 dataset. We evaluate PS-Track against several association methods, detailed in Table 1. Our comparisons include TRR, TrackRCNN, and other methods using the MOTS dataset.

Our method differs from TRR in its adaptive data association approach, despite utilizing the same Mask-IoU and ReID components. We find that PS-Track improves TRR's sMOTA metric from 55.0 to 61.0, IDF1 from 57.3 to 58.7,

Table 1. Comparison of PS-Track with the methods on the MOTS20 test set, +MG indicates Mask R-CNN generation mask using domain fine-tuning [35]. (Optimal data are marked in bold).

Scheme	Method	sMOTSA↑	MOTSA↑	FPS↑
predetermined	TRR[2]	55.0	68.3	54.5
	TRR+RReID[2]	55.8	69.1	36.4
	Track R-CNN [28]	40.6	55.2	2.0
	jCC+MG[16]	48.3	63.0	-
	FWT+MG[14]	49.3	64.0	-
	MHT-DAM+MG[17]	48.0	62.7	-
deep-learning	TraDeS[32]	50.8	65.5	-
	TrackFormer[24]	54.9	69.9	-
	MOTSNet[25]	56.8	69.4	-
	PointTrack[34]	58.09	70.58	-
	MOTDT+MG[8]	47.8	61.1	-
	MPNTrackSeg[1]	58.6	73.7	2.3
adaptive	PS-Track(ours)	**61.0**	**76**	**76.5**

and FPS from 54.5 to 76.5. This highlights the importance of fusion based on pre-matching and demonstrates PS-Track's ability to reduce the impact of weak appearance similarity on association results.

TrackRCNN serves as the MOTS benchmark method. We find that compared to TrackRCNN, under the same detector, PS-Track's results also have high gains. This indicates that designing association mechanisms can improve tracking performance and achieve better sMOTA and FPS by adding accurate enough pre-matching. We note that in cases of severe occlusion, Re-ID features are easily affected, which may lead to identity switches, while the motion model's behavior is more reliable.

As shown in Table 1, compared to methods based on MOTS benchmarks, we find that, in terms of FPS, the performance difference is significant when compared to deep learning methods. As shown in Figure 2, PS-Track demonstrates a remarkable ability to strike a balance between efficiency and cost by adaptive data association. This optimization leads to improved tracking speed while maintaining effective tracking results. In comparison to solely relying on the single cue or utilizing a predetermined combination scheme with muti-cues, our method has also demonstrated promising results on the test set.

Ablation Studies on Pre-matching. We explore various similarity metrics for PS-Track's pre-matching and selective association processes. The results are shown in Table 2. In pre-matching, both Re-ID and Mask-IoU are considered viable choices for similarity#1, with Mask-IoU achieving better IDF1 and Re-ID achieving higher sMOTA. To enrich the metric scale, we fuse the cost of

Table 2. Comparison of different types of similarity metrics used in the pre-matching and selective association on the MOTS20 train set with equal effort. The best results are shown in bold.

similarity#1	similarity#2	sMOTSA↑	IDF1↑	ID Sw.↓
Mask-IoU	Mask-IoU & ReID	59.47	58.51	310
ReID	Mask-IoU & ReID	**59.72**	58.19	**244**

Re-ID and Mask-IoU. However, in dense crowd scenes, while reliable appearance features can improve tracking accuracy, similar appearance features may lead to errors in the association process due to severe occlusion or motion blur. Mask-IoU can be used to obtain more robust association results, in this case. Therefore, selecting a suitable cue for similarity#2 in the selective association process is crucial. To address this, we use Re-ID to divide tracks into two groups during pre-matching: those with high appearance similarity and those with low appearance similarity. Pre-matching is used to select reliable appearance cues for fusion with motion information. From Table 3, it can be observed that introducing pre-matching to select reliable appearance cues for fusion results in an approximate increase of 4.9 in sMOTA and a 33% reduction in IDSW compared to solely utilizing predetermined schemes of Mask-IoU or Re-ID & Mask-IoU.

Table 3. The results in the MOTS test set for different schemes.

Scheme	sMOTSA↑	MOTSA↑	IDF1↑	IDSW↓
Mask-IoU	49.5	67.0	53.6	797
ReID & Mask-IoU	55.1	68.8	53.4	826
PSTarck	**61.0**	**75.6**	**56.2**	**619**

Table 4. Ablation Studies on different weights of appearance and motion.

Mask-IoU	ReID	sMOTSA↑	MOTSA↑	IDF1↑	IDSW↓
1	0	58.80	72.16	**62.91**	491
0.9	0.1	59.05	72.41	60.86	423
0.8	0.2	59.23	72.58	64.59	376
0.7	0.3	**59.72**	**73.08**	58.19	**244**
0.2	0.8	59.41	72.77	58.16	325
0.1	0.9	59.37	72.73	56.62	336
0	1	59.35	72.71	57.19	343

Ablation Studies on Combinations. We try different combinations of weights for fusing Mask-IoU and Re-ID in PS-Track. Refer to Table 4 for the outcomes of the MOTS20 train set. It's evident that in MOTS, both Mask-IoU and Re-ID alone can be effective choices for measuring similarity. Mask-IoU performs better in IDF1, while Re-ID performs better in sMOTA and IDSW. On the MOTS test set, the weight of Mask-IoU in fusion should be higher than Re-ID. This is because MOTS encompasses a substantial amount of pedestrian motion, and in densely populated areas, motion occlusion is inevitable, leading to occasional unreliability of appearance cues. In densely crowded environments, especially when facing occlusion or motion blur, it is crucial to incorporate both Mask-IoU and Re-ID for similarity measurement on the dataset. This joint consideration of motion and appearance enhances the reliability of similarity measurement. From Table 4, it can be observed that compared to using Mask-IoU alone, incorporating Mask-IoU as the primary measure supplemented with Re-ID can increase sMOTA by approximately 1.0, indicating that reliable appearance cues can enhance tracking accuracy.

5 Conclusion

PS-Track stands out as an uncomplicated yet versatile data association method specifically crafted for multi-object tracking and segmentation. It adaptively combines appearance and motion cues to facilitate tracking matching, thereby elevating tracking performance. A key strength of PS-Track lies in its inclusion of a pre-matching step, strategically designed to mitigate the influence of unreliable appearance data through Selective Association. On the MOTS benchmark's test set, PS-Track showcases remarkable results, achieving a MOTS score of 61.0 and an impressive MOTSA score of 76.0. The commendable performance of PS-Track, characterized by its harmonious blend of accuracy, speed, and simplicity, positions it as an appealing choice for practical applications in real-world scenarios. Additionally, the method's efficacy prompts avenues for further research, encouraging exploration into parameter fine-tuning, including thresholds, and the investigation of diverse methods for effectively fusing appearance and motion information.

Acknowledgement. This work is supported by Xiamen Natural Science Foundation(Grant No.3502Z202372034), the research startup foundation of Huaqiao university(Grant No.20201XD022) and Quanzhou Science and Technology Projects(Grant No.2023N013).

References

1. Aharon, N., Orfaig, R., Bobrovsky, B.Z.: Bot-sort: Robust associations multi-pedestrian tracking. arXiv preprint arXiv:2206.14651 (2022)
2. Ahrnbom, M., Nilsson, M.G., Ardö, H.: Real-time and online segmentation multi-target tracking with track revival re-identification. In: VISIGRAPP (5: VISAPP). pp. 777–784 (2021)

3. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP Journal on Image and Video Processing **2008**, 1–10 (2008)
4. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-Convolutional Siamese Networks for Object Tracking. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 850–865. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_56
5. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE international conference on image processing (ICIP). pp. 3464–3468. IEEE (2016)
6. Brasó, G., Cetintas, O., Leal-Taixé, L.: Multi-object tracking and segmentation via neural message passing. Int. J. Comput. Vision **130**(12), 3035–3053 (2022)
7. Cao, J., Pang, J., Weng, X., Khirodkar, R., Kitani, K.: Observation-centric sort: Rethinking sort for robust multi-object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9686–9696 (2023)
8. Chen, L., Ai, H., Zhuang, Z., Shang, C.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE international conference on multimedia and expo (ICME). pp. 1–6. IEEE (2018)
9. Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., Meng, H.: Strongsort: Make deepsort great again. IEEE Transactions on Multimedia (2023)
10. Farkh, R., Alhuwaimel, S., Alzahrani, S., Al Jaloud, K., Quasim, M.T.: Deep learning control for autonomous robot. Computers, Materials & Continua **72**(2) (2022)
11. Gao, Y., Xu, H., Zheng, Y., Li, J., Gao, X.: An object point set inductive tracker for multi-object tracking and segmentation. IEEE Trans. Image Process. **31**, 6083–6096 (2022)
12. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
13. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
14. Henschel, R., Leal-Taixé, L., Cremers, D., Rosenhahn, B.: Fusion of head and full-body detectors for multi-object tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 1428–1437 (2018)
15. Ke, L., Li, X., Danelljan, M., Tai, Y.W., Tang, C.K., Yu, F.: Prototypical cross-attention networks for multiple object tracking and segmentation. Adv. Neural. Inf. Process. Syst. **34**, 1192–1203 (2021)
16. Keuper, M., Tang, S., Andres, B., Brox, T., Schiele, B.: Motion segmentation & multiple object tracking by correlation co-clustering. IEEE Trans. Pattern Anal. Mach. Intell. **42**(1), 140–153 (2018)
17. Kim, C., Li, F., Ciptadi, A., Rehg, J.M.: Multiple hypothesis tracking revisited. In: Proceedings of the IEEE international conference on computer vision. pp. 4696–4704 (2015)
18. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1–2), 83–97 (1955)
19. Li, Y., Zhang, J., Ma, D., Wang, Y., Feng, C.: Multi-robot scene completion: Towards task-agnostic collaborative perception. In: Conference on Robot Learning. pp. 2062–2072. PMLR (2023)
20. Liu, D., Cui, Y., Tan, W., Chen, Y.: Sg-net: Spatial granularity network for one-stage video instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9816–9825 (2021)

21. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
22. Lucarno, S., Zago, M., Buckthorpe, M., Grassi, A., Tosarelli, F., Smith, R., Della Villa, F.: Systematic video analysis of anterior cruciate ligament injuries in professional female soccer players. Am. J. Sports Med. **49**(7), 1794–1802 (2021)
23. Luxem, K., Sun, J.J., Bradley, S.P., Krishnan, K., Yttri, E., Zimmermann, J., Pereira, T.D., Laubach, M.: Open-source tools for behavioral video analysis: setup, methods, and best practices. elife **12**, e79305 (2023)
24. Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C.: Trackformer: Multi-object tracking with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8844–8854 (2022)
25. Porzi, L., Hofinger, M., Ruiz, I., Serrat, J., Bulo, S.R., Kontschieder, P.: Learning multi-object tracking and segmentation from automatic annotations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6846–6855 (2020)
26. Sun, P., Cao, J., Jiang, Y., Zhang, R., Xie, E., Yuan, Z., Wang, C., Luo, P.: Transtrack: Multiple object tracking with transformer. arXiv preprint arXiv:2012.15460 (2020)
27. Viswanath, P., Sistu, G., Ilie, M., Yogamani, S.K., Horgan, J.: Early fusion of dense optical flow with image for semantic segmentation in autonomous driving. In: AICS. pp. 126–137 (2018)
28. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B.B.G., Geiger, A., Leibe, B.: Mots: Multi-object tracking and segmentation. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition. pp. 7942–7951 (2019)
29. Wang, S., Sheng, H., Yang, D., Zhang, Y., Wu, Y., Wang, S.: Extendable multiple nodes recurrent tracking framework with rtu++. IEEE Trans. Image Process. **31**, 5257–5271 (2022)
30. Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., Xia, H.: End-to-end video instance segmentation with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8741–8750 (2021)
31. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP). pp. 3645–3649. IEEE (2017)
32. Wu, J., Cao, J., Song, L., Wang, Y., Yang, M., Yuan, J.: Track to detect and segment: An online multi-object tracker. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12352–12361 (2021)
33. Xu, W., He, Y., Li, J., Zhou, J., Xu, E., Wang, W., Liu, D.: Robotization and intelligent digital systems in the meat cutting industry: from the perspectives of robotic cutting, perception, and digital development. Trends in Food Science & Technology (2023)
34. Xu, Z., Yang, W., Zhang, W., Tan, X., Huang, H., Huang, L.: Segment as points for efficient and effective online multi-object tracking and segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **44**(10), 6424–6437 (2021)
35. Xu, Z., Zhang, W., Tan, X., Yang, W., Huang, H., Wen, S., Ding, E., Huang, L.: Segment as Points for Efficient Online Multi-Object Tracking and Segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 264–281. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_16

36. Yang, F., Wang, Z., Wu, Y., Sakti, S., Nakamura, S.: Tackling multiple object tracking with complicated motions-re-designing the integration of motion and appearance. Image Vis. Comput. **124**, 104514 (2022)
37. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. In: European conference on computer vision. pp. 1–21. Springer (2022)
38. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: European conference on computer vision. pp. 474–490. Springer (2020)

Oracle Bone Script Recognition Based on Multi-scale Feature Fusion and Knowledge Distillation

Jiaoyan Wang[1], Xuebin Xu[1], Alimjan Aysa[1,2], Jun Ding[1],
and Kurban Ubul[1,2,3](✉)

[1] School of Computer Science and Technology, Xinjiang University, Ürümqi 830046, China
kurbanu@xju.edu.cn
[2] Xinjiang Multilingual Information Technology Key Laboratory, Xinjiang University, Ürümqi 830046, China
[3] Joint International Research Laboratory of Silk Road Multilingual Cognitive Computing, Xinjiang University, Ürümqi 830046, China

Abstract. Oracle bone scripts, being the oldest and most developed writing system discovered in China till date, have been meticulously studied by numerous scholars. This paper presents the Multi–scale Feature Fusion Attention Net (MFFA–Net) as a solution to the issue of variant characters in the oracle bone affecting recognition accuracy. The network is based on ResNet18 and employs asymmetric convolution alongside a refined coordinate attention technique to acquire image features. It also integrates perceptual field information of varying sizes through hierarchical bilinear pooling. Furthermore, knowledge from the Wide_ResNet101 and DenseNet169 is transferred to MFFA-Net using knowledge distillation techniques. Finally, to validate the effectiveness of our proposed method, we conducted rigorous experiments on the OBC306, OBC265, and EOBC datasets, comparing our results with those of existing methods. The experimental results demonstrate that our method obtains remarkable performance in oracle character recognition, reaching state-of-the-art Top–1 accuracy with 92.42%, 94.78%, and 98.82% on the OBC306, OBC265, and EOBC datasets, respectively.

Keywords: Oracle Recognition · Attention · Feature Fusion · Knowledge Distillation

1 Introduction

Oracle bone script recognition (OBS) employs computer technology to classify oracle bone inscription (OBI) images, which hold crucial significance in various fields, such as archaeology and palaeography. However, OBI images are noisy and fragmented due to the prolonged burial and erosion of the tortoise bones. In addition, during the Yin and Shang dynasties, script norms were non-uniform,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 269–284, 2025.
https://doi.org/10.1007/978-3-031-78312-8_18

and the ongoing evolution of the script led to numerous variants of each oracle bone character. Furthermore, a few categories account for a significant portion of the total number of OBI images, resulting in long-tail distribution issues. All these challenges collectively hinder the accurate identification of oracle bones.

To solve the above problems, various researchers have carried out research on oracle recognition using traditional methods and deep learning methods respectively. Among them, [1] fused low–level features extracted by Gabor Filter with middle–level features obtained from sparse encoders, utilizing convolutional neural network for classification, leading to promising recognition results. Although it obtained a high accuracy rate, it used copied oracle bone characters rather than the original oracle bone topographies. Besides, [2] employed cross–modal deep metric learning approach and Generative Adversarial Network (GANs), achieving an accuracy of 86.7% on the OBC306 dataset. Despite these advancements, existing methods still encounter challenges in accurately recognizing partially similar OBI images. Although these techniques have achieved promising results, they often suffer from low accuracy and poor generalization when dealing with OBI images containing significant noise, cracks, missing fonts, numerous variant characters, and uneven category distribution.

In this paper, we introduce a multi-scale feature fusion attention network (MFFA-Net) as a solution to the aforementioned challenges. Given the rectangular structure of oracle bone characters and the high similarity among different classes, we innovate by incorporating asymmetric convolution and developing a multi-scale fusion approach. Additionally, utilizing knowledge distillation compression technology to guide the learning of MFFA-Net. The primary contributions of this paper are outlined as follows:

- The OBC265 and EOBC datasets were constructed based on the OBC306 dataset. In creating OBC265, we corrected misclassified images and eliminated those contaminated with noise that was challenging for the human eye to distinguish, ultimately yielding a dataset encompassing 265 distinct categories. Furthermore, the EOBC dataset builds upon OBC265 through a series of data augmentation techniques, resulting in a comprehensive collection of 483,805 images.
- The introduction of asymmetric convolution reduces the information redundancy and increases the feature representation capability of square convolution. Furthermore, by introducing coordinate attention, we enrich the model's focus on spatial information. By designing a multi–scale feature fusion module, we achieve a robust fusion of low-level, medium-level, and high-level features. This enables our network to capture detailed position information while emphasizing semantic content during feature extraction, thereby boosting the recognition accuracy of OBS.
- After using the knowledge distillation model compressiontechnique, the generalisation ability of MFFA-Net and the recognition of oracle bone characters were enhanced.

2 RELATED WORKS

The objective of OBS recognition is twofold: categorizing oracle bone characters and deciphering their meanings. Currently, the identification technology of OBS can be divided into two types: traditional recognition techniques and deep learning-based techniques.

2.1 Traditional OBS Recognition

Traditional OBS recognition technology primarily involves extracting features from OBI fonts or topological graphics. [3] treated oracle bone characters as undirected graphs, proposing a multi-level graph theory feature code for hierarchical OBS recognition. [4] considered OBI as graphical symbols, utilizing Fourier descriptors derived from contour curvature histograms as novel features to calculate similarity. However, the feature description obtained by this method is not sufficiently accurate. [5] utilized the graph isomorphism determination algorithm for OBS recognition. However, it's algorithm complexity is high. [6] suggested employing the Hough transforms and clustering to extract line feature points and calculating the corresponding minimum distance to recognition, but there was not enough experimental work.

2.2 Deep Learning-based OBS Recognition

Since AlexNet [7] secured victory in the 2012 ILSVRC [8], deep neural networks have made significant strides in vision tasks such as object recognition, image classification, and semantic segmentation. As a result, an increasing number of scholars are exploring the application of deep convolutional neural networks in OBS recognition. Deep learning technology can enhance the accuracy and efficiency of OBS recognition by automatically extracting features from OBI images, eliminating the need for human-defined features or criteria.

The oracle bone character dataset OBC306 [9], boasting the most labelled samples to date, was jointly developed by Anyang Normal University and the South China University of Technology. In their tests on this dataset, they employed neural networks such as VGG-16, ResNet-50, and Inception-v4 to benchmark the recognition accuracy of each network, setting a standard for future research. [10] constructed an ancient Chinese character image dataset (ACCID), encompassing annotations at both the radical and character levels, and proposed a baseline method for zero-shot OCR. The experimental outcomes quantitatively and qualitatively validate the efficacy of ACCID and the baseline model. To mitigate the long-tail distribution issue, [11] proposed a generative adversarial framework to augment oracle characters in problematic classes. Experimental results showcased the notable performance of the proposed algorithm in oracle character recognition. Moreover, [12] proposed a novel unsupervised domain adaptation method, which achieved state-of-the-art result on the Oracle-241 dataset surpassing the recently proposed network by 15.1%.

[13]explored the integration of mixup data augmentation with a triplet loss for further improvement.

On two oracle bone character datasets, [14] conducted experiments utilizing nearest neighbor classification and deep metric learning algorithms, resulting in accuracy rates of 92.43% and 83.47%, respectively. Additionally, by combining category masking with automatic recognition correction, [15] pioneered the integration of OBS detection and recognition.

[16] proposed a structure-texture separation network (STSN), an end-to-end learning framework for joint disentanglement, transformation, adaptation and recognition. Experiments on Oracle-241 dataset demonstrated that STSN outperforms other adaptation methods, effectively boosting recognition performance.

3 METHODOLOGY

This paper adopts ResNet18 [17] as the backbone network and further enhances its performance. The overall architecture of the proposed network is illustrated in Fig. 1. Below, we provide a detailed description of our method.

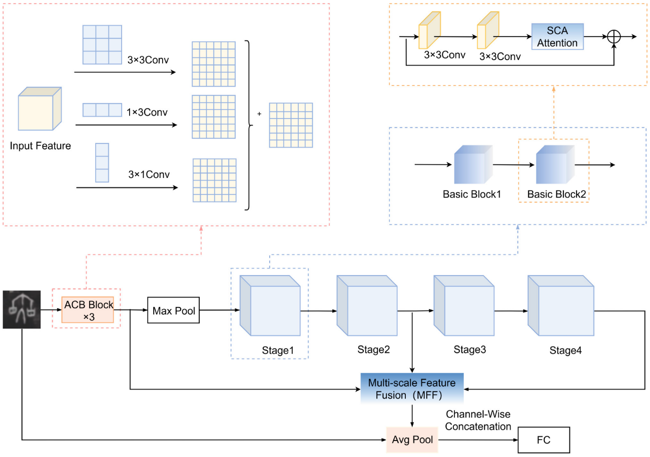


Fig. 1. The network architecture.

3.1 Asymmetric Convolutional

The length-to-height ratio of the oracle bone character is mostly 1:2, while the majority of conventional convolutional kernels are square. Compared with symmetric convolution, asymmetric convolution has a different perceptual field, and can effectively make use of the four-neighbourhood framework information in the feature image. [18] raised the asymmetric convolutional network (ACNet), whose core is the Asymmetric Convolution Block (ACB). ACB attains efficient feature extraction via parallel K×K, 1×K and K×1 convolution kernels. In this paper, we replace ResNet18's initial 7×7 convolution with three 3×3 standard convolutions. Subsequently, these three 3×3 convolutions are substituted with ACBs, aiming to capture the directional features of oracle characters more efficiently. Fig. 1 illustrates the design of ACB.

3.2 Spatial Coordinate Attention Mechanism

The goal of OBS recognition is to accurately classify the oracle bone characters present in an image. However, oracle bone images often contain irrelevant background and noise, and they face the problem of high similarity between distinct classes and low similarity between similar classes. Attention mechanisms enable the network model to focus on relevant local information, aiding in oracle bone identification. [19] introduced a novel type of attention mechanism known as Coordinate Attention (CA). It can acquire location information and enhance the target localization capability of the network. The execution process of the CA mechanism can be summarized into four key steps:

1. For input x, we use two one-dimensional pooling kernels ((H, 1) and (1, W)) to encode each channel in the vertical and horizontal directions, respectively.
2. We concatenate the feature maps obtained from these two directions and feed them into a 1×1 convolutional layer for dimensionality reduction. The resulting batch-normalized feature map, denoted as F_1, is then passed through a sigmoid activation function to generate a feature map f.
3. f is split along the spatial dimension to yield two separate feature maps, each of which undergoes a 1x1 convolution operation. Subsequently, the attention weights for the feature map in both width and height dimensions are obtained via another sigmoid activation function.
4. Finally, the original feature map is weighted by the attention weights obtained in the previous step.

Given that distinguishing highly similar oracle bone characters requires the model to have excellent local detail feature extraction capabilities, this paper introduces an additional branch to the CA mechanism. In this branch, for the input feature maps, we first perform adaptive maximum pooling and adaptive average pooling operations for each channel, computing the maximum eigenvalue and average eigenvalue for each channel, respectively. Subsequently, a 1×1 convolution is applied to obtain the attention feature map, which is then passed through an activation function to generate the attention weights. Multiply this resultant map with the original feature graph to get the spatial attention output U. Finally, we add U to the original CA attention.

The enhanced CA attention, named SCA, is depicted in the implementation flow chart shown in Fig. 2. SCA can capture the most critical fine features of the oracle bone characters while effectively suppressing the background noise in the oracle bone images. This enhancement significantly boosts the network's feature extraction capability. In our approach, SCA is incorporated after the second convolution within the basic block of ResNet18. Additionally, SCA is employed in the multi-scale feature fusion module.

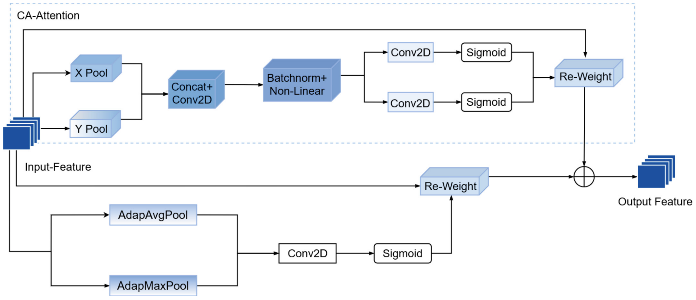


Fig. 2. Implementation process of SCA attention based on CA attention improvement.

3.3 Multiscale Feature Fusion

Given the absence of a standardized writing norm for oracle bone characters, their forms evolve continuously over time. Therefore, it is possible for the same text to have different writing styles, while unrelated texts may share significant similarities. This variability poses significant challenges for oracle bone recognition. Fine-grained image categorization is primarily concerned with discriminating highly similar objects on a macroscopic level, yet belonging to distinct subcategories on a more granular level. Upon careful analysis, it becomes evident that recognizing oracle bone characters that share strong similarities yet belong to different categories can indeed be framed as a fine-grained image classification task.

[20] introduced a hierarchical bilinear pooling(HBP) method, which can efficiently capture complementary information and inter-layer feature interactions from various convolutional layers. Given its relevance to extracting features for similar oracle bone characters, this work integrates HBP into the ResNet18 architecture. Furthermore, this paper proposes an Multiscale Feature Fusion(MFF) module based on HBP. The model structure of MFF is depicted in Fig. 3. Initially, the features extracted from the third asymmetric convolutional block, stage 2, and stage 4 of the ResNet18 are individually processed by the SCA mechanism. This ensures that the network focuses on the target regions while ignoring background noise. Subsequently, adaptive average pooling and 1 × 1 convolution are applied to standardize the feature sizes across different levels. Next, the feature map is up-dimensioned in the channel using 1 × 1 convolution, and the three features are integrated through elemental multiplication. Finally, high-dimensional features are compressed into compact representations

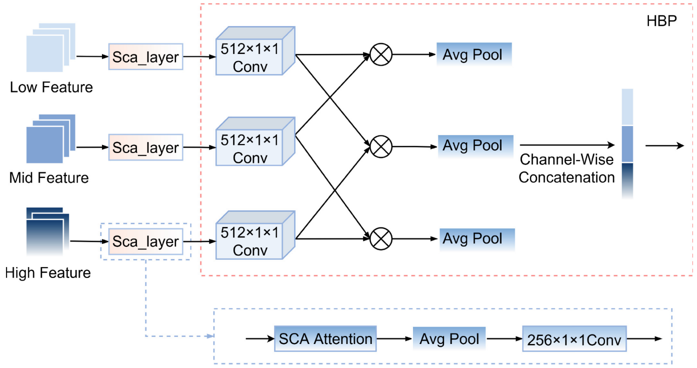


Fig. 3. Network Model Structure of MFF.

using adaptive averaging pooling, and concatenated along the channel dimension. The MFF module effectively fuses low-level, medium-level, and high-level features. This fusion approach helps preserve detailed information from oracle scripts, aiding in the discrimination of variant and similar characters within oracle scripts.

3.4 Multi-teacher Knowledge Distillation

Addressing the challenges posed by limited scenarios due to the intricate design of current models and their extensive parameter counts, this paper introduces knowledge distillation technology to the realm of oracle recognition for the first time. This technique enables the compression and optimization of complex models, significantly enhancing their generalization capabilities.

Knowledge distillation [21], employs a Teacher-Student framework, where the teacher model is more complicated than the student model, with more generalizable knowledge. To enhance the learning of MFFA-Net, this article applies two teacher models, namely Wide_ResNet101 and DenseNet169. Fig. 4 illustrates the training procedure of multi-teacher knowledge distillation: (1) Training teacher models, (2) Generating soft targets by utilizing high–temperature T, (3) Training the student model using soft and hard targets simultaneously.

Knowledge distillation introduces a temperature parameter T (serving as a modulator to fine-tune the softening level of the probabilistic output) into the original softmax function. By doing so, it augments the informational richness that each sample contributes to the student network, The refined SoftMax function is formulated as (1).

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}. \tag{1}$$

where q_i is the probability of each category output, and z_i denotes the probability of belonging to category I.

The loss function comprises two components: distillation loss, which corresponds to the soft target, and student loss, which corresponds to the hard target. Distillation loss is further divided into two categories: distillation loss 1 and distillation loss 2. Here, we present only the formula for distillation loss 1, as the formula for distillation loss 2 is similar to distillation loss 1. (2) is the formula for distillation loss.

$$L_{distill1} = -\sum_{i}^{N} p_i^T log(q_i^T) \tag{2}$$

Where

$$p_i^T = \frac{exp(v_i/T)}{\sum_k^N exp(v_k/T)}, q_i^T = \frac{exp(z_i/T)}{\sum_k^N exp(z_k/T)} \tag{3}$$

v_i and z_i represent the probabilities predicted by the teacher model and the MFFA-Net, respectively, indicating the likelihood of an image belonging to class I. Here, N denotes the total number of classes.

The student loss value is calculated by the cross-entropy between the SoftMax output and the ground truth value of MFFA Net at T=1, as shown in (4).

$$L_{student} = -\sum_{i}^{N} c_i log(q_i^1), \tag{4}$$

where c_i is the true value of class I, $c_i \in \{0, 1\}$. The final Loss function consists of three parts:

$$L = \alpha L_{distill1} + \beta L_{distill2} + \gamma L_{student}. \tag{5}$$

In the experimental setup, $\alpha = \beta = 0.2$, $\gamma = 0.6$.

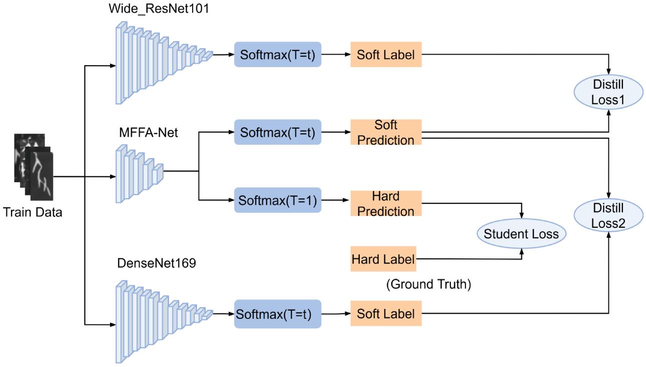


Fig. 4. Training process for multi-teacher knowledge distillation.

4 Experiments

4.1 Datasets

OBC306 Dataset OBC306, the first publicly available dataset with numerous oracle bone characters, is also the largest publicly available dataset for OBS identification research. It comprises 309,551 OBI images, encompassing 306 oracle bone character categories. Notably, a minority of high-frequency words, totaling 70 categories, accounting for 83.82% of the total image count. Conversely, the remaining 236 low-frequency words taking up 16.18%. In our study, we excluded 29 classes with a single sample from OBC306, utilizing the remaining 277 classes for our experiments. Fig. 5 presents a partial sample of the OBC306 dataset. Numerous images in this dataset are exceedingly noisy, rendering them unrecognizable. This poses a significant challenge for OBS recognition.

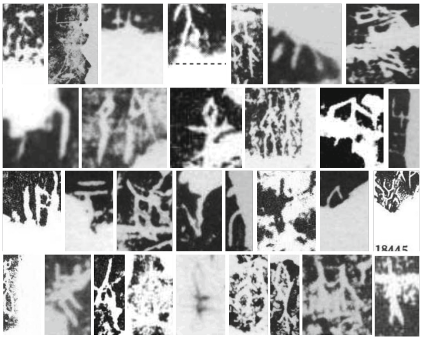


Fig. 5. Partial sample presentation of the OBC306 dataset.

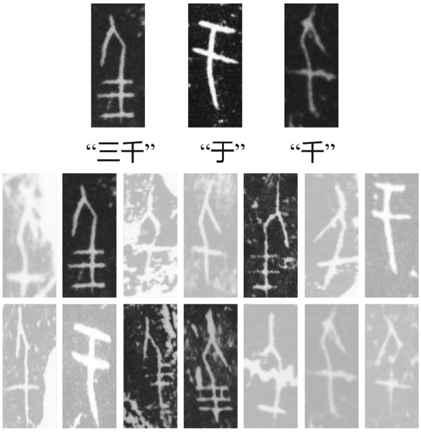


Fig. 6. Display of the characters “于”, “三千” and “千”, as well as the misclassification in the “三千” category. The grey-shaded areas are the misclassified images.

OBC265 Dataset In accordance with [22], this article has reorganized the OBC306 dataset, renaming it as OBC265. The reorganization process involves the following steps:

1) Remove images that were excessively noisy or severely damaged to the point of being indistinguishable by the human eye. This ensured the quality and clarity of the dataset.

2) Correcting misclassified OBI images. The OBC306 dataset contained some samples of misclassification, as illustrated in Figure 6. In the category of "三千", there are two types of images that do not belong to "三千": "于" and "千". Such errors have the potential to significantly undermine the accuracy of OBS recognition.
3) The deletion method was chosen for categories with a sample size of less than 10.

EOBC Dataset OBC265 has a severe issue of uneven sample distribution, as demonstrated in Fig. 7. To address the long-tailed distribution issue, this paper augments the train and validation sets corresponding to classes with less than 1000 samples in the OBC265 dataset. Data enhancement operations include rotate, contrast enhancement, affine transformation, erosion, etc. An example of a transformed image is presented in Figure 8(a). After a series of data augmentations, a total of 483,805 OBI images were obtained, and the augmented dataset is named EOBC. To validate the effectiveness of the data augmentation techniques, this paper utilizes the OBC265 test set as the EOBC test set.

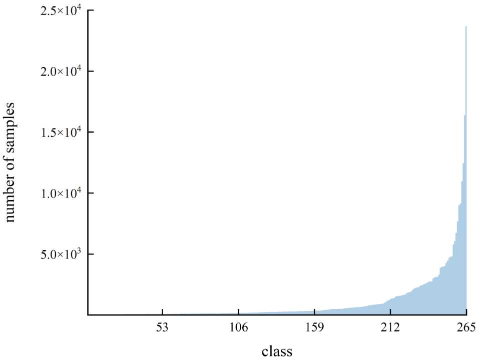


Fig. 7. Number of samples in each category.

4.2 Denoising

To minimize the adverse effects of background noise on OBS recognition, the images were denoised using bilateral filtering [23] and non-local means [24]. The comparison images before and after processing are shown in Fig. 8(b-c), clearly demonstrating the positive impact of the denoising process. The preprocessing operations were only used on the OBC265 and EOBC datasets, while the OBC306 dataset was unprocessed.

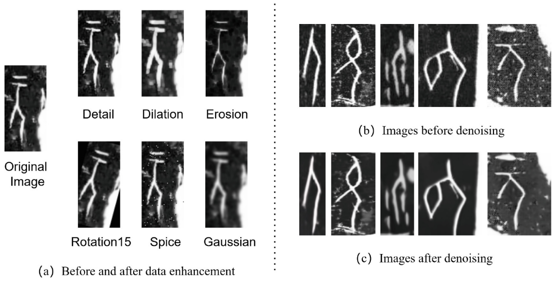

Original Image
Detail
Dilation
Erosion
Rotation15
Spice
Gaussian
(a) Before and after data enhancement
(b) Images before denoising
(c) Images after denoising

Fig. 8. Comparison of images before and after denoising and data enhancement.

4.3 Experimental Settings

In all relevant experiments conducted in this work, a uniform data format was employed. Specifically, all images were resized to 128 × 128 × 3 and subsequently normalized to the range of [0,1]. Set the batch size to 16 and the epoch to 15 during training. The AdamW optimization algorithm [25] was chosen to efficiently update the model parameters. To dynamically adjust the learning rate, the CosineAnnealingLR scheduler [26] was utilized, with a maximum number of iterations set to 3 and an initial learning rate of 0.001.

4.4 Experimental Results and Analysis

Table 1. Recognition results of different methods on the OBC306 dataset.

Model	Top–1 Acc	Top–5 Acc	Precision	Recall	F1–score	parameters
ShuffleNet_v2	89.715	97.376	74.309	68.446	70.261	**1.63**M
Wide_ResNet50	89.216	97.373	74.870	68.629	70.534	67.35M
DenseNet121	91.074	97.908	77.376	71.101	72.937	7.21M
MobileNet_v2	89.793	97.545	74.480	69.740	70.860	2.55M
ResNet50	89.569	97.431	76.174	70.508	72.203	24.05M
Wide_ResNet101	90.127	97.470	75.640	70.423	72.087	125.38M
DenseNet169	90.620	97.833	78.678	72.083	73.022	12.91M
Ours	**92.423**	**98.385**	**79.470**	**74.054**	**75.925**	12.85M

Compare with some typical CNN models In this section, MFFA–Net will be evaluated against renowned models such as ShuffleNet_v2, WideResNet50, DenseNet121, DenseNet169, MobileNet_v2, ResNet50 and Wide_ResNet101. This evaluation will be conducted on three datasets: OBC306, OBC265, and

Table 2. Recognition results of different methods on the OBC265 dataset.

Model	Top–1 Acc	Top–5 Acc	Precision	Recall	F1–score
ShuffleNet_v2	92.406	98.463	80.770	73.985	75.806
Wide_ResNet50	92.620	98.341	81.554	76.454	77.445
DenseNet121	94.317	98.840	81.992	78.974	79.593
MobileNet_v2	93.042	98.566	82.174	77.954	78.944
ResNet50	93.433	98.603	83.212	78.380	79.546
Wide_ResNet101	93.049	98.363	83.335	78.372	79.506
DenseNet169	93.917	98.710	84.750	80.516	81.298
Ours	**94.778**	**99.024**	**88.162**	**81.716**	**83.613**

Table 3. Recognition results of different methods on the EOBC dataset.

Model	Top–1 Acc	Top–5 Acc	Precision	Recall	F1–score
ShuffleNet_v2	94.564	99.169	87.190	96.295	90.305
Wide_ResNet50	97.835	99.797	95.911	98.860	97.111
DenseNet121	97.620	99.686	93.931	98.308	95.419
MobileNet_v2	95.399	99.375	87.609	97.019	90.841
ResNet50	98.108	99.749	96.066	98.774	97.124
Wide_ResNet101	98.005	99.793	96.476	98.532	97.157
DenseNet169	97.846	99.782	95.545	98.791	96.795
Ours	**98.821**	**99.863**	**97.112**	**99.389**	**98.012**

EOBC. The evaluation criteria will encompass Top–1 and Top–5 accuracy, Precision, Recall, and F1–score, ensuring a comprehensive assessment of our method's performance.

Tables 1, 2, and 3 present the experimental results on the OBC306, OBC265, and EOBC datasets, respectively. The highest scores are shown in bold, and the sub–highest scores are underlined. On the OBC306 dataset, our approach achieves a remarkable Top–1 accuracy of 92.423%, surpassing all other methods. Similarly, it demonstrates superiority in Precision, Recall, and F1–score, with respective improvements of 0.792%, 1.971%, and 2.903% compared to the sub-optimal model DenseNet169. On the Top–5 accuracy metric, our method also outperforms the competition, improving by 0.477% compared to the sub-optimal DenseNet121.

The trend continues on the OBC265 and EOBC datasets. Our method achieves the highest scores in all evaluation metrics, demonstrating its consistent superiority across different datasets. On the OBC265 dataset, the Top–1 accuracy of our approach stands at 94.778%, while on the EOBC dataset, it reaches an impressive 98.821%. These results firmly establish the state-of-the-art performance of our method compared to other leading models. The reason why our method outperforms other methods is that we take a multi-scale feature fusion approach fusing different levels of features and retaining rich semantic information. We also use the knowledge distillation model compression technique to further enhance the generalisation ability and performance of the model in this paper.

Meanwhile, we have conducted experiments on the OBC306 dataset regarding the parameters of the different methods to be used, and the results of the experiments are shown in Table1. It will be analysed next.In terms of the number of parameters in the model, this paper's method is much lower than that of Wide_ResNet101, which is about 125M, and this paper's method is only about 10% of it, but the accuracy of this paper's method is 4.6% higher than that of Wide_ResNet101, respectively. Although the number of parameters of this method is not as low as that of the ShuffleNet_v2 model, the accuracy of this method is 5.161% higher than that of the ShuffleNet_v2 model. Comprehensively, the method in this chapter can maintain a high level of oracle bone character recognition effect while effectively reducing the number of parameters in the model, which verifies the effectiveness and practicality of the method in this paper.

Comparison with Existing Oracle Image Domain Methods To validate the advancement of the method proposed in this paper, comparison experiments with some latest algorithms with better comprehensive identification performance are carried out on the public dataset OBC306. The comparison results are detailed in Table 4.

Table 4. Verification of model advancement.

Methods	Number of categories	Top-1 Accuracy
[2]	241	86.7
[9]	306	70.28
[11]	277	93.86
[12]	241	62.2(Scan),93.6(Handprint)
[13]	277	91.59
Ours	277	92.423
Ours	265	**94.778**

From Table 4, it's evident that among the comparison algorithms tested on the same public dataset, they recognize a minimum of 241 classes and up to 306 classes. Notably, when the number of recognition classes for this paper's algorithm is set at 265, it achieves the highest recognition rate in terms of performance. However, when the number of recognition classes was set to 277, our method ranked second, trailing slightly behind the algorithm proposed in [11]. We think the probable reason for this discrepancy lies in the fact that

[11] employed Generative Adversarial Networks for data augmentation on the OBC306 dataset. This augmentation technique likely led to a more balanced distribution of samples, resulting in superior recognition performance. Despite this minor gap, it is worth emphasizing that the method outlined in this paper still demonstrates effectiveness.

Ablation Experiment To further investigate the impact of different modules on the overall performance of our method, we conducted ablation experiments. The ablation study results are presented in Table 5.

Effectiveness of ACB. The introduction of ACB resulted in a notable enhancement in overall performance. Compared to the baseline, the utilization of ACB led to improvements in Top-1 accuracy, Precision, Recall, and F1-score by 1.038%, 2.934%, 4.139%, and 3.301%, respectively. This suggests that asymmetric convolution proves more effective than square convolution in extracting features from rectangular oracle images.

Effectiveness of MFF. By incorporating the multi-scale feature fusion module (MFF) with CA attention introduced in this study, we observed a significant enhancement in Top-1 accuracy and Precision, with respective improvements of 0.477% and 1.809% compared to the baseline. This clearly demonstrates the efficacy of the MFF module. Furthermore, when leveraging the enhanced CA mechanism, namely SCA, we witnessed even more improvements in performance. Specifically, Top-1 accuracy, Top-5 accuracy, Recall, and F1-score increased by 0.107%, 0.288%, 1.742%, and 2.42%, respectively. These findings further corroborate the effectiveness of the SCA attention mechanism we proposed.

Effectiveness of Multi-teacher Knowledge Distillation. When compared to MFFA-Net without knowledge distillation, the model incorporating multi-teacher knowledge distillation exhibited remarkable improvements in multiple performance metrics. Specifically, there was an increase of 0.391% in Top-1 accuracy, 0.292% in Top-5 accuracy, 3.424% in Precision, 2.645% in Recall, and 3.164% in F1-score. These findings clearly demonstrate the significant role of multi-teacher knowledge distillation in enhancing various evaluation indicators, highlighting its crucial importance in improving model performance.

Table 5. Ablation studies of different components on the OBC265 dataset. All ratings are reported as percentages.

Configurations	Top-1	Top-5	Precision	Recall	F1–score
ResNet18	92.783	98.333	79.783	76.274	77.077
+ACB Block	93.821	98.577	82.717	80.413	80.378
+MFF (CA)	93.260	98.282	81.592	75.185	77.055
+MFF (SCA)	93.367	98.570	84.598	76.927	79.475
+ACB+MFF (SCA)	94.387	98.732	84.738	79.071	80.449
+ACB+MFF (SCA) +KD	94.778	99.024	88.162	81.716	83.613

5 Conclusion

In this paper, we introduce the MFFA-Net, a framework designed to capture the intricate multi-scale information inherent in OBI images. Firstly, we incorporate asymmetric convolution into the Resnet architecture, enabling it to effectively extract features from rectangular OBI images in both width and height directions. Secondly, we design a multi-scale feature fusion module by combining the improved coordinate attention and bilinear pooling mechanism. In addition, we leverage the knowledge distillation technique to transfer the knowledge acquired from sophisticated models such as Wide_ResNet101 and DenseNet169 to the MFFA-Net, significantly enhancing its generalization capabilities and OBS recognition performance. Extensive experiments on three datasets validate the superiority of our proposed MFFA-Net, outperforming other state-of-the-art models by a significant margin.

Acknowldgements. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62266044 and 61862061. It was also supported by the "Tianshan Talents" Leading Talents Program for Scientific and Technological Innovation in Xinjiang Uygur Autonomous Region(2023TSYCJ0025), and the Open Project of Key Laboratory of Oracle Bone Inscription Information Processing, Ministry of Education (OIP2021E004).

References

1. Guo, J., Wang, C.H., Roman-Rangel, E., Chao, H.Y., Rui, Y.: Building hierarchical representations for oracle character and sketch recognition. IEEE Trans. Image Process. **25**(1), 104–118 (2016)
2. Zhang, Y.K., Zhang, H., Liu, Y.G., Liu, C.L.: Oracle character recognition based on cross-modal deep metric learning(in Chinese). Zidonghua Xuebao Acta Auto. Sin. **47**(4), 10 (2021)
3. F. Yang, “Recohnition of jia gu wen based on graph theory(in Chinese),” *J. Electron. Inf. Technol.*, 1996
4. X. Q. Lv, N. Mo, K. W. Cai, X. Wang, and Y. M. Tang, “A graphic-based method for chinese oracle-bone classification(in Chinese),” *Journal of Beijing Information Science and Technology University*, 2010
5. Li, Q.S., Yang, Y.X.: Recognition of inscriptions on bones or tortoise shells based on graph isomorphism(in Chinese). Comput. Eng. Appl. **47**(8), 112–114 (2011)
6. L. Meng, “Recognition of oracle bone inscriptions by extracting line features on image processing,” in *Proc. Int. Conf. Pattern Recognit. Appl. Methods*, 2017
7. A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” *Adv. Neural. Inf. Process. Syst.*, vol. 25, no. 2, 2012
8. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, “Imagenet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, pp. 1–42, 2014
9. S. P. Huang, H. B. Wang, Y. G. Liu, X. S. Shi, and L. W. Jin, “Obc306: A large-scale oracle bone character recognition dataset,” in *Proc. Int. Conf. Doc. Anal. Recognit.*, pp. 681–688, 2019

10. Diao, Xiaolei and Shi, Daqian and Li, Jian and Shi, Lida and Yue, Mingzhe and Qi, Ruihua and Li, Chuntao and Xu, Hao,"Toward Zero-shot Character Recognition: A Gold Standard Dataset with Radical-level Annotations,"in *Proceedings of the 31st ACM International Conference on Multimedia.*, pp. 6869-6877, 2023
11. Li, Jing and Wang, Qiu Feng and Huang, Kaizhu and Yang, Xi and Zhang, Rui and Goulermas, J. Y , "Towards better long-tailed oracle character recognition with adversarial data augmentation,"*Pattern Recognit.*,vol. 140, pp. 109534, 2023
12. Mei Wang, Weihong Deng, Sen Su, "Oracle character recognition using unsupervised discriminative consistency network,"*Pattern Recognit.*,vol. 148, 2024
13. J. Li, Q. Wang, R. Zhang, K. Huang,"Mix-up augmentation for oracle character recognition with imbalanced data distribution," in *International Conference on Document Analysis and Recognition. Springer, Cham*, pp. 237-251, 2021
14. Y. K. Zhang, H. Zhang, Y. G. Liu, Q. Yang, and C. L. Liu, "Oracle character recognition by nearest neighbor classification with deep metric learning," in *Proc. Int. Conf. Doc. Anal. Recognit.*, pp. 309–314, 2019
15. S. Yan, F. Liu, D. M. Sun, and H. B. Li, "Knowledge diffusion and activating inheritance of the oracle bone script based on artificial intelligence in museums(in Chinese)," *Chinese Museum*, vol. 000, no. 003, pp. P.110–116,144, 2021
16. Wang, Mei and Deng, Weihong and Liu, Cheng-Lin,"Unsupervised Structure-Texture Separation Network for Oracle Character Recognition," *IEEE Trans. Image Process.*, vol. 31, pp. 3137-3150, 2022
17. K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016
18. X. H. Ding, Y. C. Guo, G. G. Ding, and J. G. Han, "Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks," 2019
19. Q. B. Hou, D. Q. Zhou, and J. S. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 13708–13717, 2021
20. C. J. Yu, X. Y. Zhao, Q. Zheng, P. Zhang, and X. G. You, "Hierarchical bilinear pooling for fine-grained visual recognition," in *Proc. Eur. Conf. Comput. Vis.*, September 2018
21. Gou, J.P., Yu, B.S., Maybank, S.J., Tao, D.-C.: Knowledge distillation: A survey. Int. J. Comput. Vision **129**, 1789–1819 (2021)
22. Key Laboratory of Oracle Bone Inscriptions Information Processing, Ministry of Education of China. "Yin qi wen yuan," http://jgw.aynu.edu.cn/, 2019
23. Papari, G., Idowu, N., Varslot, T.: Fast bilateral filtering for denoising large 3d images. IEEE Trans. Image Process. **26**(1), 251–261 (2016)
24. Ballester, Coloma, Fedorov, and Vadim, "Affine non-local means image denoising," *IEEE Trans. Image Process.*, 2017
25. I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent., ICLR*, 2017
26. I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," in *Proc. Int. Conf. Learn. Represent., ICLR - Conf. Track Proc.*,2016

Check for updates

Combining Frequency-Based Smoothing and Salient Masking for Performant and Imperceptible Adversarial Samples

Amon Soares de Souza[1](✉), Andreas Meißner[2], and Michaela Geierhos[1]

[1] Research Institute CODE, University of the Bundeswehr Munich,
Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
amon.soares@unibw.de

[2] ZITiS, Big Data, Zamdorfer Str. 88, 81677 Munich, Germany

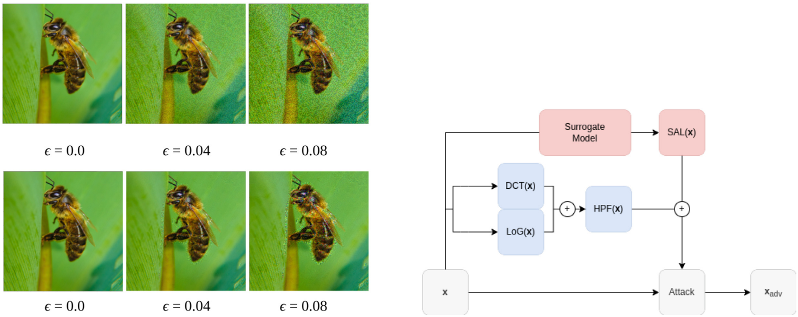
Abstract. Adversarial attacks provide a simple and effective way to fool neural networks by applying subtle perturbations to the network's input. However, to ensure a misclassification by an image classifier, the attacker must often apply a significant amount of perturbation to the input image, resulting in the characteristic noisy appearance of adversarially perturbed images. This essentially reveals the attack to the human visual system, limiting the use of adversarial attacks to applications without human supervision. To address this issue, we present a novel approach to disguise adversarial attacks on images with high-pass filtering based on some assumptions of JPEG compression. Unlike other smoothing approaches based on variation, we not only provide the ability to locally adjust the amount of distortion, but also incorporate information about salient regions to preserve the attack information in critical parts of the input. Our frequency-aware method provides a more flexible attack and higher imperceptibility compared to its vanilla counterparts. At the same time, it preserves most of the attack performance, occasionally even outperforming the standard attack. Finally, our model allows for superior performance retention compared to related attack smoothing approaches due to the inclusion of salient regions of the surrogate model, while achieving smoothing results comparable to the state-of-the-art. The code to reproduce the experiments can be found here: https://github.com/amonsoes/salient-hpf.

Keywords: Image Classification · Adversarial Attacks · Smoothing

1 Introduction

In image processing, adversarial examples are obtained by making per-pixel adjustments to the input image [1,12,21]. This effectively creates a noise pattern that can be subtle or quite easy to detect, depending on the size of the perturbation [1,24]. The attacker must choose a reasonably high perturbation magnitude to ensure a successful attack. This often results in an image where

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 285–302, 2025.
https://doi.org/10.1007/978-3-031-78312-8_19



a Comparison of noise patterns

b Graph of our HPF attacks

Fig. 1. (a) This figure shows a comparison of the resulting noise patterns for different perturbation magnitudes ϵ. The adversarial examples in the top row are generated with vanilla FGSM, while those in the bottom row are generated with FGSM using the proposed HPF extension. (b) A graph of our HPF extension. Elements in the standard attack are colored gray, smoothing elements are colored blue, and performance preserving elements are colored pink.

the characteristic noise pattern is easy to detect by human perception since the noise pattern tends to become more obvious with a higher adaptation rate [24]. To address this problem, we present a novel approach that hides adversarial attacks by creating a mask that reduces or increases the per-pixel adaptation in certain regions depending on the local frequency information. These regions are chosen based on assumptions about the capabilities and limitations of human visual perception that form the basis of compression techniques such as GIF and JPEG [9].

Exploiting this shortcoming of the human visual perception system, we design a high-pass filter (HPF) by using a combination of the discrete cosine transform (DCT) and the Laplacian of Gaussian (LoG) filters commonly used in edge detection algorithms [19]. The HPF mask scales down the per-pixel adjustment in low-frequency regions while maintaining the full perturbation magnitude for pixel adjustments in high-frequency regions. In addition, we incorporate information about salient regions of the surrogate model, which further improves performance. Applying this concept to image classification, we find that our approach makes the attack dramatically less perceptible to the human eye while retaining most of the effectiveness of the attack. In some cases, the extended attack even outperforms the vanilla attack in performance while being less disruptive. For a comparison of the standard attack and our extended attack, see Figure 1a. Our contributions are as follows:

- We present a novel smoothing approach that makes adversarial attacks in images less perceptible. Although frequency-based smoothing has been introduced before for attacks based on optimization [16,24], this approach successfully includes variation computation into gradient-projection attacks like BIM [12], providing smoothing for attacks with other use cases.

- We achieve superior attack retention for the extended attacks in our experiments by incorporating salient regions of the surrogate model, leading to attack performance preservation in regions that are critical to the attack performance.
- We address the previously unaddressed problem of frequency-based smoothing in lossy compression scenarios by designing boosting methods and combining frequency filtering with chrominance isolation techniques which, to the best of our knowledge, has not been attempted before.
- We show that this new approach can be used as a straightforward extension for most existing pixel-based adversarial attacks. To prove that our samples are smoother, we measure perceptibility using the most apparent distortion (MAD) metric [14] and conduct a user study that supports our claim and demonstrates the effectiveness of our approach.

2 Related Work

2.1 Adversarial Attacks

The goal of adversarial attacks is to generate a malicious example $\boldsymbol{x}^{adv}$ from an input $\boldsymbol{x}$ that causes a DNN to predict either a predefined target class y_t or any class $y \in Y \setminus \{y_{true}\}$ that is not the true class y_{true}. To accomplish this, it uses the gradient estimation of a DNN trained on similar data [1,12,21,24].

It is by the definition of the gradient estimation network $\hat{\phi}$ and the target network ϕ that one usually separates white-box attacks and black-box attacks. In a white-box attack, $\hat{\phi} = \phi$, which means that the attacker has access to the network and its weights. In a black-box attack, the network type and its weights are unknown, so $\hat{\phi} \neq \phi$ [21].

White-Box Attacks and Black-Box Attacks. By linearizing the cost function in the input space and performing pixel-wise perturbations in a single step, the Fast Gradient Sign Method (FGSM) [12] generates an adversarial example. Let $\boldsymbol{x}^{adv}$ be the perturbed version of the image $\boldsymbol{x}$, ϵ the perturbation magnitude, $\nabla_{\boldsymbol{x}} J$ the gradient of the loss function J with respect to $\boldsymbol{x}$, and $sign()$ a function that returns the sign of the input [12].

$$\boldsymbol{x}^{adv} = \boldsymbol{x} + \epsilon \cdot sign\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y; \theta)\right) \tag{1}$$

An iterative variant of this method has been proposed in [12]. The Basic Iterative Method (BIM) extends the idea of slightly changing the input with a gradient estimation in a single step to a multi-step variant.

$$\boldsymbol{x}_{t+1}^{adv} = clip_{\boldsymbol{x},\epsilon}(\boldsymbol{x}_t^{adv} + \alpha \cdot sign(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y; \theta))) \tag{2}$$

$$clip_{\boldsymbol{x},\epsilon}(\boldsymbol{x}^{adv}) = min(\boldsymbol{x} + \epsilon, max(\boldsymbol{x}^{adv}, \boldsymbol{x} - \epsilon)) \tag{3}$$

For each time step $t \in T$, BIM applies pixel-wise perturbations with a step size of α. To limit the perturbation size to ϵ, $clip()_{\boldsymbol{x},\epsilon}$ lets every pixel value be within the ϵ neighborhood of the original image $\boldsymbol{x}$ [12].

Others [21] build on the iterative approach by adding a momentum term to each perturbation step. Similar to the usage for adaptive optimization, the momentum term contains gradient information from previous steps up to iteration t. Let μ be a decay factor and $\boldsymbol{g}_t$ be the momentum term at time step t [21].

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y; \theta)}{||\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y; \theta)||_1} \tag{4}$$

$$\boldsymbol{x}_{t+1}^{adv} = \boldsymbol{x}_t^{adv} + \alpha \cdot sign(\boldsymbol{g}_{t+1}) \tag{5}$$

White-box attacks are inefficient in black-box settings without additional modifications [21]. To increase the portability of iterative gradient-based attack variants, Wang et al. [21] propose a new method called variance tuning. Their approach builds on the iterative variant and adds a variance term in addition to the momentum term. It uses gradient information in the neighborhood by randomly sampling $\boldsymbol{x}^i$ in a predefined range of $\boldsymbol{x}$ from the previous data point to adjust the gradient of the current data point at each iteration [21].

$$V(\boldsymbol{x}) = \frac{1}{N}\sum\nolimits_{i=1}^{N} \nabla_{\boldsymbol{x}^i} J(\boldsymbol{x}^i, y; \theta) - \nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y; \theta) \tag{6}$$

$$\boldsymbol{v}_{t+1} = V(\boldsymbol{x}_t^{adv}) \tag{7}$$

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t \frac{\hat{\boldsymbol{g}}_{t+1} + \boldsymbol{v}_t}{||\hat{\boldsymbol{g}}_{t+1} + \boldsymbol{v}_t||_1} \tag{8}$$

2.2 Attack Smoothing

Applying adversarial attacks to images with a strong magnitude usually results in a pattern that is easy to detect. Work on attack smoothing often attempts to hide this perturbation pattern by applying frequency transformations [10,16,24]. Another approach is to perturb the image with an L_0 constraint, which means that only a few pixels are changed [8]. To address the issue of visible noise patterns in adversarial attacks, Jia et al. [10] apply changes to the spectrum of images by computing the DCT. Zhang et al. [24] have proposed a perturbation smoothing method closely related to our approach. They present a technique for generating perturbed images that locally match the smoothness (or roughness) of the original image. A similar method has been proposed in [16], where instead of scaling perturbations locally by means of frequency transformations, they compute regions of high variation and hide the perturbations in these regions. Croce and Hein [3] use constraints based on local variation to generate smooth black-box adversarial samples. Recently, Luo et al. [17] presented an attack that produces smooth samples by perturbing on semantic similarity and using a discrete wavelet transform in the constrained optimization. Similar to our approach, Liu et al. [15] used a combination of salient regions and the DCT to compute smooth adversarial samples.

Our method is conceptually similar. However, it differs from these approaches:

- Like the related work on frequency-based smoothing [16,24], we integrate frequency information to locally scale the attack, but we additionally include salient regions of the surrogate model to preserve attack performance.
- Like Liu et al. [15], we use salient regions to identify critical areas for the perturbation and use a DCT, but we do not need to perform a search to determine the frequency cut-off and use the DCT localized. Additionally, we include the LoG for a more diverse masking computation. Lastly, we also include black-box evaluations, which are not present in [15].
- Unlike Zhang et al. [24] and their sC&W attack, our mask can smooth a variety of other attacks and does not require difficult integration into the attack's loss function. We prove that a much simpler a-posteriori smoothness constraint by masking works as well.
- Similar to Jia et al. [10], we hide the attack by using the DCT of the image. However, we perturb directly in the RGB space.
- Different to Croce et al. [3], our method extends other attacks. Our extension also includes information about salient regions from the surrogate model.
- Some attacks perturb the input minimally [1,16,17,24]. However, such attacks usually require many iterations to converge, whereas ours requires only one iteration in the case of HPF-FGSM.
- In addition, we present a novel way to locally perturb samples not only on the basis of frequency but also on the basis of color, which, to the best of our knowledge, has not been attempted before.

3 Approach

3.1 HPF Mask

To ensure a misclassification on a given input, an adversary typically uses a high perturbation magnitude for an adversarial attack, which makes the changes perceptible to human observers [10,24]. Inspired by previous work on adversarial smoothing [16,24], our model aims to hide perturbations in regions with many high-frequency components. Our approach constructs a mask based on local frequency components and information about salient regions from the surrogate model.

DCT and LoG Coefficient Mask. Similar to JPEG compression, we perform a patch-wise DCT, where each patch is 8×8 (4×4 for small image resolution such as CIFAR data [11]) pixels in size. 8×8 pixels is optimal for capturing pixel dependencies at smaller resolutions (our images are scaled to 224×224), while perturbations for larger images may require larger window sizes $p \times p$ [9].

An 8×8 pixel patch results in an 8×8 matrix of coefficients representing the frequency components of the original patch.

$$\boldsymbol{X}_{i,j}^{coeff} = \sum_{i=0}^{p} \sum_{j=0}^{p} |\boldsymbol{X}_{i,j}^{masked}| \tag{9}$$

$$\boldsymbol{X}_{i,j}^{masked} = \begin{cases} \boldsymbol{X}_{i,j}^{DCT} & max(i,j) > \psi, \\ 0 & else \end{cases} \tag{10}$$

Where $\boldsymbol{X}^{coeff} \in \mathbb{R}^{8\times 8}$ are the corresponding perturbation coefficients for the original patch $\boldsymbol{X}$, $\boldsymbol{X}^{DCT}$ is the transform of the original patch, and ψ defines the cutoff of the low frequency components. This procedure is now called DCT_{mask}.

For more flexibility in computing appropriate coefficients, we construct a second mask LoG_{mask} by applying a Laplacian of Gaussian filter to the input image. By identifying areas of rapid intensity change, the LoG mask will have high values for edges and low values for areas of uniform intensity [6].

Combination and Tradeoff. The final HPF mask should contain coefficients to appropriately reduce the effect of the perturbation. Therefore, it makes sense to calculate these coefficients in $[0, 1]$. We tried several ways of combining them, including linearly combining the normalized masks with coefficients that add up to 1. However, it turns out that this adds an unnecessary layer of complexity and another set of hyperparameters. So the addition of the two masks is therefore simply projected to $[0, 1]$ by the *clip* function, which performs value clamping as $clip(x_i) = min(max(x_i, 0), 1)$. Figure 2 shows a visual representation of the masking process.

$$HPF(\boldsymbol{x}) = clip_{\boldsymbol{x}}(DCT_{mask}(\boldsymbol{x}) + LoG_{mask}(\boldsymbol{x})) \tag{11}$$

Including Salient Regions of the Surrogate Model. To preserve the performance of the attack, we include information about salient regions of the target model as a mask. Essentially, this mask should contain coefficients in [0,1] that define how much a data point affects the loss of the target model. Let SAL be a function that returns the normalized absolute value of the gradient of the loss function with respect to the input image $\boldsymbol{x}$

$$SAL(\boldsymbol{x}) = \frac{|\nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y; \theta)|}{max(|\nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y; \theta)|)} \tag{12}$$

The target model ϕ, which is used in the computation of $\boldsymbol{J}$ is only available in the white-box case, since $\phi = \hat{\phi}$. Due to the property of transferability of adversarial attacks [2], $\nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y; \theta) \approx \nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y; \hat{\theta})$, where $\hat{\theta}$ are the parameters of the surrogate model. This approach can therefore also be used for black-box attacks. However, this means that our method requires an approximation of the gradient of ϕ to work. The mask coefficients of $SAL(\boldsymbol{x})$ are added to the coefficients in $HPF(\boldsymbol{x})$ and clipped to [0,1].

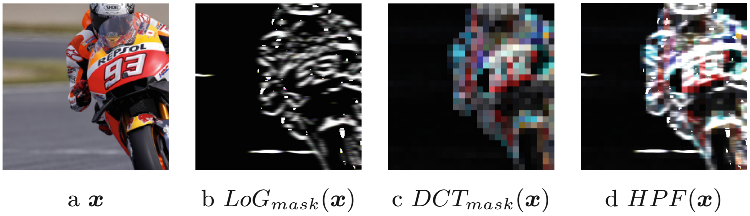

a $\boldsymbol{x}$ b $LoG_{mask}(\boldsymbol{x})$ c $DCT_{mask}(\boldsymbol{x})$ d $HPF(\boldsymbol{x})$

Fig. 2. Example of the HPF mask applied to an image. The last image shows the HPF mask, which is a clipped linear combination of the LoG mask and the DCT mask. White areas indicate full perturbation magnitude, while completely black areas indicate that no perturbation will be applied.

Epsilon Adjustment. Using the mask coefficients in the range [0,1] as scalars results in a weaker perturbation magnitude for most pixels. To account for this, we must adjust the total ϵ upward. For additional performance retention, we can adjust ϵ according to the mean of the inverted HPF mask. The same can be done with α for iterative attacks. However, the ϵ bounds must be set on a pixel-by-pixel basis to allow for a higher perturbation at high-frequency regions and lower ϵ at low frequencies. Note that even though we internally adjust ϵ, our attack is *always* less perturbing according to the CAD metric (see Section 4).

$$HPF_{inv}(\boldsymbol{x}) = 1 - HPF(\boldsymbol{x}) \tag{13}$$

$$\epsilon_{HPF} = \epsilon(1 + \mu(HPF_{inv})) \tag{14}$$

3.2 Extending Attacks

The HPF mask provides a simple extension for many pixel-based attacks. It locally scales the perturbation magnitude of both single-step methods and iterative methods to obtain the desired perturbation in high-frequency regions and less adjustment in low-frequency regions. For a single-step method like FGSM, the mask coefficients scale the perturbation magnitude ϵ directly.

$$\boldsymbol{x}^{adv} = \boldsymbol{x} + \epsilon_{HPF} HPF(\boldsymbol{x}) \cdot sign\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y; \theta)\right) \tag{15}$$

Where $HPF(\boldsymbol{x})$ is defined as in Equation 11 (including the salient mask), and $J(\boldsymbol{x}, y; \theta)$ is the loss determined by some network ϕ_θ. Similarly, iterative attacks can be extended by scaling the step size α, which is adjusted at each iteration by the HPF mask.

$$\boldsymbol{x}_{t+1}^{adv} = clip_{\boldsymbol{x},\epsilon}\left(\boldsymbol{x}_t^{adv} + \alpha_{HPF} HPF(\boldsymbol{x}) \cdot sign\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y; \theta)\right)\right) \tag{16}$$

Where $clip_{\boldsymbol{x},\epsilon}$ is defined as in the calculation in BIM [12] in Equation 3.In equations Equation 15 and Equation 16, the subscript HPF denotes the adjustment of the perturbation parameter according to Equation 14.The added perturbation is scaled by the high and low frequency characteristics of the original image $\boldsymbol{x}$. Thus, it is not necessary to recompute the HPF mask at each iteration, which minimizes the computational effort required and dramatically improves performance. This procedure is not limited to FGSM and BIM, but works with most pixel-based perturbations.

3.3 LF and CbCr Boosting

Although the HPF variants perform unexpectedly well on their own (even in black-box scenarios), they can be further improved at the cost of some additional attack visibility. This is especially helpful in cases where we can expect some sort of lossy compression, which typically removes high frequency details [9]. We introduce two boosting variants that reintroduce a small amount of noise in the low to mid frequency range.

LF Boosting. LF boosting aims to reintroduce some perturbation into the mid-to-low frequency ranges according to their frequency distribution. The HPF mask can be inverted to specifically target and boost the mid-to-low frequencies. Since high frequency components will have high coefficients, the inverted mask $HPF_{inv}(\boldsymbol{x})$ ensures that high frequency components are not affected by the LF boost. Let $\rho \in [0, 1]$ be a scaling factor that controls the amount of LF boosting.

$$HPF_{LF}(\boldsymbol{x}) = \rho HPF_{inv}(\boldsymbol{x}) + HPF(\boldsymbol{x}) \tag{17}$$

CbCr Boosting. JPEG compression also downsamples color information by a predefined factor. This is possible because of the previous color conversion from RGB to YCbCr, where Y contains brightness information and (Cb, Cr) contains all color information. This can be used for boosting since Cb and Cr can be perturbed with a higher magnitude without increasing the perceptibility of the attack. We define a function $h : \mathbb{R}^{c \times m \times n} \rightarrow \mathbb{R}^{c \times m \times n}$ which maps an image $\boldsymbol{x}_{RGB}$ to its counterpart in YCbCr $\boldsymbol{x}_{YCbCr}$ and its inverse[1] h^{-1}. Instead of working directly in the YCbCr color space, we obtain the HPF mask and the gradient estimation from the original RGB input $\boldsymbol{x}_{RGB}$. After obtaining the intermediate adversarial example $\hat{\boldsymbol{x}}^{adv}$ from each attack, we map it to YCbCr with h and obtain the perturbation map $\boldsymbol{\Delta}$ by subtracting $h(\hat{\boldsymbol{x}}^{adv})$ from $h(\boldsymbol{x})$.

$$\boldsymbol{\Delta} = h(\boldsymbol{x}) - h(\hat{\boldsymbol{x}}^{adv}) \tag{18}$$

$$\boldsymbol{\Delta}'_Y = \boldsymbol{\Delta}_Y \cdot HPF(\boldsymbol{x}) \tag{19}$$

$$\boldsymbol{x}^{adv} = h^{-1}(\boldsymbol{x} + \boldsymbol{\Delta}') \tag{20}$$

[1] Note that, strictly speaking, color conversion from RGB to YCbCr with rounding errors is not a bijective function [4]. However, this can be ignored since an approximate approach is sufficient for our purposes.

Having isolated the perturbation in the brightness details (Y of YCbCr) from that in the color details, we can improve performance by applying HPF only to the Y channel of the perturbation map $\boldsymbol{\Delta}$. We leave the chrominance channels Cb, Cr of $\boldsymbol{\Delta}$ unscaled. Finally, we add $\boldsymbol{\Delta}'$ to $\boldsymbol{x}$ and invert the color conversion.

Implementation Details. We isolate luminance from chrominance directly in the RGB color space. This can be done by subtracting the grayscale representation $\boldsymbol{x}^{lum}$ of the image from the original image $\boldsymbol{x}$. Given the assumptions of JPEG compression, we combine the chrominance information of the original attack output $\boldsymbol{x}^{adv}_{crom}$ with the luminance information of the HPF attack output $\boldsymbol{x}^{HPF}_{lum}$. This results in the adversarial example $\boldsymbol{x}^{adv}_{CbCr}$, which carries the full perturbation magnitude in the color information and the scaled perturbation magnitude by the HPF mask in the luminance information. Let g be a grayscale transformation $g : \mathbb{R}^{3\times m\times n} \rightarrow \mathbb{R}^{1\times m\times n}$

$$\boldsymbol{x}^{adv}_{crom} = \boldsymbol{x}^{adv} - g(\boldsymbol{x}^{adv}) \tag{21}$$

$$\boldsymbol{x}^{adv}_{CbCr} = \boldsymbol{x}^{adv}_{crom} + g(\boldsymbol{x}^{HPF}) \tag{22}$$

4 Experiments

4.1 Data

Table 1. ASR results in a white-box scenario with varying ϵ for the datasets NIPS17 and CIFAR100, denoted as N17 and CI100, respectively, and separated by a pipe symbol. No LF boosting was used.

N17 |CI100	ϵ	V	HPF	$\Delta_{\bar{D}}$
FGSM	0.129 |0.036	0.501 |0.901	**0.542** |0.901	4.100 |0.035
BIM	0.010 |0.006	0.903 |0.927	**0.904** |**0.960**	0.502 |0.011
VMIFGSM	0.008 |0.007	**0.899** |**0.908**	0.896 |0.900	0.276 |0.011

ImageNet and CIFAR. Our experiments are performed on the NIPS 2017 adversarial competition dataset (N17) [13] and the CIFAR10/100 (CI10/CI100) dataset [11]. The NIPS 2017 dataset consists of 1,000 images from the ImageNet-1K challenge, which contains a wide variety of image classes [13] and presents a challenging and realistic problem. In addition to benchmarking against vanilla attacks, this dataset provides the means to compare our method to related approaches.

4.2 Evaluation Metrics

Attack Success Rate (ASR). An appropriate metric should measure how often an attack has successfully influenced the target network to force a misclassification. To obtain a faithful metric of the performance of the attack, we need to define a subset $\boldsymbol{X}_t$ of the original test dataset $\boldsymbol{X}$, consisting of data points that the target network correctly classified. On this subset, ASR defines the rate of data points where the attack successfully forced a misclassification. Let t be the ground truth of a data point $\boldsymbol{x}$, ϕ the target network, N the number of data points in $\boldsymbol{X}_t$ and α the attack [21,24].

$$\boldsymbol{X}_t = \{\boldsymbol{x} \in \boldsymbol{X} | \phi(\boldsymbol{x}, \theta) = t\} \tag{23}$$

$$\boldsymbol{X}_t^{success} = \{\boldsymbol{x} \in \boldsymbol{X}_t | \phi(\alpha(\boldsymbol{x}), \theta) \neq t\} \tag{24}$$

$$ASR(\phi(\boldsymbol{X}_t, \theta), T) = \frac{|\boldsymbol{X}_t^{success}|}{N} \tag{25}$$

Conditional Average Rate ($\bar{D}$). In addition to ASR, $\bar{D}$ measures the average distance of an adversarial example $\hat{\boldsymbol{x}} = f(\boldsymbol{x})$ from the original data point $\boldsymbol{x}$, where $\boldsymbol{x} \in \boldsymbol{X}_t^{success}$. This metric, the L2 norm, is chosen as a distortion measure to compare with Zhang et al. [24], who used the metric as a distortion measure.To compare distortion results to [15], we benchmark against the reported results in their work using the peak signal-to-noise ratio (PSNR).

$$\bar{D}(f, \boldsymbol{X}_t^{success}) = \frac{1}{|\boldsymbol{X}_t^{success}|} \sum\nolimits_{\boldsymbol{x} \in \boldsymbol{X}_t^{success}} |f(\boldsymbol{x}) - \boldsymbol{x}|_2 \tag{26}$$

4.3 Attack Evaluations

The model providing the gradient ($\hat{\phi}$) is a ResNet [22] that was pre-trained on either ImageNet [13] or CIFAR10/100 [11]. For the white-box attack, $\hat{\phi}$ will be the target model ϕ. To keep comparisons fair, we additionally report the difference in average distortion, and all ϵ were chosen from an ϵ-grid with intervals of 1×10^{-3}, so that the vanilla attacks achieve a robust performance of $ASR >= 0.90$. For the comparatively weaker FGSM attack, $ASR >= 0.50$ (for CIFAR $ASR >= 0.90$). For the black-box evaluations, we extend the prior-guided RGF [2] and VMIFGSM [21], which can also be used for black-box attacks. We use white-box methods with and without our extension to attack hardened models [18,23] and compare our approach with results obtained in two closely related approaches [15,24]. Finally, we test our boosting methods in lossy compression scenarios, which pose a challenging problem for frequency-based smoothing approaches and are usually left unaddressed. In all experiments and tables, V refers to the vanilla version of the attack and HPF refers to our extended version of the attack.

White-Box Attacks. Table 1 shows the ASR results in a white-box scenario and the difference in average distortion ($\Delta_{\bar{D}} = \bar{D}_V - \bar{D}_{HPF}$). In our experiment, the HPF attack almost completely maintained the vanilla attack's performance for both datasets. To our surprise, sometimes the smooth extension **increased** the performance of the attack while still having a much lower average distortion. This is possible due to the internal epsilon adjustment and the emphasis of the perturbation in salient regions of the model. We also observed that the average distortion $\bar{D}$ does not increase even though ϵ is internally adjusted in the HPF attack.

Table 2. ASR results in a back-box scenario for the datasets NIPS17 and CIFAR100, denoted as N17 and CI100, respectively, and separated by a pipe symbol. For PG-RGF, we used the L2 bound version and a fixed λ. The other parameters were taken from the experiments in [2]. For VMIFGSM, p is L_{inf}.

N17 |CI100	ϵ	α	p	V	HPF	$\Delta_{\bar{D}}$
PG-RGF [2]	7.0	2.0	L2	**0.141** |**0.953**	0.132 |0.951	0.110 |0.059
VMIFGSM [21]	0.07	2/255	Linf	**0.462** |**0.870**	0.327 |0.842	4.924 |0.094

Black-Box Attacks. Table 2 shows the ASR results in a black-box scenario and the difference in average distortion ($\Delta_{\bar{D}} = \bar{D}_V - \bar{D}_{HPF}$). For our black-box experiments, the target model was InceptionV3 (VGG for CIFAR). The extended methods are PG-RGF [2] and VMIFGSM [21]. PG-RGF [2] uses a prior to improve the computation of the gradient. In our experiments, we used a fixed lambda λ to define the tradeoff between the prior and the RGF computation, we used the L2 version and the parameters that were used in the original paper. For VMIFGSM [21], $N = 10$ samples are taken in the neighborhood of $\boldsymbol{x}$ and used to compute the final gradient. We chose a larger ϵ compared to our white-box experiments to account for the increased difficulty of the black-box adversarial attack. The HPF extension was able to almost completely preserve the ASR for PG-RGF, while being less distorted according to the CAD norm. While the adversarial sample was much less distorted by the HPF extension of VMIFGSM (see Δ_D), the extension of the attack also resulted in a greater decrease in ASR.

Evaluation on Adversarial Training. Table 3 shows the ASR results in a scenario where the attacked model has been hardened by adversarial training. We also show the difference in average distortion ($\Delta_{\bar{D}} = \bar{D}_V - \bar{D}_{HPF}$). In our experiments, the model denoted as PGD refers to a model that was adversarially pretrained using one of the most common adversarial training protocols that uses PGD and a min-max optimization [18]. The model denoted as FBF [23] uses RFGSM to train competitively hardened models much faster [23]. For both models, the HPF variant performed only slightly worse than the vanilla attack, while having a lower average distortion on ImageNet. For CIFAR, the

Table 3. ASR results for models hardened by adversarial training for the datasets NIPS17 and CIFAR10, denoted as N17 and CI10, respectively, and separated by a pipe symbol. The values for ϵ were taken from the white-box experiments. No boosting techniques were used.

PGD [18]	V	HPF	$\Delta_{\bar{D}}$	FBF [23]	V	HPF	$\Delta_{\bar{D}}$
FGSM	**0.341** |0.336	0.319 |**0.360**	0.490 |0.153	FGSM	**0.446** |0.419	0.386 |**0.438**	0.429 |0.143
BIM	**0.317** |0.050	0.309 |0.050	0.187 |0.007	BIM	**0.416** |0.067	0.391 |0.067	0.294 |0.006
VMIFGSM	0.183 |0.041	**0.197** |**0.042**	0.063 |0.024	VMIFGSM	0.224 |0.053	**0.234** |**0.056**	0.033 |0.022

HPF attacks sometimes surpassed the ASR of the vanilla attack while being less distorted. Surprisingly, for VMIFGSM [21], the HPF attack performed better in both experiments while still being less distorted.

Comparison with other Smoothing Approaches. To provide a benchmark against other smoothing approaches presented for attacks based on gradient projection, we report the performance of our approach on the NIPS 2017 untargeted attack competition [13] and compare it to (1) sPGD$_{L2}$ presented in [24] and to (2) RGLF-FGSML$_2$ presented in [15]. We do not use attacks where the base attack obtained a score of 1.0 in [15], since it is unclear if the ϵ is set too high, resulting in a skewed comparison of the base attack and the extended attack. For (1), we perform a white-box attack experiment using Inception-V3 [24] with an accuracy of 0.953 and its adversarially trained counterpart [22]. Because we use different model weights for Inception-V3 [24][2], ResNet and different implementations of PGD$_{L2}$ and FGSML_2, we had to report the metrics in a different way to keep the comparison fair.

Table 4. Comparison of our smoothing approach with [24] and [15] . *Base* shows the indicated ASR of the base attack (PGD$_{L2}$ for [24], FGSML$_2$ for [15]), *Ext* shows the ASR of the respective extensions, and Δ_{ASR} shows the difference between the ASR of the base version and the extended smooth version. Finally, $\Delta_{\bar{D}} = (\ \bar{D}_V - \bar{D}_{Smooth})$ shows the difference in average distortion, and $\Delta_{PSNR} = \overline{PSNR_{smooth}} - \overline{PSNR_V}$ shows the average difference in PSNR, with higher values being better for both Δ.

Inception	Zhang et al. [24]	Ours	AdvInc	Zhang et al. [24]	Ours	Resnet50	Liu et al. [15]	Ours
Base	1.000	1.000	Base	1.000	0.988	Base	0.934	0.375
Ext	0.960	0.999	Ext	0.690	0.994	Ext	0.887	0.343
Δ_{ASR}	0.040	**0.001**	Δ_{ASR}	0.310	**0.005**	Δ_{ASR}	0.047	**0.032**
$\Delta_{\bar{D}}$	-0.300	**0.118**	$\Delta_{\bar{D}}$	-3.970	**0.239**	Δ_{PSNR}	**15.730**	2.708

[2] Our ported PyTorch weights perform slightly differently than the TensorFlow weights used in [24].

Instead of using ASR directly as the basis for comparing the smoothing approaches, we instead report the difference in ASR (Δ_{ASR}) and $\Delta_{\bar{D}}$ ($\bar{D}_V - \bar{D}_{Smooth}$) from the original PGD$_{L2}$ to the respective smoothed version ([24]'s sPDG$_{L2}$ and ours). For comparison, we used the same settings for PGD$_{L2}$ as those reported in [24]. The step size α is set to 3 and ϵ is set to 5. Our Inception-V3 (or AdvInception-V3) achieves a base accuracy of 0.953 (0.867) on the dataset, compared to 0.96 (0.94) in [24]. For (2) we compare RLGF-FGSML$_2$, HPF-FGSML$_2$ and FGSML$_2$. Since the average distortion L_2 is not reported in [15],we instead report the difference in peak-signal-to-noise ratio (PSNR), denoted as $\Delta_{PSNR} = \overline{PSNR_{smooth}} - \overline{PSNR_V}$, along with Δ_{ASR} from FGSML$_2$ to the respective smooth version, where the base attacks from both works achieved approximately the same average PSNR.

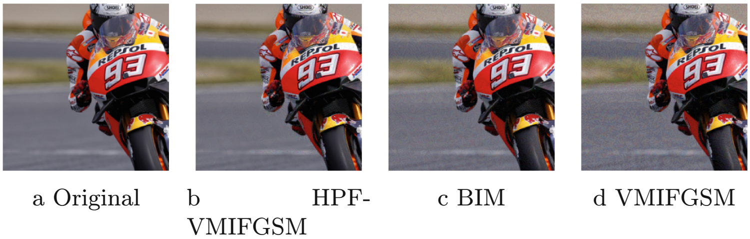
Comparing our method to Zhang et al. [24] in Table 4, our smoothing approach was able to better maintain performance in the case of PGD_{L2} while at the same time providing better distortion reduction compared to the original attack ($\Delta_{\bar{D}} = \bar{D}_V - \bar{D}_{Smooth}$). Furthermore, we note that our HPF extension actually *increased* the performance of the vanilla attacks, showing that at least some adversarial training protocols result in models that are susceptible to the different properties of our adversarial samples. Comparing our method to Liu et al [15], we note that our method is much less smooth according to PSNR, but provides slightly better performance retention without the need to perform a search over several cutoff-frequencies, as done in [15]. Furthermore, some studies [5] also suggest that PSNR may not be the best metric to mimic the HVS, and therefore measure perceived distortion.

Table 5. Comparison of our boosting methods in compression scenarios. The compression rate for JPEG compression was set to 0.5. In these experiments, FGSM was extended and used to attack a ResNet trained on ImageNet. $\hat{\phi}$ is the target model and ρ was set to 0.5.

$\tau = 0.5$	lower ϵ	ASR	$\bar{D}$	higher ϵ	ASR	$\bar{D}$
HPF	0.0129	0.190	0.743	0.04	0.323	3.892
HPF+CbCr	0.0129	0.177	0.845	0.04	0.308	4.526
HPF+LF	0.0129	**0.224**	1.053	0.04	**0.341**	4.952
Vanilla	0.0129	0.216	1.097	0.04	0.338	5.286

Compression Experiments. Methods to generate adversarial samples that are robust against JPEG compression [7,20] work by using a differentiable approximation of the non-differentiable JPEG compression [20]. However, the performance of these attacks depends heavily on the exact compression algorithm and even the exact compression rate used [20]. Although the issue of compression is rarely addressed in work on adversarial smoothing, it is a critical aspect to consider for approaches that perform smoothing by local variation. Therefore, we examine how HPF attacks perform with and without boosting. Table 5 shows the ASR results for the standard HPF attack and its boosted counterparts. For this experiment, all images were taken from the ImageNet subset and compressed with a high compression rate of $\tau = 0.5$ and samples were created with varying ϵ. For LF boosting, ρ was set to 0.5. Although standard HPF masking was able to maintain most of the performance of its vanilla counterpart, LF boosting further improved the ASR and surpassed the ASR of the vanilla attack, while being less distorted. CbCr boosting performed slightly worse than even the standard HPF mask, while still being less noticeable than LF-boosted counterparts with high values of ρ.

4.4 Measuring Imperceptibility



a Original b HPF-VMIFGSM c BIM d VMIFGSM

Fig. 3. Qualitative comparison of an HPF adversarial sample to the samples produced by iterative attacks with no smoothing in addition to the comparison of patters to FGSM found in Figure 1. For all attacks, ϵ was set to 0.02 for a stronger perturbation to adjust for the small image resolution. Zooming in helps to identify the perturbations better.

Imperceptibility is subjective and difficult to capture in a metric. In the adversarial attack literature, L_2 norms are usually presented as a measure of perturbation size and perceptibility [1]. However, the L_2 norm only measures the mean distance of pixel values of the original image $\boldsymbol{x}$ and the adversarial example $\boldsymbol{x}^{adv}$, which has some disadvantages [24]. Instead, one can use a metric that was designed to mimic the human visual system's perception of distortion [14,24]. We use MAD [14] as a numerical estimator of the smoothing benefit of our approach. In addition, we present a human evaluation study where users were asked to choose the less noisy image from $\{\boldsymbol{x}^{adv}, \boldsymbol{x}^{HPF}\}$ where the perturbation magnitude ϵ is the same for both adversarial samples and taken from Table 1, where both the vanilla attack and the HPF counterpart achieve approximately the same performance for the same ϵ. A more detailed description of the study can be found in the supplementary material. Figure 3 shows a visual comparison of the perturbation patterns produced by a HPF attack (b) compared to attacks with no smoothing, which can be best observed upon magnifying. Looking at areas with low variation, the adversarial sample produced by (b) has much less noise. Zooming in on the motorcyclist with a lot of high-variation detail, it can be observed that (b) contains more noise compared to the other samples, as the HPF mask shifted the noise into areas of high variation.

Numerical Evaluation. MAD was designed to closely mimic the human visual system in the perception of distortion [14]. Fezza et al. [5] compared fifteen image quality metrics to the subjective perceptual judgments of a group of users and concluded that MAD best mimicked their judgments [5,24]. We use the parameters in our white box experiments to compute average MAD scores for both vanilla attacks (denoted as V) and attacks with our HPF extension (denoted as HPF), with lower scores indicating higher fidelity. Note that for ASR, the results obtained by the respective ϵ were approximately the same and that HPF *outperformed* the vanilla attack in two out of three cases (see Table 1).

Table 6. MAD scores with different ϵ. α has been set to 2/255 for BIM and VMIFGSM. Lower scores indicate higher fidelity. In addition, we present the results of the survey. Columns V and HPF show the percentage of people who consider the vanilla version and the HPF version to be less noisy, respectively. UND denotes "undecided".

MAD	ϵ	V	HPF	Survey	V	HPF	UND
FGSM	0.129	206.69	**177.11**	FGSM	0.125	**0.825**	0.05
BIM	0.010	6.01	**2.70**	BIM	0.19	**0.775**	0.031
VMIFGSM	0.008	5.77	**3.36**	VMIFGSM	0.209	**0.398**	0.392

Table 6 shows the average MAD scores for the vanilla attacks and their HPF counterparts, which were computed using the same ϵ (and α) values that were used to compute the ASR scores of Table 1. For all attack comparisons, the HPF variant achieved a lower MAD score than the vanilla attack, indicating that our HPF extension makes attacks less visually detectable.

Survey Evaluation. Table 6 shows the results of our survey. V denotes the image produced by the vanilla attack, HPF denotes the adversarial sample produced by our HPF counterpart, and UND denotes that the user could not decide on a sample. The results show that the adversarial example with the HPF attack appears less noisy than the vanilla counterparts in all cases. However, due to the overall small perturbation magnitudes ϵ needed for the white-box attack, many users were undecided in the VMIFGSM case. In contrast, the perturbation magnitude for the white-box attack of FGSM was set so high that perturbations were visible in both cases. Since hardly any participant chose the vanilla adversarial sample as the less noisy one, our survey results are consistent with our numerical evaluation.

5 Ethical Discussion and Conclusion

Adversarial attacks, which can manipulate a model into producing incorrect outputs, have the potential to be exploited for malicious purposes. Therefore, any research conducted in this field has ethical implications. However, seeing that more and more applications rely on deep learning, it is important to expose and address any flaws these models have before they can be safely used in any high-security context. Smoothing methods like ours show that simply ignoring the threat of adversarial samples due to the fact that they exhibit a characteristic noise pattern does not hold anymore.

HPF masks provide a straightforward extension to most pixel-based adversarial attacks that, unlike the standard perturbation size ϵ hyperparameter found in regular attacks, provides a more flexible tradeoff between attack performance and attack detectability. Including salient regions can sometimes *increase* the performance of the attack while still being (1) less biased according to the CAD metric and (2) less detectable by the HVS according to the MAD metric. Our experiments demonstrate that, for the attacks we tested, our method can maintain or even enhance attack performance more effectively than comparable frequency-based smoothing approaches., while providing a comparable reduction in distortion compared to the state-of-the-art.

References

1. Carlini, N., Wagner, D.A.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017. pp. 39–57. IEEE Computer Society (2017). https://doi.org/10.1109/SP.2017.49
2. Cheng, S., Dong, Y., Pang, T., Su, H., Zhu, J.: Improving black-box adversarial attacks with a transfer-based prior. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. pp. 10932–10942 (2019)
3. Croce, F., Hein, M.: Sparse and imperceivable adversarial attacks. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 4723–4731. IEEE (2019). https://doi.org/10.1109/ICCV.2019.00482
4. Domanski, M., Rakowski, K.: Color transformations for lossless image compression. In: 10th European Signal Processing Conference, EUSIPCO 2000, Tampere, Finland, September 4-8, 2000. pp. 1–4. IEEE (2000)
5. Fezza, S.A., Bakhti, Y., Hamidouche, W., Déforges, O.: Perceptual evaluation of adversarial attacks for cnn-based image classification. In: 11th International Conference on Quality of Multimedia Experience QoMEX 2019, Berlin, Germany, June 5-7, 2019. pp. 1–6. IEEE (2019). https://doi.org/10.1109/QOMEX.2019.8743213
6. Haralick, R., Shapiro, L.: Computer and robot vision. No. Bd. 2 in Computer and Robot Vision, Addison-Wesley Pub. Co. (1993), http://books.google.de/books?id=LfVRAAAAMAAJ

7. He, W., Wei, J., Chen, X., Carlini, N., Song, D.: Adversarial example defense: Ensembles of weak defenses are not strong. In: Enck, W., Mulliner, C. (eds.) 11th USENIX Workshop on Offensive Technologies, WOOT 2017, Vancouver, BC, Canada, August 14-15, 2017. USENIX Association (2017)
8. He, Z., Wang, W., Dong, J., Tan, T.: Transferable sparse adversarial attack. CoRR **abs/2105.14727** (2021), https://arxiv.org/abs/2105.14727
9. Hudson, G., Léger, A., Niss, B., Sebestyén, I., Vaaben, J.: JPEG-1 standard 25 years: past, present, and future reasons for a success. J. Electronic Imaging **27**(04), 040901 (2018). https://doi.org/10.1117/1.JEI.27.4.040901
10. Jia, S., Ma, C., Yao, T., Yin, B., Ding, S., Yang, X.: Exploring frequency adversarial attacks for face forgery detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 4093–4102. IEEE (2022). https://doi.org/10.1109/CVPR52688.2022.00407
11. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009), https://api.semanticscholar.org/CorpusID:18268744
12. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings. OpenReview.net (2017)
13. Kurakin, A., Goodfellow, I.J., Bengio, S., Dong, Y., Liao, F., Liang, M., Pang, T., Zhu, J., Hu, X., Xie, C., Wang, J., Zhang, Z., Ren, Z., Yuille, A.L., Huang, S., Zhao, Y., Zhao, Y., Han, Z., Long, J., Berdibekov, Y., Akiba, T., Tokui, S., Abe, M.: Adversarial attacks and defences competition. CoRR **abs/1804.00097** (2018), http://arxiv.org/abs/1804.00097
14. Larson, E.C., Chandler, D.M.: Most apparent distortion: full-reference image quality assessment and the role of strategy. J. Electronic Imaging **19**(1), 011006 (2010). https://doi.org/10.1117/1.3267105
15. Liu, J., Lu, B., Xiong, M., Zhang, T., Xiong, H.: Low frequency sparse adversarial attack. Comput. Secur. **132**, 103379 (2023). https://doi.org/10.1016/J.COSE.2023.103379, https://doi.org/10.1016/j.cose.2023.103379
16. Luo, B., Liu, Y., Wei, L., Xu, Q.: Towards imperceptible and robust adversarial example attacks against neural networks. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pp. 1652–1659. AAAI Press (2018). https://doi.org/10.1609/AAAI.V32I1.11499, https://doi.org/10.1609/aaai.v32i1.11499
17. Luo, C., Lin, Q., Xie, W., Wu, B., Xie, J., Shen, L.: Frequency-driven imperceptible adversarial attack on semantic similarity. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 15294–15303. IEEE (2022). https://doi.org/10.1109/CVPR52688.2022.01488, https://doi.org/10.1109/CVPR52688.2022.01488
18. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018)
19. Marr, D., Hildreth, E.: Theory of Edge Detection. In: Proceedings of the Royal Society of London. Series B: Biological Sciences. vol. 207, pp. 187–217 (1980). https://doi.org/10.1098/rspb.1980.0020

20. Shin, R.: Jpeg-resistant adversarial images (2017), https://api.semanticscholar.org/CorpusID:204804905
21. Wang, X., He, K.: Enhancing the transferability of adversarial attacks through variance tuning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 1924–1933. Computer Vision Foundation / IEEE (2021). https://doi.org/10.1109/CVPR46437.2021.00196
22. Wightman, R.: Pytorch image models (2019). https://doi.org/10.5281/zenodo.4414861
23. Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: Revisiting adversarial training. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020)
24. Zhang, H., Avrithis, Y., Furon, T., Amsaleg, L.: Smooth adversarial examples. EURASIP J. Inf. Secur. **2020**(1), 1–12 (2020). https://doi.org/10.1186/s13635-020-00112-z

Maximizing Coverage over a Surveillance Region Using a Specific Number of Cameras

M. S. Sumi Suresh[1,2](✉), Vivek Menon[2], Srirangaraj Setlur[1],
and Venu Govindaraju[1]

[1] Department of Computer Science and Engineering, University at Buffalo, Buffalo,
NY 14260, USA
{sumisure,setlur,govind}@buffalo.edu

[2] Department of Computer Science and Engineering, Amrita School of Computing,
Amrita Vishwa Vidyapeetham, Amritapuri, Kollam 690525, India
{sumisuresh,vivekmenon}@am.amrita.edu

Abstract. An efficient approach to camera deployment enhances the cost-effectiveness and functionality of a multi-camera surveillance network. Traditional camera placement strategies involve maximizing total coverage over a surveillance region with predefined camera locations by optimizing the camera orientations. However, some modern approaches allow users to specify the desired total coverage over the surveillance region, and the algorithms subsequently determine the optimal number of cameras, as well as their exact locations and orientations, to meet this specified coverage. The Reward Penalty Score (RPS) algorithm and Extended Greedy Grid Voting (EGGV) algorithm are two innovative algorithms designed to attain the coverage constraint by proposing an optimal number of cameras and their locations and orientations. In certain scenarios, the number of cameras available for designing the surveillance network is limited, which should be treated as an input constraint rather than a specified coverage requirement. Under such conditions, the primary objective becomes maximizing total coverage with the given number of cameras by determining their optimal locations and orientations. Currently, there are no widely recognized algorithms specifically designed to handle this particular scenario. In this paper, we effectively modify the RPS and EGGV algorithms (m-EGGV and m-RPS), adapting them to optimally deploy the specified number of cameras over the entire surveillance region in an efficient manner to maximize coverage. Additionally, by employing the m-EGGV and m-RPS algorithms, we address the scenario of coverage loss resulting from the failure of one or more cameras. These modified algorithms facilitate the relocation of a subset of potential cameras which can alleviate this loss in coverage caused by the failure of cameras. The m-EGGV and m-RPS algorithms demonstrate a robust performance through extensive testing in diverse simulation environments.

Keywords: Camera placement · Coverage optimization algorithms · Greedy search methods · Video surveillance · Visual sensors

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 303–320, 2025.
https://doi.org/10.1007/978-3-031-78312-8_20

1 Introduction

Advancements in camera technology, along with strides in image processing and machine learning, have significantly contributed to the expansion of intelligent video surveillance applications [13]. As security and safety become increasingly critical, the use of video cameras has expanded into various areas such as homeland security, surveillance, smart housing, and robotics [19] [17] [16] [20]. The optimal arrangement of cameras is crucial, as it determines the scope of surveillance coverage, impacting the system's effectiveness and cost [12]. Iteratively refining the position and orientation of cameras are essential to achieving the broadest possible coverage with a given number of cameras [22].

In scenarios involving fixed camera locations, camera placement algorithms typically suggest optimal camera orientations to achieve specific coverage objectives [12]. However, the coverage outcomes are often less than ideal due to constraints and preferences involved in the original choice of camera locations. Although some methods optimize both the locations and orientations of cameras, their focus is restricted to maximizing coverage within certain predefined areas [14]. For truly optimal camera placement, the algorithm should minimize the number of cameras used while fulfilling the required coverage specifications [24].

Effectively, the various optimization scenarios related to coverage maximization over a surveillance network under various constraints can be summarized as follows: (i) with an existing surveillance network where the cameras are already installed,(ii) without an existing surveillance network, constrained by the number of cameras, (iii) without an existing surveillance network, constrained by coverage requirement, and (iv) with an existing surveillance network in the event of camera failure. Table 1 outlines these scenarios, detailing their input parameters and the parameters targeted for optimization.

In our previous work [23] addressing Scenario 1, we had introduced two innovative and efficient camera placement algorithms, the Alternate Global Greedy (AGG) algorithm and the Greedy Grid Voting (GGV) algorithm, to determine the optimal configuration of cameras in a surveillance network with fixed camera positions. Thereafter, addressing Scenario 3, we proposed two new algorithms: the Reward Penalty Score (RPS) algorithm and the Extended Greedy Grid Voting (EGGV) algorithm, to optimize coverage across a surveillance area without an existing network limited by a specific coverage goal [22]. The RPS algorithm is a versatile and scalable approach for the optimal placement of cameras, addressing both their locations and orientations to meet particular coverage criteria with the fewest cameras possible. Building on the principles of the RPS algorithm, the EGGV algorithm enhances the Greedy Grid Voting (GGV) algorithm by expanding its scope to not only optimize camera orientations but also their numbers and locations. Additionally, as part of our previous research, addressing Scenario 4, we proposed the Visibility Graph Reduction (VGR) algorithm, which identifies a subset of potential neighboring cameras that are capable of addressing the coverage loss due to the failure of cameras [21].

In certain circumstances, the number of cameras available for setting up the surveillance network is limited. Therefore, the primary objective for this paper

Table 1. Scenarios related to coverage maximization over a surveillance network

Scenario	Optimizing Problem	Input Parameters	Optimizing Parameters	Algorithms
1	Maximize total coverage	No. of Cameras, Camera Location	Camera Orientation	LG, GG, AGG, GGV
2		No. of Cameras	Camera Location, Camera Orientation	m-EGGV, m-RPS
3		Coverage Requirement	No. of Cameras, Camera Location, Camera Orientation	EGGV, RPS
4	Reduce coverage loss due to camera failure	Damaged Camera Location, Active Camera Locations	Total Coverage, Camera Orientation	VGR+GGV VGR+RPS
			Total coverage, Location, Orientation	VGR+m-EGGV VGR+m-RPS

lies in addressing Scenario 2, where the goal is to maximize total coverage over the surveillance region by proposing optimal locations and orientations for a pre-specified number of cameras. In this paper, we have modified the EGGV and RPS algorithms to maximize total coverage in the surveillance region by determining optimal locations and orientations for a specific number of cameras. The key objective of this paper is to determine the maximum coverage result with the available resources through the optimal placement of a pre-specified number of cameras. The modified-EGGV (m-EGGV) and modified-RPS (m-RPS) algorithms generate a set of possible camera locations and corresponding orientations to attain maximum coverage, provided a plan image of a surveillance region along with the number of available cameras as input. Additionally, we have also evaluated these modified algorithms in Scenario 4 for regaining coverage lost due to the failure of one or more cameras. Initially, the VGR algorithm [21] is used to identify the potential subset of cameras among the active cameras, which can optimally alleviate the loss in coverage resulting from the failure of one or more cameras. Subsequently, the m-EGGV or the m-RPS algorithm is used to propose the new locations and orientations for the VGR subset of cameras to alleviate the coverage loss. The key focus areas of this paper are as follows:

- Optimize the camera coverage for a surveillance scenario using a pre-specified number of cameras.
- Relocate a potential subset of cameras in the event of coverage loss due to failure of cameras.
- Evaluate the proposed algorithms over diverse indoor and outdoor surveillance scenarios.

The remainder of this paper is structured as follows: Section 2 examines related work on camera placement and coverage optimization in surveillance networks. Section 3 describes the modifications to the RPS and the EGGV algorithms to address the problem under consideration. Section 4 delves into our simulation experiments and their outcomes, and section 5 concludes the paper.

2 Literature Review

The objective of the coverage optimization problem is to maximize overall coverage with the minimum number of cameras. The optimization of camera coverage has garnered considerable attention in research circles due to its critical impact on sensor planning, selection, calibration, and optimal placement [15,18]. Modeling camera coverage is crucial to ensure that the deployment and operation of cameras adequately fulfill the specified requirements. Two principal methods for modeling camera coverage are deterministic modeling [25] and probabilistic modeling [1]. In deterministic modeling, coverage is quantified in a fixed manner and the resulting coverage is used to evaluate performance. On the other hand, probabilistic modeling employs a probabilistic function that incorporates variables like camera range, visibility, and angle to calculate coverage.

Most studies on camera placement optimization algorithms, as highlighted in previous research [10], concentrate on enhancing coverage within surveillance networks with fixed cameras, employing methods from basic greedy techniques to advanced dynamic programming strategies [2]. In these settings, greedy algorithms are frequently chosen for their high efficiency. The effectiveness of coverage optimization often depends on the sequence in which cameras are optimized, with greedy algorithms such as Local Greedy (LG), Global Greedy (GG), and Alternate Global Greedy (AGG), typically starting at the first camera location and proceeding in raster scan order. Departing from the typical greedy methods that operate in a raster scan sequence, the GGV algorithm emphasizes covering unique and strategically critical areas that fixed-sequence approaches might neglect [23].

Research on optimizing camera placement, which involves finding the best locations and orientations has employed a variety of methods, including Integer Linear Programming (ILP) [4], Genetic Algorithms [11], Binary Optimization techniques [6], and iterative algorithms [8]. Furthermore, studies aiming to identify optimal locations and orientations to fulfill specific coverage goals with the fewest cameras have utilized Genetic Algorithms (GA) [7], probabilistic optimization frameworks [5], and combinatorial algorithms [3]. Generally, most of the methods in the literature propose optimal camera locations within a surveillance region to attain user-specific or application-specific constraints. The existing research on surveillance scenarios is often limited by its task-specific nature or user-imposed constraints, leading to a lack of generalizability and scalability. This issue was tackled in our previous work [22], where we introduced the Reward Penalty Score (RPS) and the Extended Greedy Grid Voting (EGGV) algorithms. These algorithms are generic and scalable, effectively addressing the camera placement problem in varied indoor and outdoor settings. They optimize the number of cameras, along with their locations and orientations, to fulfill specific coverage requirements efficiently. Although our algorithms are designed to be versatile and not confined to particular tasks or domains, they are also adept at managing a wide array of constrained situations, from areas needing redundant coverage for security purposes to those requiring no coverage to protect privacy.

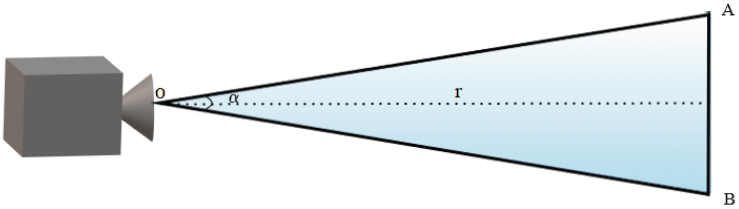


Fig. 1. A triangular layout of camera coverage area

Research on optimal camera reconfiguration in response to failures within surveillance networks is limited. Our previous work involves a novel, cost-effective, and computationally efficient graph-based algorithm that substantially minimizes human intervention in adjusting cameras across a range of practical settings [21]. The lack of adequate research focused on maximizing coverage over a surveillance region using a pre-specified number of cameras forms a key motivation behind this paper.

3 Proposed Method

3.1 Problem Formulation

Let η denote the constraint in the case of camera coverage optimization problem, which for scenario 3 is the specified number of cameras given for deployment N_{giv}. In a surveillance environment ξ with coverage constraint η, the camera coverage optimization problem involves determining the optimal locations L and their corresponding orientations θ to maximize total subject to η. The objective is to pinpoint the coordinate positions (x, y) for optimal camera locationsL_i, along with their respective orientations θ_j, using the specified number of cameras N_{giv} across the designated surveillance region. The following assumptions regarding the surveillance network have been made for practical convenience : (i) surveillance region is limited to 2D, (ii) similar camera with same Field of View (FoV) α, focal length f, and range r (iii) each camera can take only a finite number of possible orientations $O = \{\theta_1, \ldots, \theta_P\}$.

It is essential that there exists some function $f(C_{tot} \mid \eta, \xi)$ that maximizes the total coverage C_{tot} in the surveillance region ξ satisfying coverage constraint η, by optimally placing the given number of cameras N_{giv}; $C_{tot} = \sum_{i=1}^{N_{giv}} C_i$. Here, C_i represents the coverage provided by a camera optimally positioned at location L_i with orientation θ_j. Thus, C_i is defined by the function $f(L_i, \theta_i)$ and the optimization problem can be expressed as

$$\Gamma = \underset{C_{tot}}{\arg\max}\, f(C_{tot} \mid \eta, \xi) \tag{1}$$

We have modeled the camera coverage using a triangular layout as in Fig. 1 and is calculated deterministically [25] considering the camera parameters such as field of view α, focal length f of the camera, and range r. A camera's field of view defines the visible area which is calculated as.

$$\alpha = 2 \times \tan^{-1}\left(\frac{W_I}{2f}\right) \tag{2}$$

where w_I is the width of the image sensor and f is the focal length of the camera. Let A and B be the coordinate positions of the triangular coverage area shown in 1 and can be represented as (r, y) and $(r, -y)$ respectively. The value of y will be evaluated as:

$$y = r \times \tan\left(\frac{\alpha}{2}\right) \tag{3}$$

We use Bresenham's line algorithm [9] for ray tracing to handle occlusions in the surveillance region while drawing the triangular coverage of a camera. The Bresenham algorithm assesses the visibility of a point from the camera's viewpoint, considering the ray portion beyond the point where it intersects with the boundary of an occluding object as invisible to the camera.

3.2 Optimization Approaches

In this paper, we have modified the EGGV and RPS algorithms to maximize total coverage over the surveillance region by determining optimal locations and orientations for a specific number of cameras. The major function of the RPS algorithm is to select the minimum number of cameras, its corresponding location, and orientations to attain the coverage requirement over the surveillance region. In some scenarios, the constraint may be to deploy given number of cameras over the surveillance region optimally instead of a specified coverage requirement. The algorithms Extended Greedy Grid Voting Algorithm (EGGV) & Reward Penalty Score Algorithm (RPS) [22] can be modified to maximize total coverage with given number of cameras by changing the loop exit criteria as these algorithms does not have any predefined execution order. The modified EGGV (m-EGGV) and modified RPS (m-RPS) algorithm is shown in Algorithm 1 & Algorithm 2. Here, we provide the number of cameras specified N_{giv}, as input to the algorithms. Both the modified algorithms place all N_{giv} given cameras over the surveillance region optimally.

The GGV algorithm is a novel grid-based two-stage voting method for optimizing camera coverage across diverse scenarios. The GGV algorithm prioritizes covering unique and strategically important areas. Our Extended Greedy Grid Voting (EGGV) algorithm capable of optimizing camera location and orientaiton, is an extended version of our GGV algorithm, which was restricted to optimizing only camera orientations. The EGGV algorithm generates all the potential camera locations from a building blueprint or sketch map. In the next step, it assumes that a camera is placed at each location identified in the previous step. Thereafter, a 2-stage process of forward and reverse voting is performed for each coverage field C produced by these cameras at various orientations. The

Algorithm 1: Modified EGGV Algorithm (m-EGGV)

```
Input  : Camera range $r$, width of the image sensor $W_I$, plan image of
         surveillance region $I$, and number of cameras given $N_{giv}$
Output: A set $LO$ containing optimal location orientation pairs $(L_i, \theta_j)$ for the
         $N_{giv}$ cameras
Initialize set of orientations $O = \{\theta_1, \ldots, \theta_P\}$
Initialize set of configured cameras $SC = \{\}$
Initialize $N_{fix} = 0$;
Compute field of view $\alpha$ using Eq. 2
while $N_{fix} <= N_{giv}$ do
    call the EGGV algorithm to get a priority location and orientation.
      $(L_i, \theta_j) = EGGV(r, \alpha, I)$
    Compute coverage C of selected $(L_i, \theta_j)$ pair using Eq. 3.
    $LO = SC \cup \{(L_i, \theta_j)\}$
    $N_{fix} = N_{fix} + 1$
    $SC = SC \cup \{C\}$
```

coverage field with highest vote is chosen for deployment and is considered as configured. This process repeats until the number of configured cameras N_{fix} matches the given number of cameras N_{giv}.

The RPS algorithm begins by generating all potential camera locations from a building blueprint or sketch map. In the next step, it posits that a camera is placed at each location identified in the initial stage. The coverage field C, produced by these cameras at various orientations is then evaluated and scored. The coverage field with the highest score S_C, is chosen for deployment. The camera corresponding to this optimal coverage field is deemed configured, and this procedure is repeated until the number of configured cameras N_{fix}, matches the given number of cameras N_{giv}.

We evaluate the complexity of the m-EGGV and m-RPS algorithms using a 300 × 300 computer-generated map image as the surveillance region, with no predefined cameras, and specifying the given number of cameras, $N_{giv} = 16$. Both m-RPS and m-EGGV are enhanced versions of the original RPS and EGGV algorithms. Consequently, they share the same computational complexity $O(n^2)$, as depicted in Fig. 2

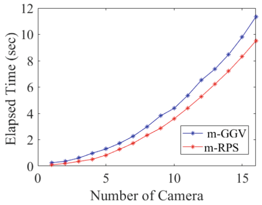
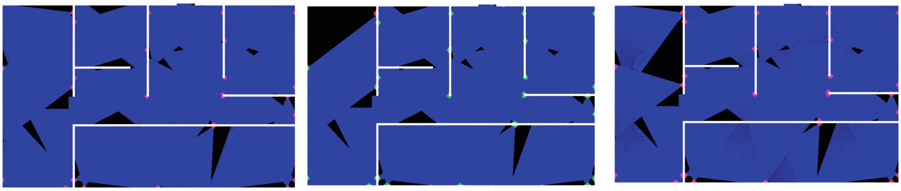


Fig. 2. Computational complexity analysis: m-EGGV and m-RPS

Algorithm 2: Modified RPS Algorithm (m-RPS)

```
Input  : Camera range $r$, width of the image sensor $W_I$, plan image of
         surveillance region $I$, and number of cameras given $N_{giv}$
Output: A set $LO$ containing optimal location orientation pairs $(L_i, \theta_j)$ for the
         $N_{giv}$ cameras
Initialize set of orientations $O = \{\theta_1, \ldots, \theta_P\}$
Initialize set of configured cameras $SC = \{\}$
Initialize $N_{fix} = 0$;
Compute field of view $\alpha$ using Eq. 2
while $N_{fix} <= N_{giv}$ do
    call the RPS algorithm to get a priority location and orientation.
      $(L_i, \theta_j) = RPS(r, \alpha, I)$
    Compute coverage C of selected $(L_i, \theta_j)$ pair using Eq. 3.
    $LO = SC \cup \{(L_i, \theta_j)\}$
    $N_{fix} = N_{fix} + 1$
    $SC = SC \cup \{C\}$
```



(a) (b) (c)

Fig. 3. Indoor Environment: Plan image of an office (a) Coverage results before failure (b) Coverage results after failure of one camera (c) Coverage results after relocating a neighbouring camera using m-EGGV

3.3 Alleviating Coverage Loss in a Surveillance Network after Failure of Cameras

Consider a scenario where a surveillance network faces the failure of one or more cameras. Typically, due to cost considerations, the layout of cameras within the network is not reconfigured unless necessary to address coverage gaps caused by such failures. Rather than replacing the malfunctioning cameras, a more economical approach involves repositioning some of the adjacent cameras to reduce the impact of the coverage loss. Our Visibility Graph Reduction (VGR) algorithm [21] utilizes a graph-based method to identify which active cameras are best suited to compensate for the lost coverage. This two-stage graph reduction process prioritizes cameras based on overlapping coverage areas and visibility, selecting an optimal subset of active cameras for repositioning. Subsequently, using our modified Extended Greedy Grid Voting (m-EGGV) or modified Reward Penalty Score (m-RPS) algorithm, we can strategically relocate these selected neighboring cameras to a new setup that effectively mitigates the coverage deficit.

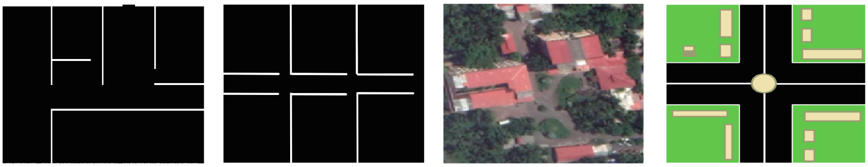

(a) (b) (c) (d)

Fig. 4. Simulation Environments (a) Plan image of an office (b) Plan image of a University building (c) Google map image of a campus (d) Map image of a crossroad junction

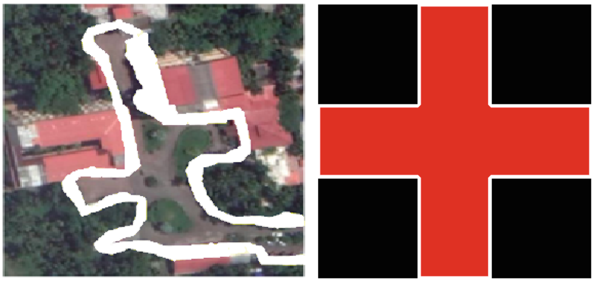
Table 2. Camera specifications

Scenarios	Camera Model	Sensor	Focal Length	FoV
Plan Image of Office	AXISP1357	1/3.2" RGBCMOS	2.8-8 mm	54 °
Plan Image of University	AXISP1357	1/3.2" RGB CMOS	2.8-8 mm	54 °
Google Map	BIPRO-540LA	1/3.2" SONY CCD	3.6 mm	74 °
Crossroad Junction	DS2CE16C2T-IR	1/3" CMOS	6 mm	54 °

Fig. 3a shows the existing surveillance network across the office floor, equipped with 27 cameras achieving 95% coverage, where areas covered by cameras are indicated in blue. Fig. 3b demonstrates a reduction in coverage to 84.43% following the failure of one camera, with operational cameras highlighted in green and the damaged camera in red. Fig. 3c displays the coverage outcome after relocating a neighboring camera using the m-EGGV algorithm, which restored coverage to 91%.

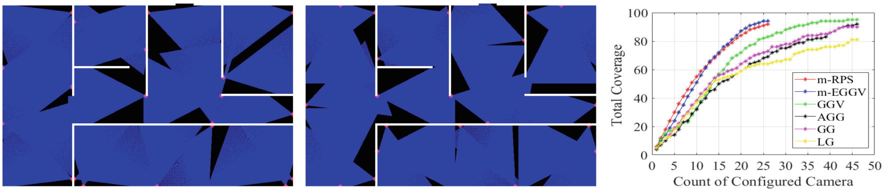
4 Experimental Results

We perform simulation experiments across diverse indoor and outdoor environments, as shown in Fig. 4. Our initial focus is on indoor environments ranging from small to large surveillance regions. Thereafter, we focus on outdoor environments such as a university campus and a crossroad junction. Traditionally, indoor surveillance networks rely on wall-mounted cameras. Therefore, building walls are regarded as the optimal locations for camera placement. A primary challenge while designing an outdoor surveillance network is to identify the optimal locations for fixing cameras. Therefore, the original EGGV and RPS algorithms use a sketch map of the outdoor surveillance region as an input to the algorithm, instead of a regular plan image. The sketch map is an image of a surveillance region showcasing the appropriate areas for camera placement. The sketch map for the Google map image of a campus (Fig. 5a) and the map image of a crossroad junction (Fig. 5b) are illustrated in Fig. 5, where the appropriate areas for placing cameras are highlighted in white. The cameras and their configurations for conducting simulation experiments over these surveillance environments are listed



(a) (b)

Fig. 5. Sketch maps: (a) Google map image of a campus (b) Map image of a crossroad junction



(a) (b) (c)

Fig. 6. Indoor environment : Plan image of an office (a) Coverage results of m-EGGV algorithm (b) Coverage results of m-RPS algorithm (c) Comparison of coverage results

in Table 2. In this experiment, we consider 72 orientations for each camera by initializing the set of possible camera orientations as $O = \{0°, 5°, 10°, \ldots, 355°\}$.

Across all these environments, we initially focus on coverage maximization by proposing optimal locations and orientations for a user-specified number of cameras. Subsequently, we perform a failure analysis of cameras in the network. Across all the experimental scenarios, we consider the failure of any one or two of the cameras in the network and analyze the corresponding loss in coverage. Subsequently, we propose new locations and orientations to the VGR algorithm-generated subset of neighboring cameras using the m-EGGV and m-RPS algorithms and evaluate the coverage results. We use boxplots to assess the coverage outcomes from our failure analysis. The yellow box represents scenarios where a single camera fails, while the green box indicates scenarios where two cameras fail simultaneously within the surveillance network. The cyan horizontal line acts as a baseline, showing the total coverage across the surveillance area before any failures.

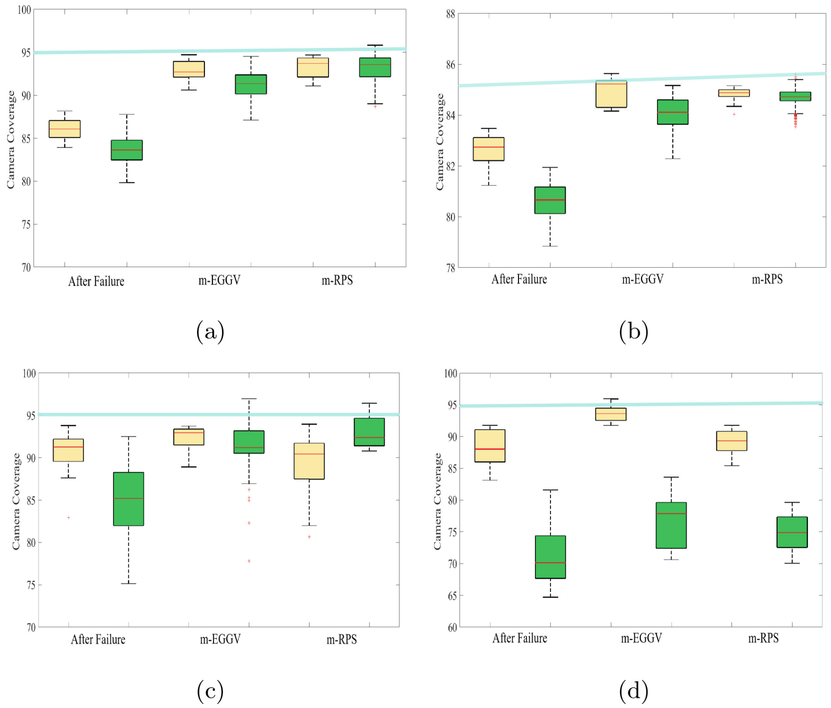
Table 3. Comparison of coverage results of optimization algorithms on various surveillance environments

Algorithms	Plan image of Office		Plan image of university building		Google map	
	No. of Cameras	Total Coverage	No. of Cameras	Total Coverage	No. of Cameras	Total Coverage
LG	46	81	54	72	17	81
GG		90		78		87
AGG		92		79		88
GGV		95		82		92
m-EGGV	26	94	44	78	14	93
m-RPS	26	92	44	82	14	94

4.1 Indoor Environment: Plan Image of an Office

We simulate a preinstalled surveillance network with 46 ‘AXIS P1357’ static network cameras in an office building of 3800 square feet area as shown in Fig. 4a. Initially, we evaluate the performance of coverage optimization algorithms such as LG, GG, AGG, and GGV by proposing optimal orientation to the 46 cameras mounted at predefined camera locations over the plan image of the Office. The LG, GG, AGG, and GGV algorithms attained 81%, 90%, 92%, and 95% coverage, respectively. Subsequently, we explore coverage optimization in the same surveillance region without any pre-installed cameras, with only a limited number of cameras available for deployment. In this scenario, we evaluate the performance of the m-EGGV and m-RPS algorithms, using a user-specified input of 26 cameras($N_{giv} = 26$). Both algorithms propose locations and orientations for these 26 cameras, achieving total coverage of 94%, and 92%, respectively, as illustrated in Fig. 6a and Fig. 6b. Fig. 6c presents the iterative coverage results for each camera placement across various algorithms, with the red plot representing the m-RPS algorithm and the blue plot representing the m-EGGV algorithm. In this comparison, the m-EGGV algorithm yields superior coverage by proposing optimal locations and orientations for the 26 cameras provided by the user. The Table 3 lists the total coverage obtained with these algorithms over the plan image of the office environment.

Following the coverage optimization experiments, we conduct a camera failure analysis on the office building’s plan image, which has an existing surveillance network of 27 cameras achieving 95% total coverage. We consider all possible scenarios involving the failure of one or two cameras, represented by combinations ${}_{27}C_1$ and ${}_{27}C_2$ respectively. Upon camera failure within the network, the VGR algorithm identifies a potential subset of neighboring cameras that could alleviate the loss in coverage. Subsequently, we propose new locations and orientations for the VGR-identified subset of cameras using the m-EGGV and m-RPS algorithms and analyze the resultant coverage. Fig. 7a displays a box plot illus-



(a)

(b)

(c)

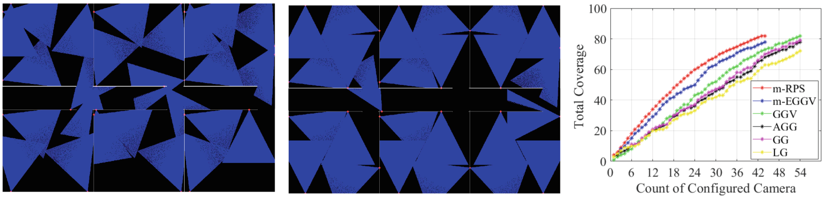
(d)

Fig. 7. Coverage results on surveillance environments after camera failure and after relocation (a) Plan image of an office (b) Plan image of a University building (c) Google map image of a campus (d) Map image of a crossroad junction

Table 4. Coverage results on various surveillance scenarios after failure of cameras

Scenarios	No. of Cameras	Total Coverage Before Damage	No. of Damaged Cameras	No. of Samples	Total Coverage After Damage	No. of Active Cameras	VGR Subset	Relocate VGR	
								m-EGGV	m-RPS
Plan Image of Office	27	95%	1	${}_{27}C_1 = 27$	86.07%	26	3	93.24%	94.71%
			2	${}_{27}C_2 = 351$	83.63%	25	7	91.82%	94.54%
Plan Image of University	56	85%	1	${}_{56}C_1 = 56$	82.74%	55	6	85.22%	84.88%
			2	${}_{56}C_2 = 1540$	80.66%	54	11	84.11%	84.73%
Google Map of Campus	15	95%	1	${}_{15}C_1 = 15$	90.88%	14	4	93.12%	90.89%
			2	${}_{15}C_2 = 105$	85.99%	13	6	91.23%	92.53%
Map Image of Crossroad	8	95%	1	${}_{8}C_1 = 8$	88.02%	7	4	93.62%	89.35%
			2	${}_{8}C_2 = 28$	70.15%	6	5	77.8%	74.9%

trating the coverage analysis post-camera failure and post-reconfiguration (in terms of new locations and orientations proposed for the VGR set of cameras). Additionally, Table 4 lists the median coverage values from this failure analysis. Both algorithms demonstrate their ability to reduce the loss in coverage resulting from camera damage.



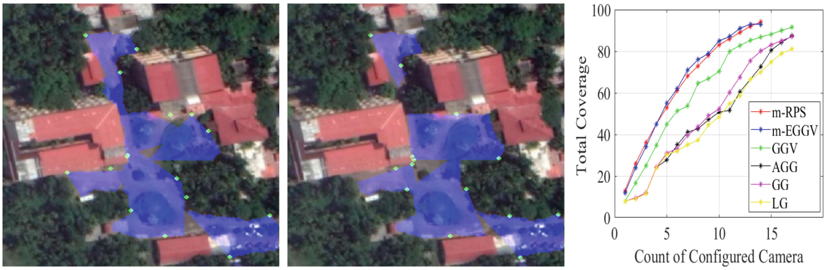
(a) (b) (c)

Fig. 8. Indoor environment: Plan image of a university building (a) Coverage results of EGGV algorithm (b) Coverage results of RPS algorithm (c) Comparison of coverage results

4.2 Indoor Environment: Plan Image of University Building

We simulate an existing surveillance network with 54 preinstalled 'AXIS P1357' static network cameras in a 7800 square feet area of a university building as shown in Fig. 4b. Initially, we evaluated the performance of coverage optimization algorithms such as LG, GG, AGG, and GGV by proposing optimal orientation to the 54 cameras mounted at predefined camera locations over the plan image of the university building. The LG, GG, AGG, and GGV algorithms attained 72%, 78%, 79%, and 82% coverage, respectively. Subsequently, we explore coverage optimization in the same surveillance region without any preinstalled cameras, with only a limited number of cameras available for deployment. In this scenario, we evaluate the performance of the m-EGGV and m-RPS algorithms, using a user-specified input of 44 cameras($N_{giv} = 44$). Both algorithms propose locations and orientations for these 44 cameras, achieving total coverage of 78%, and 82%, respectively, as illustrated in Fig. 8a and Fig. 8b. Fig. 8c presents the iterative coverage results for each camera placement across various algorithms, with the red plot representing the m-RPS algorithm and the blue plot representing the m-EGGV algorithm. In this comparison, the m-RPS algorithm yields superior coverage by proposing optimal locations and orientations for the 44 cameras provided by the user. Table 3 lists the total coverage obtained with these algorithms over the plan image of the university building.

Following the coverage optimization experiments, we conduct a camera failure analysis on the university building's plan image, which has an existing surveillance network of 56 cameras with 85% total coverage. We consider all possible scenarios involving the failure of one or two cameras, represented by combinations ${}_{56}C_1$ and ${}_{56}C_2$ respectively. Upon camera failure within the network, the VGR algorithm identifies a potential subset of neighboring cameras that could alleviate the loss in coverage. Subsequently, we propose new locations and orientations for the VGR-identified subset of cameras using the m-EGGV and m-RPS algorithms and analyze the resultant coverage. Fig. 7b displays a box plot illustrating the coverage analysis post-camera failure and post-reconfiguration (in terms of new locations and orientations proposed for the VGR set of cameras). Additionally, Table 4 lists the median coverage values from this failure



(a) (b) (c)

Fig. 9. Indoor environment: Google map image of a campus (a) Coverage results of EGGV algorithm (b) Coverage results of RPS algorithm (c) Comparison of coverage results

analysis. Both algorithms demonstrate their capability to alleviate the loss in coverage resulting from camera damage.

4.3 Outdoor Environment: Google Map Image of a Campus

In our third surveillance simulation, we focus on an outdoor setting with an existing surveillance network consisting of 17 preinstalled ‘BIPRO-540L4’ static network cameras, utilizing a Google map image of a campus environment shown in Fig. 4c. Initially, we evaluated the performance of coverage optimization algorithms such as LG, GG, AGG, and GGV by proposing optimal orientation to the 17 cameras mounted at predefined camera locations over the Google image of a campus. The LG, GG, AGG, and GGV algorithms attained 81%, 87%, 88%, and 92% coverage, respectively. Subsequently, we explore coverage optimization in the same surveillance region without any pre-installed cameras, with only a limited number of cameras available for deployment utilizing the sketch map of the surveillance region as shown in Fig. 5a. In this scenario, we evaluate the performance of the m-EGGV and m-RPS algorithms, using a user-specified input of 14 cameras($N_{giv} = 14$). Both algorithms propose locations and orientations for these 14 cameras, achieving total coverage of 93%, and 94%, respectively, as illustrated in Fig. 9a and Fig. 9b. Fig. 9c presents the iterative coverage results for each camera placement across various algorithms, with the red plot representing the m-RPS algorithm and the blue plot representing the m-EGGV algorithm. In this comparison, the m-RPS algorithm yields superior coverage by proposing optimal locations and orientations for the 14 cameras provided by the user. The Table 3 lists the total coverage obtained with these algorithms over the Google map image of a campus.

Following the coverage optimization experiments, we conduct a camera failure analysis on the Google map image of the campus, which has an existing surveillance network of 15 cameras achieving 95% total coverage. We consider all possible scenarios involving the failure of one or two cameras, represented by combinations ${}_{15}C_1$ and ${}_{15}C_2$ respectively. Upon camera failure within the

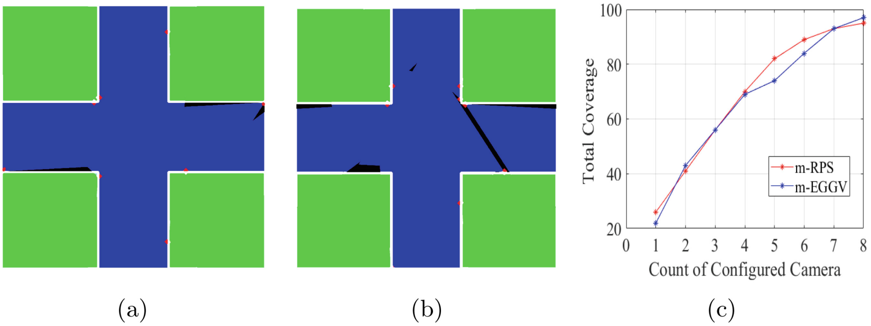


Fig. 10. Oudoor environment : Map image of a crossroad junction (a) Coverage results of EGGV algorithm (b) Coverage results of RPS algorithm (c) Comparison of coverage results

network, the VGR algorithm identifies a potential subset of neighboring cameras that could alleviate the loss in coverage. Subsequently, we propose new locations and orientations for the VGR-identified subset of cameras using the m-EGGV and m-RPS algorithms and analyze the resultant coverage. Fig. 7c displays a box plot illustrating the coverage analysis post-camera failure and post-reconfiguration (in terms of new locations and orientations proposed for the VGR set of cameras). Additionally, Table 4 lists the median coverage values from this failure analysis. Both algorithms demonstrate their capability to alleviate the loss in coverage resulting from camera damage.

4.4 Outdoor Environment: Map Image of Crossroad Junction

Finally, we simulate a crossroad junction that spans over 500 square meters without any existing surveillance network. This scenario is illustrated in Fig. 4d and the sketch map highlighting the suitable locations for camera placement is shown in Fig. 5b. In this scenario, we evaluate the performance of the m-EGGV and m-RPS algorithms, using a user-specified input of 8 DS2CE16C2T-IR cameras($N_giv = 8$). Both algorithms propose locations and orientations for these 8 cameras and attain total coverage of 95% as illustrated in Fig. 10a and Fig. 10b. Fig. 10c presents the iterative coverage results for each camera placement across these algorithms, with the red plot representing the m-RPS algorithm and the blue plot representing the m-EGGV algorithm. Both algorithms exhibit similar performance over the map image of the crossroad junction.

Following the coverage optimization experiments, we conduct a camera failure analysis on the map image of the crossroad junction, which has an existing surveillance network with 8 cameras achieving 95% total coverage. We consider all possible scenarios involving the failure of one or two cameras, represented by combinations ${}_8C_1$ and ${}_8C_2$ respectively. Upon camera failure within the network, the VGR algorithm identifies a potential subset of neighboring cameras that could alleviate the loss in coverage. Subsequently, we propose new locations and

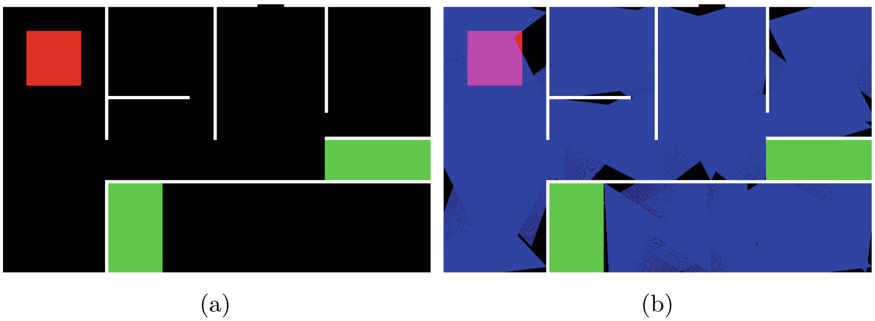


Fig. 11. Plan image of (a) Office space with user-specified constraints (b) Coverage results of m-EGGV algorithm

orientations for the VGR-identified subset of cameras using the m-EGGV and m-RPS algorithms and analyze the resultant coverage. Fig. 7d displays a box plot illustrating the coverage analysis post-camera failure and post-reconfiguration (in terms of new location and orientation proposals for the VGR set of cameras). Additionally, Table 4 lists the median coverage values from this failure analysis. Both algorithms demonstrate their capability to mitigate the coverage loss caused by camera damage.

4.5 Indoor Surveillance Environments with Constraints

Creating an efficient indoor surveillance network, while taking into account factors like critical regions and privacy-sensitive regions, presents a significant challenge. Critical regions are high-priority areas, such as bank vaults, that mandate redundant or backup camera coverage for security purposes. In contrast, privacy-sensitive regions, like restrooms, are areas where camera coverage should be avoided to protect privacy. These specific regions are highlighted on the building plan images, which are then provided as modified inputs for the m-EGGV and m-RPS algorithms, as detailed in [22]. In the context of surveillance, privacy-sensitive regions are treated as occlusions. In our experiments, areas requiring no coverage are marked in green to denote privacy sensitivity, while critical areas are highlighted in red to indicate a need for enhanced surveillance, as shown in Fig. 11a.

The original EGGV and RPS algorithms create a priority map from the input sketch map as described in [22]. In this priority map, critical regions are assigned as m, regions requiring privacy are set to 0, and the remaining regions are assigned as 1. These priorities guide the EGGV and RPS algorithms in optimizing coverage considering user constraints. Similarly, our modified versions, the m-EGGV and m-RPS algorithms, are designed to manage various constraints within the surveillance area while configuring a specified number of cameras N_{giv} Fig. 11b shows the coverage outcomes from our m-EGGV algorithm when assigned to deploy 25 cameras, ($N_{giv} = 25$). The algorithm strategically places cameras to ensure coverage of critical regions while avoiding privacy-sensitive areas.

5 Conclusions

In this work, we have presented robust algorithms for deploying multi-camera surveillance networks, emphasizing cost-effectiveness and functionality. Our modified versions of the Extended Greedy Grid Voting (m-EGGV) and Reward Penalty Score (m-RPS) algorithms demonstrate a significant advancement in optimizing camera placement. As part of this work, we have effectively tackled two key challenges in coverage optimization across diverse indoor and outdoor environments, (i) maximizing coverage with a specified number of cameras by identifying optimal locations and orientations when the number of cameras available for setting up the surveillance network is limited, and (ii) alleviating coverage loss when one or more cameras in a surveillance network is damaged, by proposing new locations and orientations for potential cameras that can compensate for the coverage loss. Overall, our findings contribute valuable insights into efficient surveillance design, proving especially useful for environments where camera resources are limited, but broad coverage is imperative.

References

1. Akbarzadeh, V., Gagné, C., Parizeau, M., Argany, M., Mostafavi, M.A.: Probabilistic sensing model for sensor placement optimization based on line-of-sight coverage. IEEE Trans. Instrum. Meas. **62**(2), 293–303 (2012)
2. Altahir, A.A., Asirvadam, V.S., Hamid, N.H.B., Sebastian, P., Saad, N.B., Ibrahim, R.B., Dass, S.C.: Optimizing visual surveillance sensor coverage using dynamic programming. IEEE Sensors Journal **17**(11), 3398–3405 (jun 2017). https://doi.org/10.1109/jsen.2017.2694385
3. Bouyagoub, S., Bull, D.R., Canagarajah, N., Nix, A.: Automatic multi-camera placement and optimisation using ray tracing. In: 2010 IEEE International Conference on Image Processing. IEEE (sep 2010). https://doi.org/10.1109/icip.2010.5649559
4. Chakrabarty, K., Iyengar, S., Qi, H., Cho, E.: Grid coverage for surveillance and target location in distributed sensor networks. IEEE Transactions on Computers **51**(12), 1448–1453 (dec 2002). https://doi.org/10.1109/tc.2002.1146711
5. Dhillon, S., Chakrabarty, K.: Sensor placement for effective coverage and surveillance in distributed sensor networks. In: 2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003. IEEE. https://doi.org/10.1109/wcnc.2003.1200627
6. Erdem, U.M., Sclaroff, S.: Automated placement of cameras in a floorplan to satisfy task-specific constraints. Boston University, Boston December (2003)
7. Geissler, F., Grafe, R.: Optimized sensor placement for dependable roadside infrastructures. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE (oct 2019). https://doi.org/10.1109/itsc.2019.8917197
8. Hänel, M.L., Schönlieb, C.B.: Efficient global optimization of non-differentiable, symmetric objectives for multi camera placement. IEEE Sensors Journal (2021)

9. Hearn, D.D., Baker, M.P., Carithers, W.: Computer graphics with open GL. Prentice Hall Press (2010)
10. Hörster, E., Lienhart, R.: On the optimal placement of multiple visual sensors. In: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks - VSSN '06. ACM Press (2006). https://doi.org/10.1145/1178782.1178800
11. Indu, S., Chaudhury, S., Mittal, N., Bhattacharyya, A.: Optimal visual sensor placement using evolutionary algorithm. In: National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics. pp. 1–5 (2008)
12. Kritter, J., Brévilliers, M., Lepagnot, J., Idoumghar, L.: On the optimal placement of cameras for surveillance and the underlying set cover problem. Applied Soft Computing **74**, 133–153 (jan 2019). https://doi.org/10.1016/j.asoc.2018.10.025
13. Liu, J., Sridharan, S., Fookes, C.: Recent advances in camera planning for large area surveillance: A comprehensive review. ACM Computing Surveys (CSUR) **49**(1), 1–37 (2016)
14. Liu, J., Sridharan, S., Fookes, C., Wark, T.: Optimal camera planning under versatile user constraints in multi-camera image processing systems. IEEE Transactions on Image Processing **23**(1), 171–184 (jan 2014). https://doi.org/10.1109/tip.2013.2287606
15. Mavrinac, A., Chen, X.: Modeling coverage in camera networks: A survey. Int. J. Comput. Vision **101**(1), 205–226 (2013)
16. Mohan, L., Menon, V.: Modelling large scale camera networks for identification and tracking: an abstract framework. IET Comput. Vision **14**(7), 426–433 (2020)
17. Pechenkin, V., Dolinina, O., Brovko, A., Korolev, M.: Analysis of 3d scene visual characteristics based on virtual modeling for surveillance sensors parameters pp. 328–340 (2020)
18. Rehder, J., Siegwart, R.: Camera/IMU calibration revisited. IEEE Sensors Journal **17**(11), 3257–3268 (jun 2017). https://doi.org/10.1109/jsen.2017.2674307
19. Shahbaz, A., Jo, K.H.: Improved change detector using dual-camera sensors for intelligent surveillance systems. IEEE Sens. J. **21**(10), 11435–11442 (2020)
20. Spielberg, A., Amini, A., Chin, L., Matusik, W., Rus, D.: Co-learning of task and sensor placement for soft robotics. IEEE Robotics and Automation Letters **6**(2), 1208–1215 (apr 2021). https://doi.org/10.1109/lra.2021.3056369
21. Suresh, M.S.S., Menon, V.: An efficient graph based approach for reducing coverage loss from failed cameras of a surveillance network. IEEE Sens. J. **22**(8), 8155–8163 (2022)
22. Suresh, M.S.S., Menon, V.: A generic and scalable approach to maximize coverage in diverse indoor and outdoor multicamera surveillance scenarios. IEEE Transactions on Systems, Man, and Cybernetics: Systems **53**(2), 1172–1182 (2022). https://doi.org/10.1109/TSMC.2022.3194209
23. Suresh, M.S.S., Narayanan, A., Menon, V.: Maximizing camera coverage in multicamera surveillance networks. IEEE Sensors Journal **20**(17), 10170–10178 (sep 2020)
24. Sušanj, D., Pinčić, D., Lenac, K.: Effective area coverage of 2d and 3d environments with directional and isotropic sensors. IEEE Access **8**, 185595–185608 (2020)
25. Wang, Y.C., Tseng, Y.C.: Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage. IEEE Trans. Parallel Distrib. Syst. **19**(9), 1280–1294 (2008)

DSTNet: Distinguishing Source and Target Areas for Image Copy-Move Forgery Detection

Kaiqi Zhao[1], Xiaochen Yuan[2](✉), Guoheng Huang[3], and Kun Liu[4]

[1] School of Cyberspace Security, Shandong University of Political Science and Law, Jinan, China

[2] Faculty of Applied Sciences, Macao Polytechnic University, Macau SAR, China
xcyuan@mpu.edu.mo

[3] School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China
kevinwong@gdut.edu.cn

[4] School of Information Science and Engineering, University of Jinan, Jinan, China
liukun@ujn.edu.cn

Abstract. In copy-move forgery detection, most relevant studies concern locating the copy-move areas without the distinction of source and target regions. This paper proposes an end-to-end network, DSTNet, to identify the source and target based on consistency detection between the copy-move region and the non-copy-move region. The DSTNet is composed of two stages, the Pre-processing stage and the Discrimination stage. Pre-processing Stage extracts internal information of copy-move and non-copy-move areas and conducts a series of operations to meet the requirements of network input. Discrimination stage allows multiple patches for input and classifies the input patches. Specifically, the Pre-processing stage, contains the Copy-move Patches Selection (CM Patches Selection) and Genuine Patches Selection, can select pairs of copy-move and none copy-move patches. We train the proposed DSTNet on two large synthetic datasets and use the public datasets CASIA and Comofod for evaluation. The experiment shows that our method achieves excellent results. Particularly, we achieve a 5.4% higher $F1$ based on ground-truth of copy-move mask (GT-CM) on CASIA dataset.

Keywords: Copy-move forgery detection · Source and target areas distinguishing · Deep learning for forensics · Siamese network

1 Introduction

Due to the development of digital media technology, some image-processing tools have been extensively applied to society. Most of them change the content

This work was supported by the Science and Technology Development Fund of Macau SAR (Grant number 0045/2022/A), and the Macao Polytechnic University (Project No. RP/FCA-12/2022).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 321–333, 2025.
https://doi.org/10.1007/978-3-031-78312-8_21

of images to satisfy people's needs, such as removal, splicing, and copy-move. Although image processing can improve the viewing of the content and attract the attention of many people, if misused, it can lead to many negative behaviors. In recent years, image forgery detection methods are designed to settle these problems, including image copy-move forgery detection (CMFD), which is copying one or more source areas and pasting it or them to the target areas of the same image.

Most CMFD methods can only detect the copy-move mask of Fig. 1, without distinguishing the source and target areas. These CMFD methods contains traditional and deep learning methods. In traditional methods, the substance of block-based methods is to compartmentalize the forged image into several blocks and expose the copy-move areas. Some approaches of feature extraction, such as DCT [8], DWT [19], and Zernike [4] have been applied in overlapping blocks. To decline the computational cost, the keypoint-based methods extract the key points by SIFT [10] and SURF [5]. In addition to these hand-crafted feature based methods, recent deep learning methods have also been proposed in CMFD. [17] presents a dense-inception net by combine DenseNet [9]and InceptionNet [12] to locate the copy-move areas. Later, [18] develops an adaptive attention module to focus more attention on the principal features. Although some methods have contributed to CMFD, they only locate the copy-move areas to produce the binary copy-move mask but do not distinguish between source and target areas. According to our investigation, there are currently three CMFD methods that can distinguish source and target regions. BusterNet [15] uses part of splicing images for training on the second branch, resulting in a limited ability to detect forged regions. As is shown as Fig. 1, Copy-move mask is a binary graph, the black part represents genuine regions and the white part represents copy-move regions. In our experiment, it can be the copy-move binary ground-truth in the dataset or the result detected by the copy-move modules (CM-Module). Copy-move patches are the smallest external rectangles clipped in copy-move images according to the copy-move mask. In Fig. 1(a), CMSTD [2] uses the image classification, however, copy-move forgery usually goes through a lot of post-processing, and it is difficult to learn the corresponding features only for individual targets or sources. Unlike image classification based on semantic information, the features of the target and source are not simple and contain major distinguishing information. In [1], which is shown as Fig. 1(b), only copy-move images corresponding to copy-move masks containing two independently connected domains can be detected.

Given the shortcomings exposed by the above methods, we proposed DST-Net, a discriminate method to distinguish the source and target regions in copy-move areas, and contributions are (1) Since it is hard to detect the boundary of copy-move regions with 100% accuracy in the CM-Module, we chose to distinguish the source and target regions based on content. And we cut corresponding background patches from images according to the size of the copy-move areas to ensure that the size of each pair is consistent before entering the network In the Genuine Patches Selection. (2) in the Discrimination stage, we use consistency detection based on the Siamese network. The Discrimination stage accepts input from multiple patches, therefore, we can detect altered images containing multiple sources or targets.

The rest of this paper is organized as follows: Section 2 focuses on the proposed method in detail. The training setting, ablation studies, experimental results, and comparisons are illustrated in Section 3. At last, the conclusion is given in Section 4.

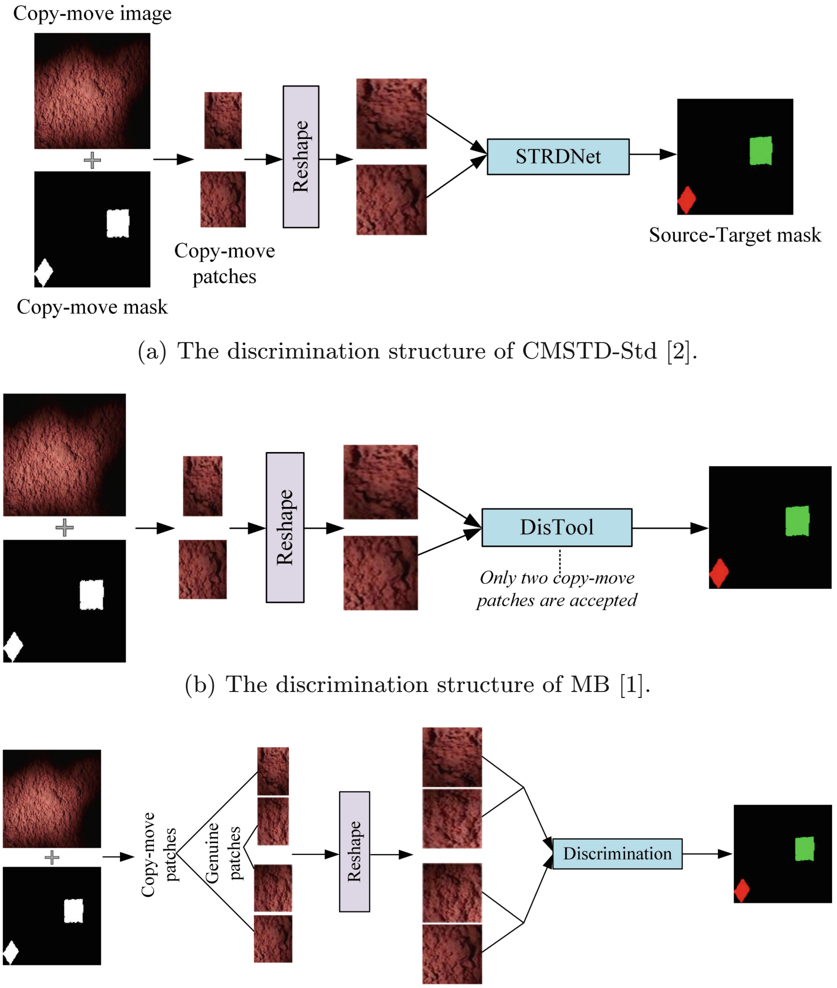
2 Method

In CMFD, it is difficult to locate directly in the whole image because of the high similarity between source and target region and the lack of obvious forgery features. Therefore, we propose an idea to divide cmfd into two phases, namely CM-Module and source-target distinction module (ST-Module). We have described the methods of the first phase in detail in the previous article, which are used to detect all copy-move regions without distinguishing between source and target. In this section, for the second phase, we propose a ST-Module based on the internal features of copy-move region and the consistency of genuine regions. Our approach consists of two stages, the Pre-processing Stage and the Discrimination Stage, which are shown in Fig. 2.

2.1 Pre-Processing Stage

Before the discrimination, we need to perform some operations on the image, which is shown in Fig. 2. Pre-processing stage includes CM Patches Selection and Genuine Patches Selection. Given a copy-move binary mask, we first find the connected domains in the Mark Connected Domains step of CM Patches Selection. Since the detection results of CM-Module may not be completely accurate, the copy-move binary mask is usually rough. In the case of this copy-move binary mask with many interference regions, we use morphological processing to clear the smaller interference regions before looking for connected domains. We clip the connected domain before selecting the copy-move image patches by the step of Mark Internal Mask to find the internal areas, and the margin threshold of clipping was set to 10-20 experimentally. Furthermore, we add the internal mask and copy-move image to generate the copy-move patches.

When addressing the issue of copy-move patches, both [2] and [1] directly alter the shape of the image, thereby making the source region susceptible to tampering as well. In our method, we opt for the genuine area as the reference to compare the consistency for each pair of copy-move region and genuine area. In the process of Generate genuine patches, we ensure their consistency with the copy-move area, so that during the reshape, each pair of genuine and copy-move



(a) The discrimination structure of CMSTD-Std [2].

(b) The discrimination structure of MB [1].

(c) The discrimination structure of the proposed method.

Fig. 1. The motivations of the proposed method. Image classification is carried out after the copy-move areas are reshaped in (a), which is also a tampering of the source areas. (b) uses bounds and internal detection, it can only accept copy-move areas that contain two independently connected domains. (c) is our proposed method, we accept one or more copy-move areas, and use the genuine patch as the third party by reshaping both genuine and copy-move patches to reduce the error.

patches has an equivalent degree of changes and errors to a certain extent. To reduce errors, we determine the size of each copy-move area patch generated at the CM Patches Selection to select genuine patches of the same size. Specifically, in the Discriminate stage, each input patch needs to be the size of 128 × 128. If the sizes of the genuine and copy-move patches are different, during the reshaping

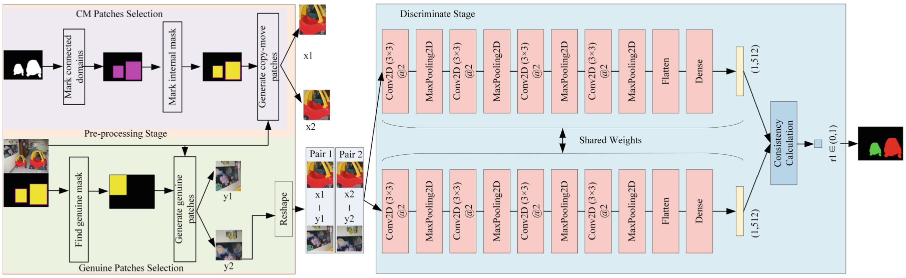


Fig. 2. The Structure of DST-Net containing Pre-processing and Discriminate Stages, the Pre-processing stage is formed from CM Patches Selection and Genuine Patches Selection. xi is the copy-move patch, yi is the corresponding ordinary patch. ri is the detection result with the range of (0,1).

process to the size of 128 × 128, there will be an out-of-sync stretch, which can also be seen as a form of image forgery to some extent. If they are the same size, during the reshape, the original copy-move forgery regions and none forgery regions are uniformly stretched, and the original copy-move tamper error can be ignored. After the CM Patches Selection and Genuine Patches Selection, we reshape the copy-move patch and corresponding genuine patch to $128 \times 128 \times 3$, each pair is entered into the network in turn.

2.2 Discrimination Stage

We first detect the copy-move area through CM-Module. Then, the source and target in the copy-move area are determined by checking the consistency between the copy-move and the genuine areas. In addition, after CM-Module determined the copy-move regions, we could start with the statistical characteristics of the boundary and content.

It is worth mentioning that in the Fig. 2, we only show the copy-move mask containing two separate connected domains. However, our method can also detect overlapping connected domains and copy-move masks that contain more than two connected domains. Our network structure can accept multiple pairs of inputs. The Pre-processing Stage can generate multiple pairs, assuming a $Pair\ i$, it has a copy-move region xi and a corresponding genuine area yi. In the Discrimination stage, we use Siamese network to detect the consistency of each pair. For the backbone, we use vgg16 convolutional network structure, Siamese network has two inputs xi and yi, and one output ri. Among them, xi and yi output a 512-dimensional vector after passing through the same network layer, and then we test the consistency of these two 512-dimensional vectors. SoftMax function is applied to this distance, and the output result ri is obtained, which ranges from 0 to 1. The larger the value of ri is, the higher the similarity of $Pair\ i$ is.

3 Experiments and Discussions

The experiments are implemented with TensorFlow on a GPU with Tesla V100-SXM3-32GB. We compile the network by using the binary_crossentropy loss, and the optimizer we select is Adam with an initial learning rate of 0.0001. We train the model for approximately 30 epochs, and the batch size of each epoch is 8. Next, we introduce the details of training/testing datasets, evaluation metrics, and comparisons.

3.1 Datasets

Training Dataset: We select 756 and 1923 images and their corresponding ground truth in the SPACMFD [16] and Uscisi [15] datasets, respectively. Fig. 3 are several examples from the SPACMFD [16]. As mentioned in Section 2.1, we extract 2679 source patches, 2757 target patches, and corresponding genuine patches respectively. The source patch and its corresponding genuine patch form a positive pair. The target patch and its corresponding genuine patch form a negative pair. To enrich the dataset and increase the robustness of the network, we added Gauss Blur (GB) and Gauss Noise (GN) to each pair to form 16308 pairs.

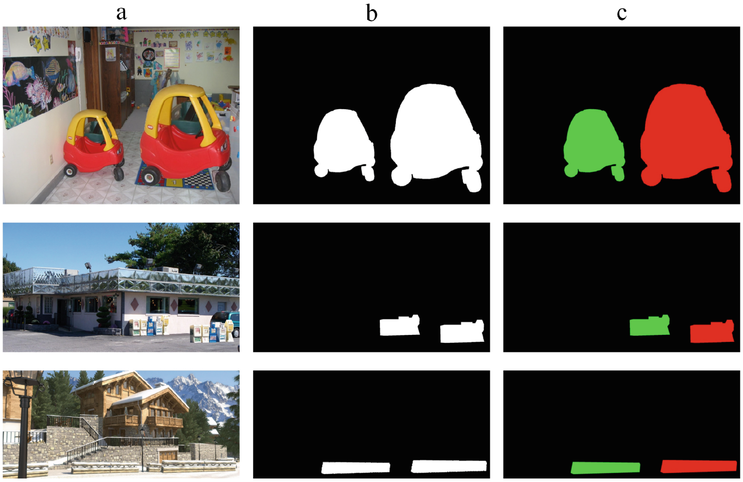


Fig. 3. Copy-move images (a) and their corresponding binary copy-move ground-truth (b) and source-target ground-truth (c) on SPACMFD [16].

Testing Dataset: To evaluate the results between existing and our method on CASIA V2 [6] and CoMoFoD dataset [13]. CASIA v2 [6] has 7,491 genuine and 5,123 forgery images divided into splicing and copy-move. We refer to [15], and 1313 copy-move forgery images are used to evaluate the performance. The binary mask is provided by [15]. Furthermore, in [15], the source-target ground-truth (ST-GT) is labeled source and target areas. CoMoFoD [13] contains 200 basic images and 4800 copy-move forged images generated by applying various post-processing approaches, including JPEG compression, CA, NA, IB, BC, and color reduction (CR). In our experiments, we use CA, CR, IB, and JC to illuminate the performance of the proposed DST-Net.

3.2 Evaluation Metrics

Among the existing methods that can distinguish, copy-move areas are detected as either source or target. Therefore, we can evaluate the pixel-level recognition of one of the two (source and target areas). The evaluation metrics [3,7,14] we use are Precision, Recall and F1, their calculation formulas are as follows:

$$Presicion = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

To evaluate the performance of the proposed method, similar to CMSTD [2], We regard the correct distinguishment number of copy-move images and the accuracy rate as an evaluation criterion. The accuracy rate refers to the ratio of the correct distinguishment number to the number of detected images. When calculating the correct distinguishment number, we adopt the following strategy: set the number of pixels in the source (green) and target (red) regions in the ST-GT as P_{s-gt} and P_{t-gt} respectively, let the source (green) and target (red) regions in the detection result be P_{s-det} and P_{t-det}. On the condition that $P_{s-gt} \cap P_{s-det}$ and $P_{t-gt} \cap P_{t-det}$ are more than 50, we hold the view that this is a correct prediction, otherwise it could be considered a failure.

3.3 Performance and comparison based on known copy-move binary ground-truth

In the case of perfectly correct detection of copy-move regions, we view copy-move images and ground-truth of copy-move mask (GT-CM) of CASIA and CoMoFoD-based datasets as the inputs of DST-Net. To stick out the advantages of DST-Net, we consider DST-Net as the module for distinguishing the source and target (ST-Module) and compare it with the other two methods, CMSTD-Std [2] and MB [1]. Table 1 shows the comparisons of probing the source areas at the pixel level and image level, the metrics in pixel level are Precision, Recall, and

F1. Initially, our method performs best, followed by CMSTD-Std [2]. It should be noted that MB [1] having the worst result. In CASIA and CoMoFoD, GT-CM not only contains two non-overlapping copy-move areas but also contains the following two cases: (1) copy-move areas overlap, that is, only one connected region; (2) there are more than two copy-move areas. However, MB [1] rules input containing both cases as discarded. In contrast to our method, we can detect all the above three cases. One reason is that our network supports the detection of multiple patch pairs of copy-move area and genuine area. Additionally, we evaluate the results of image level, the total number of images in CASIA and CoMoFoD datasets are 1313 and 200, the number of correctly distinguishing the source and target areas in CMSTD-Std [2], MB [1] and proposed DST-Net as *Corr.*. Compared with CMSTD-Std [2] and MB [1]. Our method has a better performance in the two datasets. Besides, we also take a test on CoMoFoD dataset to evaluate the robustness of CMSTD-Std [2], MB [1] and proposed DST-Net, which is shown in Fig. 4. In most post-processing, our method correctly distinguishes the largest number of images. In particular, some copy-move forgery images in CoMoFoD dataset have more than one source region and one target region, they may have one source and multiple targets.

Table 1. Comparisons of pixel level and image level based on GT-CM, which are in terms of *Precision*, *Recall*, *F1*, *Corr.*, and *Accuracy*. The values in bold and underlined represent the first and second values in this evaluation..

Dataset	ST-Modules	GT-CM				
		Precision	Recall	F1	Corr.	Accuracy
CoMoFoD	CMSTD-Std [2]	0.505	0.456	0.479	108	54%
	MB [1]	0.438	0.423	0.431	93	46.5%
	Ours	**0.521**	**0.461**	**0.489**	**121**	**60.5%**
CASIA	CMSTD-Std [2]	0.431	0.487	0.457	542	41.28%
	MB [1]	0.364	0.417	0.389	543	41.36%
	Ours	**0.474**	**0.553**	**0.511**	**658**	**50.1%**

3.4 Performance and comparison based on CM-Modules

To evaluate DST-Net under the condition that the location of copy-move regions is not ideal, we combine the effective methods of locating copy-move regions and the proposed DST-Net as an end-to-end detection. We use the CMSTD-Cm [2] as the location method of copy-move regions, named CM-Module, and we compare DST-Net with CMSTD-Std [2] and MB [1] to verify the ability

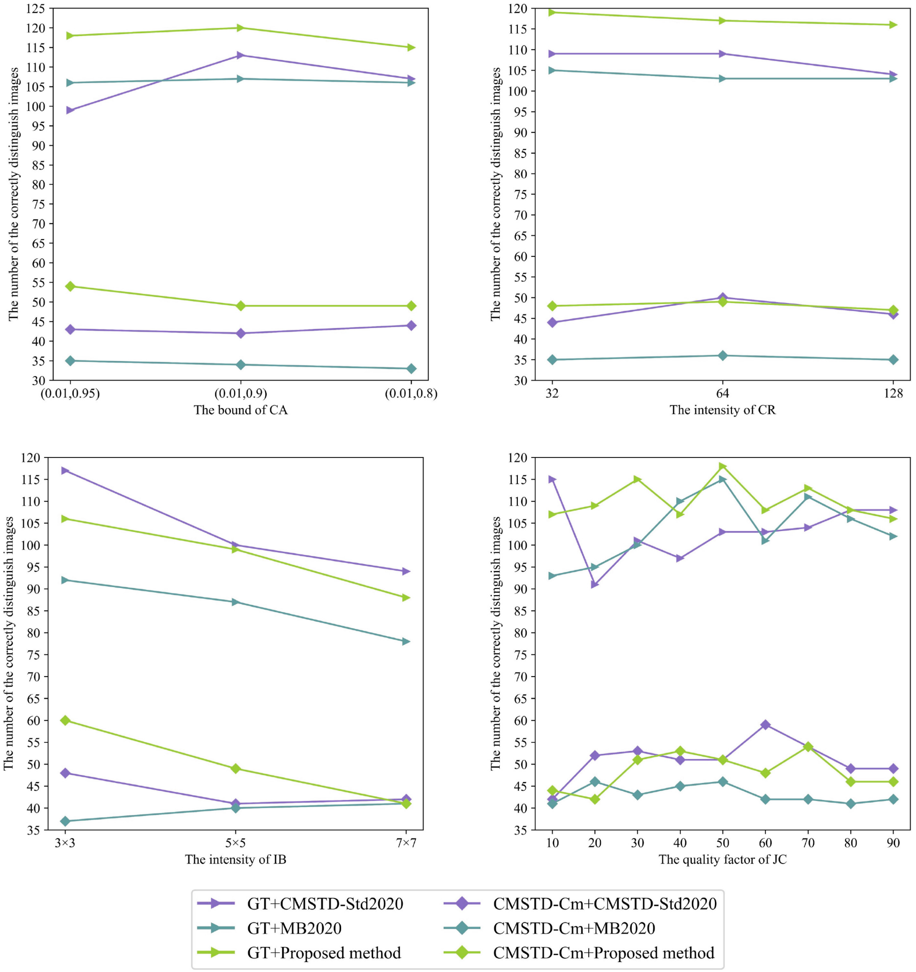


Fig. 4. The comparisons in the number of correctly distinguishing images based on GT-CM and CMSTD-Cm [2] on CoMoFoD-post datasets.

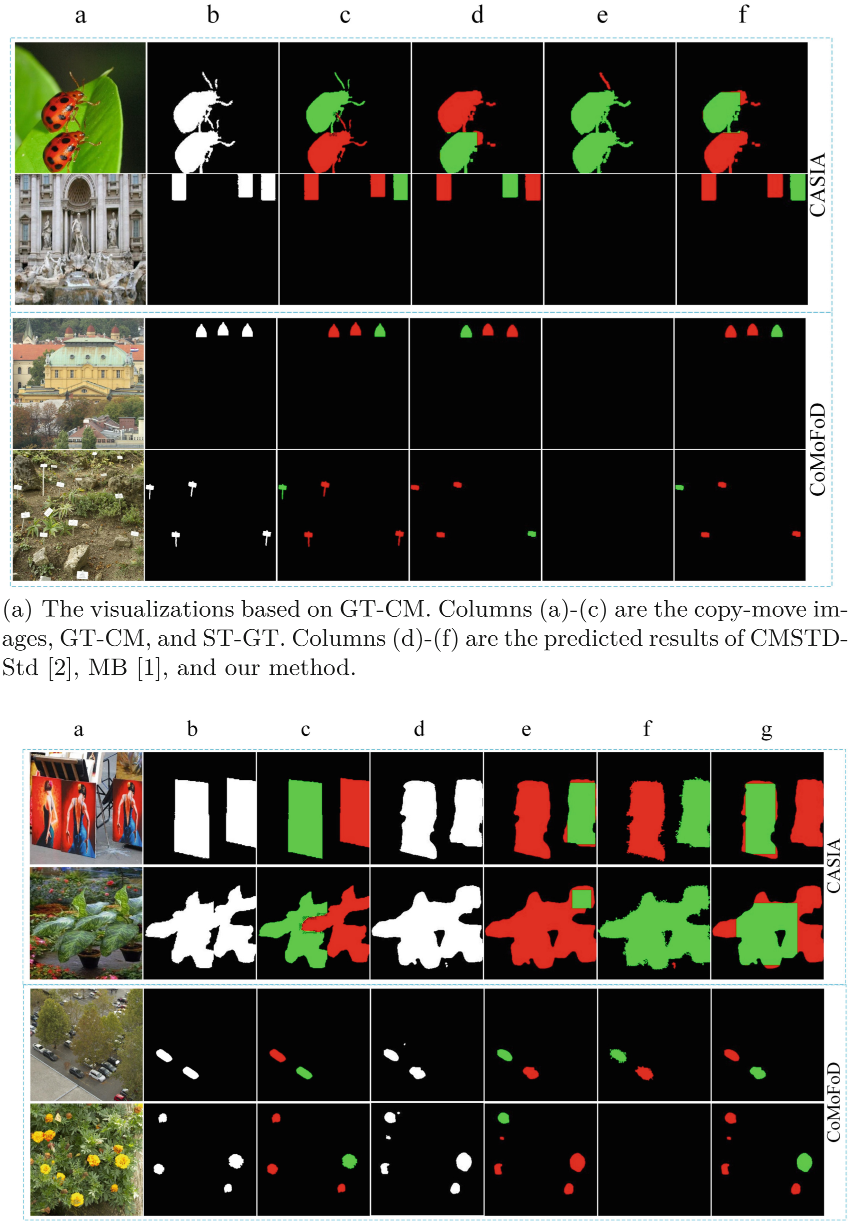
of DST-Net. Additionally, when the F1 score of the result in CM-Module is larger than 0.5, we think the location of copy-move areas is meaningful, then we bring the results of the binary mask to the DST-Net to generate a mask with the source and target labeled. Table 2 exposes pixel-level comparisons in distinguishing the source areas. Our method ranks the top two in both CASIA and CoMoFoD. In particular, compared with the other two methods, we achieve the highest F1 value in CASIA. CMSTD-Std [2] also performs well on these test datasets, and it is reasonable to assume that this is because CMSTD-Cm and CMSTD-Std [2] are originally two serial structures in the same network. CMSTD-std [2] is somewhat inferior to the detection combined with its original Cm-module, CMSTD-Cm [2]. The results based on GT-CM and CMSTD-Cm [2] further demonstrate that our method has better generalization. Table. 2 shows the evaluation of image level, the numbers of meaningful results from the CM-Modules on CASIA and CoMoFoD datasets are 729 and 109, respectively. And the number of correctly distinguishing images based on Cm-Modules as *Corr.* . We also take robustness tests on the CoMoFoD-post dataset, and the results are shown in Fig. 4.

Table 2. Comparisons of pixel level and image level based on CM-Modules, which are in terms of *Precision*, *Recall*, *F1*, *Corr.*, and *Accuracy.* The values in bold and underlined represent the first and second values in this evaluation.

Dataset	ST-Modules	CMSTD-Cm				
		Precision	Recall	F1	Corr.	Accuracy
CoMoFoD	CMSTD-Std [2]	**0.377**	**0.409**	**0.392**	**67**	**61.47%**
	MB [1]	0.245	0.273	0.258	33	30.28%
	Ours	0.325	0.291	0.307	65	59.63%
CASIA	CMSTD-Std [2]	0.388	0.488	0.432	278	38.13%
	MB [1]	0.298	0.413	0.347	247	33.88%
	Ours	**0.409**	**0.492**	**0.447**	**344**	**47.19%**

3.5 Visualizations

To observe the results of our method more directly, we list the detection results based on GT-CM and CMSTD-Cm [2] in different datasets, which are shown in Fig. 5(a) and Fig. 5(b) respectively. Compare with CMTSD-Std [2] and MB [1], the detection results of our method are more accurate when the copy-move region overlaps or the copy-move region contains more than two independent connectivity areas. What's more, our method is outstanding in the distinction of images containing transformed target regions, such as geometric transform and blur, because we choose the genuine region as the third party, which is mainly used to find the difference between the target region and it. MB [1] discards images that it cannot detect, we mark its visualization in black. Since MB [1] only be able to distinguish the copy-move region that has only two independently connected areas, whereas CMTSD-Std [2] and our method can accept inputs from other cases.



(a) The visualizations based on GT-CM. Columns (a)-(c) are the copy-move images, GT-CM, and ST-GT. Columns (d)-(f) are the predicted results of CMSTD-Std [2], MB [1], and our method.

(b) The visualizations based on CMSTD-Cm [2]. Columns (a)-(c) are the copy-move images, GT-CM, and ST-GT. Column (d) denotes the results of CMSTD-Cm [2]. Columns (e)-(g) are the predicted results of CMSTD-Std [2], MB [1], and our method.

Fig. 5. The visualizations based on GT-CM and CMSTD-Cm [2].

4 Conclusion

In this paper, we have introduced a CMFD method to distinguish the source and target regions in copy-move forgery images. We utilize a third party, the genuine region, to calculate the consistency between them. In this process, we use the Siamese network to extract key feature information to perform the discrimination. The experiments show that our method outperforms other methods in CASIA and CoMoFoD datasets, whether based on GT-CM or binary copy-move mask from CM-Modules. In addition, our method, which has a good generalization, can be used as a plug-and-play module in combination with other deep learning based and traditional methods that cannot distinguish source and target areas but can locate copy-move areas. Although our method has achieved good results, when the target region has not been transformed by any geometric processing, the accuracy of our network still needs to be improved. In future work, we tend to use deep learning explanations, such as Grad-cam [11], to generalize consistent or non-consistent features between target/source areas and genuine areas, thereby focusing more attention on important features.

Acknowledgements. This work was supported by the Science and Technology Development Fund of Macau SAR (Grant number 0045/2022/A), and the Macao Polytechnic University (Project No. RP/FCA-12/2022).

References

1. Barni, M., Phan, Q.T., Tondi, B.: Copy move source-target disambiguation through multi-branch cnns. IEEE Trans. Inf. Forensics Secur. **16**, 1825–1840 (2020)
2. Chen, B., Tan, W., Coatrieux, G., Zheng, Y., Shi, Y.Q.: A serial image copy-move forgery localization scheme with source/target distinguishment. IEEE Trans. Multimedia **23**, 3506–3517 (2020)
3. Christlein, V., Riess, C., Jordan, J., Riess, C., Angelopoulou, E.: An evaluation of popular copy-move forgery detection approaches. IEEE Trans. Inf. Forensics Secur. **7**(6), 1841–1854 (2012)
4. Cozzolino, D., Poggi, G., Verdoliva, L.: Efficient dense-field copy-move forgery detection. IEEE Trans. Inf. Forensics Secur. **10**(11), 2284–2297 (2015)
5. Dhivya, S., Sangeetha, J., Sudhakar, B.: Copy-move forgery detection using surf feature extraction and svm supervised learning technique. Soft. Comput. **24**, 14429–14440 (2020)
6. Dong, J., Wang, W., Tan, T.: Casia image tampering detection evaluation database. In: 2013 IEEE China summit and international conference on signal and information processing. pp. 422–426. IEEE (2013)
7. Emam, M., Han, Q., Niu, X.: Pcet based copy-move forgery detection in images under geometric transforms. Multimedia Tools and Applications **75**, 11513–11527 (2016)
8. Fridrich, J., Soukal, D., Lukas, J., et al.: Detection of copy-move forgery in digital images. In: Proceedings of digital forensic research workshop. vol. 3, pp. 652–63. Cleveland, OH (2003)

9. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
10. Prakash, C.S., Panzade, P.P., Om, H., Maheshkar, S.: Detection of copy-move forgery using akaze and sift keypoint extraction. Multimedia Tools and Applications **78**, 23535–23558 (2019)
11. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
12. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
13. Tralic, D., Zupancic, I., Grgic, S., Grgic, M.: Comofod-new database for copy-move forgery detection. In: Proceedings ELMAR-2013. pp. 49–54. IEEE (2013)
14. Wu, Y., Abd-Almageed, W., Natarajan, P.: Deep matching and validation network: An end-to-end solution to constrained image splicing localization and detection. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 1480–1502 (2017)
15. Wu, Y., Abd-Almageed, W., Natarajan, P.: Busternet: Detecting copy-move image forgery with source/target localization. In: Proceedings of the European conference on computer vision (ECCV). pp. 168–184 (2018)
16. Zhao, K., Yuan, X., Xie, Z., Xiang, Y., Huang, G., Feng, L.: Spa-net: A deep learning approach enhanced using a span-partial structure and attention mechanism for image copy-move forgery detection. Sensors **23**(14), 6430 (2023)
17. Zhong, J.L., Pun, C.M.: An end-to-end dense-inceptionnet for image copy-move forgery detection. IEEE Trans. Inf. Forensics Secur. **15**, 2134–2146 (2019)
18. Zhu, Y., Chen, C., Yan, G., Guo, Y., Dong, Y.: Ar-net: Adaptive attention and residual refinement network for copy-move forgery detection. IEEE Trans. Industr. Inf. **16**(10), 6714–6723 (2020)
19. Zimba, M., Xingming, S.: Dwt-pca (evd) based copy-move image forgery detection. International Journal of Digital Content Technology and its Applications **5**(1), 251–258 (2011)

Dual Hypergraph Convolution Networks for Image Forgery Localization

Jiahao Huang, Xiaochen Yuan^(✉), Wei Ke, and Chan-Tong Lam

Faculty of Applied Sciences, Macao Polytechnic University, Macao 999078, China
{p2316936,xcyuan,wke,ctlam}@mpu.edu.mo

Abstract. The continual advancement of image editing techniques has made manipulated images easier to create. Improper use may lead to the proliferation of forged images. In order to detect and locate forged regions within forged images, existing research utilizes various feature views to capture subtle forgery traces. However, forged images exhibit complex higher-order relationships, such as group interaction among regions. The interaction reflects inconsistencies among regions. Therefore, we propose a novel Dual Hypergraph Convolution Network (DHC-Net) to enhance the localization of forged regions by representing group interactions using hypergraphs. The DHC-Net constructs region-wise and edge-wise hypergraph convolution branches to refine the localization of forged region. We validate the DHC-Net on four widely used public datasets, including CASIA1.0, NIST, Columbia, and Coverage. The results demonstrate that the proposed DHC-Net achieves advance localization accuracy.

Keywords: Image Forgery Localization · Hypergraph · Hypergraph Convolution Networks

1 Introduction

Image editing techniques manipulate images through pixel-level content alterations. Due to the rapid advancement of image editing technologies, this capability has been widely applied in domains such as entertainment and advertising creativity. However, improper use has led to the inappropriate purposes of forged images. For instance, modifying images in news reports can distort their original meaning, while altering photos can facilitate insurance fraud. Consequently, the development of methods for identifying and localizing forged regions is increasingly crucial. Image forgery localization typically encompasses various forgery types, such as copy-move, splicing, and inpainting. The manipulation leave subtle traces in certain image areas, distinguishing them significantly from tasks like semantic segmentation.

This work was supported by the Science and Technology Development Fund of Macau SAR (Grant number 0045/2022/A), and the Macao Polytechnic University (Project No. RP/FCA-12/2022).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 334–345, 2025.
https://doi.org/10.1007/978-3-031-78312-8_22

Existing methods [26,29,33] for image forgery localization focus on capturing subtle artifacts within forged regions to differentiate them from authentic ones. Traditional methods [9,23] capture clues of forgery through inconsistencies in JPEG compression artifacts or noise distributions. These approaches offer meticulously handcrafted feature methods, whereas deep learning-based methods provide more generalized solutions. RGB-N [37] constructs a dual-stream network using Fast R-CNN and SRM [7] filters for end-to-end image forgery detection. CR-CNN [1] proposes a constrained convolutional operator for capturing local subtle forgery traces. HP-FCN [15] introduces a deep neural network model based on high-pass filtering residuals. DenseFCN [38] proposes a deep fully connected network demonstrating that sufficiently deep networks can generalize forgery traces that are not easily observable. ManTra-Net [30] provides a robust image forgery localization method, training on 385 manipulation types. SPAN [12] proposes a spatial pyramid attention network encoding spatial correlations between image blocks at different scales. GSR-Net [36] divides the image forgery localization task into three steps: generation, segmentation, and refinement, with specific network structures designed for each step. Furthermore, some methods approach network design from multi-view and multi-scale perspectives. For example, MVSS-Net [3] constructs a multi-view approach based on noise, RGB, and edge supervision, learning the consistency of multiple feature maps across views. PSCC-Net [17] introduces a progressive spatial-channel correlation network for localizing target regions of varying sizes. NEDB-Net [35] introduces a dual-branch network based on noise and edges, incorporating an effective edge merge module design.

However, existing methods have not yet fully considered the complex higher-order relationships within forged images. The higher-order relationship manifest spatially as group interaction among regions. By modeling group interactions among regions, networks can better learn the inconsistencies between authentic and forged regions. Therefore, to address this problem, we propose a novel Dual Hypergraph Convolution Networks (DHC-Net). Hypergraphs are data structures used to represent complex higher-order relationships among objects, widely applied in fields such as complex network analysis [5] and recommendation systems [31]. The proposed DHC-Net consists of three components: a dual-view feature extraction module (DFEM), a region-wise hypergraph convolution branch (RHCB), and an edge-wise hypergraph convolution branch (EHCB). The dual-view feature extraction module comprises a feature extraction networks and an edge extraction block. The region-wise and edge-wise hypergraph convolution branches construct dual-channel group interaction learning, focusing on the forgery traces within regions and edges. Our contributions can be summarized as follows.

– We propose a Dual Hypergraph Convolution Networks (DHC-Net) for image forgery localization. We explore for the first time the potential of hypergraphy in image forgery localization.

– We design a spatial hypergraph modeling approach, constructing hypergraph convolution branches at the region-wise and edge-wise for group interaction. This approach preserves rich information of regions and edge details.
– Experimental results demonstrate that the proposed DHC-Net achieves advanced localization accuracy in image forgery localization. Visualization with Grad-CAM showcases the attention region of the hypergraph convolution branches.

2 Related Works

2.1 Image Forgery Detection

In this section, we primarily introduce deep learning-based methods for image forgery detection. Due to the remarkable performance achieved by deep neural networks in various tasks, it has become a popular practice to learn subtle forgery traces through neural networks. In earlier works such as RGB-N [37], HP-FCN [15], and CR-CNN [1], forgery traces were predominantly captured through specific filters. For instance, RGB-N captures particular noise in forged images using the SRM filter, which has been widely applied in subsequent works. As the SRM filter involves manual parameter tuning, CR-CNN proposes trainable constrained convolutional operators to locally capture specific noise. Furthermore, HP-FCN devises a high-pass filtering residual approach to capture high-frequency information of artifacts. ManTra-Net [30] adopts a unique training strategy by generalizing forgery traces across 385 manipulation types, offering a robust method for image forgery localization. Additionally, MVSS-Net [3] and PSCC-Net [17] approach detection from the perspectives of multi-view and multi-scale respectively. Specifically, MVSS-Net constructs edge view and noise view using Sobel operators and constrained convolutional operators, while PSCC-Net designs a top-down spatial channel correlation network to accommodate scale variations in forged images. Some works [10,20] integrate graph structures to transform forgery localization into a community detection problem. In contrast, we propose a deep learning solution based on hypergraph neural networks, capturing inconsistencies between regions. Moreover, certain works focus on the robustness of image forgery detection. CAT-Net [14] and OSN [28] respectively devise methods to counter post-processing attacks and online media compression in image forgery detection.

2.2 Hypergraph Neural Networks

Hypergraphs [2] are used to model higher-order relationships among components in complex systems. Unlike graphs, hypergraphs enable interaction between groups through message passing among node-hyperedge-node. HGNN [6] is the first neural network method based on hypergraphs. HGNN utilizes the normalized hypergraph Laplacian for convolution operations and designs hypergraph convolutional layers. Subsequently, numerous works propose variants of hypergraph neural networks, such as UniGNN [13], HyperGCN [32], and others. Hypergraph neural networks make significant progress in various fields, including recommendation systems [31], protein prediction [21], and computer vision [16,34]. For example, Wadhwa et al. [25] devise a hypergraph method for hyperrealistic image inpainting to learn contextual information from images. Fu et al. [8] employ hypergraph convolutional networks for continuous image deraining. These works demonstrate the significant role of hypergraphs in computer vision, yet the potential of hypergraph remains to be fully explored.

3 Methodology

Fig. 1 illustrates the framework of the proposed DHC-Net. DHC-Net consists of three components: a dual-view feature extraction module, a region-wise hypergraph convolution branch, and an edge-wise hypergraph convolution branch. In the dual-view feature extraction module, we employ the pre-trained ConvNeXt as the feature extraction network to extract high-level feature maps. Then, we utilize an edge extraction block for edge learning. In the region-wise and edge-wise hypergraph convolution branches, we first model the group interaction relationships of feature maps using spatial hypergraph. To achieve group interaction learning, we utilize hypergraph convolution in both two branches. Subsequently, we merge the feature maps from both branches and proceed with localization. Finally, we optimize the network through region-wise and edge-wise loss functions.

Given an input image $X \in \mathbb{R}^{C\times H\times W}$, we employ ConvNeXt [18] as the feature extraction network to obtain high-level feature maps $X_r \in \mathbb{R}^{768\times \frac{H}{32}\times\frac{W}{32}}$. Subsequently, we extract edge feature maps $X_e \in \mathbb{R}^{768\times \frac{H}{32}\times\frac{W}{32}}$ using an edge extraction block. As illustrated in Fig. 1, the edge extraction block consists of three convolutional layers with kernel sizes of $\{1,3,3\}$. We utilize the high-level feature maps X_r and edge feature maps X_e as inputs to the region-wise and edge-wise hypergraph convolution branches, respectively, forming a dual-view feature extraction.

Given the feature map X_r, we construct the region-wise hypergraph from the perspective of spatial correlation. As shown in Fig. 1, we further extract X_r into node features X_N, and hyperedge features X_{HE}, through two convolutional layers with kernel sizes $\{1,3\}$. This approach captures spatial correlations by exploiting resolution differences of feature maps, representing high-order relationships among regions. We reshape X_N and X_{HE} into $\mathbb{R}^{512\times\frac{HW}{32*32}}$ and $\mathbb{R}^{512\times\frac{HW}{64*64}}$, respectively. Therefore, the process of constructing the region-wise hypergraph is represented as follows:

$$\mathcal{H} = |X_N^T X_{HE}| \tag{1}$$

Next, we construct the hypergraph convolution layer [6] by the hypergraph Laplacian operator. The hypergraph Laplacian operator is defined as follows:

$$\mathcal{G} = \mathbf{I} - \mathbf{D}_v^{-1/2}\mathcal{H}\mathbf{W}_e\mathbf{D}_e^{-1}\mathcal{H}^T\mathbf{D}_v^{-1/2} \tag{2}$$

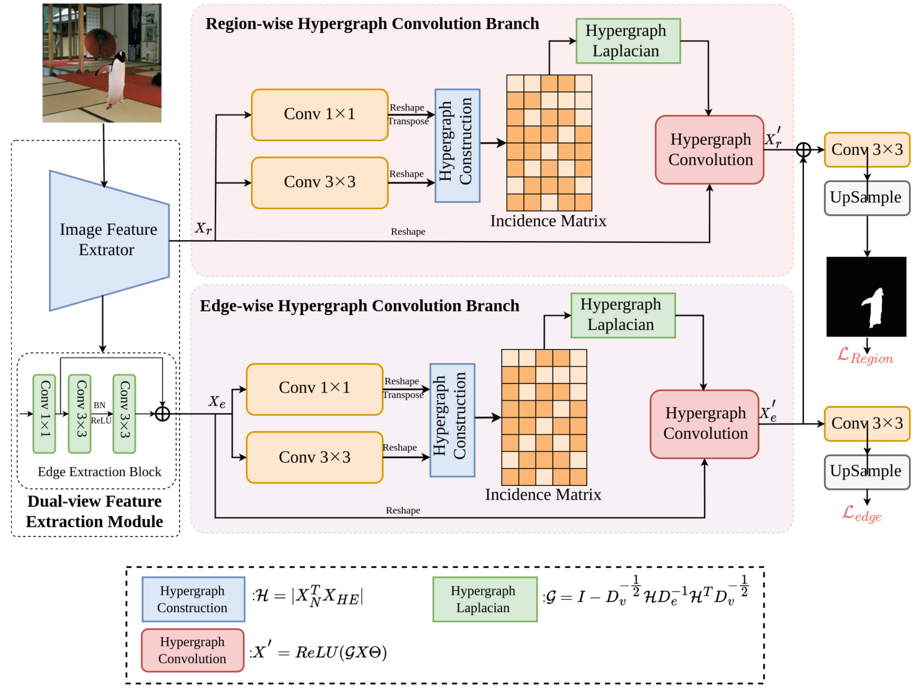


Fig. 1. The framework of the proposed DHC-Net. DHC-Net consists of dual-view feature extraction module (DFEM), region-wise hypergraph convolution branch (RHCB), and edge-wise hypergraph convolution branch (EHCB). Specifically, DFEM comprises a feature extraction network and an edge extraction block. The structures of RHCB and EHCB are consistent, modeling group interactions through spatial hypergraphs and further learning via hypergraph convolution. Before outputting the prediction map for localization, we fuse the feature maps from the two branches.

where $\mathbf{D}_v$ denotes the node degree, $\mathbf{D}_e$ denotes the hyperedge degree, and $\mathbf{W}_e$ represents the hyperedge weight set to 1. Thus, the hypergraph convolution layer is represented as follows:

$$X_r^{'} = ReLu(\mathcal{G} X_r \Theta) \tag{3}$$

where Θ represents the learnable parameter matrix. We reshape X_r and input it into the hypergraph convolution layer to obtain the feature map $X_r^{'}$, facilitating the learning of group interaction relationships among regions. To enable $X_r^{'}$ for forgery localization, we restore its dimension to a low-resolution feature map. Consistent with the region-wise hypergraph convolution branch, we construct a hypergraph for the edge feature maps X_e. By employing hypergraph convolution layer, the model learn group interaction relationships at the edge feature maps, resulting in the feature map $X_e^{'}$.

Through the region-wise and edge-wise hypergraph convolution processes, we perform element-wise addition of the feature maps $X_r^{'}$ and $X_e^{'}$ to obtain feature map X_{seg} . Subsequently, we utilize a convolution layer with a kernel size of 3 to reduce the number of channels of X_{seg} to 1, followed by upsampling to restore it to the original input size for localization. Thus, we construct the region-wise loss function using binary cross-entropy (BCE). To supervise the edge-wise hypergraph convolution branch, we reduce the number of channels of $X_e^{'}$ to 1 using a convolution layer with a kernel size of 3, and then upsample it to 1/4 of the original input size. We utilize binary cross-entropy for edge-wise loss function. Therefore, the loss function of the DHC-Net is represented as follows:

$$\mathcal{L} = \mathcal{L}_{region}(X_{seg}, X_{gt}) + \alpha(\mathcal{L}_{edge}(X_e^{'}, X_{gt}^{'})) \tag{4}$$

where $X_{gt}^{'}$ denotes the edge map of the ground truth, following the edge map generation of [3]. For the balance parameter α, we empirically set it to 0.5.

4 Experiments and Discussions

4.1 Datasets and Evaluation metric

To evaluate the performance of the proposed DHC-Net, we employed widely used cross-dataset evaluation methods and publicly available datasets. Specifically, we utilized CASIA2.0 [4] as the training set, comprising 5063 pairs of forged images and ground truth images. The test set includes CASIA1.0 [4], NIST [11], Columbia [22], and Coverage [27] datasets. CASIA1.0 contains 920 images of splicing and copy-move types, NIST contains 564 images of splicing, copy-move, and inpainting types. Columbia and Coverage are single-type datasets, consisting of 180 splicing-type images and 100 copy-move-type images, respectively. Additionally, we randomly sample 2000 images from DEFACTO [19] as a validation set to supplement the deficiencies in the validation set. Detailed summaries are provided in Table 1.

We employ commonly used pixel-level metrics as in previous works [3,35]: Area Under Curve (AUC), F1 score (F1) and Intersection over Union (IoU). Specifically, the calculation approach for pixel-level F1 score is as follows:

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{5}$$

The calculation approach for IoU is as follows:

$$IoU = \frac{P \cap G}{P \cup G} \tag{6}$$

4.2 Implementation Details

We implement the proposed DHC-Net using PyTorch. For the feature extraction network, we utilize the ConvNeXt-Small which pre-trained on the ImageNet1k

Table 1. Summary of the datasets.

	Datasets	Total	Copy-move	Splicing	Inpainting
Train	CASIA2.0	5063	3235	1828	0
Validation	DEFACTO2k	2000	500	1000	500
Test	CASIA1.0	920	459	461	0
	NIST	564	68	288	208
	Columbia	180	0	180	0
	Coverage	100	100	0	0

dataset. We use Adam as the optimizer, with a learning rate set to $3e-5$, and a batch size of 16. We set the input image size to 512×512. The maximum number of training epochs was set to 100, and we employ early stopping to select the model weights corresponding to the best AUC score on the validation set. Common data augmentations used on the training set include image compression, Gaussian blurring, scaling and flipping, and more.

4.3 Comparisons and Visualizations

Table. 2 presents the AUC, F1 and IoU scores of the proposed DHC-Net and the comparative methods on four test datasets. For HP-FCN, ManTra-Net, and NEDB-Net, we conduct testing using the model weights provided by the authors. For MVSS-Net, GSR-Net and SPAN, we refer to the testing results provided by [3]. For PSCC-Net, we train the model on the CASIA2.0 dataset using the code provided by the authors. The results in Table. 2 indicate that DHC-Net outperforms previous methods in most metrics. Specifically, on the CASIA1.0 dataset, DHC-Net achieve improvements of 10.7%, 0.9% and 2.5% in AUC, F1 and IoU metrics, respectively. Compared to previous methods, the results of DHC-Net demonstrate that learning group interaction relationships can enhance the accuracy of forgery localization. Overall, the results in Table. 2 show the advance performance of DHC-Net in forgery localization, indicating that leveraging hypergraph convolution to capture group interaction relationships is a viable solution.

Fig. 2 presents the output visualizations of DHC-Net and the comparative methods across four testing datasets. Compared to previous methods, DHC-Net demonstrates more accurate localization of forged regions and richer edge details. Furthermore, for authentic regions, DHC-Net is less prone to misjudgments. This may be attributed to the learning of group interactions, which can effectively capture the differences between authentic and forged regions. Overall, the proposed DHC-Net exhibits superior performance in forged region localization, validating the positive role of hypergraphs in forgery localization.

Table 2. Comparison on CASIA1.0, NIST, Columbia and Coverage datasets. The image forgery localization results are measured in terms of AUC(%), F1(%) and IoU(%) scores. Best test results are highlighted in bold while the second best are underlined.

Method	CASIA1.0			NIST			Columbia			Coverage		
	AUC	F1	IOU	AUC	F1	IOU	AUC	F1	IOU	AUC	F1	IOU
HP-FCN [15]	53.0	11.2	6.5	50.6	11.7	7.1	50.7	29.8	17.8	49.6	1.6	0.8
ManTra-Net [30]	54.8	15.1	8.7	60.0	12.2	7.5	53.0	40.9	26.4	51.0	43.5	27.8
SPAN [12]	-	18.4	-	-	22.1	-	-	48.7	-	-	17.2	-
GSR-Net [36]	-	38.7	-	-	28.3	-	-	61.3	-	-	28.5	-
PSCC-Net [17]	72.8	30.7	21.8	70.1	18.9	13.2	75.0	34.2	24.3	54.5	20.1	12.1
MVSS-Net [3]	-	51.3	-	-	**30.4**	-	-	66.0	-	-	**48.2**	-
NEDB-Net [35]	81.2	51.1	44.1	71.3	29.7	22.3	88.9	76.9	68.2	80.5	42.6	**30.7**
Proposed DHC-Net	**91.9**	**52.2**	**46.6**	**83.3**	28.2	**22.6**	**90.0**	**77.3**	**71.2**	**89.1**	31.7	25.1

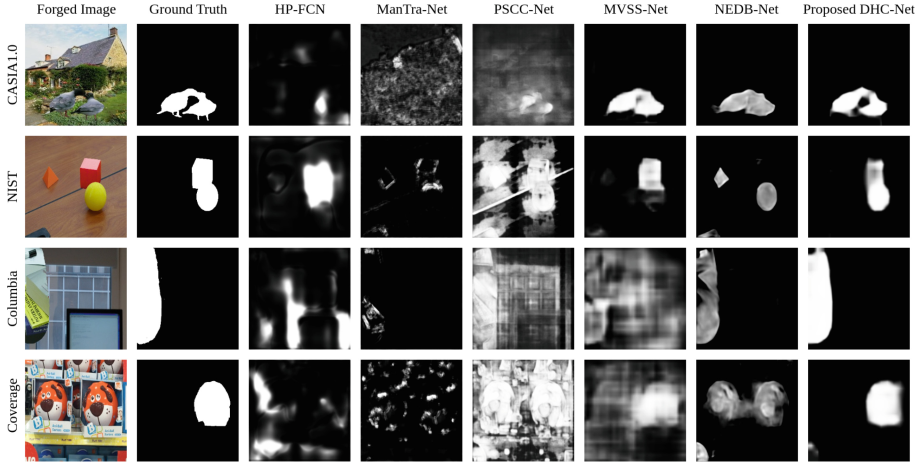


Fig. 2. Visualization results for proposed DHC-Net and compared methods. From top to bottom, we show examples from CASIA1.0, NIST, Colmnbia, and Coverage datasets. We present the visualization results of HP-FCN [15], ManTra-Net [30], PSCC-Net [17], MVSS-Net [3], NEDB-Net [35], and the proposed DHC-Net in columns respectively.

4.4 Ablation Study

Table. 3 presents the ablation results of DHC-Net, where RHCB denotes the removal of edge-related components, including edge extraction block, EHCB, and edge-wise loss function. RHCB+EHCB represents the complete DHC-Net. We report the AUC, F1, and IoU scores on CASIA1.0, NIST, Columbia, and Coverage datasets in Table. 3. It can be observed from Table. 3 that the results of RHCB+EHCB surpass RHCB in all cases. This indicates the significant role of edge-related components in forgery localization, where these components are utilized to enhance the edge details of localization. The EHCB supervised by edge-wise loss not only enhances the edge details but also, through fusion with the RHCB, improves the accuracy of localization.

Fig. 3 illustrates the gradient-based class activation map (Grad-CAM) [24] of different components of DHC-Net. We visualize the weights of the first block of DFEM, RHCB, EHCB, and the output layer to observe the attention region of each component on the forged image. In the activation map visualizations across the four datasets, we observe that the first block of the DFEM primarily learns low-level features of the images, such as textures and edges. These low-level features aid the network in further extracting forgery traces. In the visualization of RHCB, we notice that region-wise hypergraph can maintain localization of the forged regions, with its weights focusing more on the center of the forgery. On the other hand, EHCB exhibits a broader localization towards the edge portions of the forged regions, indicating that the design of EHCB enables it to learn the edges of the forged regions. The output layer combines the feature maps from RHCB and EHCB, thus resulting in more accurate final output results focusing on both the region and the edge of the forged image.

Table 3. Ablation study for DHC-Net. RHCB represents the DHC-Net without edge-related components, including edge extraction block, EHCB, and edge-wise loss function. RHCB+EHCB represents the full DHC-Net. The results include the AUC(%), F1(%), and IoU(%) scores on CASIA1.0, NIST, Columbia, and Coverage datasets.

Method	CASIA1.0			NIST			Columbia			Coverage		
	AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU
RHCB	91.0	49.2	43.3	80.9	27.1	21.1	86.7	72.1	65.0	86.6	31.5	23.6
RHCB + EHCB	91.9	52.2	46.6	83.3	28.2	22.6	90.0	77.3	71.2	89.1	31.7	25.1

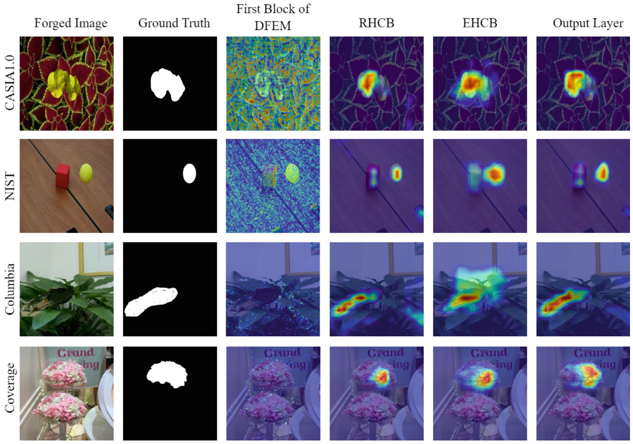


Fig. 3. Activation map for proposed DHC-Net. We utilize the gradient-based class activation map (Grad-CAM) for visualization. From top to bottom, we show examples from four datasets by different part: the first block of DFEM, RHCB, EHCB, and output layer. The activation map shows the response of the module's parameters to the forged regions.

5 Conclusion

Due to the harm caused by forged images in social media, news, and other domains, methods for locating forged regions in images have received widespread attention. In order to address the problem of group interaction among forged regions, we propose a dual hypergraph convolution networks (DHC-Net) by introducing hypergraphs. DHC-Net learns group interaction relationships between regions through region-wise and edge-wise hypergraph learning. Through cross-dataset validation on publicly available datasets include CASIA1.0, NIST, Columbia, and Coverage, DHC-Net surpasses previous methods in most metrics, demonstrating its effectiveness. In future work, we will consider fully exploiting the potential of hypergraph structures in image forgery localization, such as dynamic hypergraph structure learning, to further enhance the expressive power of complex relationships.

Acknowledgment. This work was supported by the Science and Technology Development Fund of Macau SAR (grant number 0045/2022/A), and the Macao Polytechnic University (project No. RP/FCA-12/2022).

References

1. Bayar, B., Stamm, M.C.: Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. IEEE Trans. Inf. Forensics Secur. **13**(11), 2691–2706 (2018)
2. Bretto, A.: Hypergraph theory. An introduction. Mathematical Engineering. Cham: Springer **1** (2013)
3. Dong, C., Chen, X., Hu, R., Cao, J., Li, X.: Mvss-net: Multi-view multi-scale supervised networks for image manipulation detection. IEEE Trans. Pattern Anal. Mach. Intell. **45**(3), 3539–3553 (2022)
4. Dong, J., Wang, W., Tan, T.: Casia image tampering detection evaluation database. In: 2013 IEEE China Summit and International Conference on Signal and Information Processing. pp. 422–426. IEEE (2013)
5. Estrada, E., Rodríguez-Velázquez, J.A.: Subgraph centrality and clustering in complex hyper-networks. Physica A **364**, 581–594 (2006)
6. Feng, Y., You, H., Zhang, Z., Ji, R., Gao, Y.: Hypergraph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3558–3565 (2019)
7. Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. IEEE Trans. Inf. Forensics Secur. **7**(3), 868–882 (2012)
8. Fu, X., Xiao, J., Zhu, Y., Liu, A., Wu, F., Zha, Z.J.: Continual image deraining with hypergraph convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. **45**(8), 9534–9551 (2023)
9. Gan, Y., Zhong, J., Vong, C.: A novel copy-move forgery detection algorithm via feature label matching and hierarchical segmentation filtering. Information Processing & Management **59**(1), 102783 (2022)
10. Gardella, M., Musé, P.: Image forgery detection via forensic similarity graphs. Image Processing On Line **12**, 490–500 (2022)

11. Guan, H., Kozak, M., Robertson, E., Lee, Y., Yates, A.N., Delgado, A., Zhou, D., Kheyrkhah, T., Smith, J., Fiscus, J.: Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In: 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW). pp. 63–72. IEEE (2019)
12. Hu, X., Zhang, Z., Jiang, Z., Chaudhuri, S., Yang, Z., Nevatia, R.: Span: Spatial pyramid attention network for image manipulation localization. In: Computer Vision–ECCV 2020: 16th European Conference. pp. 312–328. Springer (2020)
13. Huang, J., Yang, J.: Unignn: a unified framework for graph and hypergraph neural networks. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021. pp. 2563–2569 (2021)
14. Kwon, M.J., Nam, S.H., Yu, I.J., Lee, H.K., Kim, C.: Learning jpeg compression artifacts for image manipulation detection and localization. Int. J. Comput. Vision **130**(8), 1875–1895 (2022)
15. Li, H., Huang, J.: Localization of deep inpainting using high-pass fully convolutional network. In: proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8301–8310 (2019)
16. Li, X., Li, Y., Shen, C., Dick, A., Van Den Hengel, A.: Contextual hypergraph modeling for salient object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3328–3335 (2013)
17. Liu, X., Liu, Y., Chen, J., Liu, X.: Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. IEEE Trans. Circuits Syst. Video Technol. **32**(11), 7505–7517 (2022)
18. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11976–11986 (2022)
19. Mahfoudi, G., Tajini, B., Retraint, F., Morain-Nicolier, F., Dugelay, J.L., Marc, P.: Defacto: Image and face manipulation dataset. In: 2019 27Th European Signal Processing Conference (EUSIPCO). pp. 1–5. IEEE (2019)
20. Mayer, O., Stamm, M.C.: Exposing fake images with forensic similarity graphs. IEEE Journal of Selected Topics in Signal Processing **14**(5), 1049–1064 (2020)
21. Murgas, K.A., Saucan, E., Sandhu, R.: Hypergraph geometry reflects higher-order dynamics in protein interaction networks. Sci. Rep. **12**(1), 20879 (2022)
22. Ng, T.T., Hsu, J., Chang, S.F.: Columbia image splicing detection evaluation dataset. Columbia Univ CalPhotos Digit Libr, DVMM lab (2009)
23. Pun, C.M., Yuan, X.C., Bi, X.L.: Image forgery detection using adaptive oversegmentation and feature point matching. IEEE Trans. Inf. Forensics Secur. **10**(8), 1705–1716 (2015)
24. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 618–626 (2017)
25. Wadhwa, G., Dhall, A., Murala, S., Tariq, U.: Hyperrealistic image inpainting with hypergraphs. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3912–3921 (2021)
26. Wang, J., Wu, Z., Chen, J., Han, X., Shrivastava, A., Lim, S.N., Jiang, Y.G.: Objectformer for image manipulation detection and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2364–2373 (2022)
27. Wen, B., Zhu, Y., Subramanian, R., Ng, T.T., Shen, X., Winkler, S.: Coverage-a novel database for copy-move forgery detection. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 161–165. IEEE (2016)

28. Wu, H., Zhou, J., Tian, J., Liu, J., Qiao, Y.: Robust image forgery detection against transmission over online social networks. IEEE Trans. Inf. Forensics Secur. **17**, 443–456 (2022)
29. Wu, Y., Abd-Almageed, W., Natarajan, P.: Busternet: Detecting copy-move image forgery with source/target localization. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 168–184 (2018)
30. Wu, Y., AbdAlmageed, W., Natarajan, P.: Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9543–9552 (2019)
31. Xia, X., Yin, H., Yu, J., Wang, Q., Cui, L., Zhang, X.: Self-supervised hypergraph convolutional networks for session-based recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 4503–4511 (2021)
32. Yadati, N., Nimishakavi, M., Yadav, P., Nitin, V., Louis, A., Talukdar, P.: Hypergcn: A new method for training graph convolutional networks on hypergraphs. Advances in Neural Information Processing Systems **32** (2019)
33. Zhai, Y., Luan, T., Doermann, D., Yuan, J.: Towards generic image manipulation detection with weakly-supervised self-consistency learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22390–22400 (2023)
34. Zhang, S., Cui, S., Ding, Z.: Hypergraph-based image processing. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 216–220. IEEE (2020)
35. Zhang, Z., Qian, Y., Zhao, Y., Zhang, X., Zhu, L., Wang, J., Zhao, J.: Noise and edge based dual branch image manipulation detection. In: Proceedings of the 2023 4th International Conference on Computing, Networks and Internet of Things. pp. 963–968 (2023)
36. Zhou, P., Chen, B.C., Han, X., Najibi, M., Shrivastava, A., Lim, S.N., Davis, L.: Generate, segment, and refine: Towards generic manipulation segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 13058–13065 (2020)
37. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Learning rich features for image manipulation detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1053–1061 (2018)
38. Zhuang, P., Li, H., Tan, S., Li, B., Huang, J.: Image tampering localization using a dense fully convolutional network. IEEE Trans. Inf. Forensics Secur. **16**, 2986–2999 (2021)

A Whale Falls, All Thrive: Mitigating Attention Gap to Improve Adversarial Transferability

Qiang Wan[1], Biwei Chen[2], Anjie Peng[1], and Hui Zeng[1](✉)

[1] Southwest University of Science and Technology, Mianyang, China
zengh5@mail2.sysu.edu.cn
[2] Beijing Normal University, Zhuhai, China

Abstract. Deep neural networks (DNNs) are deemed vulnerable to adversarial examples (AEs). Transfer-based attacks enable attackers to craft adversarial images based on local surrogate models without feedback from remote ones. One of the promising attacks is to distract the attention map of the surrogate model that is likely to be shared among remote models. However, we find that the attention maps calculated from a local model are usually over-focus on the most critical area, which limits the transferability of the attacks. In response to this challenge, we propose an enhanced image transformation method (EIT), which guides adversarial perturbations to distract not only the most critical area but also other relevant regions. The proposed EIT effectively mitigates the differences in attention maps between multiple models and better neutralizes model-specific features, thereby avoiding getting stuck in local optima specific to the surrogate model. Experiments confirm the superiority of our approach to the state-of-the-art benchmarks. Our implementation is available at: github.com/britney-code/EIT-attack.

Keywords: Deep Neural Networks · Adversarial Examples · Transfer-based Attack · Attention Map

1 Introduction

It is widely recognized that adversarial examples (AEs) help identify the deep neural networks' (DNNs) vulnerability, which is essential for security-sensitive applications. Since the details of the victim model are usually unavailable, attackers have to craft AEs according to a local surrogate model.

Recently, tremendous efforts have been dedicated to improving the transferability (from the surrogate model to the victim model) of AEs [1–8]. Among them, attention modification approaches that assume different DNNs attend to similar regions for a given image have achieved competitive performance [4, 5, 11, 12]. In this paper, we revisit this hypothesis and find it is not always valid, especially when the structure of the victim model is strikingly different from that of the surrogate model. Our experiments show that

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78312-8_23.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 346–359, 2025.
https://doi.org/10.1007/978-3-031-78312-8_23

existing attention modification-based attacks tend to overemphasize the critical region while neglecting other related areas to a certain extent (Fig. 1(b)), causing insufficient attacks in many object-relative regions. Moreover, model-specific features, i.e., gradient noise dispersed over object-irrelevant regions, distract the attack from focusing on the real object, resulting in limited transferability.

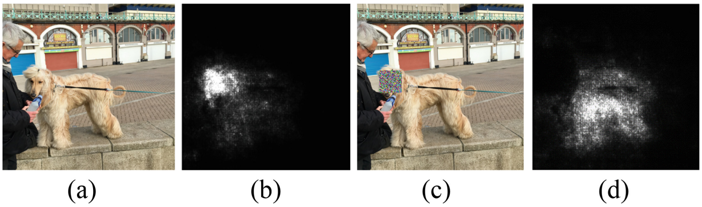


Fig. 1. Visualization of saliency maps. The saliency maps [23] are calculated based on an Inc-v3 model. (a, b) original image and its saliency map. (c, d) masked image and its saliency map.

To address these shortcomings, we propose an enhanced image transformation method (EIT) consisting of random patch masking, noise addition, and scaling. EIT encourages the attack to pay more attention to relevant but non-critical regions by continuously masking the most critical region (Fig. 1(d)) and better neutralizes model-specific information. We summarize our contributions as follows:

- We discover that existing attention modification-based attacks tend to overemphasize the critical region while neglecting other related areas. Additionally, object-irrelevant regions' noise can interfere with the effectiveness of these attacks.
- Inspired by the above observations, a novel attack method EIT is proposed to guide the perturbation to disrupt more object-related features, thereby exhibiting stronger transferability on multiple models.
- Experiments on diverse classification models demonstrate the superior transferability of the proposed EIT method compared to the benchmark attacks.

2 Related Work

Given a classification model f that outputs a label y for an input x, crafting an AE $x\prime$ that is visually indistinguishable from x can be formulated as the following optimization problem:

$$\underset{x\prime}{argmax} L(x\prime, y), s.t. \|x\prime - x\|_p \leq \epsilon \quad (1)$$

where $L(\cdot, \cdot)$ is a loss function, ϵ is the maximum perturbation and $\|\cdot\|_p$ is the l_p norm distance. In this paper, we use l_∞-norm to measure the distance between x and $x\prime$.

2.1 Adversarial Attacks

Roughly speaking, existing attacks fall into three categories: optimization, smoothing, and attention modification. Since this paper follows the last direction, we briefly summarize the first two in this subsection and comprehensively review attention modification approaches afterward.

Optimization The optimization approach utilizes the gradients of a surrogate model to optimize a standard objective function, such as maximizing an entropy loss or minimizing the logit output. Goodfellow et al. propose a one-step fast gradient sign method (FGSM) [1], which is subsequently extended to an iterative version, denoted as I-FGSM [9]. Dong et al. [10] and Lin et al. [17] incorporate a momentum term or a Nesterov momentum term into the I-FGSM, denoted as MI-FGSM or NI-FGSM, to encourage a stabilized optimization direction.

Smoothing The smoothing approach aims to avoid over-fitting the decision surface of the surrogate model. It uses smoothed gradients derived from multiple points of the decision surface or resorts to estimate the gradient from a smoother surface. Xie et al. [6] introduce a random transformation to the input, whereas Dong et al. [7] shift the input images to obtain an ensemble of AEs that can improve the performance. Wang et al. [8] utilize the gradient information obtained at the last iteration to correct the current gradient. Lin et al. [17] leverage the scale-invariant property of DNNs and average the gradients for different scaled images to update AEs. Wang et al. [19] connect AEs with flat maxima, enhancing the transferability by constraining the gradient norm within the neighborhood range. Spectrum Simulation Attack (SSA) [20] performs the model augmentation in the frequency domain to enrich the diversity of substitute models. Ge et al. [37] use style transfer networks to generate diverse images from various fields to enhance the transferability of AEs. However, finetuning the style transfer network requires access to multiple surrogate models and consumes significant time.

2.2 Attention Modification-Based Attack

The attention modification approach presumes that different DNNs classify the same image based on similar features. Consequently, AEs generated by altering the features in benign images are anticipated to be more transferable. Jacobian-based Saliency Map Attack (JSMA) [11] employs the Jacobian matrix to compute its attention map, but its objective function remains ambiguous. Attack on Attention (AOA) [5] utilizes Softmax Gradient Layer-wise Relevance Propagation (SGLRP) [15] to compute the attention map. Attention-guided Transfer Attack (ATA) [12] derives its attention map from the gradients of an objective function concerning neuron outputs. Both AOA and ATA aim to maximize the difference between the attention maps of AEs and benign samples. Transferable Adversarial Perturbations (TAP) [13] maximizes the difference of the feature maps for all layers between benign samples and AEs, whereas Intermediate Level Attack (ILA) [14] finetunes AEs by enlarging the similarity of the feature difference at a given layer. Random Patch Attack (RPA) [39] leverages random patch transformations to aggregate gradient information from feature layers, preserving important object-related regions to guide the successive AE generation.

Another critical aspect of the attention modification attack is how to calculate an accurate attention map for a given image under a given model. As a representative scheme, Integrated Gradients (IG) [16] attributes the DNN prediction by calculating a straight-line path integral of the gradients from a reference image r to x

$$IG_i(f, x, r) = (x_i - r_i) \times \int_{\eta=0}^{1} \frac{\partial f(r+\eta\times(x-r))}{dx_i} d\eta, \tag{2}$$

where *i* denotes the entrance of the image. Huang et al. [4] propose two versions of attack (TAIG-S and TAIG-R) based on IG. TAIG-S uses the original IG to generate AEs:

$$x_{n+1}^{adv} = clip_x^{\epsilon}\big(x_n^{adv} - \alpha \times sign\big(IG\big(x_n^{adv}, y\big)\big)\big), \tag{3}$$

where α is the fixed step size. Due to the sensitivity of IG to noise, TAIG-R implements a random piecewise linear path to mitigate the noise influence. Specifically, a uniformly distributed noise v with support ($-\tau$, τ) is added to all points along the straight-line path of IG, that is, $x_e = r + \frac{e}{E}(x - r) + v$, $e \in (0,1, \ldots, E)$, where E represents the number of turning points.

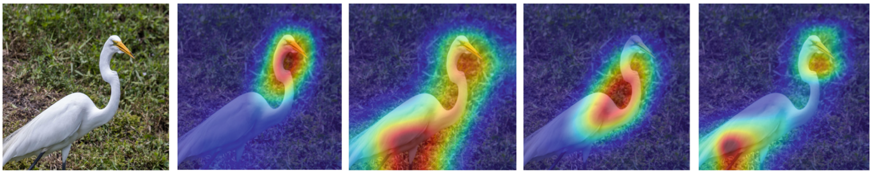


Fig. 2. Attention maps [21] calculated from different models. Starting from the second one on the left: VGG16 [24], InceptionV3 [25], ResNet152 [26], DenseNet121 [27].

3 The Proposed Method

3.1 Motivation

Dong et al. [7] point out that different models focus on different discriminative regions during recognition. As shown in Fig. 2, although all models emphasize the bird region more than the background, the regions with the highest attention vary by models. For example, the VGG16 focuses on the head of the bird only, whereas the DenseNet121 and the ResNet152 also highlight the legs. In this paper, we categorize the image into three types of regions based on its attention map: critical region, relevant but non-critical region, and irrelevant region. The critical region represents the most concerned area for classification (e.g., the bird's head). Relevant but non-critical regions represent areas relevant to decisions, yet the surrogate model pays little attention to (e.g., the feathers and legs of the bird in the attention map of VGG16). Irrelevant regions represent regions irrelevant to the decision (e.g., the grass). Existing attention modification attacks amend the features of salient attention on the surrogate model (e.g., TAIG, AOA). However, the computed attention maps tend to overemphasize the critical region, which undermines the transferability of AEs. An intuitive idea of avoiding overfitting the critical regions of a single surrogate model is to aggregate gradient information from multiple models [40]. The ensemble attack eliminates the differences between the attention maps of different models, i.e., the attention maps highlight the head, feathers, and other parts of the bird, thereby enhancing transferability. However, in practice, accessing multiple models is often challenging and costly. Therefore, we focus on the transferability under the single-model scenario here.

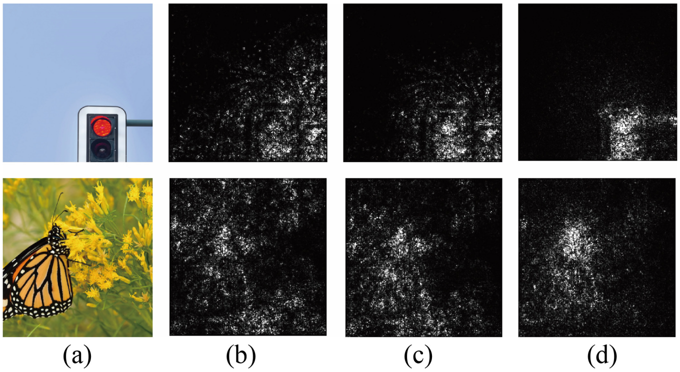


Fig. 3. Gradient saliency maps (GSM). (a) Raw input. (b) the GSM of the raw input. (c) the GSM after RPM. (d) the GSM after RPM and ARN.

3.2 Enhanced Image Transformation

To avoid the attacks overemphasizing the critical region, we propose an enhanced image transformation consisting of three steps: random masking, noise addition, and scaling.

Random Patch Masking (RPM) Since we want to highlight relevant but not-critical regions, an intuitive solution is masking the most discriminative area, forcing the surrogate model to look for more object-related regions in calculating the attention map, such as Fig. 1(c, d). However, the boundary of the most critical region in practice may not be as clear as in Fig. 1(b), making it hard to locate and mask with a single patch. Additionally, masking only the most discriminative area may not reveal enough object-related regions. Hence, we opt for a random patch masking strategy to highlight relevant but non-critical regions in attention map calculation. This idea is partially inspired by [22], in which the authors randomly masked some patches of the training images to improve the model's generalizability. Specifically, given a clean image x of size $H \times W \times 3$, we divide it into patches of size $s \times s \times 3$, where $s \in S = \{s_1, s_2, \dots s_n\}$ and S is the candidate size set. Then, we substitute each patch with a uniformly distributed noise patch according to a constant probability p_1.

There are two merits of the adopted RPM. First, it helps reveal more relevant but non-critical regions as shown in Fig. 1(d). These object-related features are less important for the surrogate model but may be important for the victim models in making decisions. Second, aggregating the attention maps for the masked images can neutralize model-specific elements (e.g., the noise in the irrelevant regions), as shown in Fig. 3(b, c).

Add Random Noise (ARN) To further suppress model-specific features, we draw inspiration from interpretability research [23] and inject random noise $\gamma \sim U\left[-p_2, p_2\right]$ into the input image to calculate the average gradient, where $U[,]$ is the uniform distribution, and p_2 represents the amplitude of the noise addition. In this manner, the calculated gradient can better represent the importance of image pixels, and regions related to decision-making will be more pronounced and clustered, as shown in Fig. 3(d).

The key to generating highly transferable AEs is to identify which features are easier to transfer. A DNN model learns features related to the object and retains model-specific features [38]. Different models result in different model-specific features and share similar object-aware features. Thus, distorting the object-aware features of relevant regions instead of noise on irrelevant regions will effectively improve adversarial transferability. Unfortunately, model-specific noise is often mixed with object-aware features in gradient saliency maps (Fig. 3(b)). Directly destroying these features is likely to lead to a model-specific local optimal solution [18]. Introducing RPM and ARN will effectively suppress model-specific features, whereas transferable object-aware features will be highlighted.

Scale Transformation (ST) Due to the generation of perturbations relies on the sign of the gradient, and the reference image r is usually a black image, we can ignore $(x_i - r_i)$ and approximate Eq. 2 as

$$\int_{\eta=0}^{1} \frac{\partial f(r+\eta\times(x-r))}{dx_i} d\eta \approx \frac{1}{N}\Sigma_{i=1}^{N} \frac{\partial f\left(\frac{\mathrm{i}}{N}\times x\right)}{dx_i} \tag{4}$$

to address the gradient saturation issue [4, 16] and provide more precise gradient information. Equation (4) can be efficiently implemented by calculating the gradient of the scale-transformed inputs $\frac{i}{N} \times x$.

Algorithm 1 Enhanced Image Transformation Attack

Input: The original clean image x and its true label y, a classification model f, the set of candidate sizes S, masking probability p_1, amplitude of noise addition p_2, total number of copies N, maximum perturbation ϵ, iteration number T.

Output: The adversarial example x^{adv}.

```
$\alpha = \epsilon/T;\ g_0 = 0, x_0^{adv} = 0$
for $t = 0 \rightarrow T - 1$ do
    Set $\bar{g} = 0$;
    for $i = 0,1,\ldots,N - 1$ do
        $x_t^i$ = RPM $(x_t^{adv})$
        $x_t^i = x_t^i + \gamma$
        $x_t^i = \frac{i}{N}(x_t^i)$
        Calculate the gradient for the current copy:
        $g' = \nabla_{x_t^i} L\left(x_t^i, y\right)$
        Accumulate gradients and average:
        $\bar{g} = \bar{g} + \frac{g'}{N}$
    end for
    Update $g_{t+1} = u \cdot g_t + \frac{\bar{g}}{\|\bar{g}\|_1}$;
    Update $x_{t+1}^{adv}$ by applying the gradient sign:
        $x_{t+1}^{adv} = Clip\{x_t^{adv} - \alpha * sign(g_{t+1})\}$
end for
return $x^{adv} = x_T^{adv}$
```

3.3 The Attack Algorithm

The general framework is shown in Fig. 4. With the enhanced image transformation presented in Sect. 3.2, an image is transformed N times, and the average gradient is calculated, denoted as the importance gradient. Guided by the importance gradient, the generated perturbations undermine the intrinsic features of these objects that can be transferred across models, making the AEs exhibit stronger transferability. We summarize the proposed algorithm in **Algorithm 1.**

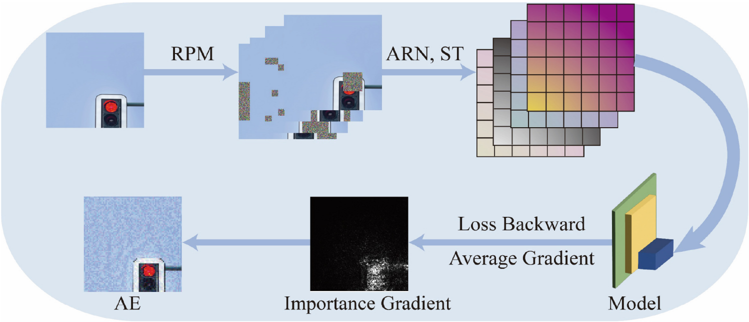


Fig. 4. General framework of the EIT.

4 Experiments

4.1 Experiment Setup

Datasets We conduct experiments on the ImageNet-compatible dataset, which contains 1000 images used for the NIPS 2017 adversarial competition.

Models We craft AEs against four surrogate models: Inc-v3 [25], Inc-v4, IncRes-v2 [28], and Res101. Due to page limitation, only the results of Inc-v4 and IncRes-v2 are reported in this paper, while the remaining are provided in the supplementary material: *supp.pdf*. For victim models, we use six normally trained models (Inc-v3, Inc-v4, IncRes-v2, Res50, Res101, and Res152 [26]), four defense models [29] (Inc-v3$_{\text{adv}}$, Inc-v3$_{\text{ens3}}$, Inc-v3$_{\text{ens4}}$, and IncRes-v2$_{\text{ens}}$), and three state-of-the-art vision transformers (ViTs) (PiT-S [30], CaiT-S-24 (CaiT-S) [31], DeiT-B [32]). We also evaluate different attacks on four defense methods, including JPEG [33], Bit-Red [34], FD [35], and NRP [36]. Four defense methods are combined with Inc-v3$_{\text{ens3}}$. We further evaluate the attacks on a real-world model: Baidu AI Cloud (https://cloud.baidu.com).

Competitors We compare the proposed EIT to diverse state-of-the-art attacks, including VT-MI [8], TAIG-S [4], TAIG-R [4], RPA [39], SSA [20], PGN [19], and STM [37]. To be fair, we add momentum terms to TAIG-S and TAIG-R. We also include combined versions of DI [6] and TI [7], denoted as DT, such as DT-TAIG-R, DT-SSA, DT-EIT, etc.

Parameter Settings In all experiments, the maximum perturbation $\epsilon = 16$, the iteration $T = 10$, and the step size $\alpha = 1.6$. For DI, we set the transformation probability $p = 0.5$. For TI, we set the kernel size $k = 7$. For VT-MI, we let the sample quantity be 20 and the sample range factor β be 1.5. For TAIG-S and TAIG-R, the number of turning points E is set to 20, and τ is set equal to ϵ. For RPA, the ensemble number N is

60, the modification probability $p_m = 0.3, 0.2$, and 0.2 when attacking normally trained models, defense models, and vision transformers. For SSA, we set the tuning factor $p = 0.5$, and the number of spectrum transformations $N = 20$. For PGN, the upper bound of neighborhood $\zeta = 3.0$, the balanced coefficient $\delta = 0.5$. For STM, we set the mixing ratio $\gamma = 0.5$, the noise upper bound $\beta = 2.0$, and the number of style transfer images $N = 20$. For the proposed EIT, we let the masking probability $p_1 = 0.1$, the set of alternative sizes $S = \{0, 20, 40, 60, 80\}$. For the amplitude of the noise addition, we set $p_2 = 0.2, 0.4$, and 0.4 when attacking the normally trained model, the defense model, and vision transformers, respectively. The number of image copies $N = 20$.

Table 1. Attack success rate (%) on six normally trained models. "*" indicates white-box attacks. The result in **bold** is the best.

	Attack	Inc-v3	Inc-v4	IncRes-v2	Res152	Res50	Res101	AVG
Inc-v4	VT-MI	78.8	99.8*	71.0	63.3	65.8	64.6	73.9
	TAIG-S	84.5	**99.9***	77.1	71.9	75.6	72.9	80.3
	TAIG-R	88.2	97.8*	84.1	80.8	82.6	79.7	85.5
	RPA	90.4	98.1*	85.7	79.2	82.7	80.1	86.0
	SSA	90.8	99.6*	86.6	81.1	83.7	82.5	87.4
	PGN	91.8	99.5*	88.1	83.8	84.5	83.9	88.6
	STM	93.9	99.0*	89.5	86.3	86.0	84.7	89.9
	EIT	**94.6**	99.7*	**91.7**	**89.0**	**88.9**	**88.4**	**92.1**
IncRes-v2	VT-MI	81.0	75.9	99.2*	66.5	69.3	68.8	76.8
	TAIG-S	87.0	82.6	98.8*	77.5	81.2	79.9	84.5
	TAIG-R	85.3	83.1	95.3*	80.3	82.0	81.5	84.6
	RPA	87.8	85.1	94.3*	80.6	83.6	82.1	85.6
	SSA	90.5	89.4	98.1*	84.6	85.5	83.9	88.7
	PGN	**93.9**	92.5	**99.8***	88.6	88.5	88.9	92.0
	STM	91.9	90.9	98.6*	88.3	87.5	87.7	90.8
	EIT	**93.9**	**93.2**	99.4*	**89.3**	**90.7**	**90.0**	**92.8**

4.2 Comparison of Transferability

This section compares the proposed EIT with the competitors against normally trained models, defense models, and vision transformers.

Attacking normally trained models As reported in Table 1, our approach significantly improves the transferability of AEs compared to state-of-the-art attacks. Specifically, AEs crafted by our proposed EIT are capable of getting a 92.5% success rate on average, outperforming VT-MI, TAIG-S, TAIG-R, RPA, SSA, PGN, and STM by 17.1%, 10.1%, 7.4%, 6.7%, 4.4%, 2.2%, 2.1%, respectively.

Attacking defense models Although many attacks can easily fool normally trained models, they may fail in attacking models with the defense mechanism. Table 2 reports the black-box attack success rates of the proposed EIT and competitors against defense models. Without the combination of DT, our proposed method trumps almost all competitors, resulting in an average success rate increase of 5.6%. Further, we combine our EIT and other baselines with DT, and our method continues to beat the other approach. For instance, considering the generated AEs on IncRes-v2, EIT's average attack success rate surpasses SSA and PGN by 4.2% and 1.8%. Although the attack performance of the STM is comparable to EIT, it is computationally much more expensive due to finetuning the style-transfer networks.

Table 2. Attack success rate (%) against four defense models.

	Attack	Inc-v3$_{adv}$	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	AVG
Inc-v4	VT-MI	40.7	40.8	41.1	27.2	37.5
	TAIG-S	47.7	48.7	46.1	32.1	43.7
	TAIG-R	66.8	67.4	65.3	52.9	63.1
	RPA	50.4	47.3	44.6	26.8	42.3
	SSA	66.8	65.0	60.8	42.4	58.8
	PGN	64.3	66.5	64.3	49.5	61.2
	STM	65.4	67.6	66.2	49.0	62.1
	EIT	**72.4**	**75.6**	**74.4**	**57.8**	**70.1**
	DT-TAIG-R	77.8	78.0	76.4	69.7	75.5
	DT-RPA	68.7	71.0	67.0	56.5	65.8
	DT-SSA	82.2	80.6	80.2	72.9	79.0
	DT-PGN	81.8	82.4	82.2	**75.8**	80.6
	DT-STM	**83.2**	83.0	82.8	74.5	80.9
	DT-EIT	**83.2**	**84.5**	**82.9**	75.5	**81.5**
IncRes-v2	VT-MI	46.2	48.4	44.7	38.3	44.4
	TAIG-S	59.1	60.1	52.8	44.2	54.1
	TAIG-R	71.3	70.6	66.0	62.3	67.6
	RPA	68.6	61.8	54.9	49.1	58.6
	SSA	73.1	73.3	68.1	61.1	68.9
	PGN	75.8	75.2	70.3	66.3	71.9
	STM	74.7	76.5	72.7	64.5	72.1
	EIT	**78.8**	**79.2**	**74.9**	**67.9**	**75.2**
	DT-TAIG-R	79.7	78.6	77.4	76.2	78.0
	DT-RPA	74.0	71.5	68.1	65.6	69.8
	DT-SSA	83.1	83.8	81.7	80.3	82.0
	DT-PGN	85.8	85.0	83.3	83.4	84.4
	DT-STM	**87.1**	87.2	**85.6**	**85.4**	**86.3**
	DT-EIT	**87.1**	**88.5**	85.2	84.0	86.2

Model ensemble [40] is widely adopted to improve the transferability against defense models. Table 3 shows that even under advanced defense methods, EIT can achieve an average attack success rate of 82.5% and exceeds PGN attack by more than 1.8%,

Table 3. Attack success rate (%) of AEs crafted on an ensemble of Inc-v3, Inc-v4, IncRes-v2, and Res-101.

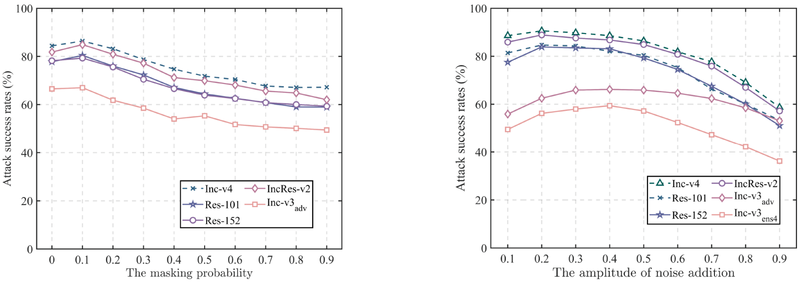
Model	Attack	Inc-v3$_{ens3}$	IncRes-v2$_{ens}$	Bit-Red	JPEG	FD	NRP	Baidu AI Cloud	AVG
Ens	TAIG-R	87.6	81.2	86.6	91.9	89.6	**28.3**	83.7	78.4
	SSA	88.9	81.6	86.8	93.0	90.4	26.8	82.5	78.6
	PGN	90.7	85.9	88.9	93.1	90.1	27.9	82.3	79.8
	STM	**94.3**	**88.4**	**92.3**	**96.9**	93.0	26.0	84.8	82.2
	EIT	93.0	**88.4**	91.7	96.2	**93.1**	27.8	**87.4**	**82.5**

Table 4. Attack success rate (%) of against vision transformers.

	Attack	PiT-S	DeiT-B	CaiT-S	AVG
Inc-v4	VT-MI	42.9	32.6	35.4	37.0
	TAIG-S	46.2	34.1	36.9	39.1
	TAIG-R	62.1	50.3	56.5	56.3
	RPA	52.0	37.4	43.6	44.3
	SSA	60.9	45.8	53.7	53.5
	PGN	64.3	50.6	57.5	57.5
	STM	66.6	50.8	62.2	59.9
	EIT	**68.5**	**54.1**	**63.8**	**62.1**
	DT-TAIG-R	63.7	53.7	61.4	59.6
	DT-RPA	58.1	46.0	57.8	54.0
	DT-SSA	68.5	56.9	66.8	64.1
	DT-PGN	**70.3**	**59.1**	68.4	65.9
	DT-STM	70.1	58.0	**70.2**	**66.1**
	DT-EIT	**70.3**	59.0	68.9	**66.1**
IncRes-v2	VT-MI	44.6	34.3	36.9	38.6
	TAIG-S	50.1	36.8	42.6	43.2
	TAIG-R	59.4	50.3	57.2	55.6
	RPA	58.4	44.6	51.1	51.4
	SSA	65.4	50.8	59.7	58.6
	PGN	67.8	53.4	62.2	61.1
	STM	**69.7**	**55.8**	**66.9**	**64.1**
	EIT	68.5	54.4	63.5	62.1
	DT-TAIG-R	60.7	52.9	59.4	57.7
	DT-RPA	60.5	49.1	59.4	56.3
	DT-SSA	69.7	60.2	70.3	66.7
	DT-PGN	72.2	63.0	71.6	68.9
	DT-STM	**73.1**	**63.1**	**75.7**	**70.6**
	DT-EIT	71.3	61.5	72.2	68.3

demonstrating our attack's effectiveness. It is worth noting that the EIT exceeds the STM attack by 2.8% when attacking real-world online models.

Attacking Vision Transformers Table 4 compares different attacks against ViTs. Previous works demonstrate that ViTs present better adversarial robustness than convolutional neural networks (CNNs), and transferring from CNNs to ViTs is even more arduous. With EIT, the black-box attack success rate has increased by 6.0%, and 2.8% on average compared to SSA and PGN. Furthermore, combining with DT can further improve the proposed EIT's transferability.



(a) (b)

Fig. 5. The impact of (a) the masking probability p_1, and (b) the amplitude of noise addition p_2.

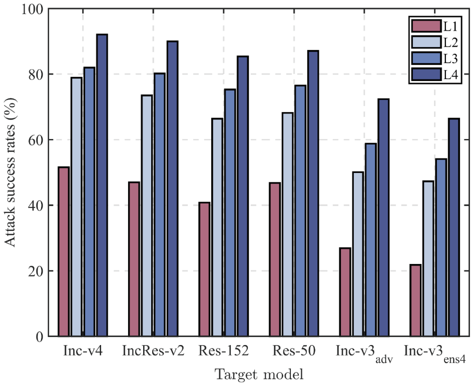


Fig. 6. The impact of each step of EIT on transferability.

4.3 Ablation Study

We conduct a series of experiments to study the impact of different parameters, including the masking probability p_1, the amplitude of noise addition p_2 and the number of image copies N, the set of alternative sizes S will be illustrated in the supplementary material. To simplify the analysis, we fix the surrogate model as the Inc-v3.

The masking probability p_1. Figure 5(a) shows the effect of the p_1 in the black-box setting. As p_1 increases, the transferability continues to increase and reaches a peak at

$p_1 = 0.1$. Excessive masking image patches may cause a large amount of information loss in the image, resulting in a significant decrease in transferability. Therefore, we set $p_1 = 0.1$.

The amplitude of noise addition p_2. Figure 5(b) studies the impact of the amplitude of noise addition p_2 on the transferability. As p_2 increases, the transferability peaks at $p_2 = 0.2$ for the normally trained models, while it peaks at $p_2 = 0.4$ for the defense model. Therefore, we set $p_2 = 0.2$ for normally trained models and $p_2 = 0.4$ for more robust models, e.g., defense and transformer-based models.

The impact of each step on transferability. In Fig. 6, we study each EIT step's impact on transferability. Specifically, we divided it into four variants: no additional steps (L1) (equivalent to the method proposed in [10]), only noise addition (L2), noise addition and scaling (L3), and the complete EIT method (L4). It can be seen that each step contributes to transferability.

5 Conclusion

We propose an enhanced image transformation method to improve the transferability of AEs. Specifically, we perform three operations for each image, namely random patch masking, noise addition, and scaling, to obtain a series of images and calculate the average gradient. When used to guide the AEs generation, the average gradient encourages the attack to pay more attention to relevant but non-critical regions and better neutralize model-specific information. Experimental results show that our method can achieve higher transferability than existing transfer-based attacks. We believe further improvements in attention-modified attacks could be obtained by fully utilizing more diverse attention maps, which is our future work.

Acknowledgements. This work was supported by the Opening Project of Guangdong Province Key Laboratory of Information Security Technology (no. 2023B1212060026) and the network emergency management research special topic (no. WLYJGL2023ZD003).

References

1. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (ICLR) (2015)
2. Madry, A., Makelov, A., Schmidt, L., et al.: Towards deep learning models resistant to adversarial attacks. arXiv:1706.06083 (2017)
3. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: IEEE Symposium on Security and Privacy (SP), pp. 39–57 (2017)
4. Huang, Y., Kong, A.W.: Transferable adversarial attack based on integrated gradients. In: International Conference on Learning Representations (ICLR) (2022)
5. Chen, S., He, Z., Sun, C., et al.: Universal adversarial attack on attention and the resulting dataset DAmageNet. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020)
6. Xie, C., Zhang, Z., Zhou, Y., et al.: Improving transferability of adversarial examples with input diversity. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2730–2739 (2019)

7. Dong, Y., Pang, T., Su, H., et al.: Evading defenses to transferable adversarial examples by translation-invariant attacks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4312–4321 (2019)
8. Wang, X., He, K.: Enhancing the transferability of adversarial attacks through variance tuning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1924–1933 (2021)
9. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. In: International Conference on Learning Representations (ICLR) (2017)
10. Dong, Y., Liao, F., Pang, T., et al.: Boosting adversarial attacks with momentum. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9185–9193 (2018)
11. Papernot, N., McDaniel, P., Jha, S., et al.: The limitations of deep learning in adversarial settings. In: IEEE European Symposium on Security and Privacy, pp. 372–387 (2016)
12. Wu, W., Su, Y., Chen, X., et al.: Boosting the transferability of adversarial samples via attention. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1161–1170 (2020)
13. Zhou, W., Hou, X., Chen, Y., et al.: Transferable Adversarial Perturbations. In: European Conference on Computer Vision (ECCV), pp. 471–486 (2018)
14. Huang, Q., Katsman, I., Gu, Z., et al.: Enhancing adversarial example transferability with an intermediate level attack. In: IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4732–4741 (2019)
15. Iwana, B.K., Kuroki, R., Uchida, S.: Explaining convolutional neural networks using softmax gradient layer-wise relevance propagation. arXiv:1908.04351 (2019)
16. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International Conference on Machine Learning (ICML), pp. 3319–3328 (2017)
17. Lin, J., Song, C., He, K., et al.: Nesterov accelerated gradient and scale invariance for adversarial attacks. In: International Conference on Learning Representations (ICLR) (2020)
18. Wang, Z., Guo, H., Zhang, Z., Liu, W., Qin, Z., Ren, K.: Feature importance-aware transferable adversarial attacks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7639–7648 (2021)
19. Ge, Z., Shang, F., Liu, H., Liu, Y., Wang, X.: Boosting adversarial transferability by achieving flat local maxima. arXiv:2306.05225 (2023)
20. Long, Y., Zhang, Q., Zeng, B., et al.: Frequency domain model augmentation for adversarial attack. In: European Conference on Computer Vision (ECCV), pp. 549–566 (2022)
21. Selvaraju, R.R., Cogswell, M., Das, A., et al.: Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: IEEE International Conference on Computer Vision (ICCV) (2017)
22. Singh, K.K., Lee, Y.J.: Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In: IEEE International Conference on Computer Vision (ICCV) (2017)
23. Smilkov, D., Thorat, N., Kim, B., et al.: SmoothGrad: removing noise by adding noise. arXiv:1706.03825 (2017)
24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR) (2015)
25. Szegedy, C., Vanhoucke, V., Ioffe, S., et al.: Rethinking the inception architecture for computer vision. In: IEEE International Conference on Computer Vision (ICCV) (2016)
26. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, (2016)
27. Huang, G., Liu, Z., Van Der Maaten, L.V., et al.: Densely connected convolutional networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700–4708 (2017)

28. Szegedy, C., Ioffe, S., Vanhoucke, V., et al.: Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: AAAI Conference on Artificial Intelligence (2017)
29. Tramèr, F., Kurakin, A., Papernot, N., et al.: Ensemble adversarial training: Attacks and defenses. In: International Conference on Learning Representations (ICLR) (2017)
30. Heo, B., Yun, S., Han, D., et al.: Rethinking spatial dimensions of vision transformers. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
31. Touvron, H., Cord, M., Sablayrolles, A., et al.: Going deeper with image transformers. arXiv: 2103.17239 (2021)
32. Touvron, H., Cord, M., Douze, M., et al.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning (ICML) (2021)
33. Guo, C., Rana, M., Cisse, M., et al.: Countering adversarial images using input transformations. In: International Conference on Learning Representations (ICLR) (2018)
34. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. In: Network and Distributed System Security Symposium (NDSS) (2018)
35. Liu, Z., Liu, Q., Liu, T., et al.: Feature distillation: DNN-oriented JPEG compression against adversarial examples. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 860–868 (2019)
36. Naseer, M., Khan, S., Hayat, M., et al.: A self-supervised approach for adversarial robustness. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 262–271 (2020)
37. Ge, Z., Shang, F., Liu, H., et al.: Improving the transferability of adversarial examples with arbitrary style transfer. In: Proceedings of the ACM International Conference on Multimedia (2023)
38. Ilyas, A., Santurkar, S., Engstrom, L., et al.: Adversarial examples are not bugs, they are features. In: Annual Conference on Neural Information Processing Systems (2019)
39. Zhang, Y., Tan, Y.A., Chen, T., et al.: Enhancing the transferability of adversarial examples with random patch. In: Proceedings of the 31th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1672–1678 (2022)
40. Liu, Y., Chen, X., Liu, C., et al.: Delving into transferable adversarial examples and black-box attacks. In: International Conference on Learning Representations (ICLR) (2017)

Splicing Localization in Digital Images Through Agglomerative Clustering on Optimized Feature Sets with Zero Training Data Dependency

Debjit Das(✉) and Ruchira Naskar

Department of Information Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, India
{debjit.rs2020,ruchira}@it.iiests.ac.in

Abstract. Most image tamper detection and localization schemes in the present day rely on huge volumes of training data to achieve perfect performance. To the best of our knowledge, the state-of-the-art schemes rely on thousands of training samples (ranging from 1K to over 84K) to localize forged image regions. In this work, we aim to come around the problem of reliance on huge volumes of training images in order to efficiently locate a tampered image region; in fact, we succeed to achieve zero training data dependency for image tamper localization. In this paper, we have shown our work on a specific class of image forgery, viz. *image splicing attack*. To state more specifically, in this paper, we propose a set of optimal image features, which are subsequently fed to a hierarchical agglomerative clustering module, thereby detecting and localizing spliced region(s) within an image. Our experiments prove that the proposed method achieves close to 90% accuracy while completely bypassing any training data requirements and solely relying on the unsupervised clustering concept.

Keywords: Feature Optimization · HAC · Splicing Localization · Unsupervised Clustering

1 Introduction

Due to the easy availability of several image modification software to the general public, image tampering has become a widespread problem. The altered photographs may be purposefully utilized for unlawful ends, such as to deceive unwary users or harm the social standing of a well-known individual or group. National security problems might result from criminals or terrorists using fake identification documents to commit damaging acts. Therefore, analysing, identifying, and detecting digital image forgeries is essential for any company or country's security. Image splicing [1,14] forgery is done by merging fragments of many source images to create a single, fabricated composite image that seems natural.

Researchers are primarily investigating various feature sets and classifiers in the current state-of-the-art, representing the identification of image splicing as a digital forensics classification task. While some work [1,5,13] only focuses on identifying a

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 360–373, 2025.
https://doi.org/10.1007/978-3-031-78312-8_24

spliced image, others focus on localizing the spliced region(s) within it [10, 21]. Among the feature engineering and machine learning-based works, Prasanna et al. [13], and Shen et al. [15] only detected image splicing without any optimization of feature sets and localization. The works in [5, 19] optimized the feature sets, but localization was not performed. Walia et al. [18] and Zhu et al. [22] localized the spliced region but without feature set optimization. All of these works needed vast training datasets as well.

Deep learning-based techniques usually provide better results, but they require extensive data resources for training and superior computing architecture. Among the recent deep learning-based approaches, in [9], the authors used local and global features in a deep neural network. For image splicing localization, Zeng et al. [21] suggested a dual path-way deep neural network with multiscale fusion. In [16], the authors proposed a MobileNetV2-based localization method, while Kadam et al. [7] suggested a splicing localization scheme implemented with MobileNetV1. In [10], Peng et al. proposed CAU-Net to localize the spliced area. All these works localize spliced regions in an image, but require very large training datasets and extensive computing. A comparative analysis of the related works with our proposed scheme is given in Table 1.

Table 1. A Comparative Analysis

Methods	**Independent of training set?**	**Optimized feature set?**	**Independent of deep learning?**	**Localization of spliced region(s)?**
Prasanna et al. [13]	×	×	✓	×
Shen et al. [15]	×	×	✓	×
Walia et al. [18]	×	×	✓	✓
Zhu et al. [22]	×	×	✓	✓
Jaiprakash et al. [5]	×	✓	✓	×
Li et al. [8]	×	✓	✓	×
Wang et al. [19]	×	✓	✓	×
Peng et al. [10]	×	×	×	✓
Peng et al. [11]	×	×	×	✓
Shi et al. [16]	×	×	×	✓
Zeng et al. [21]	×	×	×	✓
Proposed	✓	✓	✓	✓

1.1 Our Contribution

Most of the recent deep learning-based approaches require superior computing and very large training datasets. Some feature extraction-based works can localize the spliced area, but they also suffer from high dimensionality and complexity. Almost in every mechanism, the data requirement is very high for training the model, which is not available in practice and also increases the complexity and execution time of the model. Our proposed scheme, suggested in this work, attempts to address these shortcomings. The following are the major contributions of our proposed method:

- Our proposed feature engineering and unsupervised clustering-based scheme can successfully detect and locate spliced regions within a forged test image with comparatively better performance than existing schemes.
- We have established an image splicing detection and localization scheme that does not need any additional dataset like it is needed in supervised classification to train the model.
- The proposed scheme consists of a highly optimized feature set fed to the Hierarchical Agglomerative Clustering (HAC) for splicing localization from a single test image.
- Our method is independent of deep learning requirements of superior computing and extensive training data.

The rest of the document is arranged as follows: we demonstrate and discuss our proposed approach in Section 2 along with a quick summary of the feature sets we looked into. We provide and discuss the findings of the experiments we conducted in Section 3. Finally, this paper concludes with a brief overview of the potential continuation of the work in Section 4.

2 Proposed Model

Our work seeks to identify image splicing from a single test image and localize the spliced region without requiring any training dataset. In this approach, we work with two features: Histogram of Oriented Gradients (HOG) and Gray Level Co-occurrence Matrix (GLCM). The input image must first be preprocessed, and then feature extraction is performed. We have focused on feature optimization to reduce the dimension of the feature set. We have implemented an unsupervised clustering technique - HAC, for clustering-based localization of the image blocks, and thus, there is no need for a training dataset. If spliced blocks are found in the test image, it will be marked as a spliced image, and the associated spliced blocks will be localized. The overall operational flow of the proposed method is provided in the form of an overview in Section 2.1.

2.1 Overview of the Proposed Methodology

To provide a comprehensive understanding of our complete workflow, we present below a brief overview of the operational steps followed by the proposed method. Following this, Section 2.2 onwards, we discuss individual components of the proposed model in detail.

- The test image is converted into grayscale and then segmented into non-overlapping blocks of 32 × 32. The original test image without grayscale conversion is also segmented into non-overlapping blocks similarly. Each of these image blocks is considered as an image sample.
- From each color block, a set of HOG features are extracted, and correspondingly, from each grayscale block, a set of GLCM features are extracted.
- The combined feature set is formed for each image block of the test image, and then correlated features are removed.

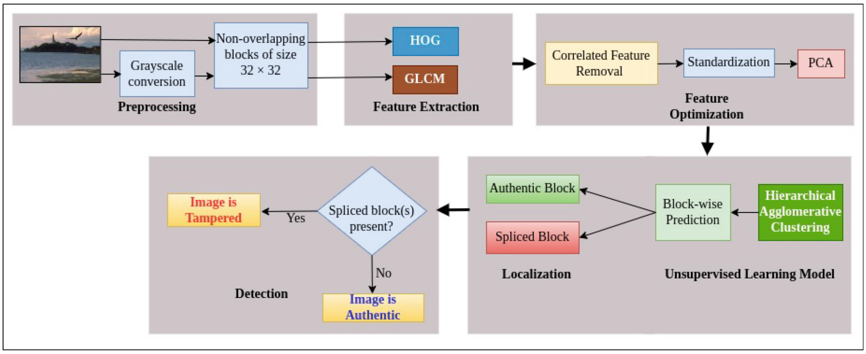


Fig. 1. Proposed Scheme: Operational Flow

- The feature set is standardized and optimised further using Principal Component Analysis (PCA) based on Cumulative Explained Variance (CEV) to improve the efficiency.
- Each image block with the optimized feature set is then fed as a singleton cluster to the HAC, and clustering is performed sequentially. The final two clusters represent the predicted authentic and spliced blocks by our model.
- If all the image blocks are mapped to form only one cluster in the threshold zone of HAC, then the test image will be marked as a completely authentic image; otherwise, it will be marked as a spliced image with localizing the spliced regions.

In Fig. 1, we provide an operational flow diagram to represent the complete workflow of the proposed model. Individual components are elaborated next.

2.2 Feature Sets Explored

In our scheme, we adopt a feature extraction-based unsupervised clustering approach. This section briefly overviews the two image feature sets extracted in this work: HOG and GLCM.

Histogram of Oriented Gradients (HOG) HOG [3,6] is frequently used as an object detection tool and as a feature descriptor of images in Computer Vision. It is advantageous as it describes the appearance and shape of the forged object by describing the intensity gradient distribution. The following formulas can be used to find the angle's magnitude and orientation for each pixel if the gradients along the axes are denoted by G_x and G_y:

$$Magnitude = \sqrt{{G_x}^2 + {G_y}^2} \tag{1}$$

$$Orientation(\theta) = tan^{-1}\frac{G_y}{G_x} \tag{2}$$

Each small region produces a different histogram, and the HOG feature of the test image is created by normalizing and concatenating all of the histograms.

Gray Level Co-occurrence Matrix (GLCM) For the study of image texture, the GLCM [17,18] has been widely employed. It is produced by calculating the frequency with which different pixel value pairs appear together in a grayscale picture. The two most important GLCM factors are direction and distance.

This research has extracted five distinct texture features-contrast, correlation, dissimilarity, energy, and homogeneity-from each GLCM. Appendix-A contains detailed definitions for these textural features.

2.3 Proposed Model for Localisation of Spliced Region Using Optimized Feature Set and Unsupervised Clustering

Here, we offer our suggested method for locating spliced areas from the input image using the feature sets mentioned above.

Preprocessing We convert a test image into grayscale, before GLCM feature extraction, by the following:

$$I = 0.299 \times R + 0.587 \times G + 0.114 \times B \tag{3}$$

where I is the grayscale intensity of the gray image, and R, G, and B denote the color components of the RGB image, respectively. After converting into grayscale, The test image is broken into multiple 32×32 equal-sized non-overlapping blocks. For HOG feature extraction, the grayscale conversion has not been applied. Our experiment treated each small image block as an individual sample for feature extraction.

Feature Extraction After the preprocessing, we perform feature extraction from each block. While extracting the HOG features, we considered the RGB image blocks, and from each block, a total of 672 HOG features are extracted.

Next, we determine each block's GLCM at three distances and four angles. As a result, a total of $3 \times 4 = 12$ unique GLCMs are calculated for each image block. We have retrieved five unique GLCM texture features from each of these, as detailed in Appendix - A. Consequently, a total of 3 × 4 × 5 = 60 GLCM texture features have been extracted from each input image block.

Feature Optimization The combined feature vector set of dimension 732 has been taken for optimization. We first eliminate the correlated features with 90% correlation or above to remove redundant data. Next, we perform standardization and implement Principal Component Analysis (PCA) [20], an unsupervised dimension reduction technique, to further optimise the feature set and represent it using HAC with much lower dimension. The number of components to be selected in PCA has been decided based on the Cumulative Explained Variance (CEV). If the initial feature set has d dimension, then the CEV of the first m principal components (m_{pc}) can be calculated as:

$$CEV_{m_{pc}} = \frac{\sum_{j=1}^{m_{pc}} \lambda_i}{\sum_{j=1}^{d} \lambda_i} \tag{4}$$

where λ_i represents the eigenvalue of i-th eigenvector. We have empirically selected the number of components in PCA that exceeds the cumulative explained variance threshold of 0.90 to preserve at least 90% of the total data's variance while optimizing the feature set further.

Algorithm 1: Block clustering for localization of spliced regions

```
Input: No. of clusters $N_c$ as total no. of image blocks
Output: Final two clusters (one representing a group of spliced blocks and the other
  representing a group of authentic blocks)
With an optimized feature set, each image block is represented
Initialize $N_c \leftarrow$ total number of blocks;
Initially, each image block forms a singleton cluster, i.e., total $N_c$ clusters, viz., $C_1$, $C_2$
  ..., $C_{N_c}$
for $i \leftarrow N_c$ to 2 do
    /* Each iteration used to merge the cluster pair
       demonstrating minimum inter-cluster distance */
    Initialize $d_{min} \leftarrow 9999$ ; /* $d_{min}$ stores the minimum of distances
      between all cluster pairs */
    foreach pair of clusters ($C_i$, $C_j$) do
        Compute $d_{i,j}$ ; /* Euclidean distance between clusters $C_i$
          and $C_j$ */
        if $d_{i,j} < d_{min}$ then
            $d_{min} \leftarrow d_{i,j}$;
        end
    end
    if $d_{min} = d_{x,y}$ then
        $C_x \leftarrow merge(C_x, C_y)$ ; /* $C_x$ and $C_y$ merged, resultant stored
          in $C_x$ */
        Discard $C_y$;
    end
    $i \leftarrow i - 1$;
end
```

Unsupervised Learning Model using HAC for Splicing Localization We employ Hierarchical Agglomerative Clustering (HAC) on the optimized feature set, which is a connectivity-based model that clusters data points close to one another according to a distance or similarity metric (*closeness of data points*). HAC follows a bottom-up approach to form the overall clustering structure. In our work, we have adopted *Euclidean Distance* to measure the closeness of data points, which are nothing but our test image blocks.

Each smallest image block is fed as an individual sample to the HAC algorithm, and clustering is performed to distinguish the authentic blocks from the spliced blocks. Each smallest image block is initially treated as a singleton cluster at the outset, and then pairs of clusters are agglomerated until all the clusters have been merged into two. The final two clusters represent the authentic and spliced image blocks, respectively. Here, the total number of clusters is set as two as it is designed as a binary classification. Each block is predicted as either an authentic or a spliced block, and finally, after complete clustering, the spliced region is localized. The block clustering-based algorithm for localizing the spliced regions is represented in Algorithm 1.

Splicing Detection and Localization If the number of clusters in the test image is predicted as two, having one authentic and one spliced cluster, then the test image is marked as a spliced image. If all the image blocks are mapped to form only one cluster, then the test image will be marked as an authentic image with no spliced region. The splicing localization output of our model has been visually presented in Fig. 2.

The predicted result of each image block is compared with the corresponding ground truth image block to evaluate our model's performance for splicing image localization. Our experimental results are presented next.

3 Experimental Results

3.1 Dataset and Implementation

In our research, we have explored the CASIA V1.0 [4] and the CASIA V2.0 [4, 12], two color image forgery datasets. The CASIA V2.0 comprises 3274 copy-move, 1849 spliced, and 7491 authentic color images. The CASIA V1.0 is made of 1721 images, out of which 800 are pristine and the remaining 921 are tampered with. We have selected the spliced images from the 1849 spliced image samples of CASIA V2.0 and from the 921 tampered images of CASIA V1.0 as our input test image.

With the aid of Python 3.7.6, the Jupyter Notebook IDE, and the scikit-learn library, we implemented our suggested model into practice. The workstation used for this experiment has a 4th generation Intel i3-4005U CPU running at 1.70 GHz processor base frequency.

3.2 Performance Evaluation Metrics

We have taken only spliced images as the input image in each set of experiments. The performance is evaluated by comparing the predicted result of each block with the corresponding ground truth image block, using the generated confusion matrix having four values: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), for each test image. Different performance metrics such as accuracy, precision, recall, F1-Score, and Matthews Correlation Coefficient (MCC) are computed from the confusion matrix. An assessment of the model's overall correctness is provided by accuracy as follows:

$$Accuracy\ (\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{5}$$

Model Precision and Recall is computed as follows:

$$Precision = \frac{TP}{TP + FP}, \;\; Recall = \frac{TP}{TP + FN} \tag{6}$$

F1-Score and MCC When the input samples are not properly balanced, i.e. if class imbalance is present among the input samples, then F1-Score and MCC can be used as more accurate performance evaluation metrics. The formulation of the F1-Score is as follows:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

The MCC [2] score effectively overcomes the unbalanced dataset problem and ranges between -1 and +1. It is calculated as:

$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{8}$$

3.3 Testing Protocol

We run the proposed clustering (Algorithm 1) on each test image, which aims to identify each smallest image block as either *spliced* or *authentic*. Ideally, all spliced blocks constitute $cluster_{Spl}$, and all authentic blocks merge to form $cluster_{Auth}$. In our experiments, while validating the correctness of the proposed clustering method, we perform the validation by comparing the clustering output on a certain image block by the proposed method (whether it belongs to $cluster_{Auth}$ or $cluster_{Spl}$) against the information present in the ground truth image, corresponding to that respective image block. Each image block is fed to HAC initially as a singleton cluster, which gradually forms the final two clusters $cluster_{Auth}$ and $cluster_{Spl}$.

3.4 Experimental Findings and Analysis

This section includes a detailed presentation of our experimental results and a performance analysis of the suggested image-splicing localization methodology. We conducted our experiments on the selected spliced images and documented the performance of our proposed model accordingly. Fig. 2 presents the visualization of the localization results of our proposed scheme, where the first row represents the spliced test images, the second row represents the corresponding ground truth images, and the third row shows the detected spliced blocks of each image by our method.

A dendrogram that shows the hierarchical relationship between the clusters and is formed based on the Euclidean distance between the clusters while initially taking each block as a separate cluster is represented in Fig. 3. The threshold selection zone for finally having two clusters is shown in red. The clustering of spliced image blocks over all iterations at different threshold distances until two clusters are formed for individual test images is represented in Table 2. Fig. 4 displays the visual representation of cluster formation at different distance thresholds for our experiments.

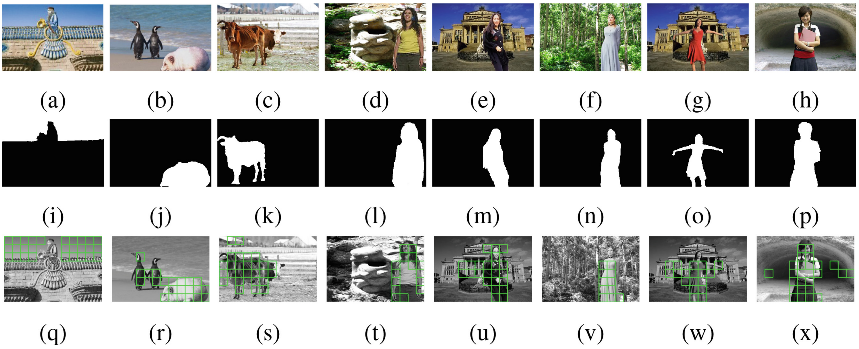


Fig. 2. Visual representation of localization results. (a)-(h) Spliced test images. (i)-(p) Ground truth of the spliced test images. (q)-(x) Spliced region localization of the test images.

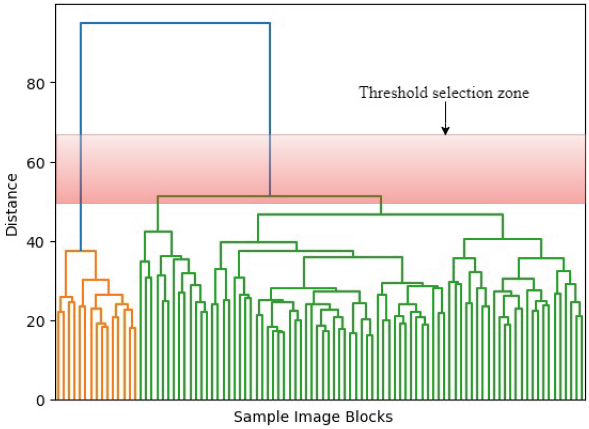


Fig. 3. Evolution of clusters starting from initial iteration, where each image block represents one cluster, upto the final iteration, where all blocks are grouped into two clusters: one representing spliced image blocks (orange), and the other representing authentic image blocks (green). The figure shows the gradual convergence of all blocks into two clusters through subsequent cluster merging based on inter-cluster distances. Two clusters formed within threshold inter-cluster distance in the range [48,65].

An evaluation of our proposed model's effectiveness in comparison to other recent cutting-edge techniques for image splicing localization is given in Table 3, where the performance metrics for comparison are taken as F1-Score, MCC, dependency on deep learning and number of training samples. We have also considered the corresponding operating principle and the dataset used for the respective scheme. The F1-Score and MCC serve as two appropriate performance indicators, as in most test situations, image datasets or image blocks are not adequately balanced. The results of Table 3 prove

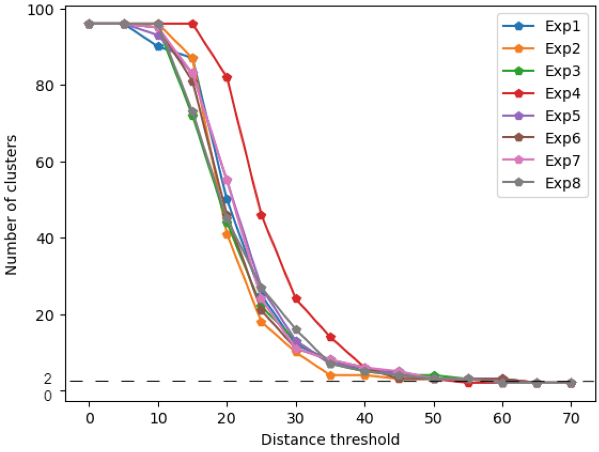


Fig. 4. Cluster formation at different distance thresholds.

Table 2. Performance of the proposed model for spliced image block clustering over $0 - 94$ iterations (in steps of 8) in terms of distance threshold (T) vs. number of clusters formed (N_c)

Test	It. 0 (Initial)		It. 8		It. 16		It. 24		It. 32		It. 40		It. 48		It. 56		It. 64		It. 72		It. 80		It. 88		It. 94	
	T	N_c	T	N_c	T	N_c	T	N_c	T	N_c	T	N_c	T	N_c	T	N_c	T	N_c	T	N_c	T	N_c	T	N_c	T	N_c
1	-	96	13.78	88	15.76	80	16.99	72	18	64	19.01	56	20.25	48	21.66	40	23.66	32	25.04	24	28.18	16	32.97	8	63.92	2
2	-	96	14.63	88	15.69	80	16.37	72	17.42	64	18.20	56	19.39	48	20.10	40	21.50	32	23.37	24	25.90	16	30.70	8	64.25	2
3	-	96	12	88	13.86	80	14.95	72	16.73	64	17.90	56	18.74	48	20.63	40	22.41	32	24.11	24	27.29	16	34.45	8	57.59	2
4	-	96	18.21	88	20.04	80	21.32	72	22.71	64	24.06	56	24.58	48	26.42	40	27.98	32	29.71	24	32.30	16	37.41	8	51.19	2
5	-	96	12.51	88	15.79	80	17.68	72	18.82	64	19.87	56	21.32	48	22.12	40	23.32	32	25.41	24	28.09	16	34.54	8	60.51	2
6	-	96	13.79	88	15.22	80	15.83	72	16.90	64	18.09	56	18.71	48	20.80	40	22.30	32	23.68	24	26.75	16	33.45	8	61.01	2
7	-	96	12.31	88	16.22	80	17.47	72	18.79	64	19.93	56	20.62	48	21.98	40	22.90	32	24.44	24	27.68	16	34.99	8	55.03	2
8	-	96	12.69	88	13.74	80	15.09	72	16.35	64	17.35	56	18.63	48	20.60	40	22.30	32	26.07	24	29.67	16	34.65	8	58.98	2

the superiority of the proposed model as compared to the majority of the state-of-the-art (except the scheme of Shi et al. [16] which is closely followed by our method, when tested on CASIA V1.0), despite our model being independent of training samples requirement as well as overcoming the computational intensity inherent in deep learning models. While all other compared works require extensive training samples, our method does not need any training data set, as it can directly locate the spliced regions from a single test image. Additionally, our method minimizes the complexity of the modelling structure as it is based on unsupervised clustering, so it does not need any training dataset and is designed with an optimized feature set.

To understand the performance of our model on individual test subsets, we have conducted multiple different experiments on varied subsets of the test dataset.

Table 3. Performance Analysis and Comparison

Author, Year	**Operating Principle**	**Dataset**	**No. of Training samples**	**F-1 Score**	**MCC**	**Extensive computing needs for deep learning**
Kadam et al. [7] (2021)	Mask R-CNN, MobileNetV1	CASIA V1.0	2505	0.64	-	✓
Li et al. [9] (2022)	Multi-scale guided learning	CASIA V1.0	5123	0.64	0.59	✓
Peng et al. [10] (2023)	CAU-Net	CASIA V2.0	4488	0.58	-	✓
Shi et al. [16] (2023)	MobileNetV2, SRM	CASIA V1.0	4780	0.75	0.73	✓
Peng et al. [11] (2023)	GP-Net, TcFusion	CASIA V2.0	5078	0.61	-	✓
Zeng et al. [21] (2024)	Dual-path model, multiscale fusion	CASIA V1.0	84288	0.65	-	✓
Proposed method	Feature extraction, optimization, HAC[†]	CASIA V1.0	0	0.71	0.66	×
Proposed method	Feature extraction, optimization, HAC[†]	CASIA V2.0	0	0.78	0.74	×

[†] Our method eliminates the requirement of the training data set and deep learning environments requiring extensive computing and data, which is in contrast to the other schemes shown in the table

Table 4. Performance of our proposed model for five sets of experiments

Exp. No.	**Accuracy**	**F1- Score**	**Precision**	**Recall**	**MCC**
1	0.84	0.70	0.79	0.65	0.66
2	0.81	0.66	0.73	0.63	0.61
3	0.89	0.78	0.90	0.72	0.74
4	0.82	0.67	0.77	0.61	0.62
5	0.87	0.75	0.84	0.71	0.70

Here, we report the performance of our model on five selected subsets of test images, each of which consists of five test images. We have performed our test on five randomly selected test images in each experiment and have reported the average performance of these as the result of each experiment set in Table 4. The proposed method achieves the best performance with an accuracy of 0.89, an F1-Score of 0.78, Precision and Recall of 0.90 and 0.72, respectively, and a MCC score of 0.74. The composite bar diagram representing the proposed model's performance is given in Fig. 5.

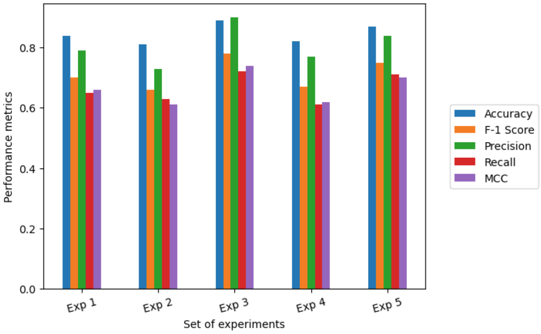


Fig. 5. Composite bar diagram representing the performance of our proposed model

4 Conclusion and Future Scope

This work proposes an effective method to detect image splicing and locate the spliced regions from an input test image without the requirement of a training dataset. We have extracted HOG and GLCM features from each block of a test image after it is subdivided into multiple non-overlapping blocks, the combined feature vector is optimized, and finally, an unsupervised clustering method named HAC is implemented for clustering the authentic blocks and spliced blocks of the detected spliced image. If the number of clusters in the resultant dendrogram is two, the test image is detected as a spliced image, and the spliced blocks will be localized; otherwise, if all image blocks form a single cluster, it will be marked as an authentic image. We assess the efficiency of our suggested method by comparing the predicted result of each image block with the associated ground truth image. The experiment's findings indicate that our scheme can successfully detect whether the test image is spliced or authentic and also locates the spliced regions of the spliced image with better performance without requiring any additional training dataset. The future direction of this work includes exploring the suitability of HAC in large-scale test data scenarios as well as investigating the robustness of unsupervised splicing detectors against post-processing image attacks.

Appendix-A

Contrast: It calculates the intensity difference between a pixel and its surrounding pixels over the whole image. If contrast is high, it indicates that the image is visually more clear [15] and sharp. It is formulated as follows:

$$Contrast = \sum_{i,j=0}^{N-1} G_{ij} \cdot (i - j)^2 \tag{9}$$

Correlation: It measures how a pixel value is correlated to its neighbour pixel over the entire image. A high value in correlation means the elements in GLCM are uniform [15]. The range of value of correlation is -1 to 1. If μ represents the GLCM mean, and σ^2 depicts the variance of intensities within the GLCM, then the correlation will be calculated as:

$$Correlation = \sum_{i,j=0}^{N-1} G_{ij} \cdot \frac{(i - \mu) \cdot (j - \mu)}{\sigma^2} \tag{10}$$

Dissimilarity: Weight in dissimilarity is calculated as $|i - j|$, and, therefore, as a pixel shifts away from the diagonal, its weights grow linearly. Consequently, the formulation of dissimilarity is as follows, if G_{ij} is the element at (i, j)th position in the GLCM and N is the number of gray levels:

$$Dissimilarity = \sum_{i,j=0}^{N-1} G_{ij} \cdot |i - j| \tag{11}$$

Energy: Energy calculates the sum of squared elements in GLCM. Energy has a value between 0 and 1. It is computed as follows if G_{ij} is the element at location (i, j) in the normalized GLCM and N is the number of gray levels:

$$Energy = \sum_{i,j=0}^{N-1} (G_{ij})^2 \tag{12}$$

Homogeneity: It estimates how much the distribution of GLCM elements resembles that of the GLCM diagonal. It is expressed as follows, and it has a range of 0 to 1.

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{G_{ij}}{1 + (i - j)^2} \tag{13}$$

References

1. Abd El-Latif, E.I., Taha, A., Zayed, H.H.: A passive approach for detecting image splicing based on deep learning and wavelet transform. Arab. J. Sci. Eng. **45**, 3379–3386 (2020)
2. Chicco, D., Jurman, G.: The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC Genomics **21**(1), 1–13 (2020)
3. Das, D., Naskar, R., Chakraborty, R.S.: Image splicing detection with principal component analysis generated low-dimensional homogeneous feature set based on local binary pattern and support vector machine. Multimedia Tools and Applications pp. 1–18 (2023)
4. Dong, J., Wang, W., Tan, T.: CASIA image tampering detection evaluation database. In: 2013 IEEE China Summit and International Conference on Signal and Information Processing. IEEE (Jul 2013). https://doi.org/10.1109/chinasip.2013.6625374, https://doi.org/10.1109/chinasip.2013.6625374
5. Jaiprakash, S.P., Desai, M.B., Prakash, C.S., Mistry, V.H., Radadiya, K.L.: Low dimensional dct and dwt feature based model for detection of image splicing and copy-move forgery. Multimedia Tools and Applications **79**, 29977–30005 (2020)
6. Jaiswal, A.K., Srivastava, R.: A technique for image splicing detection using hybrid feature set. Multimedia Tools and Applications **79**, 11837–11860 (2020)
7. Kadam, K., Ahirrao, S., Kotecha, K., Sahu, S.: Detection and localization of multiple image splicing using mobilenet v1. IEEE Access **9**, 162499–162519 (2021)
8. Li, C., Ma, Q., Xiao, L., Li, M., Zhang, A.: Image splicing detection based on markov features in qdct domain. Neurocomputing **228**, 29–36 (2017)
9. Li, Z., You, Q., Sun, J.: A novel deep learning architecture with multi-scale guided learning for image splicing localization. Electronics **11**(10), 1607 (2022)
10. Peng, J., Li, Y., Liu, C., Gao, X.: The circular u-net with attention gate for image splicing forgery detection. Electronics **12**(6), 1451 (2023)
11. Peng, J., Liu, C., Pang, H., Gao, X., Cheng, G., Hao, B.: Gp-net: Image manipulation detection and localization via long-range modeling and transformers. Appl. Sci. **13**(21), 12053 (2023)
12. Pham, N.T., Lee, J.W., Kwon, G.R., Park, C.S.: Hybrid image-retrieval method for image-splicing validation. Symmetry **11**(1), 83 (2019)
13. Prasanna, G.S., Pavani, K., Singh, M.K.: Spliced images detection by using viola-jones algorithms method. Materials Today: Proceedings **51**, 924–927 (2022)

14. Ren, R., Niu, S., Jin, J., Zhang, J., Ren, H., Zhao, X.: Multi-scale attention context-aware network for detection and localization of image splicing: Efficient and robust identification network. Applied Intelligence pp. 1–20 (2023)
15. Shen, X., Shi, Z., Chen, H.: Splicing image forgery detection using textural features based on the grey level co-occurrence matrices. IET Image Proc. **11**(1), 44–53 (2017)
16. Shi, X., Li, P., Wu, H., Chen, Q., Zhu, H.: A lightweight image splicing tampering localization method based on mobilenetv2 and srm. IET Image Processing (2023)
17. Thakur, A., Aggarwal, A., Walia, S., Saluja, K.: Localisation of spliced region using pixel correlation in digital images. In: 2019 International Conference on Signal Processing and Communication (ICSC). pp. 153–157. IEEE (2019)
18. Walia, S., Kumar, K.: Characterization of splicing in digital images using gray scale co-occurrence matrices. In: 2019 Twelfth International Conference on Contemporary Computing (IC3). pp. 1–6. IEEE (2019)
19. Wang, R., Lu, W., Li, J., Xiang, S., Zhao, X., Wang, J.: Digital image splicing detection based on markov features in qdct and qwt domain. In: Digital Forensics and Forensic Investigations: Breakthroughs in Research and Practice, pp. 61–79. IGI Global (2020)
20. Xiao, B.: Principal component analysis for feature extraction of image sequence. In: 2010 International conference on computer and communication technologies in agriculture engineering. vol. 1, pp. 250–253. IEEE (2010)
21. Zeng, N., Wu, P., Zhang, Y., Li, H., Mao, J., Wang, Z.: Dpmsn: A dual-pathway multiscale network for image forgery detection. IEEE Transactions on Industrial Informatics (2024)
22. Zhu, N., Li, Z.: Blind image splicing detection via noise level function. Signal Processing: Image Communication **68**, 181–192 (2018)

End-to-End User-Defined Keyword Spotting Using Shifted Delta Coefficients

V. Kesavaraj(✉), M. Anuprabha, and Anil Kumar Vuppala

Speech Processing Laboratory, International Institute of Information Technology Hyderabad, Hyderabad, India
{kesavaraj.v,anuprabha.m}@research.iiit.ac.in, anil.vuppala@iiit.ac.in

Abstract. Identifying user-defined keywords is crucial for personalizing interactions with smart devices. Previous approaches of user-defined keyword spotting (UDKWS) have relied on short-term spectral features such as mel frequency cepstral coefficients (MFCC) to detect the spoken keyword. However, these features may face challenges in accurately identifying closely related pronunciation of audio-text pairs, due to their limited capability in capturing the temporal dynamics of the speech signal. To address this challenge, we propose to use shifted delta coefficients (SDC) which help in capturing pronunciation variability (transition between connecting phonemes) by incorporating long-term temporal information. The performance of the SDC feature is compared with various baseline features across four different datasets using a cross-attention based end-to-end system. Additionally, various configurations of SDC are explored to find the suitable temporal context for the UDKWS task. The experimental results reveal that the SDC feature outperforms the MFCC baseline feature, exhibiting an improvement of 8.32% in area under the curve (AUC) and 8.69% in terms of equal error rate (EER) on the challenging Libriphrase-hard dataset. Moreover, the proposed approach demonstrated superior performance when compared to state-of-the-art UDKWS techniques.

Keywords: shifted delta coefficients · mel spectrogram · cross-attention · user-defined keyword spotting

1 Introduction

Advancements in deep learning technology have transformed voice-activated interactions with machines from science fiction to reality. The proliferation of voice assistants like Amazon's Alexa, Apple's Siri, Google's Assistant, and Microsoft's Cortana are good proof of this [1]. These voice assistants are activated using a technology called spoken keyword spotting, or simply keyword spotting, which detects specific wake-up words within a continuous audio stream [2]. This helps to avoid activating the more computationally intensive automatic speech recognition (ASR) when unnecessary. For instance, Google's voice search

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 374–386, 2025.
https://doi.org/10.1007/978-3-031-78312-8_25

responds to the phrase "Okay Google," while Apple's conversational assistant is activated with the phrase "Hey Siri" [3]. However, these keywords are not personalized.

With the growing demand for personalized voice assistants, user-defined keyword spotting (UDKWS) [4,5], also known as custom keyword detection or open vocabulary keyword spotting, has gained considerable attention. Unlike closed vocabulary keyword spotting [6], where only predetermined keywords are recognized, open vocabulary keyword spotting deals with the challenge of identifying random keywords that the model may not have encountered during training, adding an additional layer of complexity to the task.

Over the years, various techniques have been explored for UDKWS. One of the earliest approaches involves the use of large-vocabulary continuous speech recognition (LVCSR) systems [7,8]. These systems decode the speech signal, after which the keyword is searched in the generated lattices. Another approach is keyword/filler hidden markov model (HMM) [9,10]. In this approach, separate HMMs are trained to model keyword and non-keyword audio segments. While these architectures allow for customization of the keyword by modifying the decoding graph, the computational requirements remain significant.

Recent works in UDKWS have focused on developing end-to-end systems that take two inputs: the enrolled keyword references and the speech data to be detected. One such classical approach is the query-by-example (QbyE) [11,12] approach, which involves matching input queries with pre-enrolled examples. However, the effectiveness of the QbyE method relies heavily on the similarity between the recorded speech during enrollment and the subsequent evaluated speech recordings. Challenges such as diverse vocal characteristics among users and background noise in different environments can significantly impact the consistency of the QbyE method's performance. To address these challenges, researchers have explored text enrollment-based methods [4,13].

For instance, [14] proposes an ASR-free end-to-end system that generates audio embedding and keyword embedding using an acoustic encoder and a keyword encoder, respectively. These embeddings are then merged into a multilayer perceptron for keyword existence prediction. In [13], an attention-based cross-modal matching approach is proposed to learn the agreement between audio and text keyword at the utterance level. In [15], a novel zero-shot UDKWS is proposed to learn the audio-phoneme relationship of the keyword through phoneme-level detection loss. Also, [16] introduced dynamic sequence partitioning to optimally partition the audio embedding sequence into the same length as the text sequence. These recent end-to-end techniques [13,15,16], primarily depend on evaluating speech and text representations in a common latent space, demonstrated promising results in the custom keyword spotting task. Despite these significant advancements in the field of UDKWS, the predominant focus has been on the advancement of deep learning models and training approaches. There has been relatively limited exploration of feature engineering which plays a crucial role in enhancing the performance of any speech application. This observation has motivated us to perform feature-level exploration for UDKWS task.

In literature, mel-scale related features such as MFCC and mel spectrogram are the most commonly used features in UDKWS [13,15,16]. While these features provide a good estimation of the local spectra, they may not fully capture the temporal dynamics, such as changes in pronunciation over time, present in spoken keywords. Incorporating contextual information could help the system in capturing this pronunciation variability by modeling frame-level dependencies. Notably, in some studies [17,18], SDC is well-regarded for their ability to capture long-term temporal information (stacking delta features across several frames) for language identification tasks. Motivated by this fact, we propose to use the SDC feature to enhance the robustness of UDKWS, especially in distinguishing between similar pronunciations of audio-text pairs. To the best of our knowledge, this is the first study that explored the importance of long-term temporal information at feature level for UDKWS task. The key contributions of this study include:

- Performance comparison of SDC features with commonly used short-term spectral features, namely MFCC, mel spectrogram, perceptual linear prediction coefficients (PLP), and relative spectral-perceptual linear prediction coefficients (RASTA-PLP) in a common experimental setup.
- Exploration of different configurations of SDC features to determine the appropriate temporal context for the UDKWS task.
- Examination of the system's performance for keywords of different word lengths.
- Assessment of the efficiency of the proposed approach through comprehensive comparisons with various state-of-the-art UDKWS systems.

The organization of this paper is as follows: Section 2 provides details about the architecture, Section 3 discusses the feature extraction techniques, Section 4 describes the experimental setup, Section 5 presents the results and discussion and Section 6 concludes the study.

2 Architecture

In this section, we will discuss the details of the architecture that is used for studying different audio features for the UDKWS task, as shown in Fig. 1. The proposed architecture is adopted from [19]. It consists of four submodules: audio encoder, text encoder, pattern extractor, and pattern discriminator.

2.1 Audio Encoder

In this study, various short-term and long-term spectral features (discussed in Section 3) are used as input to the audio encoder. The encoder utilized two 2-D convolutional layers (Conv2D), each consisting of 32 filters with a kernel size of 3. To enhance computational efficiency, the initial convolution layer employs a stride of 2 to skip processing of consecutive frames. Additionally, batch normalization is applied after each Conv2D operation to ensure stable training.

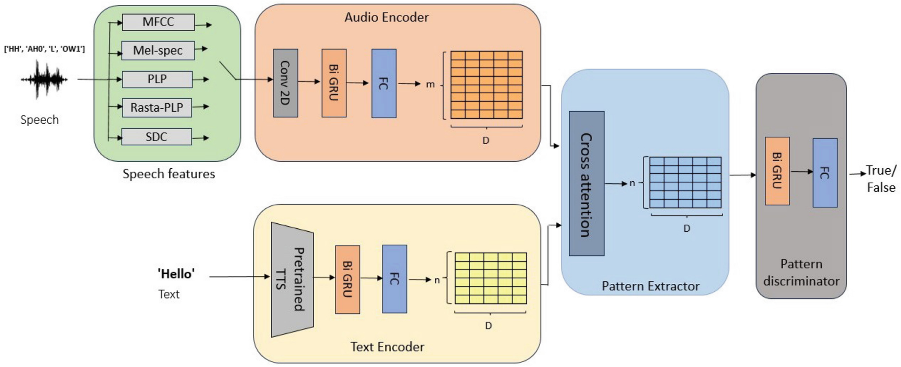


Fig. 1. Proposed architecture for user-defined keyword spotting

Following the Conv2D layers, two bidirectional gated recurrent units (Bi-GRU) with a dimension of 64 each are utilized. Finally, a 128-dimensional audio embedding is produced by passing the output from the final Bi-GRU layer to a dense layer. The output from the audio encoder is denoted as $E_a \in \mathbb{R}^{m \times D}$, where m and D denote the length of the audio (i.e. number of frames) and embedding dimension, respectively.

2.2 Text Encoder

It includes a pre-trained Tacotron 2 [20] model, a recurrent sequence-to-sequence text-to-speech (TTS) system, which takes character sequences as input to produce the corresponding audio output. The integration of a pre-trained TTS system in the text encoder is inspired by [19]. The main motivation for including the TTS system is to generate text representations that are aware of audio projections. According to [19], employing intermediate representations from the pre-trained TTS model leads to superior performance compared to using character embeddings as text features. It is also demonstrated that representations from the LSTM block of the Tacotron 2 encoder, with a dimension of 512, effectively act as text features. Consequently, this study adopts a similar approach. The resulting intermediate representations from the TTS model are then passed through a Bi-GRU layer with a dimension of 64. The output from the Bi-GRU layer is subsequently fed into a dense layer with 128 units. The output from the text encoder is denoted as $E_t \in \mathbb{R}^{n \times D}$, where n and D denote the length of the text (i.e. number of characters) and embedding dimension, respectively.

2.3 Pattern Extractor

Motivated by [21], the pattern extractor employs a cross-attention mechanism to capture temporal correlations between audio and text embeddings. All hidden states from the output of audio and text encoders are fed into the cross-attention layer to preserve the temporal information. In this setup, the audio embedding E_a functions as both the key and value, while the text embedding E_t acts as the query. The resulting context vector from the pattern extractor contains information regarding the agreement between audio and text.

2.4 Pattern Discriminator

The pattern discriminator determines whether audio and text inputs share the same keyword or not. To achieve this, it consists of a single Bi-GRU layer with a dimension of 128 that takes the context vector from the pattern extractor as input. Subsequently, the output from the last frame of the Bi-GRU layer is passed through a dense layer with sigmoid as an activation function.

3 Feature Extraction

Representing a speech utterance in a vector of parameters is defined as feature extraction [22]. The significant aim of performing this step is to derive the appropriate/relevant information. In this section, we discuss about five important feature extraction techniques, namely mel spectrograms [13], MFCC [23], PLP [24], RASTA-PLP [25], SDC [17]. These five features are selected after reviewing several works [26–30] on speech related applications.

3.1 Mel Spectrogram

The mel spectrogram is derived from the magnitude spectrogram, but it's different because the mel filter bank mimics the human ear's perception and emphasizes the lower frequency region more than the higher frequencies. Initially, the magnitude spectrogram-time-frequency representation-is obtained by segmenting the speech waveform into windowed segments and applying the fast fourier transform to each segment. Following this, the computed magnitude spectrogram is mapped to a mel-scale using 40 mel filter banks, and then subjected to a logarithmic operation to generate the mel-spectrogram.

3.2 Mel-Frequency Cepstral Coefficients

MFCC, a popular feature in speech signal processing, captures vocal tract characteristics by representing the short-term power spectrum through a linear cosine transform. This transformation operates on a logarithmic power spectrum, nonlinearly scaled to the mel frequency range. The process of calculating MFCC involves windowing the signal, calculating discrete Fourier transform (DFT) coefficients for each window, taking the logarithm of the DFT magnitude, filtering frequencies with the mel scale, and finally extracting the MFCC coefficients. The first 13 cepstral coefficients are considered as MFCC features. The first and second derivatives of 13-dimensional MFCC features have been combined with static MFCCs, referred to as MFCC+Δ+$\Delta\Delta$, to capture the temporal dynamics present in the speech signal.

3.3 Perceptual Linear Prediction

Several alternatives to MFCC have been proposed for representing short-term speech signals - One such alternative is PLP (perceptual linear prediction coefficients). PLP is a feature that gives representation conforming to a smoothed short-term spectrum that has been equalized and compressed, similar to human hearing, thus making it similar to the MFCC. The process of calculating PLP features starts with windowing the signal, computing discrete Fourier transform (DFT) coefficients for each window, and taking the logarithm of the DFT magnitude to obtain power spectral estimates. Next, a trapezoidal filter is applied at 1-bark intervals to merge overlapping critical band filter responses in the power spectrum, thereby compressing higher frequencies into a narrow band. Finally, the spectral amplitude is compressed by taking the cubic root to match the nonlinear relationship between sound intensity and perceived loudness.

3.4 Relative Spectral - Perceptual Linear Prediction

RASTA-PLP builds upon the PLP technique by introducing a crucial addition: a bandpass filter at each sub-band. By suppressing undesirable frequencies, RASTA-PLP increases the robustness of PLP to noise. The process of RASTA-PLP involves several steps. Initially, it calculates the critical-band power spectrum, followed by the application of a compressing static nonlinear transformation to the spectral amplitude. Subsequently, it filters the time trajectory of each transformed spectral component with a bandpass filter. After this, the filtered speech undergoes further transformation using an expanding static nonlinear transformation. Additionally, it includes equal loudness curve adjustment and the application of the intensity-loudness power law to replicate the human auditory system. In essence, RASTA filtering acts as a modulation-frequency bandpass filter, emphasizing the modulation frequency range most relevant to speech, while disregarding lower or higher modulation frequencies.

3.5 Shifted Delta Coefficients

SDC features, extensively utilized in language identification [18], are pivotal for capturing long-term temporal information. Inspired by this, we propose to use the SDC features for the UDKWS task. The main motivation is to enhance the model's ability to capture the pronunciation variability of the spoken keyword, thus improving the overall performance of UDKWS. In this study, SDC features are computed from Mel-spectrogram, since it gives better performance compared to all other short-term spectral features. The calculation of the SDC feature depends on four parameters, and it is represented as N-d-p-k. Here N represents the number of cepstral coefficients for every frame, d denotes the amount of shift (delay) from the current frame, p denotes the shift between the consecutive delta blocks, and k denotes the number of frames whose deltas are to be concatenated. The delta feature vector for t^{th} frame in the i^{th} iteration is computed as

$$\delta_c(t,i) = c(t + ip + d) - c(t + ip - d), \quad \text{where } 0 \leq i \leq k - 1 \tag{1}$$

These k delta computations are stacked as in (2) to form k×N dimensional shifted delta coefficients

$$SDC(t) = \begin{pmatrix} \delta_c(t,0) \\ \delta_c(t,1) \\ \vdots \\ \delta_c(t,k-1) \end{pmatrix} \tag{2}$$

The computed stacked delta features are then combined with the static mel spectrogram features to obtain the final SDC feature vector. The effect of variation in parameters (d and k), which control the amount of temporal context, has also been studied.

4 Experimental Setup

4.1 Database

The LibriPhrase dataset [13], derived from the LibriSpeech corpus [31], is utilized for both training and evaluation. It comprises short phrases with varying word lengths, ranging from 1 to 4. The training set of LibriPhrase was generated using the train-clean-100 and train-clean-360 subsets, while the evaluation set was derived from the train-others-500 subset. The evaluation set consists of 4391, 2605, 467, and 56 episodes of each word length respectively. Each episode has three positive and three negative pairs. The negative samples are further categorized into easy and hard based on Levenshtein distance [32], leading to the creation of the LibriPhrase Easy ($\mathrm{LP_E}$) and LibriPhrase Hard ($\mathrm{LP_H}$) datasets. Each example is denoted by 3 entities: (audio, text, target) where the target value is 1 for a positive pair and 0 for a negative pair.

For a comprehensive evaluation of model performance, we expanded our assessment beyond the LibriPhrase dataset by including two additional datasets: the Google Speech Commands V1 dataset (G) [33] and the Qualcomm Keyword Speech dataset (Q) [34]. The Google Speech Commands V1 dataset (G) contains speech recordings from 1,881 speakers, emphasizing 30 small keywords. From that, the validation dataset corresponding to 30 keywords is used for evaluation. On the other hand, the Qualcomm Keyword Speech dataset (Q) includes 4,270 utterances of four keywords spoken by 50 speakers. Each speaker in this dataset contributes approximately 22-23 instances for each keyword.

4.2 Implementation Details

In the feature extraction, all spectral vectors are obtained by block processing the whole speech into short segments using a window length of 25 ms and overlap of 10 ms. A pre-emphasis factor of 0.97 is applied to emphasize the amount of energy in the high frequency regions. Hamming window is used during the windowing process of feature extraction to reduce the spectral leakage. Zero padding is applied along the time dimension to ensure that the input feature representation is of equal size, as required by the input of the Conv2D layer.

The training pipeline is structured as a binary classification task with the objective of classifying the similarity of input pairs {audio, text}. The training process utilizes binary cross-entropy loss as the training criterion and employs the Adam optimizer [35] with default parameters for optimization. The model is trained with a batch size of 128 and a fixed learning rate of 10^{-4}. A dropout of 0.2 is applied after each layer in both audio and text encoders to prevent overfitting. The best-performing model is selected based on the model performance on the validation set. For training, we used four NVIDIA GeForce RTX 2080 Ti GPUs.

5 Results and discussions

The proposed approach provides comprehensive insights about the spoken keyword by leveraging SDC features, which are well-regarded for their ability to capture long-term temporal information. To validate the effectiveness of our approach, we conducted extensive experiments across diverse datasets, with the results and corresponding plots presented in this section.

5.1 Comparison of different front-end features

In this section, the performance comparison between SDC features and various short-term spectral features is discussed to study the importance of long-term temporal information in the UDKWS task. The results are presented in Table 1. Upon analysing the results, it's clear that SDC features consistently outperform all baseline features across all datasets. Furthermore, when compared to the MFCC feature, the SDC feature demonstrates a notable improvement of 8.69% in AUC and 8.32% in EER on the challenging $\mathrm{LP_H}$ dataset, which comprises similar pronunciations of audio-text pairs (e.g., "madame" and "modem"). This improvement is attributed to their ability to capture the temporal dynamics of speech signals by incorporating contextual information. On the other hand, PLP, RASTA-PLP, and mel spectrogram exhibit competitive performance with consistently good AUC scores, indicating them as preferable alternatives to SDC. In contrast, MFCC performs poorly compared to all other features. Compared to MFCC alone, MFCC+Δ+$\Delta\Delta$ which models the temporal dynamics of the speech signal to some extent by concatenating their first and second derivatives to the original MFCC feature, provides significant improvements. Overall, the observations from Table 1 suggest that speech information extracted over a longer context plays a pivotal role in the development of systems for UDKWS tasks.

5.2 SDC configuration

The calculation of SDC features relies on four parameters: N, d, p, and k. Variations in these parameters can significantly affect the amount of temporal context

Table 1. Performance comparison of SDC features with various short-term spectral features across different datasets: Google Commands V1 (G), Qualcomm Keyword Speech dataset (Q), Libriphrase-Easy (LP_E), and Libriphrase-Hard (LP_H)

Features	**EER (%)**				**AUC (%)**			
	G	**Q**	$\mathbf{LP_E}$	$\mathbf{LP_H}$	**G**	**Q**	$\mathbf{LP_E}$	$\mathbf{LP_H}$
MFCC	32.24	12.59	7.99	29.8	73.95	91.2	97.8	77.21
MFCC+Δ+$\Delta\Delta$	30.28	11.8	6.8	27.01	76.5	93.29	98.06	78.54
Mel Spectrogram	27.91	16.7	5.89	26.45	79.13	90.97	98.21	79.81
PLP	28.43	15.37	6.58	25.22	77.65	90.47	97.88	78.81
RASTA-PLP	27.4	14.32	6.42	25.84	78.05	91.24	97.82	79.7
SDC	**23.54**	**9.61**	**3.84**	**21.48**	**83.56**	**96.73**	**98.34**	**85.90**

captured. Therefore, an ablation study is conducted by varying d and k values to determine the suitable temporal context for UDKWS, with the results depicted in Fig. 2. Fig. 2 (a) & (b) illustrates the performance change in terms of AUC and EER for varying d values (amount of shift from the current frame) from 1 to 4 while keeping the other parameters fixed. It can be observed that as the d value increases, the performance drops across all datasets, indicating a loss of information when the shift is increased in calculating SDC features. Fig. 2 (c) & (d) show the effect of varying k values (number of frames whose deltas are stacked) ranging from 5 to 10, while keeping the other parameters fixed. It is evident that as the k value increases from 5 to 8, the performance in terms of AUC and EER also increases across all datasets, demonstrating the effect of the context window on UDKWS. However, beyond a k value of 8, performance either saturates or declines, indicating the need for precise contextual information to reliably recognize keywords. Overall, it is evident that the SDC configuration 40-1-3-8 exhibits the best performance with optimal temporal context for UDKWS.

5.3 Analysis on the length of keywords

The evaluation of the system across different word lengths for mel spectrogram (best performing among baseline) and SDC features is studied on the LibriPhrase evaluation dataset. The results in terms of EER, AUC and F1-score are presented in Table 2. When compared to the mel spectrogram, SDC demonstrates absolute improvements of 3.24%, 2.33%, 1.81%, and 2.02% in F1-score for word lengths 1, 2, 3, and 4 respectively. Despite the improvements, SDC encounters challenges similar to the mel spectrogram in keyword recognition as word length increases.

Table 2. Performance comparison of SDC and mel spectrogram features across different word lengths

Feature	**Word Length**	**EER (%)**	**AUC (%)**	**F 1 score (%)**
Mel Spectrogram	1	7.67	97.01	91.11
	2	8.55	96.57	90.98
	3	9.05	95.6	89.76
	4	9.25	95.34	88.30
SDC	1	5.37	98.25	94.34
	2	6.35	97.87	93.31
	3	7.24	96.91	91.57
	4	8.29	96.31	90.32

5.4 Comparison of various UDKWS techniques

In this section, the performance of the proposed approach is compared with various state-of-the-art UDKWS techniques across G, Q, $\mathrm{LP_E}$, $\mathrm{LP_H}$ datasets, and the results are presented in Table 3. Evaluation results show that among all the baselines, CMCD (cross-modlaity correspondence detector) [13] demonstrates strong performance, while Triplet [4] shows weak performance across the Q, $\mathrm{LP_E}$, and $\mathrm{LP_H}$ datasets. The attention-based QbyE method demonstrates superior performance on the G dataset due to its similarity scoring mechanism, particularly when the keyword is included in the training set. However, it shows degraded performance when the keyword is unfamiliar as observed in Q and LibriPhrase. In contrast, our proposed approach outperforms all the baselines on all datasets except G. Specifically, compared to the CMCD baseline, our proposed method demonstrates a substantial improvement in EER of 4.58% in the $\mathrm{LP_E}$ dataset and 11.42% in the $\mathrm{LP_H}$ dataset. Additionally, the model is evaluated on datasets G and Q without any fine-tuning to assess its generalization capability. We observe a consistent improvement of approximately 2% on the AUC

Table 3. Performance comparison of the proposed method with various UDKWS techniques across different datasets: Google Commands V1 (G), Qualcomm Keyword Speech dataset (Q), LibriPhrase-Easy ($\mathrm{LP_E}$), and LibriPhrase-Hard ($\mathrm{LP_H}$).

Method	**EER (%)**				**AUC (%)**			
	G	**Q**	**$\mathrm{LP_E}$**	**$\mathrm{LP_H}$**	**G**	**Q**	**$\mathrm{LP_E}$**	**$\mathrm{LP_H}$**
CTC [12]	31.65	18.23	14.67	35.22	66.36	89.69	92.29	69.58
Attention [11]	**14.75**	49.13	28.74	41.95	**92.09**	50.13	78.74	62.65
Triplet [4]	35.6	38.72	32.75	44.36	71.48	66.44	63.53	54.88
CMCD [13]	27.25	12.15	8.42	32.9	81.06	94.51	96.7	73.58
Proposed	23.54	**9.61**	**3.84**	**21.48**	83.56	**96.73**	**98.34**	**85.9**

metric and 3% on the EER metric across datasets G and Q compared to the CMCD baseline. This demonstrates the effectiveness of the proposed approach in recognizing user-defined keywords that are not seen during training.

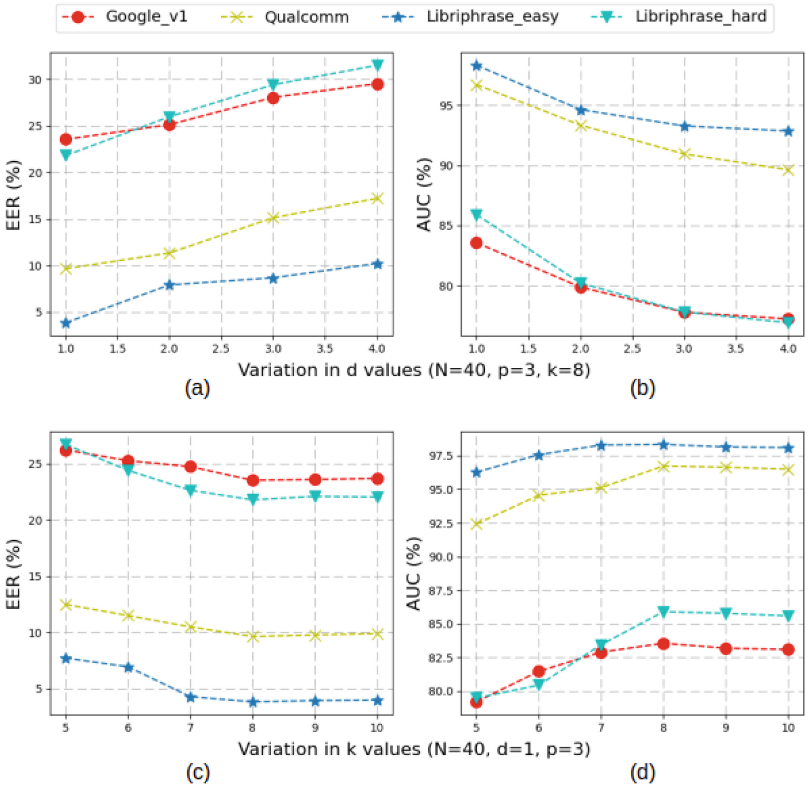


Fig. 2. Performance Analysis of SDC Configuration across four datasets. (a) & (b) illustrate the effect of varying d values. (c) & (d) illustrate the effect of varying k values.

6 Conclusion

This study presented the importance of long-term temporal information for the UDKWS task. The evaluation results indicated that SDC features outperformed the widely used short-term spectral features. Notably, it showcased its potential in distinguishing similar pronunciations of audio-text pairs in the Libriphrase hard dataset. Furthermore, the ablation study on different SDC configurations revealed that configuration 40-1-3-8 exhibited the best performance with a suitable temporal context. Moreover, the proposed approach demonstrated superior performance compared to state-of-the-art UDKWS approaches. In future work, the focus will be on improving the performance of the UDKWS system by exploring the potential of hybrid feature extraction approaches rather than relying on individual counterparts.

References

1. Matthew B Hoy. “Alexa, Siri, Cortana, and more: an introduction to voice assistants”. In: Medical reference services quarterly 37.1 (2018), pp. 81-88
2. López-Espejo, I., et al.: Deep spoken keyword spotting: An overview. IEEE Access **10**, 4169–4199 (2021)
3. Oriol Vinyals and StevenWegmann. “Chasing the metric: Smoothing learning algorithms for keyword detectionâĂİ. In: Proc. ICASSP. IEEE. 2014, pp. 3301-3305
4. Niccolò Sacchi et al. “Open-Vocabulary Keyword Spotting with Audio and Text Embeddings”. In: Proc. Interspeech. 2019, pp. 3362-3366
5. Krishna Gurugubelli, Sahil Mohamed, and Rajesh Krishna KS. “Comparative Study of Tokenization Algorithms for End-to-End Open Vocabulary Keyword Detection”. In: Proc. ICASSP. IEEE. 2024, pp. 12431-12435
6. Tara N. Sainath and Carolina Parada. “Convolutional neural networks for small-footprint keyword spotting”. In: Proc. Interspeech. 2015, pp. 1478- 1482
7. David RH Miller et al. “Rapid and accurate spoken term detection”. In: Proc. Interspeech. 2007
8. Guoguo Chen et al. “Using proxies for OOV keywords in the keyword search task”. In: Proc. Automatic Speech Recognition and Understanding (ASRU). IEEE. 2013, pp. 416-421
9. Richard C Rose and Douglas B Paul. “A hidden Markov model based keyword recognition system”. In: Proc. ICASSP. IEEE. 1990, pp. 129-132
10. Jan Robin Rohlicek et al. “Continuous hidden Markov modeling for speakerindependent word spotting”. In: Proc. ICASSP. IEEE. 1989, pp. 627-630
11. Jinmiao Huang et al. “Query-by-example keyword spotting system using multi-head attention and soft-triple loss”. In: Proc. ICASSP. IEEE. 2021, pp. 6858-6862
12. L. Lugosch, S. Myer, and V. S. Tomar. “DONUT: CTC-based Query-by- Example Keyword Spotting”. In: NeurIPS Workshop on Interpretability and Robustness in Audio, Speech, and Language. Montreal, Canada, Dec. 2018
13. Hyeon-Kyeong Shin et al. “Learning Audio-Text Agreement for Openvocabulary Keyword Spotting”. In: Proc. INTERSPEECH. 2022, pp. 1871- 1875
14. Kartik Audhkhasi et al. “End-to-end ASR-free keyword search from speech”. In: IEEE Journal of Selected Topics in Signal Processing 11.8 (2017), pp. 1351-1359
15. Yong-Hyeok Lee and Namhyun Cho. “PhonMatchNet: Phoneme-Guided Zero-Shot Keyword Spotting for User-Defined Keywords”. In: Proc. INTERSPEECH. 2023, pp. 3964-3968
16. Kumari Nishu, Minsik Cho, and Devang Naik. “Matching Latent Encoding for Audio-Text based Keyword Spotting”. In: Proc. INTERSPEECH. 2023, pp. 1613-1617
17. Ravi Kumar Vuddagiri, Hari Krishna Vydana, and Anil Kumar Vuppala. “Improved Language Identification Using Stacked SDC Features and Residual Neural Network”. In: Proc. 6th Workshop on Spoken Language Technologies for Under-Resourced Languages. 2018, pp. 210-214
18. Pedro A Torres-Carrasquillo et al. “Approaches to language identification using Gaussian mixture models and shifted delta cepstral features.” In: Proc. Interspeech. 2002, pp. 89-92

19. Kesavaraj V and Anil Vuppala. Open vocabulary keyword spotting through transfer learning from speech synthesis. In: 2024 International Conference on Signal Processing and Communications (SPCOM). IEEE. 2024, pp. 1–5
20. Jonathan Shen et al. “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions”. In: Proc. ICASSP. IEEE. 2018, pp. 4779- 4783
21. Ashish Vaswani et al. “Attention is all you need”. In: Advances in neural information processing systems 30 (2017)
22. Deepti Deshwal, Pardeep Sangwan, and Divya Kumar. “A language identification system using hybrid features and back-propagation neural network”. In: Applied Acoustics 164 (2020), p. 107289
23. Zrar Kh Abdul and Abdulbasit K Al-Talabani. “Mel frequency cepstral coefficient and its applications: A review”. In: IEEE Access 10 (2022), pp. 122136-122158
24. Hynek Hermansky. “Perceptual linear predictive (PLP) analysis of speech”. In: the Journal of the Acoustical Society of America 87.4 (1990), pp. 1738- 1752
25. H Hermansky et al. “RASTA-PLP speech analysis technique”. In: [Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing. Vol. 1. IEEE. 1992, pp. 121-124
26. Amit Meghanani, CS Anoop, and AG Ramakrishnan. “An exploration of log-mel spectrogram and MFCC features for Alzheimer’s dementia recognition from spontaneous speech”. In: IEEE spoken language technology workshop (SLT). IEEE. 2021, pp. 670-677
27. HK Palo, Mihir Narayana Mohanty, and Mahesh Chandra. “Use of different features for emotion recognition using MLP network”. In: Computational Vision and Robotics: Proceedings of ICCVR. Springer. 2015, pp. 7-15
28. Ben Milner. “A comparison of front-end configurations for robust speech recognition”. In: Proc. ICASSP. IEEE. 2002, pp. I-797
29. Savitha S Upadhya, AN Cheeran, and Jagannath H Nirmal. “Thomson Multitaper MFCC and PLP voice features for early detection of Parkinson disease”. In: Biomedical Signal Processing and Control 46 (2018), pp. 293- 301
30. Md. Sahidullah, Tomi Kinnunen, and Cemal Hanilçi. “A comparison of features for synthetic speech detection”. In: Proc. Interspeech. 2015, pp. 2087- 2091
31. Vassil Panayotov et al. “Librispeech: an asr corpus based on public domain audio books”. In: Proc. ICASSP. IEEE. 2015, pp. 5206-5210
32. Vladimir I Levenshtein et al. “Binary codes capable of correcting deletions, insertions, and reversals”. In: Soviet physics doklady. Vol. 10. 8. Soviet Union. 1966, pp. 707-710
33. Pete Warden. “Speech commands: A dataset for limited-vocabulary speech recognition”. In: arXiv preprint arXiv:1804.03209 (2018)
34. Byeonggeun Kim et al. “Query-by-example on-device keyword spotting”. In: Proc. ASRU. IEEE. 2019, pp. 532-538
35. Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: arXiv preprint arXiv:1412.6980 (2014)

Comparative Analysis of Voice Conversion in German

Karla Schäfer[1,2](✉), Jeong-Eun Choi[1,2], and Martin Steinebach[1,2]

[1] Fraunhofer Institute for Secure Information Technology, Darmstadt, Germany
`{karla.schaefer,jeong-eun.choi,martin.steinebach}@sit.fraunhofer.de`
[2] National Research Center for Applied Cybersecurity, Darmstadt, Germany

Abstract. Voice Conversion (VC) has gained attention due to its rapid development and increased accessibility. However, this also brings a potential threat for misuses. Consequently, it is crucial to thoroughly assess the performance of VC models. Current research, however, predominantly focuses on the evaluation of VC models on English, neglecting other languages during evaluation and focusing on one conversion scenario. To address this research gap, this paper aims to evaluate four VC models, namely kNN-VC, FreeVC, QuickVC, and RVC, on German speakers across three different conversion settings: any-to-any conversion (i.e., without fine-tuning), VC with speaker fine-tuning, and VC with language (German) fine-tuning. Additionally, we examined the influence of target speaker audio length using data ranging from 10 to 2400 seconds for generation.

Keywords: Voice Conversion · German · Quality · Similarity

1 Introduction

With the increasing use of social media platforms like Instagram or TikTok, high-quality recordings of individuals can be easily found on the internet. Together with the advances in artificial intelligence, the quality of audio deepfakes is rapidly improving, posing a potential threat for malicious purposes. For instance, it becomes feasible to attain believable syntheses, even when only a limited number of short high-quality recordings are available from the target speaker. The synthesized voices can then be used as impersonation attack to fool people or automatic speaker verification/recognition systems, for example to log into a bank account [20]. Voice conversion (VC) is one method of creating such audio manipulations and uses audio recordings of the target and source speakers in order to alter the voice of a source speaker to a target style, such as speaker identity, prosody and emotion, while keeping the linguistic content unchanged [7]. Another method is called text-to-speech synthesis (TTS). It can be assumed

This research work was supported by the National Research Center for Applied Cybersecurity ATHENE as well as within the joint project SecMedID with the Federal Office for Information Security.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 387–402, 2025.
https://doi.org/10.1007/978-3-031-78312-8_26

that the impact of language differences is less for VC than for TTS, as long as VC methods do not rely predominantly on the TTS method. However, this assumption has not yet been observed and discussed in detail. Instead, we can observe that most of the VC methods presented recently have largely been evaluated only on English or Mandarin speakers. An analysis of the papers presented at ICASSP 2024 revealed that, out of 11 VC papers, five papers [8,9,11,12,21] were evaluated on English speakers and four trained and evaluated on Mandarin [13,16,22,23]. Another paper was evaluated on Japanese [14] and a further on Portuguese and English [19]. The search found no publications with a focus on VC in German. A search of Google Scholar with the terms "voice conversion" and "German" yielded no results indicating the analysis of VC in the German language (as of 5 July 2024). Consequently, it can be reasonably concluded that our research represents the inaugural investigation into the applicability of state-of-the-art VC models to the German language.

This paper evaluates VC models on German speakers of two genders (male and female) in three different settings: 1) an any-to-any conversion setting, where the target and source speakers are unseen during training, 2) fine-tuning the models on the target speaker using different amounts of target speaker recordings, and 3) fine-tuning the models on a German dataset before performing any-to-any conversion. Moreover, we examined the influence of target speaker audio length using data ranging from 10 to 2400 seconds for generation, as this has not been comprehensively evaluated and compared on VC methods for the best of our knowledge. In doing this research we wanted to extend the application area of VC to German and evaluate the applicability and generalizability of models initial trained only on English data for generating German audio recordings. Given that the domains of application for VC, such as gaming or health, are not confined to the English language.

In the following, the related work is presented (Section 2), followed by an introduction of the four VC methods examined (Section 3). In Section 4 the experiments conducted are explained, followed by the results (Section 5), a comparison with the results obtained in the paper on English data (Section 6) and a conclusion (Section 7).

2 Related Work

When performing VC based on deep learning models, the typical first step is to disentangle content information and speaker information from the source and target speech. Subsequently, this information is used to convert the voice of the speech. As a result, the quality of the converted speech relies on 1) the disentanglement ability of the VC model, and 2) the reconstruction ability of the VC model [7]. A major problem in VC is the speech representation disentanglement (SRD). For SRD using text, an automatic speech recognition (ASR) model can be used to extract the content representation. Another method is the use of shared linguistic knowledge from a TTS model [7]. But these require a large amount of annotated data for training the ASR or TTS. Therefore, text-free methods were introduced. Text-free methods learn to extract content information without the guidance of text annotation. Common text-free VC approaches use for example an information bottleneck, vector quantization, and instance normalization for SRD [7]. However, their performance is, in general, lower in comparison to text-based approaches [25]. Another way to differentiate between VC models is their structure. Inspired by image style transfer in computer vision, variational autoencoders (VAEs) and generative adversarial networks (GANs) were used in VC. In this work, we focused on four state-of-the art voice conversion methods. The models were selected mainly for their ability to work with little data from the target speaker, focusing on many-to-many or any-to-any conversions, and/or high-quality audio output.

3 Methods

For the selection of the four VC models, the state-of-the-art of VC models were viewed. The RVC method was selected as the most well-known publicly available method. FreeVC and QuickVC have a similar structure, also similar to RVC, which is why they were chosen. Deviating from this, kNN-VC employs a distinct structural configuration, yet has demonstrated promising outcomes with limited data from the target speaker. In the following, the selected VC methods are introduced in more detail.

kNN-VC[1] [1], standing for k-nearest neighbours voice conversion, uses text-free speech representation disentanglement and consists of an encoder-converter-vocoder structure. kNN-VC extracts self-supervised representations of the source and target speech using WavLM [2]. To convert the source utterance to the target speaker, each frame of the source representation is replaced with its nearest neighbour in the target, whereby the average of its k-nearest neighbours in the matching set, created from the target speakers utterances, is calculated. One advantage of this model is that the kNN regression algorithm requires no training. Instead, different values for k, therefore different amounts of neighbours from the target speaker set, need to be tested to find the most optimal k. The authors obtained good and robust results at k=4. However, they also mentioned that with more reference audio (>= 10 minutes) a larger value for k (around k=10,20) is recommended to improve the quality. Therefore, we tested kNN-VC with k=4, 10, and 20. As vocoder, HiFi-GAN is used and adapted to take self-supervised features as input. For this, HiFi-GAN is trained on the LibriSpeech train-clean-100 dataset [15], which consists of 40 English speakers.

FreeVC[2] [7] adopts the end-to-end framework of the TTS method VITS [5] (a conditional VAE augmented with GAN training) for high-quality waveform reconstruction [18], but learns to disentangle content information without the need of text annotation. FreeVC uses, as kNN-VC, the pre-trained WavLM [2] model to extract linguistic features from the waveform and disentangles content information by imposing an information bottleneck to WavLM features, to

[1] https://github.com/bshall/knn-vc

[2] https://github.com/OlaWod/FreeVC

extract the content information and removing speaker information. Furthermore, spectrogram-resize-based (SR) data augmentation was proposed, which distorts speaker information without changing content information, to strengthen the disentanglement ability of the model. With this, instead of tuning the bottleneck size, SR-based data augmentation is used to help the model learn to extract the clean content information by distorting speaker information in the source waveform. A pre-trained speaker encoder, adopted from [10], was used for the target speakers encoding. During training, a discriminator is used to classify the created speech recordings as real or fake. FreeVC was trained on the VCTK dataset with 107 English speakers, the LibriTTS dataset (English) [24] was used for testing. Again, HiFi-GAN was used as vocoder.

QuickVC[3] [3] is, as FreeVC, based on VITS. Unlike the original VITS model, QuickVC uses the inverse short-time Fourier transform (iSTFT) as part of the decoder to speed up the inference, and HuBERT-Soft [4] as part of the prior encoder to extract content information features, eliminating the need for text transcription. Speaker embeddings are extracted by a speaker encoder which is trained from scratch with the rest of the model. As with FreeVC, QuickVC performs SR data augmentation on the speech in the training dataset so that the content encoder learns to better extract the content information. For training, the VCTK dataset was used (107 English speakers).

RVC[4] is similarly structured as FreeVC and QuickVC, but is known not through academia but through social media and discord server such as AIHub[5]. AIHub Discord is one of many online presences, where people are meeting and discussing novel methods for creating audio deepfakes. Originally, the server started with So-Vits-SVC[6] and then introduced RVC[7], which is a newer AI approach which usually produces audio recordings with the same or higher quality than So-Vits-SVC, but needs less training time. Retrieval-based Voice Conversion (RVC) is, again, based on VITS, but unlike FreeVC, RVC uses HuBERT for embedding. According to its GitHub repository, RVC needs around 10 minutes to 1 hour of high-quality clear voice recordings (no background noise or instrumental parts) for transformation. One can train an own voice model or use a pretrained one. Pre-trained models can be found on rvc-models.com (e.g. Donald Trump, Joe Biden, Vladimir Putin), AIHub Discord, and Huggingface. We used the pre-trained model (v2; f0G40k) provided by the creators of RVC, pre-trained on nearly 50 hours of audio from the VCTK dataset, and fine-tuned it on the respective speaker.

[3] https://github.com/quickvc/QuickVC-VoiceConversion

[4] https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI

[5] https://discord.com/invite/aihub

[6] https://github.com/svc-develop-team/so-vits-svc

[7] https://github.com/Mangio621/Mangio-RVC-Fork

4 Experiments

4.1 Experimental Setup

The experiments were performed in three different settings, for an overview of these settings see Figure 1. For all experiments, the officially provided implementation of each VC model was used (see footnotes in Section 3).

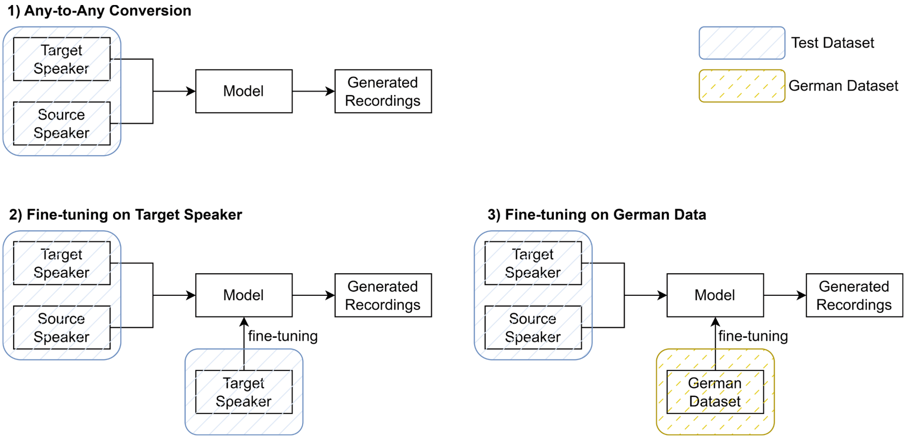


Fig. 1. The processes of the three conducted experiments. As models, kNN-VC, FreeVC, QuickVC, and RVC were used.

1) Any-to-Any Conversion First, the pre-trained models were used as they were available on the respective GitHub repositories. These models were applied on the test set (see Section 4.3) for record generation using three of the four VC models introduced in Section 3. As for RVC, the model has to be trained on the target speaker, therefore, no results for RVC are given in this setting. As all models were trained on English data, we are sure that the target and source speaker were not seen during the training of the used models. Therefore, the conversion we performed is called any-to-any conversion. To evaluate the effect of the amount of target speaker recordings used, different amounts were tested, ranging from 10 seconds to 2400 seconds. As source speaker, the same speaker was used for all experiments.

2) Fine-tuning on Target Speaker Secondly, the pre-trained models were fine-tuned on each target speaker for up to 1000 epochs. Therefore, in this setting, the target speakers are seen during model training. From the target speaker recordings, 180, 300, and 900 seconds were used for fine-tuning. After fine-tuning, the models were used to generate recordings. For kNN-VC, to evaluate the effect of different values of k, for two target speakers the models were fine-tuned with setting k=4, k=10, and k=20.

3) Fine-tuning on German Data In the last setting, the pre-trained models were fine-tuned on a German dataset for 100 and 700 epochs. With this, we tried to find out if the results improve if the models were trained on until then unseen German data, before performing any-to-any conversion, as done in the first experiment. Again, only the results of FreeVC, QuickVC, and kNN-VC are given. For the training, five hours of German audio recordings were used (see Section 4.3). As we found that as much data as 2400 seconds of the target speaker is not necessary to achieve good results, when fine-tuning the models on German data we calculated results using 10 to 900 seconds of target speaker data.

4.2 Evaluation Metrics

While evaluating, we focussed on two scores, one for the quality of the recording and the other examining the similarity of the generated recording to the target speaker. For evaluating the quality of the generated audio recordings, UTMOS [17], which achieved the first place at the VoiceMOS Challenge 2022, was used. For evaluating the similarity to the target speaker, the widely used voice similarity metric provided by resemblyzer[8] was applied, taking the generated audio recording and a 5-minute recording of the respective target speaker as input.

4.3 Datasets

As dataset, the spoken wikipedia corpora (SWC)[6] was used. The SWC corpora contains hundreds of hours of audio recordings in English, German, and Dutch from divers readers about different topics. We used the German partition for creating two subsets. One contains target and source speaker for testing. The second dataset was used for fine-tuning the models in experiment three. As in the SWC corpora each article is read only by one speaker, all recordings used contained different content.

Test Dataset In the test set, records of five persons (P1-5), 3 men and 2 women, were used as target speakers. As source speaker (S), we used the first 100 seconds of the recording named 3D from the SWC Corpora (male). In Table 1 the test set, with the five target speakers and one source speaker, is introduced. We selected target speakers with recordings in different quality. By doing this we wanted to evaluate the effect of the quality of the target speakers recording on the generated samples. The quality of the target speaker recording is given in form of the UTMOS score and a subjective evaluation in Table 1. The target speaker P1, which is a clear recording with little background noise or static, has the best quality, while P2 was the worst, with noise and a voice playing in the background[9].

German Dataset In the experiment with the third setting, the models are fine-tuned using German data. For this, another dataset was created with

[8] https://github.com/resemble-ai/Resemblyzer

[9] Listen to the recordings: https://nats.gitlab.io/swc/

Table 1. Test Dataset with 5 Target Speakers (P1-5) and 1 Source Speaker (S).

ID	Article name	Speaker	Gender	UTMOS	Subjective evaluation
S	3D	RBEReader	male	2.86	Very clear.
P1	Atom	Anhezu	male	**2.76**	Very clear, no background noise.
P2	Aeneis	Die keimzelle	male	**1.32**	Recording plays in the background, slight noise.
P3	Angriff auf Pearl Harbor	Blik Blik	male	2.35	Clear (swallowing at the beginning); slightly worse than P1.
P4	Geschichte der Juden in Braunschweig	Steffi Bütger	female	1.93	Clear, but not as clear as P1.
P5	Hanseat	Brigitte Trübenbach	female	2.11	Very clear, minor background noise, somewhat singing voice.

recordings of three persons. This dataset contains 5 hours of German recordings, with one voice (P3) shared with the test dataset. All six recordings used for fine-tuning on German data were of medium to good quality according to UTMOS, and were therefore included in this study. In both, the test and fine-tuning datasets, the first 20-30 seconds of the recording were removed to exclude the introduction of the SWC corpora, which always contains the same content.

5 Results and Analysis

The following section presents the results of the experiments conducted in the three settings, along with an overall comparison of the results. The results are presented in the form of graphs, with the amount of target speaker data used, in seconds, on the x-axis and the quality or similarity score on the y-axis.

5.1 Any-to-Any Conversion

For RVC the model has to be trained on each speaker, therefore, only the results of kNN-VC (k=4), FreeVC, and QuickVC are given in Figure 2, containing the quality evaluation using UTMOS, and Figure 3, with the evaluation results of the similarity to the original target speaker. Table 2 shows the minimum, mean, and maximum values of UTMOS (quality) and similarity to the original target speaker, calculated over the target speaker lengths.

In Figure 2 one can see that for FreeVC and QuickVC, the quality does not increase much when giving more data of the target speaker as input. For kNN-VC, the quality increased slightly, with the amount of target speaker recordings used. Furthermore, Figure 2 and Table 2 show, that the best quality recordings for all five target speakers were generated using QuickVC.

When only viewing the results of kNN-VC, on target speaker P1 the best quality results were obtained (min: 2.04; mean: 2.58; max: 2.71), compared to kNN-VC on the other target speakers. On target speaker P2 the worst quality results were reached (min: 1.33; mean: 1.71; max: 1.97), which was to be expected as P1 has the best quality input and P2 the worst. Similarly, when viewing FreeVC, the best minimum and mean value is obtained on P1, the best maximum

Table 2. VC Models in any-to-any Conversion, evaluated based on Quality (qual) and Similarity (sim). With $^+$, P1 is marked as the target speaker with the best quality audio recordings as input and P2 with $^-$ as the one with the worst quality. Best values for each target speaker are bolded. The best scores per model are highlighted in dark grey, the worst in light grey.

	kNN-VC; k=4					FreeVC					QuickVC				
	$P1^+$	$P2^-$	P3	P4	P5	$P1^+$	$P2^-$	P3	P4	P5	$P1^+$	$P2^-$	P3	P4	P5
min_{qual}	2.04	1.33	1.64	1.56	1.49	2.85	**2.57**	2.68	2.57	2.56	**2.91**	2.41	**3.02**	**2.77**	**2.56**
$mean_{qual}$	2.58	1.71	2.51	2.12	2.19	2.95	2.66	2.93	2.70	2.69	**2.98**	**2.84**	**3.28**	**2.89**	**2.76**
max_{qual}	2.71	1.97	2.74	2.28	2.41	3.03	2.74	3.09	2.82	2.78	**3.07**	**3.12**	**3.46**	**3.03**	**2.96**
min_{sim}	**0.89**	**0.87**	**0.83**	**0.90**	**0.87**	0.84	0.72	0.78	0.76	0.72	0.67	0.56	0.67	0.68	0.66
$mean_{sim}$	**0.92**	**0.88**	**0.87**	**0.91**	**0.89**	0.84	0.74	0.80	0.78	0.73	0.69	0.57	0.69	0.70	0.69
max_{sim}	**0.93**	**0.89**	**0.88**	**0.92**	**0.90**	0.85	0.75	0.81	0.79	0.74	0.70	0.63	0.71	0.71	0.70

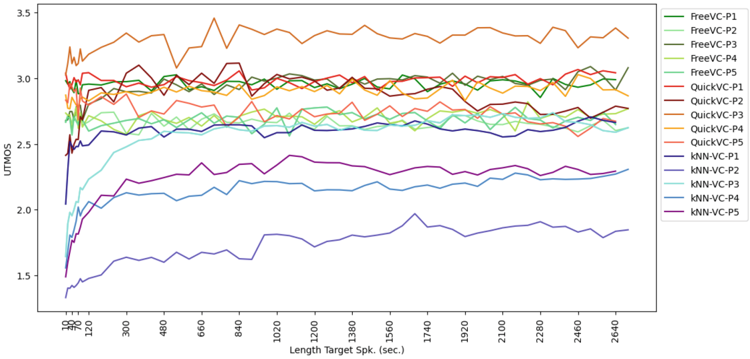


Fig. 2. Results for Quality (UTMOS) of Any-to-Any Conversion (Setting 1) with varying target speaker length.

value on P3. On P2 the worst mean and maximum value is reached, but with comparatively smaller differences than with kNN-VC (see Table 2). For QuickVC in Table 2, the best quality results can be viewed on P3. Also, P2 does not stand out with particularly poor quality results. This indicates that the quality of the target speaker input is not that important for QuickVC (and FreeVC), at least when measuring the output quality. Moreover, the overall quality results of FreeVC and QuickVC were better than them of kNN-VC.

In Figure 3 the results for the similarity evaluation are given. When viewing the similarity scores, contrary to the quality evaluation, with kNN-VC the best scores on all five target speakers were reached (see Table 2). Again, when viewing Figure 3, the scores do not improve much with more target speaker data. In this setting, overall, only slight improvements were obtained when using more target speaker data. For FreeVC and kNN-VC (Table 2), the best similarity scores were obtained on P1, which could be due to the high-quality recordings. However, different from previous observation, the worst similarity scores were obtained on P3 (min:0.83; mean:0.87; max:0.88) and not on P2. When using QuickVC, however, on target speaker P2, the similarity scores were much worse (see Table 2) and even decreased further with the use of more recordings from the

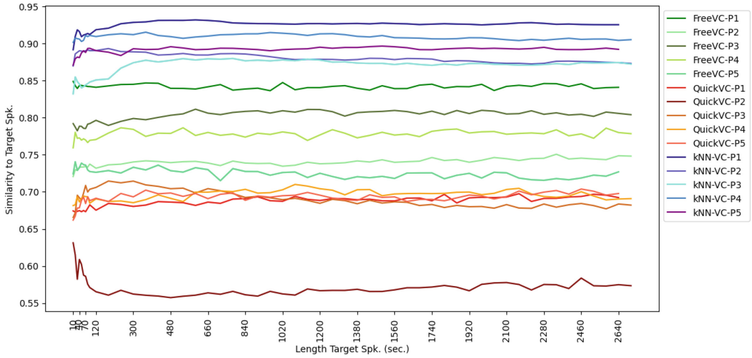


Fig. 3. Results for Similarity (Resemblyzer) of Any-to-Any Conversion (Setting 1) with varying target speaker length.

target speaker (Figure 3). This indicates that the quality of the target speaker recordings could have an impact on the similarity of the generated recordings similarity when using QuickVC in the any-to-any setting, but not kNN-VC. Thus, kNN-VC may be less dependent on the quality of the recordings.

As no improvements were visible after using 900 seconds of target speaker data, only the kNN-VC results with k=10, and k=20 for up to 900 seconds of target speaker data were used for calculating the minimum, mean and maximum value in Table 3. A higher k slightly improved the quality results of kNN-VC on target speaker P1. On the rest of the target speaker, the best minimum, mean, and maximum values over different target speaker lengths were obtained when using k=4 and k=10. Also, the best similarity score for each target person didn't improve much with a higher k. Again, the best minimum, partially mean (exception: P2), and maximum values of all five target speaker were reached when setting k=4.

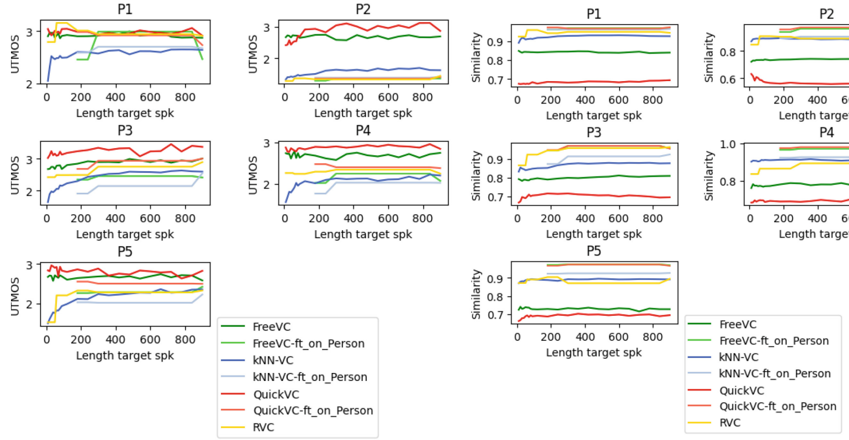
Due to space limitations, no figure is provided, but when visualizing the amount of target speaker data and the resulting UTMOS by different k's, as suggested by the authors of kNN-VC, a higher k and more data resulted in a slightly better quality, however overall k=4 achieved best results (Table 3).

Table 3. kNN-VC with k=4/10/20 in any-to-any Conversion, evaluated based on Quality (qual) and Similarity (sim). Best values for each target speaker are bolded.

	$P1^+$	$P2^-$	P3	P4	P5
min_{qual}	**2.04**/1.99/1.84	**1.33**/**1.33**/1.31	**1.64**/1.58/1.44	1.56/**1.57**/1.48	**1.49**/1.46/1.40
$mean_{qual}$	2.58/**2.61**/**2.61**	**1.71**/1.60/1.57	**2.51**/2.43/2.41	**2.12**/2.10/2.10	**2.19**/2.13/2.13
max_{qual}	2.71/2.77/**2.79**	**1.97**/1.80/1.78	2.74/2.79/**2.81**	2.28/2.28/**2.32**	2.41/2.46/**2.47**
min_{sim}	**0.89**/0.87/0.83	**0.87**/0.85/0.83	**0.83**/0.80/0.77	**0.90**/0.88/0.85	**0.87**/0.84/0.81
$mean_{sim}$	**0.92**/**0.92**/0.91	0.88/**0.89**/0.88	**0.87**/0.86/0.85	**0.91**/**0.91**/0.90	**0.89**/0.88/0.87
max_{sim}	**0.93**/**0.93**/**0.93**	**0.89**/**0.89**/**0.89**	**0.88**/**0.88**/**0.88**	**0.92**/**0.92**/0.91	**0.90**/0.89/0.89

5.2 Fine-tuned on Target Speaker

In Figure 4 the results for the quality evaluation and the similarity scores of the VC models when fine-tuned on the respective target speaker (ft_on_Person) and without are displayed. Again, only slight improvements were achieved when using more than approximately 300 seconds of target speaker data. In Table 4 the results of any-to-any conversion are compared to the results when fine-tuning the model on the target speaker.



(a) Quality (UTMOS)

(b) Similarity (Resemblyzer)

Fig. 4. Results for all Models with/without Fine-tuned on the respective Target Speaker (Setting 2). Abbreviation: ft_on_Person:fine-tuned on the respective speaker.

Among the fine-tuned models, Quick VC had the best quality for the target speakers P1, P3, P4, and P5. Considering each value instead of the mean values, the quality of the generated recordings deteriorated when using the model fine-tuned instead of the models without fine-tuning (see Figure 4). The best quality scores are reached without fine-tuning using QuickVC (see Table 4). The best similarity scores were obtained with QuickVC fine-tuned on the target speaker, and for P1, P3, and P5, FreeVC received same scores (0.97), see Table 4. Thus, the similarity scores increased significantly with FreeVC and QuickVC fine-tuned on the respective speaker. However, for quality, higher scores were obtained without fine-tuning (see FreeVC and QuickVC in Table 4). Overall, QuickVC showed a superior performance of similarity and quality on the five target speakers (see bold in Table 4).

RVC, which we were only able to evaluate in this setting, achieved comparably lower scores, followed by fine-tuned kNN-VC in terms of similarity. With RVC, similarity scores of 0.87 (P4) to 0.94 (P1) were calculated, which were all

Table 4. VC Models in any-to-any Conversion vs. fine-tuned on the Target Speaker. Best values for each target speaker are in bold. The best scores per model are highlighted in dark grey, the worst in light grey.

	Quality									
	no fine-tuning (mean)					ft_on_Person (mean)				
	P1$^+$	P2$^-$	P3	P4	P5	P1$^+$	P2$^-$	P3	P4	P5
kNN-VC; k = 4	2.58	1.71	2.51	2.12	2.19	2.68	1.39	2.14	1.98	2.03
FreeVC	2.95	2.66	2.93	2.70	2.69	2.86	1.36	2.44	2.19	2.29
QuickVC	**2.98**	**2.84**	**3.28**	**2.89**	**2.76**	2.91	1.36	2.90	2.41	2.50
RVC	-	-	-	-	-	2.96	1.35	2.61	2.27	2.13

	Similarity									
	no fine-tuning (mean)					ft_on_Person (mean)				
	P1$^+$	P2$^-$	P3	P4	P5	P1$^+$	P2$^-$	P3	P4	P5
kNN-VC; k = 4	0.92	0.88	0.87	0.91	0.89	0.96	0.90	0.91	0.93	0.93
FreeVC	0.84	0.74	0.80	0.78	0.73	**0.97**	0.96	**0.97**	0.97	**0.97**
QuickVC	0.69	0.57	0.69	0.70	0.69	**0.97**	**0.97**	**0.97**	**0.98**	**0.97**
RVC	-	-	-	-	-	0.94	0.89	0.93	0.87	0.89

lower than the ones of the other models. Regarding the quality, RVC was comparable with a UTMOS score of 2.96 on P1 which was even better than FreeVC and kNN-VC but not as good as QuickVC.

The best quality and similarity scores per model were reached on P1 when using the fine-tuned models. Also, the worst quality scores were, for all models, on target speaker P2 (see Table 4). This shows a possible effect of the quality of the used target speaker recordings, being the best for P1 and the worst for P2.

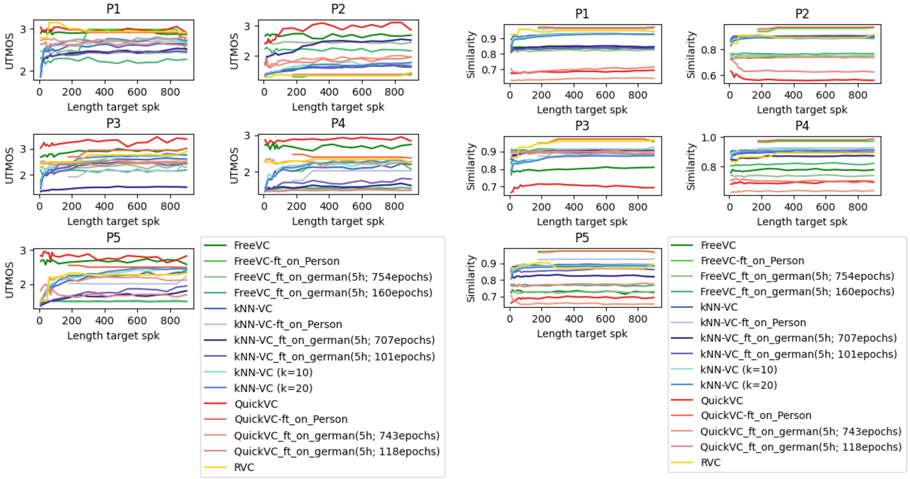
A higher k in kNN-VC fine-tuned on the respective target speaker, had only slight effects on the results and were therefore excluded from Figure 4 for better readability. Considering the similarity, for k=4, 10, and 20 no or only slight improvements of 0.01 points were viewed. The quality scores were slightly higher with a higher k. For example, for target speaker P2 with k=4 an UTMOS score of 1.38, with k=10 a score of 1.43, and with k=20 a score of 1.45 was reached. The scores were only marginal improved with varying k.

5.3 Fine-tuned on German Data

In Figure 5 the quality and the similarity scores for all settings are displayed. With ft_on_german the results of the models when fine-tuned on the German dataset are given. It should be noted that recordings of P3 are present in the German dataset used for fine-tuning, as mentioned in Section 4.3.

When fine-tuning the models on the German dataset, all results of kNN-VC are with setting k=4, as no substantial improvements were observed with a higher k. The models were fine-tuned for 100 and 700 epochs. Results for both numbers of training epochs are given in Figure 5. One can see that the results do not improve with a higher number of training epochs (100 to 700). One exception is target speaker P4 when using QuickVC fine-tuned on the German dataset, here the quality improved by 0.78 on average when fine-tuning for 743 epochs (mean: 118 epochs: 1.48; 743 epochs: 2.26). For the rest of models and target speakers the quality and similarity did not improve. Interestingly, the similarity results of FreeVC (ft_on_german) on target speaker P4 dropped by 0.07 on average when trained for more epochs (mean: 160 epochs: 0.81; 754 epochs: 0.74). Additionally, one cannot see a particular overall quality improvement in P3, despite the presence of samples of target speaker P3, in addition to other speakers, in the German training set.

Among the models fined-tuned on the German dataset, FreeVC and QuickVC have the best quality scores, and they were achieved with only 10-20 seconds of target speaker input. In terms of similarity, the best results of FreeVC and QuickVC fine-tuned on the German dataset, are, all achieved when using 10-60 seconds of target speaker input. This shows, again, that a large amount of data of the target speaker is not always required. QuickVC fine-tuned on the German dataset has the best quality results for four of the five speakers, among other models fine-tuned on the German dataset. When using kNN-VC, fine-tuned on the German dataset, the best similarity scores for four of the five speakers were achieved.



(a) Quality (UTMOS)

(b) Similarity (Resemblyzer)

Fig. 5. Results for all Models on the five Target Speakers in all three Settings. Abbreviation: ft_on_Person:Fine-tuned on the respective speaker; ft_on_german:Fine-tuned on the German dataset.

5.4 Overall Comparison

One can see that for all five target speakers the best similarity scores were reached when fine-tuning the model on the target speaker. Whereby, the fine-tuning on the target speaker was particular beneficial for FreeVC and QuickVC. Fine-tuning kNN-VC on the target speaker improved the similarity scores slightly.

When viewing the results for the quality evaluation, the quality of the results of FreeVC and QuickVC fine-tuned on the target speaker are lower than when using the models as they are (any-to-any conversion). Overall, the similarity does not change much for all models and target speakers after giving 100-200 seconds of target speaker data as input.

As with fine-tuning the models on the German dataset, the results differentiate much, depending on the target speaker and model viewed. For QuickVC, the similarity scores were improved on P2, P3, and P5 when fine-tuning QuickVC on the German dataset. For P1 and P4 (and P5 fine-tuning for 743 epochs) the similarity results deteriorated. Similarly, for FreeVC, on P3, P4, and P5 the results could slightly be improved over them of the basic model with fine-tuning on the German dataset. For both models, the best similarity scores are still reached with fine-tuning on the target speaker. For kNN-VC, the similarity worsens for four of the target speakers when fine-tuning on the German dataset, the exception being P3, which is probably because of the presence of the target speaker in the German dataset. For all five target speakers, the quality of FreeVC and QuickVC fine-tuned on the German dataset worsened compared to the basic model. The same applies to kNN-VC, except for P2, where the quality improved with the fine-tuning on the German dataset for 707 epochs.

6 Comparison with English Results

In their respective papers, kNN-VC, FreeVC and QuickVC were evaluated using subjective evaluation to measure naturalness and similarity on English datasets. This involved calculating a 5-scale mean opinion score (MOS) and a similarity mean opinion score (SMOS). Additionally, the authors of QuickVC used, as us, Resemblyzer to calculate the speaker similarity. For kNN-VC, in the paper, only results are presented for the any-to-any conversion scenario. For QuickVC, only results for the model fine-tuned on the speaker are given, and therefore presented in Table 5. The results in German in Table 5 are taken as the mean over the five speakers presented in Table 4. It is not possible to provide ratings for RVC in Table 5, as no scientific publications or ratings are available.

Table 5. Results given in the paper on English Data vs. our results in German. Abbreviations: aa: any-to-any conversion; ft: fine-tuned on target speaker.

Model	Paper Results - English			Ours - German	
	Datasets used	MOS	SMOS	Quality (UTMOS)	Similarity (resemblyzer)
kNN-VC	LibriSpeech test-clean	aa: 4.03 ± 0.08	**aa:** 2.91 ± 0.11	aa:2.22 ft: 2.04	**aa: 0.89** ft: 0.93
FreeVC	LibriTTS	**aa:** 4.06 ± 0.08 ft: 3.99 ± 0.09	aa: 2.83 ± 0.08 **ft:** 3.80 ± 0.09	**aa: 2.79** ft: 2.22	aa: 0.78 **ft: 0.97**
QuickVC	LibriSpeech, LJ Speech	**ft:** 4.28 ± 0.15	ft: 3.58 ± 0.20 (resemblyzer: 0.86)	**ft: 2.42**	**ft: 0.97**

The results between the scores in the papers (obtained on English data) and ours differed significantly. This discrepancy can be attributed to the different languages used. But, the divergence in results may also be attributed to the use of different datasets and evaluation metrics. In the aforementioned papers, subjective evaluations such as MOS and SMOS were employed, which may diverge from objective measures such as UTMOS and the similarity calculated by Resemblyzer used by us. Consequently, the results of the comparisons should be interpreted with caution. In general, it can be stated that in the any-to-any condition, viewing kNN-VC and FreeVC, FreeVC has the superior MOS/UTMOS and kNN-VC the superior SMOS/Similarity Scores in English and German. Upon examination of the results of the fine-tuned models, FreeVC and QuickVC, it was found that QuickVC exhibited superior MOS/UTMOS scores, while FreeVC (in the German setting, in conjunction with QuickVC) demonstrated superior SMOS/Similarity Scores. The authors of QuickVC [3] employed, as us, Resemblyzer for evaluation. Comparing the English and German results, our results achieved a higher score of 0.97 than the result presented in the original paper on English data (0.86). In general, when comparing English and German results based on UTMOS and MOS (both of which provide values on a 5-point scale), the English results were considerably more favourable. However, as Resemblyzer provides a value between 0 and 1, it is not possible to make a direct comparison in terms of similarity. This preliminary analysis indicates that the models exhibit similarities when compared with each other using English and German data. However, it also reveals significant discrepancies in the performance values of each model when contrasting English and German samples.

7 Conclusion

Different settings for FreeVC, QuickVC, kNN-VC, and RVC were analysed. We showed, that when using the models as they are, QuickVC achieved the best quality results and kNN-VC the best similarity scores. Whereby, only a few minutes of target speaker input is needed for acceptable results. When fine-tuning the models on the target speaker, with QuickVC the best quality and similarity scores for four of the five target speakers were achieved. Whereas, the quality decreased with the fine-tuning process and the similarity increased. On the other hand, with FreeVC comparably good similarity scores were achieved when fine-tuned on the target speaker and the results of kNN-VC were just slightly improved. When using RVC, good quality recordings were generated, but with comparatively lower similarity to the target speakers. When fine-tuning on the German dataset, again, with kNN-VC the best similarity scores were achieved, and with QuickVC the best quality. Whereby, the results of the basic models were only partially improved through the fine-tuning on German data. For kNN-VC, different parameters of k were examined, however, no major differences in the results were observed, the default setting of k=4 still being the best with little data.

In addition, we analysed target speakers with varying quality recordings as input. The target speaker P1, is represented by higher quality recordings, while

P2 is represented by lower quality recordings. We found that the quality of the recordings used as input has an effect on the generated audio recording, but primarily on the similarity of the generated recording to the target speaker. The quality of the generated recording is only marginally effected by the quality of the recording used as input.

Overall, generating German recordings with VC models trained on English data, the results of all models were rather good achieving good results on both, quality and similarity. This is also reflected in our subjective analysis, when listening to the recordings, the recordings were convincing and didn't contain accents or strangely pronounced words. Models trained on English can also be used to generate convincing German recordings. Nevertheless, it was observed that there were differences in the quality and similarity of the generated recordings when English or German was used as input.

References

1. Baas, M., van Niekerk, B., Kamper, H.: Voice conversion with just nearest neighbors. In: Interspeech (2023)
2. Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., et al.: Wavlm: Large-scale self-supervised pre-training for full stack speech processing. IEEE Journal of Selected Topics in Signal Processing **16**(6), 1505–1518 (2022)
3. Guo, H., Liu, C., Ishi, C.T., Ishiguro, H.: Quickvc: Many-to-any voice conversion using inverse short-time fourier transform for faster conversion. arXiv preprint arXiv:2302.08296 (2023)
4. Hsu, W.N., Bolte, B., Tsai, Y.H.H., Lakhotia, K., Salakhutdinov, R., Mohamed, A.: Hubert: Self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Transactions on Audio, Speech, and Language Processing **29**, 3451–3460 (2021)
5. Kim, J., Kong, J., Son, J.: Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In: International Conference on Machine Learning. pp. 5530–5540. PMLR (2021)
6. Köhn, A., Stegen, F., Baumann, T.: Mining the spoken wikipedia for speech data and beyond. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). Paris, France (may 2016)
7. Li, J., Tu, W., Xiao, L.: Freevc: Towards high-quality text-free one-shot voice conversion. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023)
8. Li, J., Guo, Y., Chen, X., Yu, K.: Sef-vc: Speaker embedding free zero-shot voice conversion with cross attention. In: ICASSP 2024. pp. 12296–12300 (2024). https://doi.org/10.1109/ICASSP48485.2024.10446160
9. Lim, J., Kim, K.: Wav2vec-vc: Voice conversion via hidden representations of wav2vec 2.0. In: ICASSP 2024. pp. 10326–10330 (2024). https://doi.org/10.1109/ICASSP48485.2024.10447984
10. Liu, S., Cao, Y., Wang, D., Wu, X., Liu, X., Meng, H.: Any-to-many voice conversion with location-relative sequence-to-sequence modeling. IEEE/ACM Transactions on Audio, Speech, and Language Processing **29**, 1717–1728 (2021)

11. Lu, H., Wu, X., Guo, H., Liu, S., Wu, Z., Meng, H.: Unifying one-shot voice conversion and cloning with disentangled speech representations. In: ICASSP 2024. pp. 11141–11145 (2024). https://doi.org/10.1109/ICASSP48485.2024.10446296
12. Luo, Y.J., Dixon, S.: Posterior variance-parameterised gaussian dropout: Improving disentangled sequential autoencoders for zero-shot voice conversion. In: ICASSP 2024. pp. 11676–11680 (2024). https://doi.org/10.1109/ICASSP48485.2024.10447835
13. Ning, Z., Jiang, Y., Zhu, P., Wang, S., Yao, J., Xie, L., Bi, M.: Dualvc 2: Dynamic masked convolution for unified streaming and non-streaming voice conversion. In: ICASSP 2024. pp. 11106–11110 (2024).https://doi.org/10.1109/ICASSP48485.2024.10446229
14. Okamoto, T., Ohtani, Y., Toda, T., Kawai, H.: Convnext-tts and convnext-vc: Convnext-based fast end-to-end sequence-to-sequence text-to-speech and voice conversion. In: ICASSP 2024. pp. 12456–12460 (2024). https://doi.org/10.1109/ICASSP48485.2024.10446890
15. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech: an asr corpus based on public domain audio books. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 5206–5210. IEEE (2015)
16. Qi, T., Zheng, W., Lu, C., Zong, Y., Lian, H.: Pavits: Exploring prosody-aware vits for end-to-end emotional voice conversion. In: ICASSP 2024. pp. 12697–12701 (2024). https://doi.org/10.1109/ICASSP48485.2024.10446191
17. Saeki, T., Xin, D., Nakata, W., Koriyama, T., Takamichi, S., Saruwatari, H.: Utmos: Utokyo-sarulab system for voicemos challenge 2022. ArXiv **abs/2204.02152** (2022), https://api.semanticscholar.org/CorpusID:247957899
18. Walczyna, T., Piotrowski, Z.: Overview of voice conversion methods based on deep learning. Appl. Sci. **13**(5), 3100 (2023)
19. Wang, Y., Su, J., Finkelstein, A., Jin, Z.: Gr0: Self-supervised global representation learning for zero-shot voice conversion. In: ICASSP 2024. pp. 10786–10790 (2024). https://doi.org/10.1109/ICASSP48485.2024.10448232
20. Wenger, E., Bronckers, M., Cianfarani, C., Cryan, J., Sha, A., Zheng, H., Zhao, B.Y.: " hello, it's me": Deep learning-based speech synthesis attacks in the real world. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. pp. 235–251 (2021)
21. Yang, Y., Kartynnik, Y., Li, Y., Tang, J., Li, X., Sung, G., Grundmann, M.: Streamvc: Real-time low-latency voice conversion. In: ICASSP 2024. pp. 11016–11020 (2024). https://doi.org/10.1109/ICASSP48485.2024.10446863
22. Yao, J., Yang, Y., Lei, Y., Ning, Z., Hu, Y., Pan, Y., Yin, J., Zhou, H., Lu, H., Xie, L.: Promptvc: Flexible stylistic voice conversion in latent space driven by natural language prompts. In: ICASSP 2024. pp. 10571–10575 (2024). https://doi.org/10.1109/ICASSP48485.2024.10445804
23. You, C.H., Dong, M.: A study on combining non-parallel and parallel methodologies for mandarin-english cross-lingual voice conversion. In: ICASSP 2024. pp. 10491–10495 (2024). https://doi.org/10.1109/ICASSP48485.2024.10446264
24. Zen, H., Dang, V., Clark, R., Zhang, Y., Weiss, R.J., Jia, Y., Chen, Z., Wu, Y.: Libritts: A corpus derived from librispeech for text-to-speech. arXiv preprint arXiv:1904.02882 (2019)
25. Zhao, Y., Huang, W.C., Tian, X., Yamagishi, J., Das, R.K., Kinnunen, T., Ling, Z., Toda, T.: Voice conversion challenge 2020: Intra-lingual semi-parallel and cross-lingual voice conversion. arXiv preprint arXiv:2008.12527 (2020)

Investigating Swimming Effect of Hologram in Mixed Reality

Subin Raj(✉), B. R. Harshitha, Amaresh Chakrabarti, and Pradipta Biswas

Indian Institute of Science, Bangalore, India
subinp@iisc.ac.in

Abstract. The continuum of eXtended Reality Displays consists of Augmented Reality, Mixed Augmented Reality or Mixed Reality and Virtual Reality systems. Among these systems, mixed reality systems promise new set of applications as it combines both 3D rendering of virtual reality and real world mapping of augmented reality interfaces. Many mixed reality headsets like Microsoft HoloLens relies on holograms to render 3D imagery on real world objects. However, ensuring the stability of these holograms, popularly known an swimming effect, is crucial, as their position appears altered from the user's perspective when viewed through a Head-Mounted Display (HMD). This paper proposed a new way of tracking object in real time and analysis of *swimming effect* of the hologram using robot. A user study involving 10 users using 50 randomly chosen viewpoints found that the proposed method is as accurate as the state of the art World Locking Tool (WLT) with additional advantage of tracking dynamic objects.

Keywords: Extended Reality · Augmented Reality · Mixed Reality · Hologram Stability · Swimming effect

1 Introduction

Recently, XR has been explored in many areas such as manufacturing, healthcare, construction, education, and so on. XR encompasses Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR). VR is computer-generated graphics where the user can navigate and interact with virtual objects with their senses. The AR refers to overlaying computer-generated graphics onto the real world. The MR is the combination of the real environment and virtual environment where the user can interact with each environment. The difference between AR and MR is the AR can just project the virtual object in the real world. In contrast to AR, the MR can blend the real and virtual world together [6]. The International Standardization Organization defined mixed and augmented reality system as a system that uses a mixture of representations of physical world data and virtual world data as its presentation medium [ISO 18039].

Traditional Augmented Reality systems can broadly be classified into video see through and Optical see through systems. In video see through systems, an external camera captures the reality while in optical see through system, the user can see the reality directly. The optical see through systems do not compromise the quality of viewing

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 403–415, 2025.
https://doi.org/10.1007/978-3-031-78312-8_27

outside environment by resolution or limited field of view of a camera. Similarly, in mixed reality headsets, Meta Quest series of headsets and Ajna Lens uses cameras mounted outside the headset to capture environment while Microsoft HoloLens uses Holograms to directly render 3D imagery on real world objects. The International Standardization Organization defined hologram as interference pattern formed between the wave emitted from the object and its coherent reference wave, which is recorded in the recording material [ISO 17901].

Hologram based mixed reality systems found many applications in industrial automation, robotic teleoperation and so on. A set of use cases are described in section III of this paper. However, all such applications require stable rendering of holograms as well as synchronous updates with users' interaction. Change in the position of the hologram with respect to the user view will lead to difficulty in interaction. The term *swimming effect* refers to the position of the hologram appearing different depending on the user's viewpoint. We require holograms to remain still, maintaining a consistent position regardless of the viewing angle. However, it has been observed that there is a significant displacement in the position of holograms when viewed from different angles. This phenomenon can be attributed to Motion Parallax [Malla 2016]. When the viewpoint changes, various objects in the scene undergo different displacements based on their depth. Motion Parallax plays a crucial role in depth perception. Various methods have been introduced to reduce the swimming effect of the hologram – the most popular one is the World Locking Tool (WLT) discussed in details in next section. However, WLT requires to set up anchors before start of mixed reality interaction. There are use cases like tracking a drone or assembling components, where new components or objects may appear within the visual field of a mixed reality headset during interaction (runtime), which cannot be tracked and locked by WLT without halting the interaction.

In this paper, we proposed a new Computer Vision based method to track objects within the visual field of a MR headset in real time and also proposed a method to compare and analyze the swimming effect of the hologram with respect to operating a fixed base robot. We compared proposed CV based method with the WLT.

2 Literature Review

The AR and MR technologies enable users to manipulate the real world by interacting with the virtual world. In such cases, the stability of the virtual controls directly impacts the accuracy of manipulating real-world objects. Holloway et al. [1997] analysed the registration error of the HMD and identified several reasons for this error, including system delay, tracker error, calibration error, and optical distortion. As the viewpoint changes, the virtual object may appear misaligned with the real object, resulting in the swimming effect that affects the user's understanding of the real and virtual objects. The researchers discovered that even slight head movements could lead to significant registration errors due to system delay. A conventional method for registering holograms onto real objects involves three steps: first, calibrating the system parameters, then tracking the object to be augmented, and finally generating appropriate virtual content to overlay onto the real world. Zheng et al. [2013] proposed a closed-loop registration method to accurately register holograms onto the real world. This method involves comparing the target image with an image that consists of both real and virtual objects, allowing for precise registration of the virtual objects onto the real world.

Vassallo et al. [2017] investigated this stability in a clinical environment while using HoloLens. They found manageable drift in the holograms. However, this drift did not significantly impact the device's usability in clinical settings, as they attributed it to the device's simultaneous localization and mapping capabilities. Liu et al. [2018] focused on analysing hologram stability during head movement, discovering that the holograms drifted along with the user's head movement. For slow head movements, the deviation of the head tracking value compared to the reference value was 0.56 cm, and for fast head movements, it was 2.63 cm. Guinet et al. [2019] examined HoloLens head tracking accuracy using a MOCAP system. Their analysis revealed a root mean square error ranging from 9 to 53 mm between the two datasets. They also found that sudden changes in trajectory increased the tracking error, while variations in head movement speed did not significantly affect HoloLens's head tracking accuracy.

Holograms can be rendered on real objects with the help of markers. Gsaxner et al. [2019] proposed a markerless virtual object registration method on the real world to assist surgeons. They utilized image-to-face mapping to register the holograms. The proposed method detects the patient's face from the image and registers the virtual object. The mean registration error of the proposed method was noted to be 9.2mm. Sun et al. [2020] proposed a method to register holograms using a combination of optic tracker data points and HoloLens depth map data. Jiang et al. [2020] evaluated the HoloLens-based vascular localization system. They used a 3D model to render the hologram and noted the registration error to be a minimum of 1.35 mm and a maximum of 3.18 mm.

Krupke et al. [2019] introduced a taxonomy for classifying MR based Robot User Interfaces (MRHUI) into four categories based on their usage as interaction, mediation, perception, and acting. In a survey by Cheng et al. [2020], challenges in implementing MR technology across various applications were identified. The challenges encompassed spatial information accuracy, User Interface (UI) design, data storage and transfer, and multiuser collaboration. Weinmann et al. [2021] conducted an experiment to assess HoloLens spatial mapping accuracy and localization in indoor environments. Comparing laser scanners with HoloLens, they found laser scanners to produce more precise data. Nevertheless, HoloLens still exhibited mapping accuracy and localization within a few centimetres. In a comparative study by Soares et al. [2021] between HoloLens and HTC VIVE tracking, HoloLens demonstrated lower accuracy compared to HTC VIVE. Consequently, HoloLens is preferred for applications where slightly reduced accuracy is tolerable in exchange for better performance. Analysing HoloLens 2 spatial mapping capabilities in vast monumental heritage environments, Teruggi et al. [2022] observed that the spatial model produced by HoloLens showed negligible deviation in close range but increased linearly with distance when compared to reference data. The study also highlighted challenges with tracking objects in low-light conditions and difficulties in tracking single architectural structures and uniform surface patterns.

Through the literature review, several factors contributing to hologram registration error were identified, including system delay, tracking error, and calibration error. Additionally, the stability of the hologram is influenced by head movement, with the extent of drift dependent on the speed of the movement. To mitigate hologram drift, the use of anchors within the hologram and the application of the World Locking Tool (WLT) have shown promise. However, there has been a lack of comparative studies on the swimming effect of holograms using different methods. The position of the hologram appears to vary depending on the viewpoint, making hologram stability crucial for applications in Human-Robot Interaction (HRI) to define waypoints for robots. In this paper, we proposed a new way of analyzing the swimming effect of the hologram. We conduct an analysis of three methods for evaluating hologram stability: spatial anchor, World Locking Tool (WLT), and the base method, which involves rendering holograms without using anchors or WLT.

3 Applications

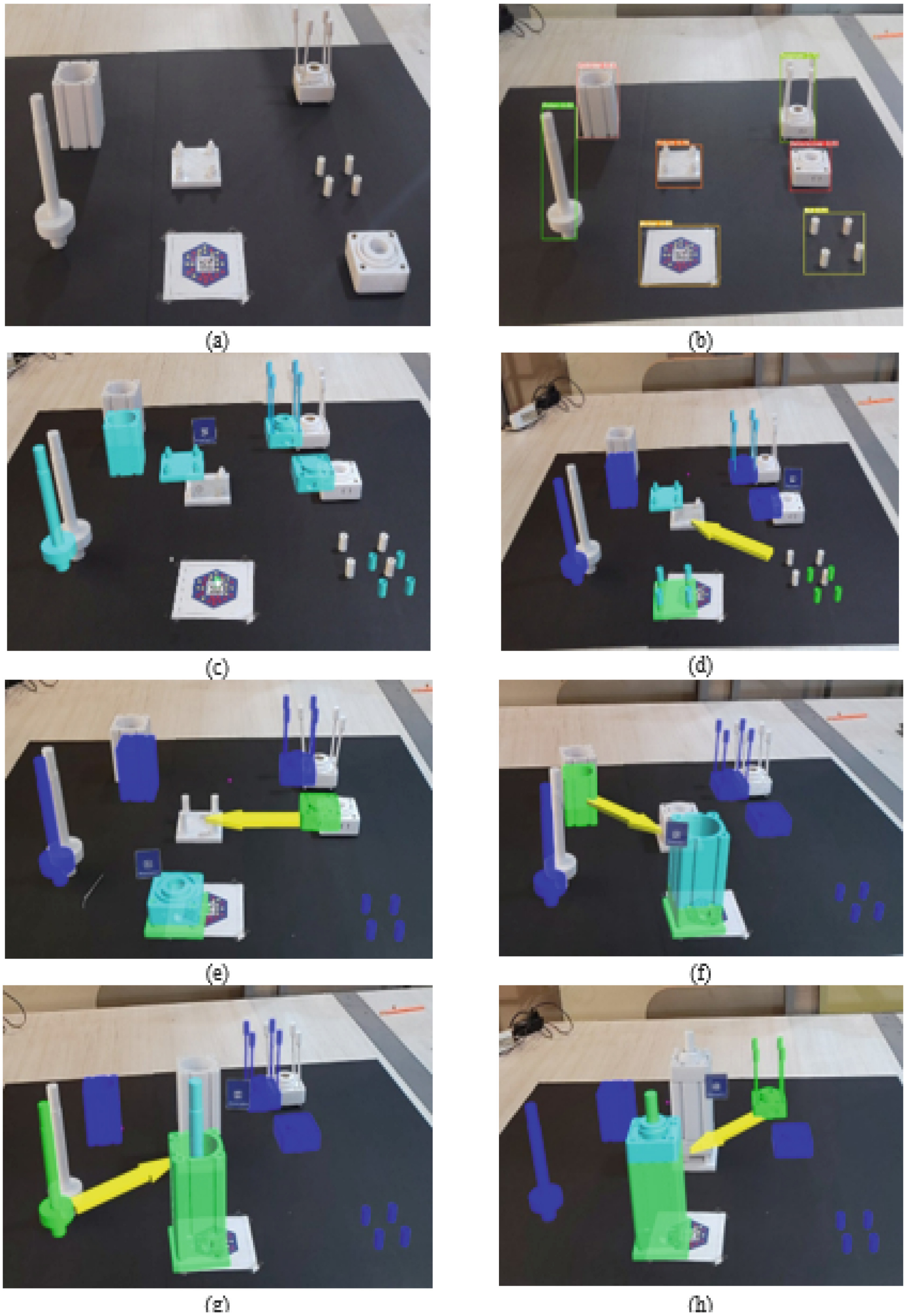
Mixed Reality systems found many applications in both research and industry. In this section, we have described a set of applications justifying the need for our subsequent study on stability of holograms.

We developed a MR based guidance system for factory shop floor workers to assist them in performing component assembly tasks obviating the need of repeated training and by not making any assumptions about the skill level of the worker (https://youtu.be/3ftXEYmZ21g). Assembly task is one of the most complex tasks performed by shop floor workers. A shop floor worker assembles various manufactured components in a structured manner during the assembly task leading to the final product. These components vary in size, geometry, weight and several parts might look similar with subtle variations in terms of geometry. A computer vision algorithm was integrated with MR to continuously map and track real objects and virtual instructions. Instructions were given to the user visually by changing color, animating a virtual hand grabbing, using audio, and displaying text messages. The multimodal user interface enabled the user to interact with the system through either voice commands or selecting buttons, reducing the complexity of interaction and allowing the user to choose a preferred interaction modality. Instructions about the next assembly step were provided to the user by changing the virtual replica's color of the real component and animating the virtual hand to pick up the right component. The animation of the assembly step allowed the user to learn the assembly step in real time and perform the assembly task. Multiple user studies confirmed users could undertake assembly tasks in the proposed method faster than a fully automated assembly and video based instruction [Raj 2023] (Fig. 1).

We extended this MR assisted manual assembly process towards a teleoperated assembly process, where users operated a robot from mixed reality environment (https://youtu.be/HeuO0WgQa0A). The robot followed a path defined through the mixed reality interface for picking up component and then placed it in appropriate location (Fig. 2) [Raj 2024].

The MR based path definition was further explored on developing a tele-operated welding assistant where the user defined an arc-welding path through his fingers and a tele-operated robot followed the exact path (Fig. 3, https://youtu.be/oZ6rcBwvrwM). We also undertook comparative studies between VR and MR assisted robotic welding system [Rao, 2023].

In all of these applications, the 3D hologram helped users to visualize and track the 6-DoF of virtual objects corresponding to real objects. It was also essential to anchor the



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 1. MR based Component Assembly System

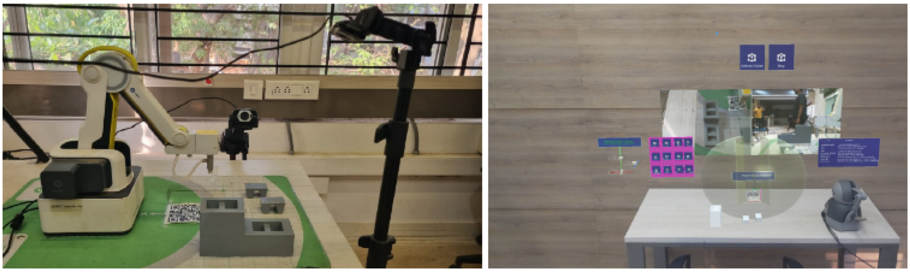


Fig. 2. MR based Robot Teleoperation

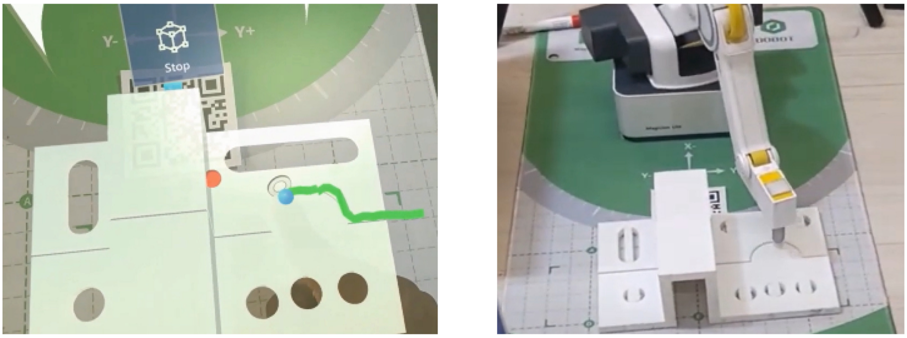


Fig. 3. TeleRobotics Welding Application

virtual object at a fixed offset from the real object as the visual field was often cluttered with multiple real and virtual objects based on the number of assembly or welding related components. The stability of holograms was essential for the user acceptance of each system. The subsequent sections present the anchoring tool and proposed study on quantitatively evaluating the stability of hologram using a robotic manipulator.

4 Computer Vision Based Tracking and Swimming Effect

MR is a user environment where physical reality and digital content are combined to enable interaction with and among real-world and virtual objects. Virtual instructions are registered on real objects by attaching markers. However, attaching and tracking markers on small and dynamic objects is challenging. We propose a CV based hologram registration method that uses image processing and machine learning techniques. This paper utilizes template matching, an image processing method, for object detection, and a regression-based mapping method to render holograms. We used opencv TM_CCOEFF_NORMED function for matching template. Template matching detects the position of objects in the real world. These position values are then sent to a regression model to predict real-world positions. To improve hologram registration accuracy, we placed four markers around the robot workspace, as shown in Fig. 4. The positions of these markers are detected from the input image and perspective transformation is applied to focus on the robot workspace, ensuring a consistent workspace regardless of the image capture position. The proposed hologram registering method does not depend on the model which we choose for detecting objects. For large number of components and cluster environment, we can choose the most suitable object detection model to detect the object such as Yolo, Faster RCNN, etc.

For training the regression model, pixel values in the image were randomly selected, and corresponding robot position values were obtained by manually moving the robot. Various regression models such as linear regression, polynomial regression, support vector regression, and random forest regression were trained with this data. We trained the model with 20 data points. The training error of different models is shown in Table 1. The linear regression model yielded superior results with an R-squared value of 0.998 compared to other models. To render the hologram, we capture an image of the robot workspace, apply perspective transformation to it, and use template matching to detect object positions in the transformed image, providing the center of the bounding box value. This bounding box value is then inputted into a regression model to predict the actual X and Y positions of the object in the real world. Finally, the system sends these X and Y values to the HMD, where they are used to render a sphere-shaped hologram with a diameter of 1 cm.

Table 1. Machine Learning Model Comparison

Model Name	**Mean Square Error (mm)**
Linear Regression	1.61
Random Forest Regression	208.51
Polynomial Regression	15.10
Support Vector Regression	834.88

The **WLT** anchors the hologram in the real world by continuously adjusting the head coordinate system. As a result, the user moves around the real world, the holograms stay precisely on the same position. The technology behind the WLT locks the entire holograph space of an application to the physical world. A hologram put in position relative to physical world features will stay fixed relative to those features, as well as remaining fixed relative to other holograms. When multiple users share the same world-locked environment, they can interact with the same virtual elements from different perspectives.

5 Evaluation

We evaluated the swimming effect of the hologram and the proposed hologram registration method in this section.

Experiment Setup

We developed an MR application to analyse the stability of holograms in the robot's workspace. We used HoloLens 2 to experience the MR environment. HoloLens2 is a standalone computer system that has a Qualcomm snapdragon 850 compute platform, and a second-generation custom build holographic processing unit. The robotic manipulator (Dobot Magician Lite) is used here to evaluate the swimming effect of the hologram. The robot has the following specifications, a maximum payload of 250g and a repeatability of ± 0.2mm. The application was created with the help of Unity and MRTK. After developing the application, it was built and deployed on HoloLens 2.

The MR-based application requires a common origin between the robot and MR application because both systems have different coordinate systems and origins. The marker (QR code) is an effective way to define the common origin for the robot and MR environment. We attached the QR code to the robot's origin, and then we rendered the hologram in the MR environment with respect to the marker, shown in Fig. 4. Now, we can define the position of the robot and hologram relative to the marker, as the marker acts as a common reference point for both the robot and hologram.

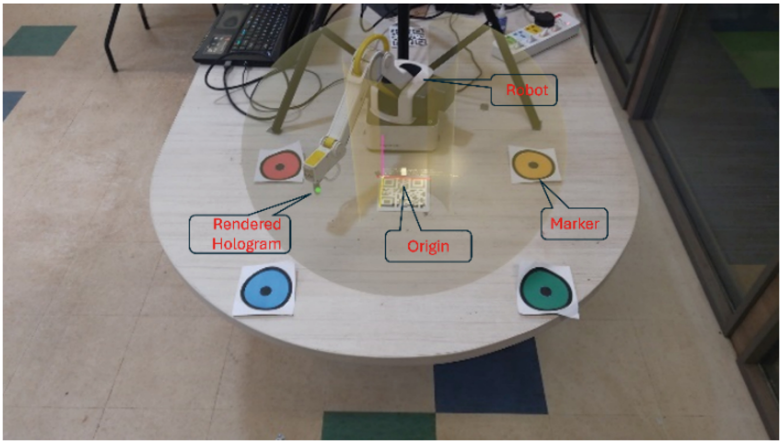


Fig. 4. Experiment Setup

User Study

We conducted a study to analyse the effect of swimming on robot path planning. We recruited 10 male participants with an average age of 25.8 years. We developed two applications in UNITY, a game engine, and deployed them on HoloLens2. In one application, we used the WLT method, while the other employed CV-based hologram rendering, which we did not use. In the WLT method, holograms were rendered randomly in the robot's workspace. For CV-based rendering, we detected randomly placed objects from input images and then rendered the hologram accordingly. Once the hologram was rendered, users manually moved the robot to its position. After aligning the robot end effector with the hologram, they clicked a virtual button to confirm alignment. We recorded both the robot end effector and hologram positions with respect to a marker. Once these positions were recorded, users moved to a different viewpoint and observed the hologram's position relative to the robot end effector. If the alignment was not satisfactory, users adjusted the robot's position to match the holograms. Otherwise, no movement was necessary. Users then recorded the position values. Similarly, users collected robot and hologram positions from 10 randomly chosen viewpoints. Additionally, users followed these steps to collect positions for 5 holograms placed at different locations in the robot's workspace. We collected data from a total of 10 users.

We analysed the data and calculated the perceived position error due to the swimming effect. The Root Mean Square Error (RMSE) due to the swimming effect between CV-based rendering and WLT is shown in Figs. 5 and 6. From the analysis, we found that the swimming effect of the hologram can impact robot path planning when placing holograms. We conducted the t test on analyzing the swimming effect. The statistical significance level was set at 5% (α = 0.05). There was no significant difference in the swimming effect between the proposed method and WLT in XYZ directions. The swimming effect of the hologram along the X axis in the proposed method compared to WLT. If we compare the swimming effect among all directions, the swimming effect in X direction is less compared to other directions. From the participant answer, they found that when they change viewpoint height, they felt more swimming effect. Therefore, the study suggests that the swimming effect of holograms should be considered while using the holograms to define the robot way points.

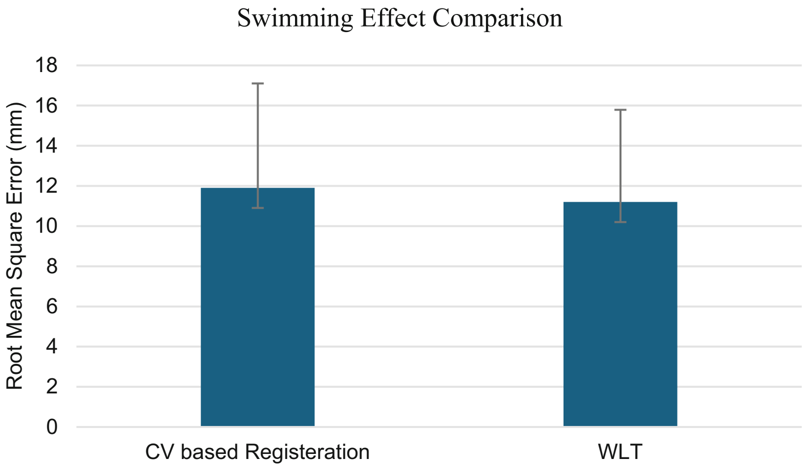


Fig. 5. Swimming effect comparison

We analyzed the discussed hologram registration method using different camera inputs: specifically, the chin camera and the external camera. We selected the HoloLens2 RGB camera as the chin camera and the Logi HD 1080p camera as the external camera, which is mounted above the robot workspace to capture images. The hologram rendering method remains consistent for both camera inputs. Next, objects were randomly placed in the robot workspace. Images of the workspace were captured from the chin camera and the external camera and then sent to the proposed method to render the hologram. The position of the object and the registered hologram with respect to the robot were recorded. We calculated the RMSE value between the chin camera and the external camera, as shown in Figs. 7 and 8. From the analysis, we found that the mapping error is lower with the external camera compared to the chin camera. We conducted the t test

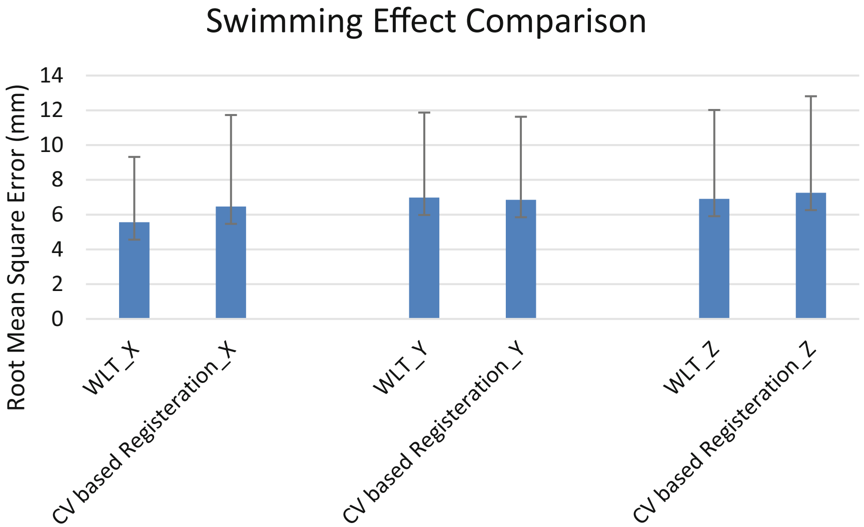


Fig. 6. Swimming Effect Comparison on each axis

to analyse the mapping error due to two cameras. The statistical significance level was set at 5% (α = 0.05). We found that there was no significant difference in the mapping error in X and Y direction. The increased mapping error along Y direction when using the chin camera results from changes in the capture pose and shape of the component, which affects the accurate prediction of drawing bounding boxes on objects and leads to registration errors. Conversely, the discrepancy between the predicted pixel position of the bounding box and the actual pixel position of the object occurs due to the object's shape. This mapping error can potentially be reduced by training the regression model by considering the centre of the bounding box.

Discussion

This paper introduces a novel approach to assess the swimming effect of holograms by aligning a robot end effector with the perceived hologram position. The state-of-the-art WLT method enhances tracking accuracy by adjusting the head's coordinate system but does not effectively mitigate the hologram's swimming effect during changes in viewpoint. This phenomenon has implications for the precision of robot path planning.

Additionally, the proposed CV-based hologram registration method successfully registers holograms without requiring markers on objects, making it suitable for environments with movement constraints. Analysis of hologram registration errors highlights the need to minimize mapping errors, which can be achieved through improved object detection models that focus on the center of bounding box values. While template matching was initially used for object detection in this study, employing advanced object detection models can enhance accuracy in capturing bounding box values for hologram registration.

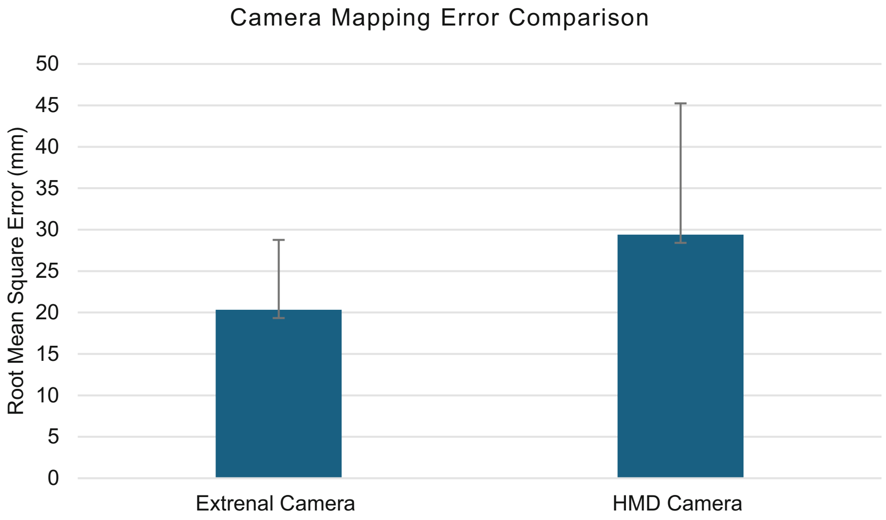


Fig. 7. Camera mapping error comparison

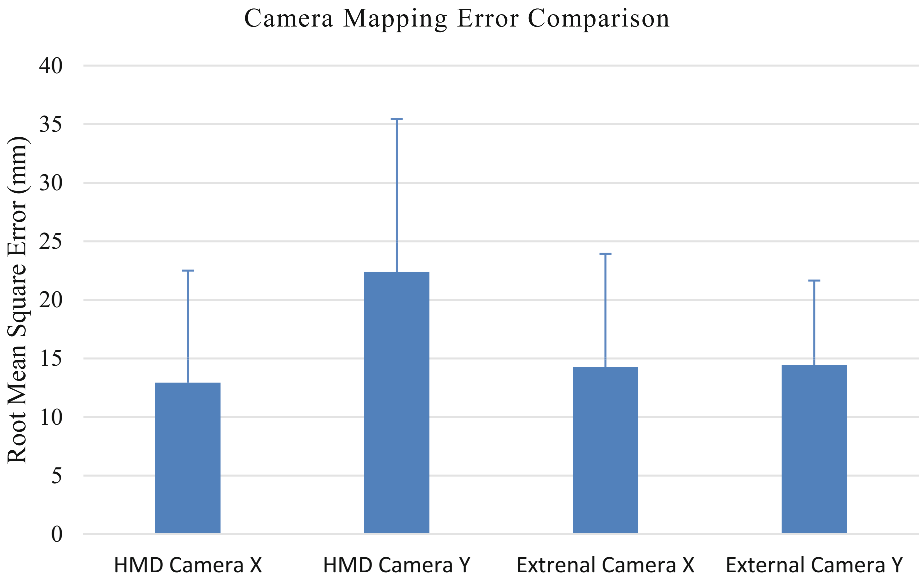


Fig. 8. Camera mapping error comparison about each axis

6 Conclusion

We introduced a novel method to analyze the swimming effect of holograms using a robot within a MR environment. This study compared the swimming effects of holograms between the WLT method and the proposed hologram registration method. The findings indicate that the swimming effect of holograms can significantly impact the definition of robot paths. Furthermore, we evaluated the registration accuracy of the proposed method using inputs from both the chin camera and an external camera. The results showed a reduction in hologram registration errors when using the external camera input. In future research, we plan to delve deeper into understanding the swimming effect of holograms by exploring additional variables such as head rotation and movement. Additionally, we enhanced mapping accuracy by incorporating shape of the object and image captured position as a factor in the regression model. This approach lays a foundation for analyzing hologram stability and hologram registration precision in MR applications.

References

1. Cheng, J. C., Chen, K., & Chen, W. (2020). State-of-the-art review on mixed reality applications in the AECO industry. *Journal of Construction Engineering and Management*, *146*(2), 03119009.
2. de la Malla, C., Buiteman, S., Otters, W., Smeets, J. B., & Brenner, E. (2016). How various aspects of motion parallax influence distance judgments, even when we think we are standing still. Journal of vision, 16(9), 8-8.
3. Gsaxner, C., Pepe, A., Wallner, J., Schmalstieg, D., & Egger, J. (2019). Markerless image-to-face registration for untethered augmented reality in head and neck surgery. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part V 22* (pp. 236–244). Springer International Publishing.
4. Guinet, A. L., Bouyer, G., Otmane, S., & Desailly, E. (2019). Reliability of the head tracking measured by Microsoft Hololens during different walking conditions. *Computer Methods in Biomechanics and Biomedical Engineering*, *22*(sup1), S169-S171.
5. Holloway, R. L. (1997). Registration error analysis for augmented reality. *Presence: Teleoperators & Virtual Environments*, *6*(4), 413–432.
6. ISO 17901–1:2015(en) Optics and photonics — Holography — Part 1: Methods of measuring diffraction efficiency and associated optical characteristics of holograms
7. ISO/IEC 18039:2019 Information technology — Computer graphics, image processing and environmental data representation — Mixed and augmented reality (MAR) reference model
8. Jiang, T., Yu, D., Wang, Y., Zan, T., Wang, S., & Li, Q. (2020). HoloLens-based vascular localization system: precision evaluation study with a three-dimensional printed model. *Journal of medical Internet research*, *22*(4), e16852.
9. Krupke, D., Zhang, J., & Steinicke, F. (2019). Impact: A holistic framework for mixed reality robotic user interface classification and design. *Multimodal Technologies and Interaction*, *3*(2), 25.
10. Liu, Y., Dong, H., Zhang, L., & El Saddik, A. (2018). Technical evaluation of HoloLens for multimedia: A first look. *IEEE MultiMedia*, *25*(4), 8-18.
11. Raj S., Murthy L.R.D., Shanmugam T.A., Kumar G., Chakrabarti A., Biswas P., Augmented reality and deep learning based system for assisting assembly process, (2024) Journal on Multimodal User Interfaces, 18 (1), pp. 119 – 133
12. Rao M.C.A., Raj S., Shah A.K., Harshitha B.R., Talawar N.R., Sharma V.K., Sanjana M., Vishwakarma H., Biswas P., Development and comparison studies of XR interfaces for path definition in remote welding scenarios (2024) Multimedia Tools and Applications, 83 (18), p. 55365 - 55404
13. Soares, I., B. Sousa, R., Petry, M., & Moreira, A. P. (2021). Accuracy and repeatability tests on HoloLens 2 and HTC Vive. *Multimodal Technologies and Interaction*, *5*(8), 47.

14. Sun, Q., Mai, Y., Yang, R., Ji, T., Jiang, X., & Chen, X. (2020). Fast and accurate online calibration of optical see-through head-mounted display for AR-based surgical navigation using Microsoft HoloLens. *International journal of computer assisted radiology and surgery*, *15*, 1907-1919.
15. Teruggi, S., & Fassi, F. (2021). Mixed Reality for the Monumental Heritage. A First Test. In *Proceedings ARQUEOLÓGICA 2.0–9th International Congress & 3rd GEORES-GEOmatics and pREServation* (pp. 538–541). Universitat Politècnica de València.
16. Teruggi, S., & Fassi, F. (2022). HoloLens 2 Spatial Mapping Capabilities in Vast Monumental Heritage Environments. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *46*(2), 489-496.
17. Vassallo, R., Rankin, A., Chen, E. C., & Peters, T. M. (2017, March). Hologram stability evaluation for Microsoft HoloLens. In *Medical Imaging 2017: Image Perception, Observer Performance, and Technology Assessment* (Vol. 10136, pp. 295–300). SPIE.
18. Weinmann, M., Wursthorn, S., Weinmann, M., & Hübner, P. (2021). Efficient 3d mapping and modelling of indoor scenes with the microsoft hololens: A survey. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, *89*(4), 319–333.
19. Zheng, F., Schubert, R., & Weich, G. (2013, March). A general approach for closed-loop registration in AR. In *2013 IEEE Virtual Reality (VR)* (pp. 47–50). IEEE.
20. Raj, S., Beri, N., Patel, D. S., Sinha, Y., Chakrabarti, A., & Biswas, P. (2024). HaM3D: generalized XR-based multimodal HRI framework with haptic feedback for industry 4.0. Journal on Multimodal User Interfaces, 1–19.

Consistent Object Removal from Masked Neural Radiance Fields by Estimating Never-Seen Regions in All-Views

Yongjoon Lee[1], Jaehak Ryu[1], Donggeun Yoon[2], and Donghyeon Cho[1](✉)

[1] Department of Computer Science, Hanyang University, Seoul, South Korea
{dyd7168,jhakryu,doncho}@hanyang.ac.kr
[2] Korea Electronics Technology Institute (KETI), Seongnam, South Korea

Abstract. Neural radiance field (NeRF) is a technique for synthesizing novel-view images based on an understanding of scene geometry. Recently, there have been studies that remove objects from NeRF, which makes it possible to synthesize novel-view images with objects removed. Most existing methods apply a pretrained inpainting model to each multi-view image to remove objects, and use these images to train the NeRF model. However, these approaches not only require a lot of feed-forward of the inpainting model, but also lead to inconsistency problems between the inpainted images. To address these limitations, we propose a method to minimize the areas that need to be filled. To this end, we estimate never-seen regions that are occluded in all images based on density, and apply inpainting only to those regions. After removing target objects, we select the images that allow the final trained NeRF to consistently fill in the removed regions. Therefore, the proposed method consistently removes target objects from NeRF, and the effectiveness of the proposed method is demonstrated through various experiments. Furthermore, we suggest practical techniques to simplify the training processes and provide a new 360° real-world dataset for inpainting in NeRF.

1 Introduction

Neural radiance field (NeRF) [16] has gained significant popularity in the task of novel view synthesis, owing to its outstanding performance. There have been previous works that aim to improve NeRF [1,2,32], which have resulted in an accurate representation of 360° real-world data. As the ability of NeRF to reconstruct real-world scenes continues to improve, a variety of practical applications are emerging. One of the promising applications is object removal from the NeRF, which can be applied to various fields such as immersive content manipulation, VR/AR algorithms and game engines. A simple way to remove the target object is to recapture the scene without the object and retrain the NeRF model.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78312-8_28.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 416–431, 2025.
https://doi.org/10.1007/978-3-031-78312-8_28

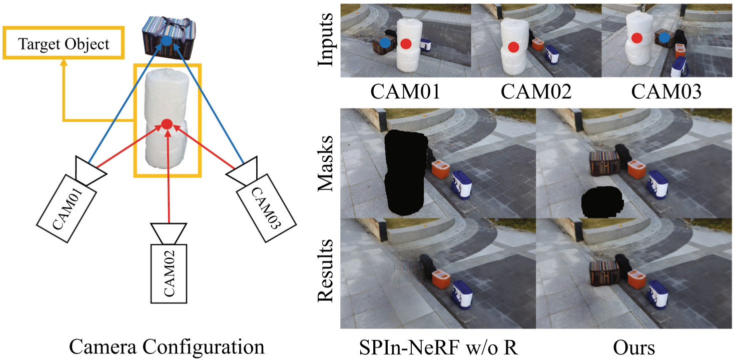


Fig. 1. Left: The bag is not visible in the image taken with `CAM02`, but is visible in the other images. Right (Inputs): Captured images using different cameras. Right (Masks): Comparison between an input object mask and an estimated never-seen mask. Right (Results): Comparison of results between SPIn-NeRF w/o R and our method. Note that the bag hidden behind the target object has been restored from the other-view image through our method.

However, this is impractical, cumbersome, and may not accurately reproduce the original scene. Therefore, a method for removing objects from the original NeRF model becomes necessary. Recently, there have been studies [14,17,28] that use an inpainting model to get rid of objects from the NeRF. Specifically, these methods remove the target objects for each image with masks, fill in erased regions using the inpainting model, and then train the NeRF by the inpainted images. These methods are very simple, but are highly dependent on the quality of the inpainting model. When the erased area becomes large and the background clutter becomes complex, inpainting quality is significantly degraded. Also, the contents generated by the inpainting model for each input image are inconsistent, which limits the performance of the NeRF.

To relieve these problems, we exploit information that is occluded by the target object but visible in the other-view images. We observe that other-view images have information beyond the object and the NeRF can fill the masked regions with contents from other-view images, as in [10,25]. As shown in Figure 1, the bag behind the target object is not visible in the image taken with `CAM02`, but it is captured in the images taken with `CAM01` and `CAM03`. Thus, there is no need to fill the area (*i.e.* bag) behind the target object with the inpainting model in the image taken with `CAM02`. In other words, we can fill those regions from other-view images by training a NeRF excluding pixels in masked regions.

However, training the NeRF cannot fill in occluded regions in the all-view images, thus we define it as never-seen regions. Based on above observations, in this paper, we propose a method to fill erased object regions by using information from other-view images as much as possible, and then apply an inpainting model to fill only the remaining never-seen regions. Since the inpainting model is applied to a minimal (*i.e.* never-seen) area, there are fewer inconsistencies between restored images and less sensitivity to the inpainting model. As a result, a method for the never-seen region estimation is required, and for this purpose, we utilize densities provided by the NeRF models. Specifically, we estimate the never-seen region using the characteristics of the original NeRF trained with input images containing the target object and the masked NeRF model trained by excluding pixels in the object region. As a result, our proposed method dramatically reduces the mask area that needs to be filled, resulting in a noticeable improvement in the performance of NeRF object removal. We train all the processes with our proposed efficient training strategy. To validate our methods, we make our new dataset, our dataset consists of real-world images captured in 360°, and includes ground truth (GT) for the evaluation of the object removal task. We validate that our proposed method is superior to the state-of-the-art object removal in NeRF [17,28] on both existing benchmarks [8,9,34] and our newly constructed data. To summarize, our paper has the following contributions.

- We introduce the concept of never-seen regions that are invisible in all-views, then remove objects from NeRF by inpainting only the never-seen regions.
- We analyze density profiles of original and masked NeRF with respect to never-seen regions, and propose a method for never-seen region estimation.
- We present an efficient training strategy that reduces the training time by finetuning the never-seen regions in the trained NeRF.
- We provide a real-world 360° dataset with GT to facilitate evaluation that can be useful in NeRF object removal tasks.

2 Related work

Image Inpainting. Image inpainting is a task of plausibly filling in damaged (*i.e.* erased) areas of an image. It is most commonly used when removing objects from the image. Traditionally, there have been many studies in image inpainting using patch-based methods [20]. With the progress of deep learning, inpainting techniques [5,38] have been actively researched and have achieved significant performance improvements in a variety of settings. Many convolution neural network based inpainting methods [3,13,23,31] have been introduced and have shown promising results, but tend to perform poorly on very large masks. Recently, diffusion-based inpainting methods [12,15,19] have emerged for more realistic reconstructions. However, these methods use information from only one image, thus they cannot restore a fully occluded object. For video inpainting, there have been methods [33,35] that use the optical flow and utilize the information in other frames. NeRF data can also be treated as a type of video, but it is not suitable to apply video inpainting because the movement between frames is large and the trajectory is often not continuous. Therefore, for multi-view images, including NeRF data, a specialized inpainting method is required.

Object Removal from NeRF. Since the first NeRF model [16] was introduced to synthesize realistic novel-view images, much follow-up research has been conducted in various directions, including synthesis quality [1,2,26,32] and generation speed [4,6,18,22]. In the case of scene manipulation field, there has also been various research on the object removal from NeRF. The general process of removing object from NeRF consists of erasing target objects and restoring space where objects were located. Typically, there are feature-based and RGB-D prior-based methods for removing objects from NeRF. Feature-based methods [7,10,24] focus on creating an accurate segmentation mask, and decompose object from NeRF. RGB-D prior-based methods [14,17,28,30] utilize the guidance of the restored color and depth maps, and restoration process normally uses inpainting; however, this method has the following two limitations. First, their performance highly depends on the quality of the inpainting model. Usually, performance of inpainting models deteriorates as the area to be filled becomes larger and the background becomes more complex. Second, inconsistencies between inpainted images cause quality degradation. In the case of 360° data with large movements between images and background clutters, objects will cause different occlusions in each view, resulting in inconsistent inpainting results. Although Weder [28] recently introduced a method to increase the consistency of object removal through view selection, it still has the limitation of having to fill a large area with the inpainting model. To overcome these limitations there are methods to improve quality by reducing the area to be filled, but they have only been effective in certain situations. Multi-view inpainting [11] uses traditional warping method and mask refinement [17] uses distance similarity to utilize other views information. These methods are pose based and they can find never-seen regions in ideal case on forward facing data. However, they are not suitable for 360° data, because of large angles between images on 360° data. To find never-seen region on 360° data, we need a pose-independent method. To this end, we propose a novel density-based never-seen region estimation method, and show its effectiveness.

3 Method

In this section, we briefly review the neural radiance field (NeRF) as background knowledge (Section 3.1) and explain a masked NeRF for the object removal (Section 3.2). Then, we describe a density-based method for estimating never-seen regions across all images using both original and masked NeRFs (Section 3.3). Finally, we describe a view selection process to make the final NeRF model consistently fill erased regions (Section 3.4). Additionally, we introduce techniques to speed up the training processes (Section 3.5). The overall pipeline of our method is illustrated in Figure 2.

3.1 Background: NeRF

Basically, the NeRF [16] is mainly used for novel-view synthesis based on understanding the 3D geometry of the scene. The NeRF consists of multi-layer perceptrons (MLPs) that take 3D position x and 2D ray direction d as inputs and estimate color c and density σ as follows.

$$F_{\Theta} : (x, d) \rightarrow (c, \sigma)\,, \tag{1}$$

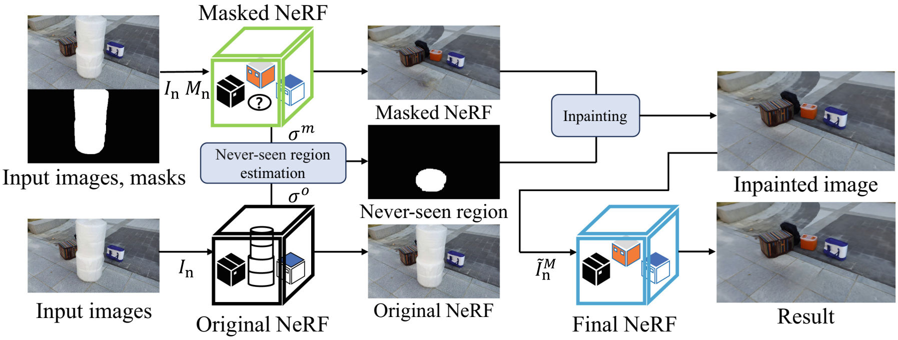


Fig. 2. Overview of our method. We take the image I_n and the mask M_n as input to train masked NeRF. We train the original NeRF starting from the weights of the masked NeRF model. Then, we use the density σ^m, σ^o of the two NeRFs to create a never-seen mask. Using an estimated never-seen mask, we apply an inpainting model to get inpainted images $\tilde{I}_n^M$. Finally, we further train the final NeRF starting from the weight of the masked NeRF using inpainted images $\tilde{I}_n^M$.

where F_Θ denotes MLPs with learnable parameters Θ. To get the color of the pixel in the image corresponding to each viewpoint, the NeRF uses a volume rendering using the color and density of the rays passing through that pixel. Specifically, since rays are continuous, the c and σ are sampled with a discrete grid, and the color of a pixel, $\hat{C}(r)$, is calculated as follows.

$$\hat{C}(r) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))c_i, \tag{2}$$

where T_i and δ_i are the transmittance and the distance between adjacent samples. Also, c_i and σ_i are the color and density of the ith sample. To train a NeRF, we use a reconstruction loss for the set of rays $\mathcal{R}$ in each batch as follows.

$$\mathcal{L} = \sum_{r \in \mathcal{R}} \left\| \hat{C}(r) - C(r) \right\|_2^2, \tag{3}$$

where $C(r)$ represents the pixel value for a ray r in the training images. Furthermore, hierarchical volume rendering [16] and distortion loss [2] are utilized to focus more on objects rather than empty space when grid sampling.

3.2 Masked NeRF for Object Removal

To remove a target object from NeRF, an inpainting model should be applied to each training image, given a mask of the object to be removed. This requires many feedforward of the inpainting model and leads to inconsistencies between inpainted images. To mitigate these drawbacks, a masked NeRF is considered as mentioned in [27–29] as a baseline method, which trains the NeRF model by excluding rays corresponding to objects that need to be removed. Specifically, when training the NeRF model based on Equation 3, we utilize only the sampled ray set $\mathcal{R}_m$, excluding rays corresponding to the target object to be removed, using masks for each training images. Since what is invisible in one-view is visible in another, it can work without applying the inpainting model, and there is less inconsistency problem. Unlike individually applying a 2D inpainting model for each image to fill the mask regions, the masked NeRF compounds information from multi-view images. Especially for 360° data with abundant multi-view information, masked NeRF outperforms inpainting models.

Even the masked NeRF restores regions that are visible in other-view images, it does not properly recover never-seen regions that are completely invisible in the all other-view images. Therefore, we introduce a method for never-seen region estimation and only apply the inpainting model to the never-seen regions. Note that all areas except never-seen regions are filled by masked NeRF.

3.3 Never-Seen Region Estimation

Our goal is to fill the never-seen region by finetuning the masked NeRF with inpainting, therefore we have to estimate never-seen region. In this paper, we analyze the density characteristics of the original NeRF and the masked NeRF for the same scene, and use them to estimate never-seen regions. As shown in Figure 3-(a), if the area to be removed is occluded in a one-view, but visible in others, the `Ray1` from the original NeRF will pass through the object and then go back into the empty space again. On the other hand, for never-seen regions that are not visible in all-views, the `Ray2` will not emerge back into the empty space. The ideal density profiles of the original NeRF for the two rays are shown in Figure 3-(b,c). Based on these density profiles, if t is the distance where the object is erased, density at the distance immediately in front of t is small for the visible region from other-views, and large for the never-seen region.

Therefore, in order to distinguish the visible region and the never-seen region, we need to estimate the distance of the end location where the object is removed. To this end, we utilize a masked NeRF trained using sampled rays in the only non-masked regions. As shown in Figure 3-(d), we observe that the density from the masked NeRF for rays passing through the region where the object is erased is empty. In particular, density profiles of both `Ray1` and `Ray2` are opaque (high density) only at the very end, as shown in Figure 3-(e,f). In fact, densities of the regions for the erased object should be unknown because they are excluded when the masked NeRF is trained. However, by using the distortion loss proposed in [2], we can take advantage of the fact that uncertain densities floating around like clouds are absorbed by opaque (dense) regions. Therefore, for each ray, based on these profiles, we can estimate the distance t when the object is erased. As a result, we can utilize the density profiles of the original NeRF and the masked NeRF to estimate the never-seen region, which is described in detail as follows.

Density Binarization. We denote the densities obtained from the original NeRF and the masked NeRF as σ^o and σ^m, respectively. Estimated densities

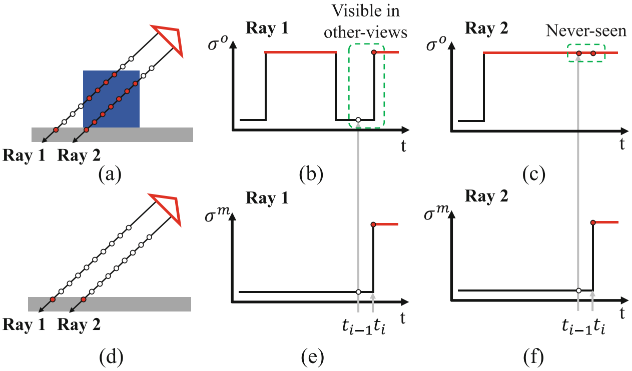


Fig. 3. (a) Ray profiles of original NeRF with objects. Density profiles of a ray passing (b) and a ray non-passing (c) through the object, in the original NeRF. (d) Ray profiles of masked NeRF without objects. Density profile of a ray passing through empty space in a masked NeRF. (e) and (f) are the rays at the same location as (b) and (c), respectively.

are continuous values, thus we perform binarization to convert them like the profiles in Figure 3. Since the densities estimated from the original and masked NeRFs have different scales, we set thresholds for each model as τ^o and τ^m, respectively. For the original NeRF, we set τ^o relatively high because there are clearly dense target objects. On the other hand, the density of the masked NeRF is relatively low, thus we set τ^o to be low. Based on thresholds τ^o and τ^m, we compute binarized densities for the original and masked NeRFs as follows.

$$\hat{\sigma}^o = \mathbb{1}\left[\sigma^o > \tau^o\right], \quad \hat{\sigma}^m = \mathbb{1}\left[\sigma^m > \tau^m\right] \times \hat{\sigma}^o, \tag{4}$$

where $\mathbb{1}[\cdot]$ is an indicator function. Note that the reason why $\hat{\sigma}^o$ is multiplied when computing $\hat{\sigma}^m$ is that $\hat{\sigma}^o$ is zero except in areas with opaque objects (high density), which suppress areas where there are no obvious objects.

Never-Seen Pixels Classification. Based on the estimated $\hat{\sigma}^o$ and $\hat{\sigma}^m$, we determine the never-seen regions. Specifically, we find the smallest index i such that $\sigma_i^m = 1$. In other words, i means the index of the nearest opaque place where the ray would hit after the target object is erased. After that, by looking at σ_{i-1}^o, we can determine if that region is completely obscured by the target object or not. If $\sigma_{i-1}^o = 1$, it is a never-seen region, otherwise, it is a region visible in the other-view images. We create a never-seen region mask by applying the above processes to all pixels in the mask regions. The entire processes for detecting never-seen regions are summarized in Algorithm 1.

3.4 View Selection for Masked NeRF

With obtained never-seen masks, a pretrained inpainting model is applied and the NeRF is trained with the inpainted images. In 360° data, masked NeRF

Algorithm 1: Never-Seen Region Estimation

```
Input: original NeRF $F^o_{\Theta_o}$, masked NeRF $F^m_{\Theta_m}$, masks $M_n$, thresholds $\tau^o$, $\tau^m$
Output: never-seen masks $M^{ns}_n$
foreach $r \in M_n$ do
    $\sigma^o \leftarrow F^o_{\Theta_o}(r)$, $\sigma^m \leftarrow F^m_{\Theta_m}(r)$
    $\hat{\sigma}^o \leftarrow \mathbb{1}[\sigma^o > \tau^o]$, $\hat{\sigma}^m \leftarrow \mathbb{1}[\sigma^m > \tau^m] \times \hat{\sigma}^o$
    $\boldsymbol{i} \leftarrow$ list of indexes having $\hat{\sigma}^m = 1$
    if $\boldsymbol{i} = \varnothing$ then
        $M^{ns}_n(r) \leftarrow 0$
    else
        $i \leftarrow \min(\mathbf{i})$
        if $\sigma^o_{i-1} = 1$ then
            $M^{ns}_n(r) \leftarrow 1$
        else
            $M^{ns}_n(r) \leftarrow 0$
```

reconstructs the entire scene, except for never-seen regions. We only need to finetune the never-seen region from masked NeRF with inpainting, and never-seen region is small and simple, thus this does not require inpainting in all-views. When using fewer images, the results are similar to using all images, but there is a computational advantage. We observed that it is better to select views evenly in all directions when training with a smaller number of inpainted images. Therefore we select 32 views in all directions along the input images. This results in consistent results with fewer inpainted images.

3.5 Training Strategies

Note that existing NeRF object removal methods [14,17,28] also have multiple trainable NeRF branches. Each branch in these methods is individually trained from scratch using all images. However, training different NeRF branches individually is time-consuming. In our method, to estimate the never-seen regions, we need to train both the original NeRF $F^o_{\Theta_o}$ and the masked NeRF $F^m_{\Theta_m}$. Therefore, to reduce training time, we train the masked NeRF first, then conduct finetuning with very small iterations only on the mask regions to get the weights of the original NeRF. Also, since the original NeRF is only used to estimate never-seen regions, it does not require a detailed texture representation as it only needs density and shape information, thus few iterations are sufficient.

Finally, even when training the final NeRF model $F^f_{\Theta_f}$ using inpainted images with a never-seen region mask, we reduce the training time by finetuning from the masked NeRF $F^m_{\Theta_m}$. To improve our result, we also apply depth and perceptual losses in [17]. This is a feasible strategy because the masked NeRF model $F^m_{\Theta_m}$ has already been trained for regions other than the masked region. This means that only the masked areas need to be further trained.

4 Experiments

4.1 Dataset

There is few public 360° data for the task of removing objects from NeRF. Therefore, we build a new dataset captured from all 360° viewing angles for object removal from novel-view synthesis. It consists of a set of images taken from 11 different real-world scenes, including objects of different sizes from indoor/outdoor for variety. Each dataset is 3,840 × 2,160 in size and each scene dataset consists of 150-200 images. For quantitative evaluation of the object removal task, we also provide ground truth (GT) data taken without the target object. We validate the proposed method on our new dataset as well as existing 360° datasets [2,26].

4.2 Evaluation Metrics

For a quantitative comparison, we leverage our new dataset containing GT images with the target object erased. To focus on more object areas, we only evaluate the region inside the bounding box of the mask to minimize the influence of non-target background. As evaluation metrics, we adopt PSNR and SSIM [9], which are traditional methods for assessing the performance of the image reconstruction. We also leverage LPIPS [34] and FID [8] to evaluate the visual quality of the synthesized image. In the case of experiments using the existing dataset, due to the absence of GT, we utilize HyperIQA [21] and MetaIQA [36] as evaluation metrics, which is one of the best blind image quality assessments.

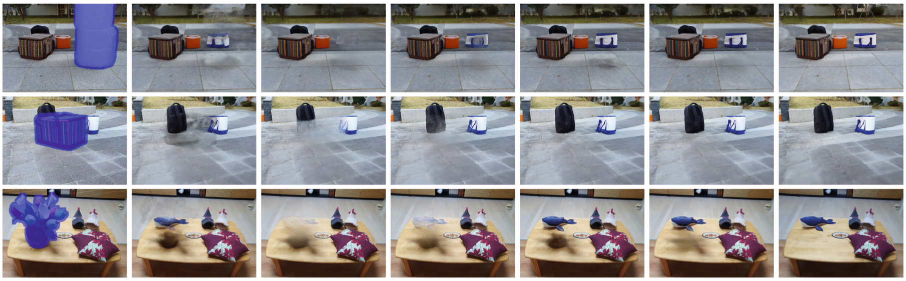
4.3 Baseline and Implementation Details

Baselines. To demonstrate the effectiveness of our method, we compare it with six baseline approaches tailored to object removal from NeRF as follows. (1) Inpainted NeRF is a method that is simply trained with inpainted images. (2) Video inpainted NeRF is similar to the inpainted NeRF, but uses a video inpainting model for consistency across the training images. (3) SPIn-NeRF is the method in [17]. (4) SPIn-NeRF w/o R is the method in [17] excluding the mask refinement method. The reason for excluding the mask refinement method is that it is likely to fail on 360° data (please see supplement for more details). It is equivalent to adding depth loss and perceptual loss to Inpainted NeRF. (5) Remove-NeRF is a method in [28]. We additionally apply perceptual loss in [17] to better performance. (6) Masked NeRF is a model that excludes regions containing objects from the training process.

Implementation details. As mentioned in Section 3.5, our method needs to train three NeRF models: $F^m_{\Theta_m}$, $F^o_{\Theta_o}$, and $F^f_{\Theta_f}$. We train all NeRF with a batch size of 8,192 using two RTX 3090. For $F^m_{\Theta_m}$, we train 100k iterations with using the reconstruction loss multiplied by the mask. Then, $F^o_{\Theta_o}$ is finetuned from $F^m_{\Theta_m}$ by $100N$ iterations, where N is the number of images. Also, $F^f_{\Theta_f}$ is updated from $F^m_{\Theta_m}$ using 32 inpainted images and depths. We trained 12,800 iterations to train

Table 1. Quantitative comparisons on 360° real-world data. The **bold score** is the best score, and the underlined score is the second.

	Our data				Existing data	
Method	PSNR↑	SSIM↑	LPIPS↓	FID↓	HyperIQA↑	MetaIQA↑
Inpainted NeRF [23]	19.53	0.76	0.33	165.06	52.91	0.278
Video inpainted NeRF [35]	20.55	0.79	0.30	191.51	47.11	0.244
SPIn-NeRF [17]	19.16	0.76	0.27	145.87	48.76	0.262
SPIn-NeRF w/o R [17]	20.14	0.78	0.24	119.48	50.19	0.277
Remove-NeRF [28]	20.74	0.80	0.19	99.41	52.31	0.285
Masked NeRF	20.07	0.78	0.20	97.14	52.18	0.269
Ours	**21.43**	**0.81**	**0.18**	**86.06**	**53.02**	**0.290**



Input SPIn-NeRF SPIn-NeRF w/o R Remove-NeRF Masked NeRF Ours GT

Fig. 4. Qualitative comparisons on our 360° real-world data. The areas overlaid in blue is the target object. Our method outperforms the baselines.

32 images 400 times each. As a result, when training with 200 images, the total iterations are 132,800. The number of iterations for the finetuning phases is much smaller compared to iterations when training the masked NeRF from scratch. For a fair comparison, we use the same model with the image inpainting as [23], video inpainting as [35], and NeRF as [2]. We apply appropriate morphological operations to the never-seen mask in each data to reduce the impact of noise.

4.4 Results

For quantitative comparisons, we compare our method to six baselines for four evaluation metrics: PSNR, SSIM, LPIPS, and FID. As reported in Table 1, the proposed method consistently achieves the best performance on all metrics. In other words, it shows that the proposed method not only restores erased areas well, but also produces visually pleasing results. In order to quantitatively evaluate the proposed method on the existing 360° datasets [2, 26], we use HyperIQA and MetaIQA as no-reference blind assessments. In Table 1, we report that our method achieves best performance in both evaluation metrics. These quantitative results on various dataset demonstrate the versatility of our method. In Figure 4, we provide qualitative comparisons of our method with four NeRF object removal methods including SPIn-NeRF [17], SPIn-NeRF w/o R [17], Remove-NeRF [28]

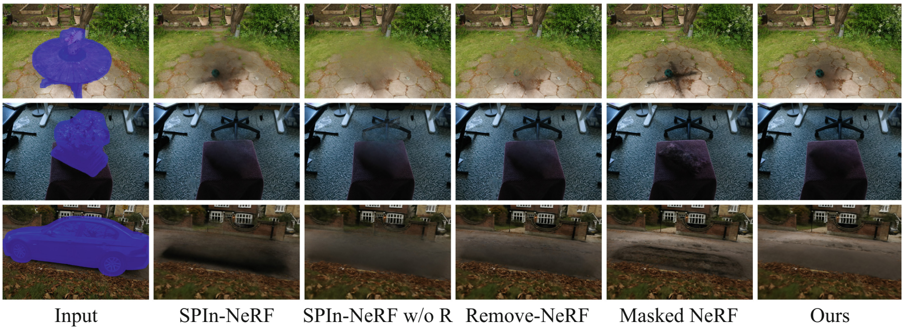


Fig. 5. Qualitative comparisons on the existing 360° datasets. These scenes are obtained from [2,26]. The areas overlaid in blue is the target object.

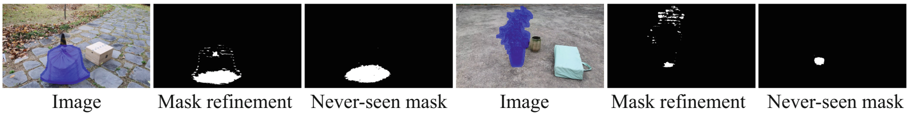


Fig. 6. Qualitative result of never-seen region estimation and mask refinement. The areas overlaid in blue is the target object. Our never-seen mask estimation produces more clear and correct masks than the mask refinement method in [17].

and Masked NeRF on our new 360° real-world NeRF object removal dataset. SPIn-NeRF can restore occluded region, but has many artifacts due to failure of mask refinement. SPIn-NeRF w/o R successfully fills in the erased regions when the background is simple or there are no objects occluding the target object. However, when there is an object occluded by the target object, it cannot properly restore occluded regions. In addition, when the mask is large and the background is complex, inpainting results for both color and depth are inaccurate, which causes smoke-like artifacts to remain where the object used to be. Remove-NeRF mitigates this problem, but artifacts still remain. Meanwhile, Masked NeRF uses information from other-views as described in Section 3.2 to fill in occluded regions, but it cannot properly restore the color and texture of never-seen regions. Finally, our method fills in the never-seen regions, showing plausible and consistent results in all directions.

Figure 5 shows qualitative results on the existing 360° datasets. These data samples are very challenging to fill in the erased areas naturally because the target objects are very large. However, our method favorably fills in the never-seen regions as well as the regions that are occluded but visible in other-view images. Figure 6 shows qualitative results of our never-seen region estimation and mask refinement [17]. Both methods aim to reduce the masked areas that needs to be erased. Mask refinement [17] makes noisy masks, however our never-seen region estimation makes clean and correct masks.

As described in Section 3.4 and Section 3.5, we simplify overall process of our method. To present time efficiency of our method, we measure the run-time with

Table 2. Comparisons on run-times.

Method	Training	Rendering	Inpainting	Mask estimation	Total
SPIn-NeRF [17]	16.92h	0.95h	∼1m	51.78h	69.67h
Remove-NeRF [28]	17.25h	0.98h	∼1m	-	18.25h
Ours	11.16h	0.93h	∼0.2m	1.95m	12.13h

Table 3. Ablation study on view selection. Our even view selection uses 32 images, but achieves similar results to using all images.

Method	PSNR↑	SSIM↑	LPIPS↓	FID↓
Masked NeRF	20.07	0.78	0.198	97.14
(a) No view selection with all views	**21.87**	**0.81**	**0.173**	**82.92**
(b) Confidence based view selection with all views	<u>21.80</u>	<u>0.81</u>	<u>0.174</u>	<u>85.11</u>
(c) No view selection with even 32 views (Ours)	21.43	0.81	0.176	86.06
(d) Confidence based view selection with even 32 views	21.39	0.80	0.178	87.26

the baselines. Note that we measure it assuming a fixed setting with 200 images of size 960 × 540 because the overall time can vary depending on the data. For the mask estimation methods, we assume 128 sampling points for each ray and a mask size of 50,000 pixels per image. As shown in Table 2, our method is faster than other baselines. Specifically, mask refinement takes a very long 51.78 hours in the official code, while never-seen mask estimation takes only two minutes.

4.5 Ablation Studies

View selection ablation. The difference between SPIn-NeRF w/o R and Remove-NeRF is the absence or presence of confidence based view selection. As repored in Table 1 Remove-NeRF outperforms SPIn-NeRF w/o R by applying confidence based view selection. To show whether confidence based view selection even effectively works in our method, we perform ablation experiments as follows. (a) No view selection with all-views. (b) Confidence based view selection [28] with all-views. (c) No view selection with even 32 views (Ours). (d) Confidence based view selection [28] with even 32 views. Note that the final NeRF in our method uses evenly selected 32 views, but SPIn-NeRF w/o R and Remove-NeRF use all-views. Figure 7 shows the results of four experiments. As reported in Table 3, we can see that the performance decreases when the confidence based view selection method is applied to our method. We guess that this is because the input view images in our final NeRF training stage are nearly consistent. In other words, there is no need to apply confidence based view selection in our method. Therefore, confidence based view selection does not provide sufficient benefit in our setting, thus we does not include it in our final method. In addition, comparison in Table 3-(a,c) shows that training with even 32 views can achieve similar results with training with all-views.

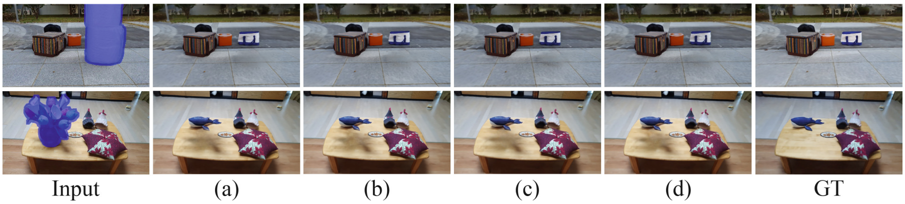


Fig. 7. Qualitative comparisons on view selection ablations. (a) No view selection with all-views. (b) Confidence based view selection [28] with all-views. (c) No view selection with even 32 views (Ours). (d) Confidence based view selection [28] with even 32 views. The areas overlaid in blue is the target object.

Table 4. Quantitative comparisons on other inpainting model. Our method achieves the best on all metrics.

	LDM				MADF			
Method	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
Inpainted NeRF [23]	19.39	0.76	0.32	157.76	19.53	0.76	0.33	174.99
SPIn-NeRF w/o R [17]	20.07	0.78	0.24	124.47	20.02	0.78	0.26	133.17
Remove-NeRF [28]	<u>20.74</u>	<u>0.80</u>	<u>0.20</u>	<u>94.97</u>	<u>20.72</u>	<u>0.81</u>	<u>0.20</u>	<u>100.09</u>
Ours	**20.83**	**0.81**	**0.18**	**87.10**	**21.31**	**0.82**	**0.19**	**93.50**

Inpainting model ablation. To show that our method is less affected by the inpainting model, we conduct an ablation study on other inpainting models. We choose the latent diffusion model (LDM) [19] as the recent model and MADF [37] as the old model. We train the same as the experiment in Section 4.4 except for the inpainting model. Existing NeRF object removal methods that do not use masked NeRF are have to fill entire mask with inpainting model, thus they highly affected by inpainting model. However, our method minimizes the mask, thus it is less affected by the inpainting model. As shown in Figure 8, while other methods have varying performance depending on the inpainting model, but ours always achieves similar performance. Comparing Table 1 with Table 4, we can see that our method has small change and still outperforms the other methods.

4.6 Limitation

Because the proposed technique aims to perform consistent inpainting by reducing the area to which the inpainting model is applied by fully utilizing information from wide viewpoints, there are fundamental limitations to forward-facing data with limited viewpoint information. As shown in Figure 9-(Left), when the angle between images is very small, the estimated never-seen mask is almost similar to the given input mask, thus the advantage of the proposed method is not revealed. In contrast, as shown in Figure 9-(Right), as the angle between input images increases, the effectiveness of the proposed method increases.

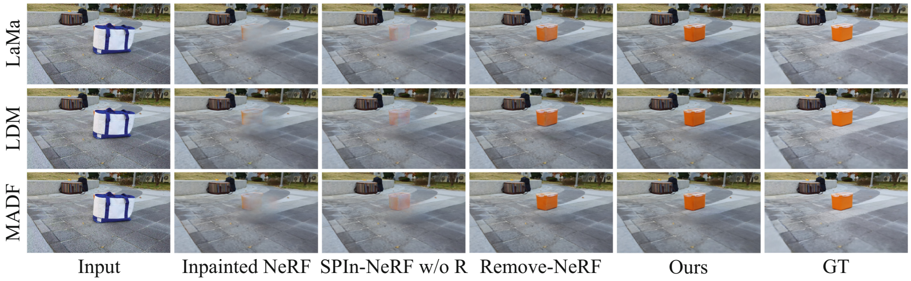


Fig. 8. Qualitative comparisons on inpainting model ablations. Ours always consistent regardless of the inpainting model, and outperforms other methods.

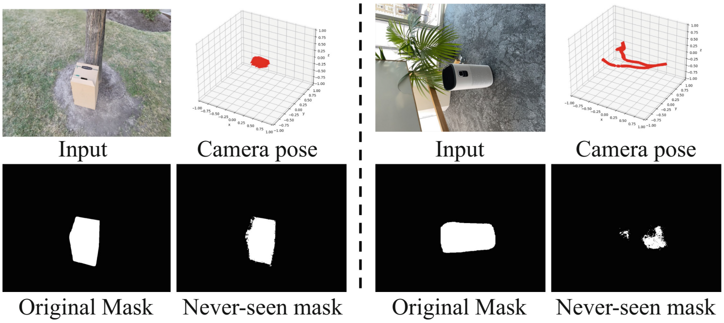


Fig. 9. Results of never-seen region estimation on forward-facing data. (Left): Narrow baseline. (Right): Wide baseline (approximately 180° angles).

5 Conclusion

In this paper, we have proposed a never-seen region estimation method for the consistent inpainting from NeRF. Specifically, the masked NeRF is trained by using only rays sampled from non-object regions. Then, we obtain the original NeRF finetuned from the masked NeRF. Based on the density profiles of the original and masked NeRFs, we find the never-seen regions. The final NeRF model is also finetuned from the masked NeRF using the inpainted images. For evaluation, we have constructed a new real-world 360° dataset for NeRF object removal. Our method has achieved competitive performance on both our new dataset as well as existing 360° dataset.

Acknowledgements. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2022-0-00022, RS-2022-II220022, Development of immersive video spatial computing technology for ultra-realistic metaverse services).

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: ICCV. pp. 5855–5864 (2021)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR. pp. 5470–5479 (2022)
3. Cao, C., Fu, Y.: Learning a sketch tensor space for image inpainting of man-made scenes. In: ICCV. pp. 14509–14518 (2021)
4. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: ECCV. pp. 333–350 (2022)
5. Demir, U., Unal, G.: Patch-based image inpainting with generative adversarial networks. arXiv preprint arXiv:1803.07422 (2018)
6. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR. pp. 5501–5510 (2022)
7. Goel, R., Sirikonda, D., Saini, S., Narayanan, P.: Interactive segmentation of radiance fields. In: CVPR. pp. 4201–4211 (2023)
8. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. NIPS (2017)
9. Hore, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: ICPR. pp. 2366–2369 (2010)
10. Kobayashi, S., Matsumoto, E., Sitzmann, V.: Decomposing nerf for editing via feature field distillation. NeurIPS pp. 23311–23330 (2022)
11. Li, F., Ricardez, G.A.G., Takamatsu, J., Ogasawara, T.: Multi-view inpainting for rgb-d sequence. In: 2018 International Conference on 3D Vision (3DV). pp. 464–473 (2018)
12. Li, H., Luo, W., Huang, J.: Localization of diffusion-based inpainting in digital images. IEEE transactions on information forensics and security pp. 3050–3064 (2017)
13. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: ECCV. pp. 85–100 (2018)
14. Liu, H.K., Shen, I., Chen, B.Y., et al.: Nerf-in: Free-form nerf inpainting with rgb-d priors. arXiv preprint arXiv:2206.04901 (2022)
15. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: CVPR. pp. 11461–11471 (2022)
16. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM pp. 99–106 (2021)
17. Mirzaei, A., Aumentado-Armstrong, T., Derpanis, K.G., Kelly, J., Brubaker, M.A., Gilitschenski, I., Levinshtein, A.: Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In: CVPR. pp. 20669–20679 (2023)
18. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM TOG pp. 1–15 (2022)
19. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
20. Ružić, T., Pižurica, A.: Context-aware patch-based image inpainting using markov random field modeling. IEEE (2014)

21. Su, S., Yan, Q., Zhu, Y., Zhang, C., Ge, X., Sun, J., Zhang, Y.: Blindly assess image quality in the wild guided by a self-adaptive hyper network. In: CVPR. pp. 3667–3676 (2020)
22. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: CVPR. pp. 5459–5469 (2022)
23. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 2149–2159 (2022)
24. Tschernezki, V., Laina, I., Larlus, D., Vedaldi, A.: Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In: 2022 International Conference on 3D Vision (3DV). pp. 443–453 (2022)
25. Tschernezki, V., Larlus, D., Vedaldi, A.: Neuraldiff: Segmenting 3d objects that move in egocentric videos. In: 2021 International Conference on 3D Vision (3DV). pp. 910–919 (2021)
26. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-nerf: Structured view-dependent appearance for neural radiance fields. In: CVPR. pp. 5481–5490 (2022)
27. Wang, Y., Wu, W., Xu, D.: Learning unified decompositional and compositional nerf for editable novel view synthesis. In: ICCV. pp. 18247–18256 (2023)
28. Weder, S., Garcia-Hernando, G., Monszpart, A., Pollefeys, M., Brostow, G.J., Firman, M., Vicente, S.: Removing objects from neural radiance fields. In: CVPR. pp. 16528–16538 (2023)
29. Yang, B., Zhang, Y., Xu, Y., Li, Y., Zhou, H., Bao, H., Zhang, G., Cui, Z.: Learning object-compositional neural radiance field for editable scene rendering. In: ICCV. pp. 13779–13788 (2021)
30. Yin, Y., Fu, Z., Yang, F., Lin, G.: Or-nerf: Object removing from 3d scenes guided by multiview segmentation with neural radiance fields. arXiv preprint arXiv:2305.10503 (2023)
31. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: ICCV (2019)
32. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492 (2020)
33. Zhang, K., Fu, J., Liu, D.: Flow-guided transformer for video inpainting. In: ECCV. pp. 74–90 (2022)
34. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018)
35. Zhou, S., Li, C., Chan, K.C., Loy, C.C.: Propainter: Improving propagation and transformer for video inpainting. In: ICCV. pp. 10477–10486 (2023)
36. Zhu, H., Li, L., Wu, J., Dong, W., Shi, G.: Metaiqa: Deep meta-learning for no-reference image quality assessment. In: CVPR. pp. 14143–14152 (2020)
37. Zhu, M., He, D., Li, X., Li, C., Li, F., Liu, X., Ding, E., Zhang, Z.: Image inpainting by end-to-end cascaded refinement with mask awareness. IEEE Trans. Image Process. **30**, 4855–4866 (2021)
38. Zhu, X., Qian, Y., Zhao, X., Sun, B., Sun, Y.: A deep learning approach to patch-based image inpainting forensics. Signal Processing: Image Communication pp. 90–99 (2018)

Neuropostors: Neural Geometry-Aware 3D Crowd Character Impostors

Mirela Ostrek[1](✉), Niloy J. Mitra[2], and Carol O'Sullivan[3]

[1] Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2] University College London, London, United Kingdom
[3] Trinity College Dublin, Dublin, Ireland

Abstract. Crowd rendering and animation was a very active research area over a decade ago, but in recent years this has lessened, mainly due to improvements in graphics acceleration hardware. Nevertheless, there is still a high demand for generating varied crowd appearances and animation for games, movie production, and mixed-reality applications. Current approaches are still limited in terms of both the behavioral and appearance aspects of virtual characters due to (i) high memory and computational demands; and (ii) person-hours needed of skilled artists in the context of short production cycles. A promising previous approach to generating varied crowds was the use of pre-computed impostor representations for crowd characters, which could replace an animation of a 3D mesh with a simplified 2D impostor for every frame of an animation sequence, e.g., Geopostors [1]. However, with their high memory demands at a time when improvements in consumer graphics accelerators were outpacing memory availability, the practicality of such methods was limited. Inspired by this early work and recent advances in the field of Neural Rendering, we present a new character representation: Neuropostors. We train a Convolutional Neural Network as a means of compressing both the geometric properties and animation key-frames for a 3D character, thereby allowing for constant-time rendering of animated characters from arbitrary camera views. Our method also allows for explicit illumination and material control, by utilizing a flexible rendering equation that is connected to the outputs of the neural network.

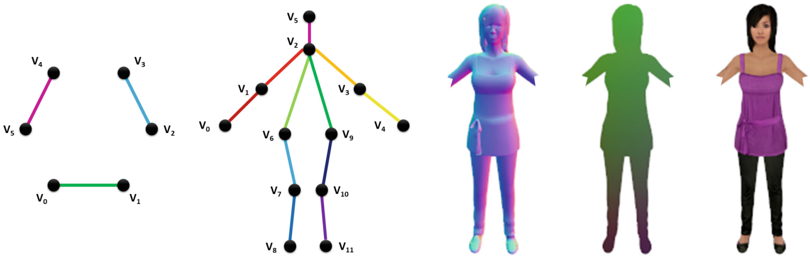
Keywords: Crowd Simulation · Virtual Characters · Neural Rendering

1 Introduction

The quality of 3D virtual characters has reached new levels of quality in recent years, with tools such as MetaHuman Creator^{TM} enabling content creators to quickly create highly realistic models of virtual humans for games and interactive applications [2]. However, most rendering and animation methods depend on a rigged 3D polygonal mesh representation, which may be costly in terms of

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 432–448, 2025.
https://doi.org/10.1007/978-3-031-78312-8_29

computational power and memory for some applications, such as crowd simulation, especially if the model itself is very detailed. Multi-view pre-computed impostor representations, where a complex 3D model is replaced by a set of an image-based 2D impostor for every frame of an animation sequence [1,3] have been proposed as an alternative, with the advantage of (i) constant-time rendering, irrespective of the complexity of the model or animation; and (ii) leveraging graphics hardware to generate appearance variety for a large number of crowd characters. However, applicability and motion variation is limited due to the memory requirements of creating and storing a full impostor set for every new animation sequence.



(a) OpenGL primitives **(b)** 2D skeleton **(c)** Normals **(d)** Vertices **(e)** Albedo

Fig. 1. We propose a hybrid classical graphics (G) and neural (N) rendering pipeline tailored for **crowd simulation**. (G) Line primitives (a) are first assembled and rendered as a stick figure (b) to animate a crowd character. (N) The skeleton is then translated into a set of intrinsic maps (c, d, e) using a small image-to-image translation module. The predicted intrinsics are later combined via the rendering equation, allowing for the generation of realistic characters and utilizing various controls from G.

Inspired by these 'Geopostors' [1] and recent developments in Neural Rendering [4], we propose a new representation (Neuropostors) that retains all the advantages of this image-based approach to character simulation, while overcoming most of the disadvantages. We propose a neural network approach that predicts the decomposition of an animated character into a set of intrinsic 2D maps that encode information such as vertices, normals and colors. (See Fig. 1). We also provide explicit illumination and material control, by utilizing a flexible rendering equation that is connected to the outputs of the neural network. Our method delivers constant speed character rendering, irrespective of the underlying model's complexity, and provides enhanced variation by allowing different characters to be rendered with the same network.

Our rendering pipeline contains three parts. (I) Synthetic data comprising 2D images of a 3D animated character at multiple keyframes, from multiple camera viewpoints (uniform sampling) is generated. These images encode vertices, normals, colors and other maps and provide the output for supervised learning. Images are also generated for 2D keypoints, camera pose and character id, as input controls for our model. (II) We train a neural network that translates from joints, camera pose and character id (in the case of multiple characters) to the normal, vertex and texture maps that are mapped onto the character. This allows for (a) pose control (unlimited interpolation between animation keyframes); (b) camera view control (unlimited interpolation between views); and (c) character control (multiple characters can be compressed and represented together with one neural network). (III) We then use a modified rendering equation to combine the previously generated maps to generate a fully shaded character representation with full (d) illumination and (e) material control. (See Fig. 2).

The main advantages of our approach are as follows: (i) a novel method that **compresses multiple characters using a neural network** and allows for pose, camera view, character, illumination, and material control – multi-view and pose-dependent appearance data is only needed when we train our model and not at runtime; (ii) **viewpoint and animation interpolation**, which overcomes the problem of limited views and animation keyframes due to memory limitations in previous methods; and (iii) the flexibility to incorporate **multiple types of maps and different rendering equations**, because memory is only needed during training, and not at run-time.

Table 1. Comparison with 3D graphics and other efficient crowd rendering methods.

Method	Visual Quality	Advantages	Disadvantages
3D Graphics	High	High visual fidelity, smooth transitions	High computational cost
Geopostors	High for close-up, low for distant	Balances detail and performance	Quality degradation for distant objects
Polypostors	Moderate	Better silhouette approximation than in Geopostors	Moderate quality for close objects
Neuropostors	High for close-up, adjustable for distant	High visual fidelity, smooth transitions, low impostor memory	NNet training and data generation costs

2 Related Work

While numerous studies exist at the intersection of Computer Graphics and Neural Rendering, conducting an exhaustive review falls beyond the scope of this paper. Our emphasis is on research pertinent to the application of neural rendering techniques for character and crowd rendering and animation, rather than real-world image processing. Notably, recent advancements in crowd simulation and Neural Rendering are detailed in [5] and [4] respectively, providing valuable insights into the current state-of-the-art in these domains.

Crowd Simulation. Crowd generation algorithms often draw from classical models such as the somatotype model proposed by Sheldon et al. [6], which delineates three primary human body types (endomorph, mesomorph, and ectomorph). By applying predefined equations to human body specifications and templates, a wide range of body shapes can be derived. However, in crowd simulation, factors beyond physical attributes, such as appearance (e.g., colors, textures, and accessories) and behavioral patterns, also play crucial roles.

One notable method incorporating these principles is Geopostors, introduced by Dobbyn et al. [1]. Building upon earlier work by Tecchia et al. [3], Geopostors pioneers a hybrid crowd rendering approach, seamlessly blending real-time rendering of 3D geometric primitives with pre-rendered 2D "impostors" of individuals. Through extensive perceptual evaluation studies, the authors ensure the perceptual realism of rendered crowds in terms of both character diversity and quantity. Specifically, characters close to the camera undergo traditional 3D rendering to maintain realism and avoid pixelation, while those further away are represented as 2D impostors, ensuring efficient rendering of large crowds. Another approach is Polypostors, introduced by Kavan et al. [7]. Polypostors convert 3D characters to 2D textured polygons, achieving significant simplification with low memory overhead, suitable for real-time crowd rendering. While they offer high rendering efficiency and enable smooth animation transitions, Polypostors may produce artifacts with overhead views or complex animations. Our approach extends the Geopostors paradigm by replacing pre-rendered 2D impostors with neural representations, allowing for more dynamic and adaptable crowd rendering while retaining efficiency (see Table 1 and Table 2).

Image Synthesis. Image synthesis techniques are broadly categorized as unconditional and conditional methods. Unconditional methods encompass earlier models such as variational autoencoders (VAEs) [8,9], autoregressive models [10,11], and Generative Adversarial Networks (GANs) [12], including variants like DCGAN [13], LS-GAN [14], Wasserstein GAN [15], and LR-GAN [16]. Recently, diffusion models have gained prominence in the field of image synthesis due to their ability to generate high-fidelity images. Models such as the Denoising Diffusion Probabilistic Model (DDPM) [17] and its variants [18], including latent diffusion models [19], have shown impressive results in generating realistic images. These models leverage a diffusion process to iteratively refine samples.

Conditional methods, on the other hand, are tailored to specific tasks such as category-to-image [20–22], text-to-image [23], sketch-to-image [24–27], and image-to-image translation. Notable examples in this category include pix2pix [28], pix2pixHD [29], CycleGAN [30], Cascaded Refinement Networks [31], CoGAN [32], and UNIT [33]. Conditional latent diffusion models like ControlNet [34] have also shown remarkable capabilities in generating realistic images. While these advances offer state-of-the-art performance in terms of image quality and fidelity, they are computationally expensive. In this work, we use a small and compact Convolutional Network that does not require a lot of memory and is fast at generating small crowd characters.

Image Decomposition for Relighting. In recent years, the rapid development of Neural Rendering has spurred a significant body of work on relighting, with notable contributions highlighted in [35] and [36]. Particularly relevant is the neural pipeline for controllable image generation introduced by Chen et al. [37]. However, their method is tailored for static objects and lacks support for articulated 3D character representation. In contrast, our focus is on replicating existing articulated 3D models of virtual humans, emphasizing efficiency and scalability.

Table 2. Analysis of impostor representation, time, and memory complexities among 3D graphics and efficient crowd rendering methods. Our method excels in (1) interpolation capabilities and (2) memory efficiency compared to geopostors and polypostors, particularly for compressing extensive animation sequences and diverse crowd characters. *Notes:* n: number of crowd characters, v: number of vertices per character, n_g: number of geometric models (close-up characters), n_i: number of impostors (distant characters), NN: Memory required for the NNet model used for impostor compression.

Method	Impostor Representation	Time Complexity	Memory Complexity
3D Graphics	Triangles for geometry	$O(n \cdot v)$	$O(n \cdot v)$
Geopostors	2D billboards	$O(n_g \cdot v + n_i)$	$O(n_g \cdot v + n_i)$
Polypostors	Body part polygons	$O(n_i)$	$O(n_i)$
Neuropostors	Neural network	$O(n_g \cdot v + n_i)$	$O(n_g \cdot v) + O(\mathrm{NN})$

3 Method

In this section, we outline the three main stages of our comprehensive 3D virtual character animation and rendering process:

1. **Synthetic Data Generation:** We provide insights into our OpenGL implementation, detailing the setup for multi-view 3D virtual character animation and rendering.
2. **Neural Decomposition Representation:** Here, we discuss our approach to neural modeling, focusing on the representation of characters through decomposition maps.
3. **Shading with a Rendering Equation:** We delve into the application of a rendering equation for shading, explaining how it is integrated into our pipeline to achieve realistic rendering results.

These three stages collectively form our methodological framework for animating and rendering virtual characters with neural techniques.

3.1 Synthetic Data Generation

In the initial stage of our pipeline, synthetic data is generated to facilitate training of our machine learning model. Utilizing a 3D model of a virtual character and a specified animation sequence, we produce a collection of images capturing the virtual character in various poses across multiple viewpoints. These images serve as the training dataset for our subsequent machine learning model.

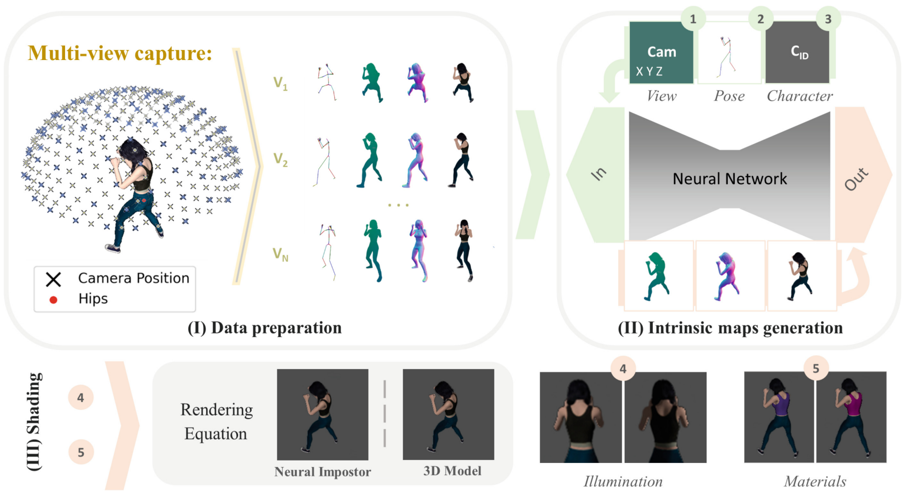


Fig. 2. System Overview: Our pipeline consists of three main stages: (I) Synthetic data generation from multiple views of a 3D character performing an animation sequence. (II) Simultaneous input and output of information including camera view ①, 2D pose ②, character ID ③, and ground truth vertex, normal, and texture maps for image-to-image translation in the ML module. (III) Integration of predicted intrinsic maps using the rendering equation to create a neural impostor, allowing for explicit control over illumination ④ and material properties ⑤.

Multi-View Camera Setup. To determine the optimal number of camera views needed for model training, we use uniform sampling. This involves dividing an icosahedron's triangular faces into smaller triangles and projecting them onto a sphere to create geodesics of varying levels. These geodesics represent densely sampled spheres of camera views. The virtual character is positioned within this sphere, typically with the hips joint at the sphere's center or slightly offset (as shown in Section 4). We focus on camera views where the camera is positioned above the character's hips, reducing the required samples by about half. While we could further reduce viewpoints for symmetric characters and animations by mirroring results from one hemisphere, we choose to use the entire upper hemisphere to accommodate asymmetric characters and animations, aligning with data augmentation principles for better model generalization.

Input Data. We collect essential information for each generated image, including the character's 2D pose, camera location, and light position. The 2D pose data is represented as a raster image in RGBA format. Camera locations and light positions are structured into a 3-channel tensor, with each channel representing the X, Y, or Z coordinate axis. In scenarios involving multiple characters, we also record each character's ID and normalize it within the $[-1, 1]$ range. This ID data is converted into a grayscale image, similar to how camera locations and light positions are converted into RGB images, to suit the Convolutional Neural Network (CNN) architecture.

Output Data. For each image, we generate three types of maps: vertex maps, normal maps, and texture maps.

1. Vertex Maps: These maps represent the 3D coordinates of each vertex in the character's mesh. We project these coordinates onto the 2D image plane to create a vertex map. Since each vertex has its own 3D position, we can interpolate between these positions to obtain smooth surfaces when rendering the character in 2D.
2. Normal Maps: Normal maps encode the surface orientation at each point on the character's surface. Similar to vertex maps, we interpolate between the normal vectors assigned to each vertex to obtain smooth surface normals across the character's body.
3. Texture Maps: Texture maps contain color information for each point on the character's surface. These maps can be directly sampled from a texture image applied to the character's mesh, providing detailed color information for rendering the character in 2D.

In summary, these maps allow us to accurately represent the geometry, surface orientation, and texture of the character in each image, enabling realistic rendering from different viewpoints. Examples of these maps can be seen in Figure 1 and will be further discussed in the evaluation section.

3.2 Neural Decomposition Pipeline

The neural decomposition pipeline, depicted in Figure 2, focuses on training a neural network to learn a mapping function $\mathcal{F} : (Img_J, Img_{C_{xyz}}) \Rightarrow Img_R$. Here, Img_J represents RGBA images containing character-specific colored 2D keypoints, while Img_R comprises decomposition maps, including vertex, normal, and texture maps, rendered based on the specified keypoints.

For our image-to-image translation task, we incorporate a rendering equation function into the neural network's output. This function allows for explicit control over illumination and materials, aligning with the task requirements.

Model Architecture: Our experiments employ the U-Net model [38], a type of Convolutional Neural Network (CNN) commonly used for tasks like image segmentation. Comprising an encoder E and decoder D, each with 5 blocks of layers, our U-Net model leverages bypass connections to capture high-resolution details from lower layers. In the encoder, each block consists of convolutional and batch normalization layers followed by ReLU activation, with subsequent max-pooling layers downsampling the tensor's width and height while doubling the number of feature maps. Once the bottleneck layer is reached, the decoder D mirrors the encoder, replacing pooling layers with upconvolutions to upsample width and height while downsampling the number of feature maps. Convolutional feature maps from D are concatenated with corresponding feature maps from E, followed by a $1x1$ convolution layer to match the desired decomposition image channels, normalized to the $[-1, 1]$ range.

Training Details: Training utilizes the Adam optimizer with batch size typically set to 32 and learning rates ranging from 0.01 to 0.001. L1 loss is employed as the error function, and training time varies based on image resolution and desired detail level, typically requiring less than 24 hours for the presented results in Section 4.

3.3 Rendering Equation

To compose the decomposition maps (vertices, normals, and textures) and create a shaded model, we utilize a series of rendering equations that consider both ambient and diffuse lighting components.

First, the ambient lighting contribution is calculated using the following equation, where the ambient reflection from the material and the ambient component of the light model are multiplied:

$$RE_A = \text{Ambient}_{\text{lightModel}} \times \text{Ambient}_{\text{material}} \tag{1}$$

Next, to account for the diffuse lighting, we compute the diffuse reflection using the equation below. This involves taking the dot product of the light vector and the vertex normal, clamping the result to a minimum of zero, and then multiplying by the diffuse components of both the light and the material:

$$\begin{aligned} RE_{\text{Diff}} = \max(&\mathbf{vector}_{\text{light}} \cdot \mathbf{normal}_{\text{vertex}}, 0) \\ &\times \text{Diff}_{\text{light}} \\ &\times \text{Diff}_{\text{material}} \end{aligned} \tag{2}$$

Finally, the vertex color is determined by combining the ambient and diffuse components, as shown in the equation below:

$$\text{VertexColor} = RE_A + RE_{\text{Diff}} \tag{3}$$

In summary, these equations work together to produce the final shaded model by considering how light interacts with the material properties at each vertex.

4 Results

In our experiments, we focus on exploring humanoid characters' morphology. We strategically position camera viewpoints around a sphere centered at the character's pelvis, mainly concentrating on the upper hemisphere for observation. Before sampling, we ensure that the data remains diverse and free from duplicates for symmetrical characters. This step ensures that our dataset accurately represents the variability within the characters. Following this, we carefully determine the sampling method and the number of viewpoints, which are crucial factors in shaping the outcome and accuracy of our experiments.

Control I: Camera View. We begin by conducting a series of experiments to determine the optimal number of camera viewpoints and the most effective sampling strategy. For this investigation, we utilize a high-quality static

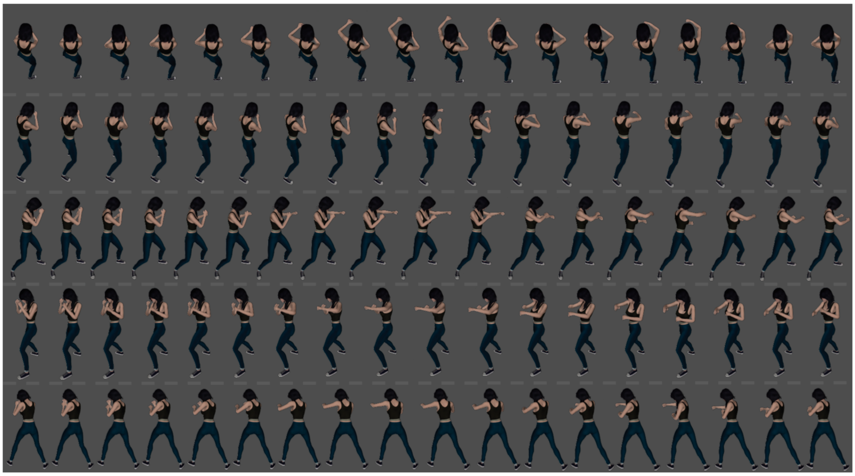


Fig. 3. Pose and Camera View Control: Punching sequence shown from 5 camera views (rows) with keyframes displayed across columns.

humanoid character of a woman (Figure 1), comprising 83,432 polygons (44,852 triangles). The image resolution is fixed at 128 × 128 pixels. The experiment is repeated five times, each time varying the geodesic levels from 1 to 5. The results are then analyzed based on the vertices of the geodesic at level 6 (Figure 5). Notably, we observe an increase in error, particularly evident around the poles and at greater distances from the original geodesic training samples. This discrepancy is visually apparent in Figure 5, where the training views are represented as clusters of blue color. Particularly on the geodesic of level 1, four distinct clusters correspond to the four camera views available in the training set only. To address the issue of increased error around the poles, an adaptive polar sampling approach may be suitable, wherein the viewpoints are denser towards the poles of the view sphere. However, our analysis reveals that increasing the number of viewpoints reduces the error around the poles. Therefore, to maintain uniformity across views and for simplicity, we adopt the uniform sampling method. Additionally, Figure 6 illustrates the training loss, indicating that using more viewpoints yields better results, as expected. Qualitative results, decomposed into vertices, normals, and textures, are presented in Figure 7.

Control II: Animation (Pose). By feeding a character's pose data into the ML model, we can animate the character with articulated movements. This approach allows us to encode animation information within the network. First, we decompose the desired animation into a series of key-frames. Then, for each key-frame, we obtain the corresponding 2D poses and additional data, as described in Section 3. However, generating animation data for numerous key-frames can be resource-intensive. To mitigate this, we employ a sampling strategy. Initially, we sample the animation at a frame rate of F_1 fps, discarding similar key-frames. Additionally, we vary the animation's starting point across different viewpoints

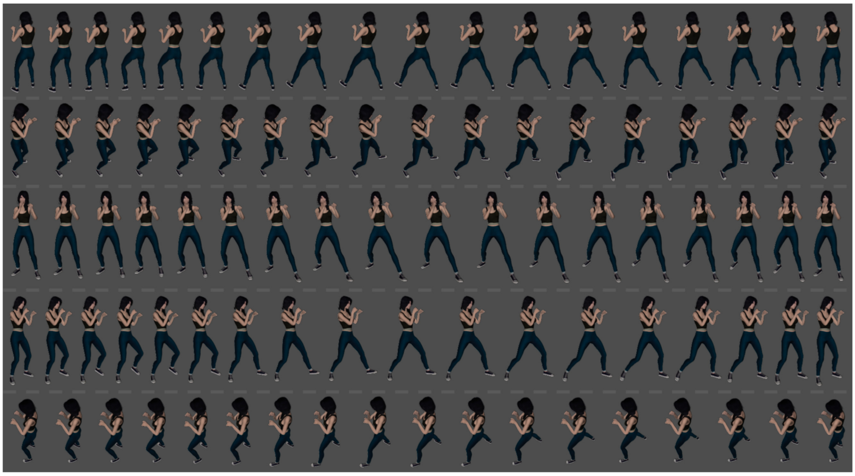


Fig. 4. Pose and Camera View Control: Walking sequence shown from 5 camera views (rows) with keyframes displayed across columns.

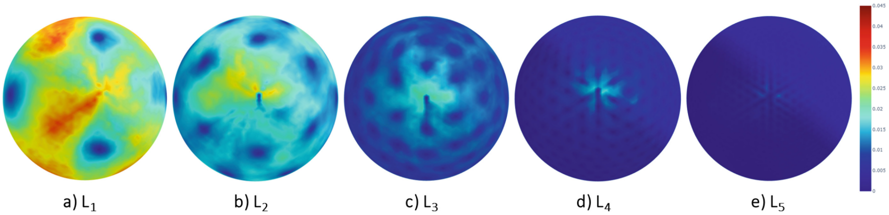


Fig. 5. Quantitative Evaluation: Losses for different numbers of views on the L1-L5 geodesics visualized on the camera dome sphere. As expected, the more views available during training, the smoother the resulting test interpolations.

to ensure robust reconstruction. For instance, by starting the animation sequence at slightly different offsets for each view, we effectively expose the model to intermediate frames between the original key-frames. We typically set F_1 to approximately 6 fps, a value that yields satisfactory results, as demonstrated in Figure 3 and Figure 4. The choice of F_1 can influence the model's animation interpolation properties. Moreover, to reduce the dataset size, we randomly discard a portion of data by assigning a dropout probability to each (keyframe, view) pair. Typically, we set this probability to 50%, halving the storage requirements. The results presented in Figure 3 and Figure 4 showcase animations of a virtual character performing punching and walking actions, respectively, from various viewpoints.

Control III: Character. In this section, we expand our model to represent multiple characters simultaneously and render them in a controllable manner, based on their IDs and from various viewpoints. To accomplish this, we initially select five female and five male characters from the MS Rocketbox 3D Avatar

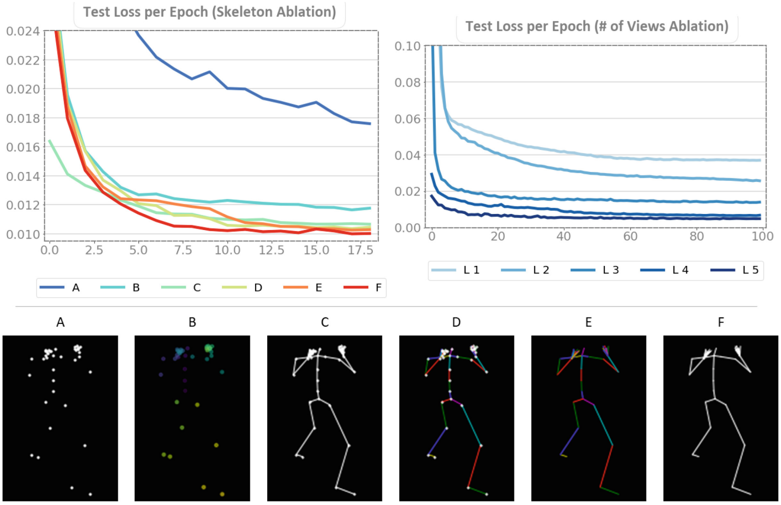


Fig. 6. Ablation Studies: *Top left - input control ablation:* test error curves over time for various input control (skeleton) representations are calculated using the L3 geodesic views. *Top right - camera views ablation:* test error curves over time for the L1-L5 geodesics camera setup with varied # of views. *Bottom:* Ablated skeleton representations (see top left for losses). We chose "E" since the distance between E and its corresponding dotted representation ("D") is low (compared to "C" and "F").

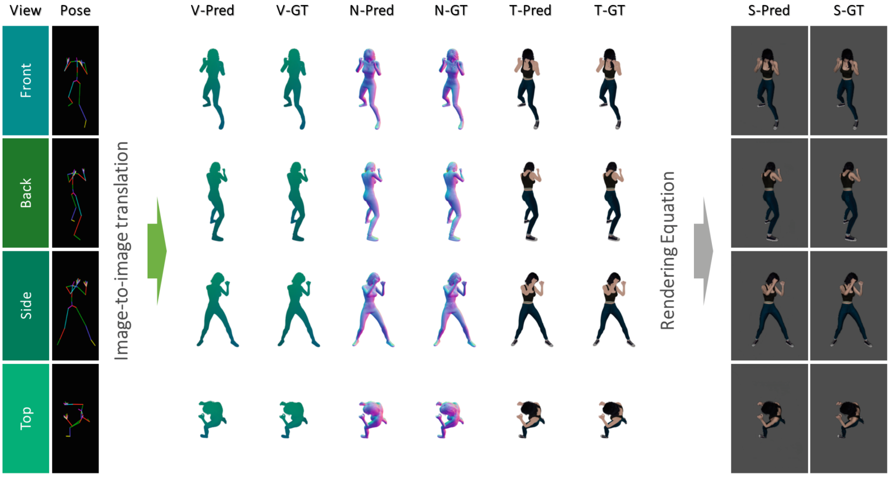


Fig. 7. Camera View & Animation (Pose) Control: Test Results showcasing camera view, 2D pose, vertices (predicted and ground truth), normals (predicted and GT), textures (predicted and GT), and the shaded model (predicted and GT).

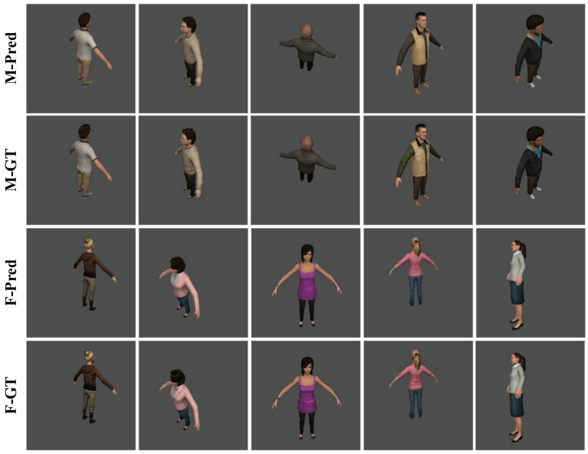


Fig. 8. Test Results for Rocketbox 10 Characters: Shaded models representing different characters are displayed in this figure. Characters are denoted as M for Male and F for Female. Our predictions are labeled as "Pred" while ground truth is "GT".

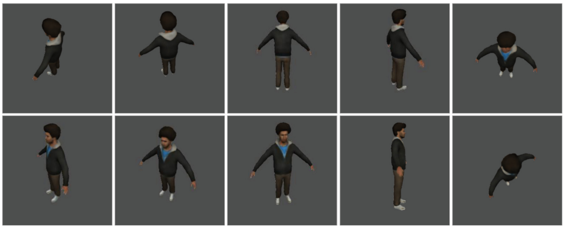


Fig. 9. One Character Across Multiple Views: This figure showcases consistent results of a single character rendered across multiple viewpoints.

dataset [39], totaling 10 characters for training. We extend our neural network architecture to include an additional input representing the character ID. This ID input, normalized to the range $[-1, 1]$, is transformed into a tensor with the same resolution as our desired output image and passed to the network's input layer. During experimentation, we observed that distinguishing between different characters in the MS Rocketbox dataset, which share similar skeletons but have varying appearances, posed a challenge for the network. Therefore, incorporating the character ID as a control parameter became necessary. To aid the network in distinguishing between characters, we adopt a progressive training approach. We begin by training the model on 32×32 image resolution and utilize these weights to initialize training for a 64×64 resolution model. Subsequently, we use the weights from the 64×64 model to initialize training for a 128×128 resolution model, and so forth up to a 256×256 output resolution. The model architecture remains unchanged throughout this process, as convolutional neural networks (CNNs) are resolution-independent. We showcase the model's ability to accurately distinguish between different characters by employing different character control IDs on the test set, as illustrated in Figure 8. Additionally, we present renderings of a single character from multiple viewpoints in Figure 9, where predictions remain consistent without any noticeable artifacts. However, it is worth noting that our current model may not capture all details present in the images, as seen in Figure 8. For instance, certain intricate details like stripes on a character's t-shirt may not be accurately reconstructed. This limitation could potentially be addressed by increasing the number of feature maps in our neural network, as discussed in Section 3. Nevertheless, for applications involving crowds comprising many small/distant characters, such adjustments may not be necessary and could significantly extend the training time.

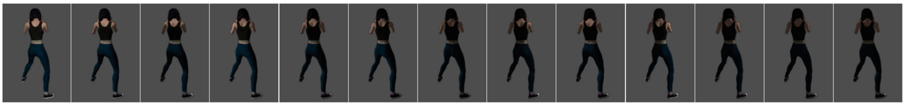


Fig. 10. Illumination Control: This figure illustrates the manipulation of illumination by adjusting the light position in the rendering equation (Eqn. 3). By varying the position of the light source, different lighting effects can be achieved, providing control over the illumination of the scene.

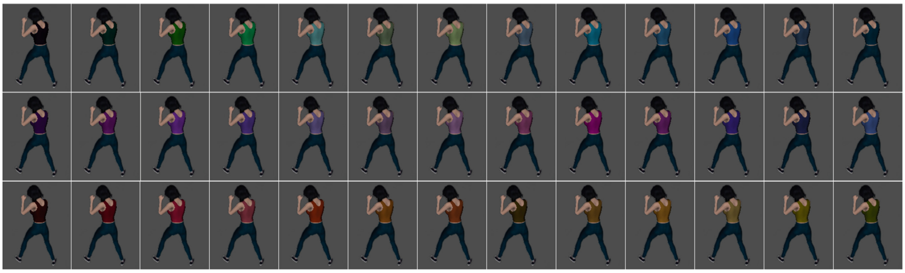


Fig. 11. Material Control: This figure demonstrates the manipulation of material properties using the rendering equation (Eqn. 3). By adjusting the equation's parameters, finer control over desired material properties can be achieved, even on specific body parts. Here, the material of the top garment is randomly altered, showcasing the versatility of the approach.

Control IV: Illumination. Illumination control is facilitated by leveraging the classical graphics rendering equation, as outlined in Section 3. Since our network predicts the decomposition of an image containing both vertex and normal information, lighting calculations can be performed. Figure 10 demonstrates this capability with different basic directional lighting configurations applied to a neuropostor.

Control V: Materials. Another property enabled by utilizing the rendering equation is explicit control over the materials. This can be achieved by masking certain parts of a character (e.g., crop top in Figure 11) and adjusting its material properties. In Figure 11, we demonstrate this by randomly changing the material, resulting in a variety of colors.

Memory Efficiency. Neuropostors excel in memory efficiency with their neural representation, offering flexible rendering and reduced memory requirements. Polypostors [7] convert 3D characters to 2D textured polygons, minimizing texture memory but potentially introducing artifacts in complex animations or overhead views. Geopostors balance detail and performance by using 3D rendering for close characters and 2D impostors for distant ones, resulting in moderate memory savings. Table 3 details memory usage across different feature map counts in Neuropostors, highlighting the trade-off between memory and visual fidelity. Adjusting feature maps for larger crowds with distant characters is straightforward, yielding significant memory savings without compromising visual quality or interpolation capabilities of the system.

Table 3. Memory consumption for different numbers of feature maps (N is set to 32 in the paper). N denotes the number of feature maps in the first convolutional block, which is doubled in each subsequent block. Lower values of N can be chosen for larger crowds where characters are more distant, balancing memory usage and visual fidelity.

No. of Feature Maps	32	16	8	1
Memory Usage	31.1 MB	7.82 MB	1.99 MB	72.8 kB

5 Conclusion

We introduced Neuropostors, a novel approach that merges classical graphics with neural rendering techniques for crowd simulation. Our method accurately decomposes animated characters into intrinsic 2D maps, offering precise control over illumination and material properties. By compressing multiple characters into a single neural network and enabling viewpoint and animation interpolation, Neuropostors deliver a rendering solution with constant speed, independent of mesh polygon count, optimizing hardware resource use.

Acknowledgements:. This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska Curie grant agreement No 860768 (CLIPE project).

References

1. Simon Dobbyn, John Hamill, Keith O'Conor, and Carol O'Sullivan. Geopostors: a real-time geometry / impostor crowd rendering system. In *Proceedings of the 2005*

Symposium on Interactive 3D Graphics and Games, I3D '05, page 95-102, New York, NY, USA, 2005. Association for Computing Machinery

2. Epic Games. Announcing metahuman creator: Fast high-fidelity digital humans in unreal engine, 2021. Accessed 2022-01-27
3. Tecchia, F., Loscos, C., Chrysanthou, Y.: Image-based crowd rendering. IEEE Comput. Graphics Appl. **22**(2), 36–43 (2002)
4. Tewari, A., Fried, O., Thies, J., Sitzmann, V., Lombardi, S., Sunkavalli, K., Martin-Brualla, R., Simon, T., Saragih, J., Nießner, M., Pandey, R., Fanello, S., Wetzstein, G., Zhu, J.-Y., Theobalt, C., Agrawala, M., Shechtman, E., Goldman, D.B., Zollhöfer, M.: State of the art on neural rendering. Computer Graphics Forum **39**(2), 701–727 (2020)
5. Daniel Thalmann and Soraia Raupp Musse. *Crowd simulation*. Springer, 2007
6. William Herbert Sheldon. The varieties of human physique. *Encyclopedia Britannica*, 2024
7. Ladislav Kavan, Simon Dobbyn, Steven Collins, Jiří Žára, and Carol O'Sullivan. Polypostors: 2d polygonal impostors for 3d crowds. In *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games*, I3D '08, page 149-155, New York, NY, USA, 2008. Association for Computing Machinery
8. Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014
9. Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 6 2014. PMLR
10. Uria, B.: Marc-Alexandre Côté. Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. CoRR, Karol Gregor (2016)
11. Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1747-1756. JMLR.org, 2016
12. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc., 2014
13. Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016
14. Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821, 2017
15. Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017
16. Jianwei Yang, A. Kannan, Dhruv Batra, and Devi Parikh. Lr-gan: Layered recursive generative adversarial networks for image generation. ArXiv, abs/1703.01560, 2017
17. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Adv. Neural. Inf. Process. Syst. **33**, 6840–6851 (2020)
18. Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502, 2020

19. Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022
20. Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014
21. Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 2642-2651. JMLR.org, 2017
22. Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV (4)*, volume 9908 of *Lecture Notes in Computer Science*, pages 776–791. Springer, 2016
23. Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1060–1069. JMLR.org, 2016
24. Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor S. Lempitsky. Few-shot adversarial learning of realistic neural talking head models. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9458–9467, 2019
25. Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9416–9425, 2018
26. Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6836–6845, 2017
27. Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch your own GAN. *CoRR*, abs/2108.02774, 2021
28. Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017
29. Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018
30. Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017
31. Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1520–1529, 2017
32. Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016
33. Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017
34. Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023

35. Carlo Innamorati, Tobias Ritschel, Tim Weyrich, and Niloy J. Mitra. Decomposing single images for layered photo retouching. *Computer Graphics Forum (Proc. Eurogr. Symp. on Rendering)*, 36(4):15–25, 7 2017
36. Sun, T., Barron, J.T., Tsai, Y.-T., Zexiang, X., Xueming, Yu., Fyffe, G., Rhemann, C., Busch, J., Debevec, P., Ramamoorthi, R.: Single image portrait relighting. ACM Trans. Graph. **38**(4), 7 (2019)
37. Chen, X., Cohen-Or, D., Chen, B., Mitra, N.J.: Towards a neural graphics pipeline for controllable image generation. Comput. Graph. Forum **40**(2), 127–140 (2021)
38. O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015
39. Mar Gonzalez-Franco, Eyal Ofek, Ye Pan, Angus Antley, Anthony Steed, Bernhard Spanlang, Antonella Maselli, Domna Banakou, Nuria Pelechano, Sergio Orts-Escolano, Veronica Orvalho, Laura Trutoiu, Markus Wojcik, Maria V. Sanchez-Vives, Jeremy Bailenson, Mel Slater, and Jaron Lanier. The rocketbox library and the utility of freely available rigged avatars. *Frontiers in Virtual Reality*, 1, 2020

InjectionNet: Realizing Information Injection for Medical Image Segmentation with Layer Relationships

Xinyu Zhu, JiaFeng Li, and Yin Wen(✉)

East China Normal University, Shanghai, China
{51255904040,51205904113}@stu.ecnu.edu.cn, ywen@cs.ecnu.edu.cn

Abstract. In current medical image segmentation tasks, the combined transformer and convolutional architectures excel in capturing global cues and local details, but still pose two main concerns from a layer-level perspective: (1) intra-layer issue: the existing methods inefficiently obtain and fuse global-local information, potentially resulting in incomplete feature extraction; (2) inter-layer issue: the most of methods follow the classical U-shape structure, which inevitably leads to information weakening in the encoder-decoder. In light of these, we propose InjectionNet from the perspective of layers, mainly comprising the Intra-layer Global-Local Injection (GLI) module and Inter-layer Weight Injection (WI) modules. GLI employs multi-scale convolution for local information extraction and flexibly uses a multi-head self-attention mechanism for efficiently capturing global information and fusing them effectively. WI enhances information transfer by injecting generated feature weights, with different variants to suit various network stages. Extensive experiments on three medical imaging public datasets demonstrate the superior performance of InjectionNet compared to previous works.

Keywords: CNNs · ViTs · Inter-layer · Intra-layer · Medical Image Segmentation

1 Introduction

Medical image segmentation is crucial for diagnosis and treatment decisions. Convolutional neural network (CNN)-based models such as UNet [1] and its variants have achieved remarkable success in medical image processing. However, CNNs can only share information within limited local regions and thus lack the capacity to establish long-range dependencies, which may be necessary for image segmentation tasks.

Visual transformers (ViTs) have made a significant impact on computer vision, excelling at modeling global relationships. Consequently, several studies have focused on hybrid CNN-Transformer architectures to combine global and local information, enhancing feature representation. For example, TransUNet [4]

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 449–460, 2025.
https://doi.org/10.1007/978-3-031-78312-8_30

adds the transformer to the high-level features of CNNs to learn global information. Similarly, SwinUNet [5] and CMT [6] combine CNNs and transformers through serial fusion, obtaining a better trade-off in accuracy and efficiency. Another popular architectures, e.g. TransCeption [7] fuses global and local information through concatenation operations. TransFuse [8] proposes the BiFusion module to fuse global dependencies and spatial details in a parallel manner. Additionally, some works such as ScaleFormer [10] and TransWnet [11] propose mitigating solutions to the problem of information weakening in the encoder part of a typical U-shaped structure. Despite the promising results of these works, three limitations persist: (***i***) global and local feature representations are not simultaneously captured at the same feature level, so the extracted feature representations may be incomplete. Medical images contain abundant local details and global cues, so how to extract the global-local information is important to obtain a comprehensive feature representation of the image. (***ii***) global and local information have differences. Therefore, it is essential to find an effective fusion method to alleviate the information gap between them and to improve the efficiency of modeling the global context while maintaining a firm grasp of the local details. (***iii***) to enhance information flow between neighboring layers, traditional methods typically use concatenation operation to reuse information from neighboring layers in the encoder. However, they may not fully realize that it is also equally important to enhance the information flow in the decoder. Moreover, concatenation operations may not be fully effective. Therefore, how to effectively mitigating information weakening to assist the encoding-decoding process becomes a problem.

To address these important issues, we design a new encoder-decoder network, called InjectionNet, from the perspective of layer relationships. (***i***) in the intra-layer, we design a Global-Local Injection (GLI) module, which aims to combines global-local information from the input feature map to achieve richer feature representations. The Global Extraction (GE) and Multi-Scale Mixed Convolution (MSMC) modules in GLI capture the global and local information respectively, and the Injection approach in GLI fuses the information from the GE and MSMC modules to obtain a more comprehensive feature representation. (***ii***) to alleviate the information flow weakening during the encoding-decoding process in the network, we propose different Inter-layer Weight Injection (WI) modules for different network phases. This includes Encoder Weight Injection (E-WI) during the encoder stage, and Decoder Weight Injection(D-WI) and Final Decoder Weight Injection (FD-WI) at the decoder stage.

In general, our contributions are three-folded: (1) We propose the Global-Local Injetion(GLI) module in intra-layers, which efficiently extracts and fuses global-local information. (2) We propose different Inter-layer Weight Injection(WI) modules in inter-layers to address the problem of information weakening during encoding-decoding. (3) We propose a novel network InjectionNet from the perspective of layer relations and verify its effectiveness on three public medical imaging datasets.

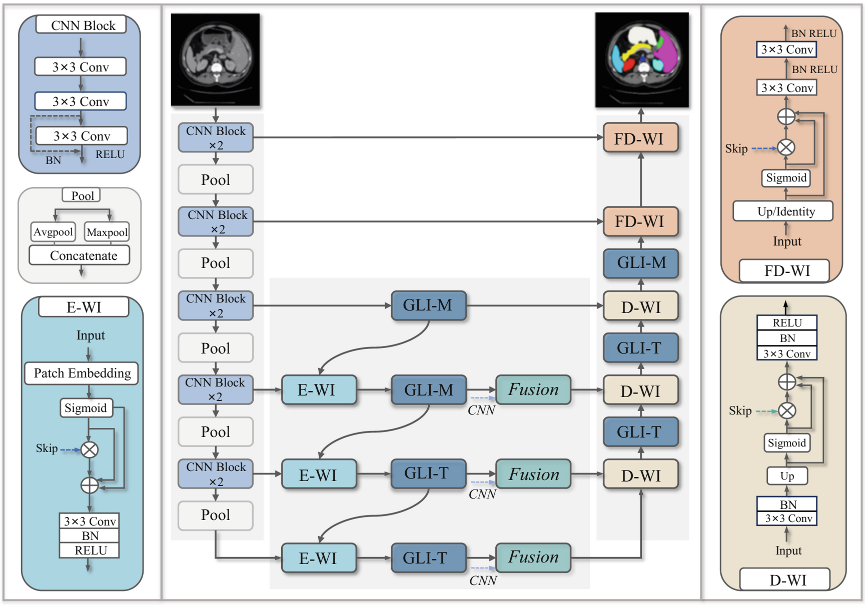


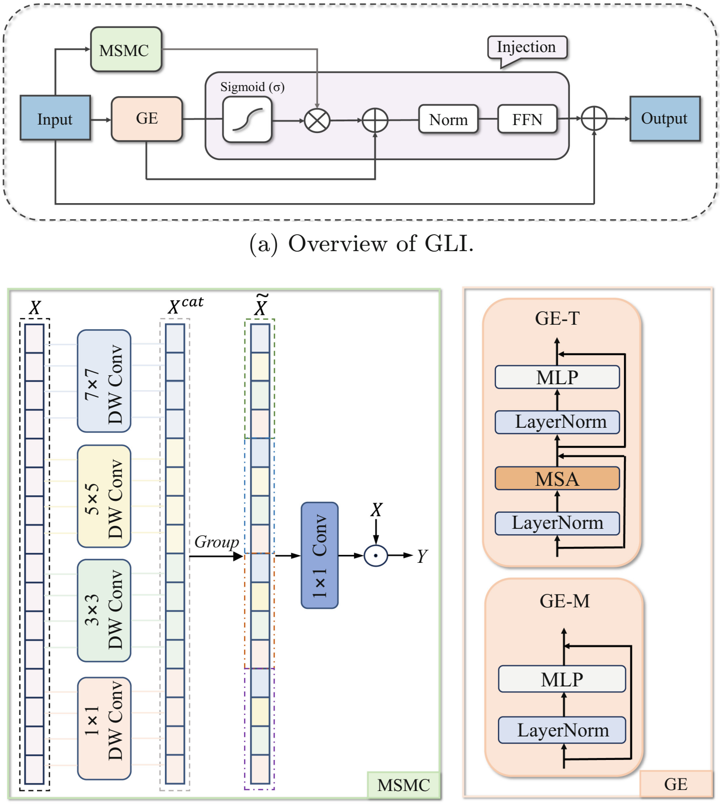
Fig. 1. Overview of InjectionNet. First, the image fuses the generated global cues and local details by GLI. Then, the interaction of upper layer information with lower layer information is realized by WI. Finally, the enhanced output is further combined with CNN features of the same scale and sent to the corresponding decoder block.

2 Methods

Overall Pipeline The proposed InjectionNet extracts global-local features within layers and enhances information flow between layers as shown in Fig. 1. The encoder utilizes the five CNN blocks for hierarchical extraction of local features, reducing feature map resolution by both max-pooled and average-pooled. The GLI module(in Sec.2.2), including GLI-M and GLI-T, extracts global and local information within layers and facilitates cross-semantic interaction between them. Then, the features enhanced by GLI are injected into the next layer by the E-WI module(in Sec.2.3). The Fusion module(in Sec.2.4) complements features with CNNs at the same feature level before transmission to the decoder. In the decoder, GLI modules are reused in separate layers, connected by corresponding WI modules(in Sec.2.3) for the final prediction. In the following, we describe each module in InjectionNet in detail.

2.1 Intra-layer Global-Local Injection

As depicted in Fig. 2(a), GLI comprises three key parts: Global Extraction (GE), Multi-Scale Mixed Convolution (MSMC), and Injection. We will describe these three parts as follows.



(a) Overview of GLI.

(b) Detailed architecture of MSMC and GE.

Fig. 2. (a) Overview of GLI. (b) MSMC denotes the Multi-Scale Mixed Convolution module, which is used to extract the local information of the features. GE denotes the Global Extraction module, which is used to extract the global information of the features.

Global Extraction. The Global Extraction (GE) module captures global information from the feature map. Recent studies such as [9] has shown that the multi-head self-attention (MSA) mechanism is less sensitive to capture global information in the early stages of the model, and there also exists a quadratic complexity problem in computing high-resolution image. To address these issues, as shown in Fig. 2(b), we employ the GE-M Block at the shallow layer to learn the global context only by MLP. Moreover, using MSA in the later stages is quite efficient. Therefore, for deeper layers, the GE-T Block strikes a balance between computational efficiency and maintaining the ability of MSA to capture long-range dependencies. GLI-M and GLI-T refer to embedding GE-M and GE-T in the GLI module, respectively.

Multi-Scale Mixed Convolution. Recognizing the limitations of single-scale convolution, we introduce Multi-Scale Mixed Convolution (MSMC) to extract spatial features at different scales in the same feature layer. As shown in Fig. 2(b), MSMC divides the input feature map into equal parts along the channel dimension (default is 4 parts) and utilizes multi-scale convolution kernels. Here we use depth-wise separable convolutions to reduce computational overhead. For larger resolutions, larger kernel sizes are favored, while smaller kernel sizes are sufficient to capture localized details. Therefore, in GLI-M, we incorporate branches with convolution kernel sizes of 5×5, 7×7, and 9×9 alongside the original 3×3 kernel. In GLI-T, we introduce additional branches with 1×1, 5×5, and 7×7. Inspired by [10], we shuffle multi-scale spatial information through a Group operation. Channels from each part are aggregated through a convolutional layer and modulate input features through the product, enhancing modeling capability. The output Y is expressed as follows:

$$X^{cat} = Concat\left(DW_{k_1 \times k_1}(X^1), \ldots, DW_{k_n \times k_n}(X^N)\right) \tag{1}$$

$$\tilde{X} = Group(X^{cat}) \tag{2}$$

$$Y = X \odot Conv_{1\times 1}(\tilde{X}) \tag{3}$$

where $X = [X^1, X^2, \ldots, X^N]$ means to split up the input feature $X \in \mathbb{R}^{H\times W\times C}$ channel C equally into N parts. k_n denotes the depth-wise separable convolutions kernel size. $Group(\cdot)$ indicates the Group operation. $\odot$ is the element-wise multiplication.

Injection operation. To endow multi-scale low-level spatial features with rich high-level global semantic information, we utilize an Injection operation. As illustrated in Fig. 2(a), we first input the global information into a sigmoid layer to generate semantic weights, which are then multiplied with multi-scale local information to dynamically adjust the attention of local information according to the relative importance of global information, while preserving the original global information. Subsequently, passing through the normalization and FFN layers. Also, shortcut connection is adopted.

2.2 Inter-layer Weight Injection Modules

Lower-layer information is typically derived from upper-layer information, so features from neighboring layers can compensate for missing information during transfer. To facilitate cross-layer interactions, we introduce WI modules (see Fig. 1), including Encoder Weight Injection (E-WI), Decoder Weight Injection (D-WI) and Final Decoder Weight Injection (FD-WI). We will describe each of these modules.

E-WI. During the encoder stage, we use the E-WI module for inter-layer information interaction. This involves inputting information from different layers: one skip branch from the CNN block and the other input branch is derived from the upper-layer feature map, which is enhanced by the corresponding GLI module. Input branch is first scaled according to the size of the skip branch to ensure proper alignment. Then, a sigmoid function is applied to the aligned input branch information to obtain attention weights. These attention weights are then combined with the CNN branch to augment the perception of crucial information in the upper layer and to focus more specifically on different regions of the input image. To minimize the weakening of information during transmission, to the greatest extent, two shortcuts are added at the end. Let i index the downsampling layer along the GLI branch, N denotes the layer of GLI, and the E-WI process is:

$$\begin{aligned} EWI(X_{cnn}^{i+1}, X_{GLI}^{i}) = Conv_{3\times 3}(X_{cnn}^{i+1} \odot \sigma(Down(X_{GLI}^{i})) \\ + Down(X_{GLI}^{i}) + \sigma(Down(X_{GLI}^{i}))) \end{aligned} \tag{4}$$

where EWI denotes the E-WI module, X_{cnn}^{i+1} denotes the branched input from the *(i+1)-th* layer CNN block, X_{GLI}^{i} denotes the *i-th* layer representation enhanced by the GLI module. $Down(\cdot)$ denotes the patch embedding, which is realized by a convolution with stride of 2 and followed by a batch normalization and a ReLU activation function. σ denotes the sigmoid function. $\odot$ is the element-wise multiplication.

D-WI. In the early stage of the decoder, we use the D-WI for inter-layer information interaction. As shown in Fig. 1, by first changing the channel through a convolutional layer, we find information from the decoder stage is important for accurate prediction, therefore, to avoid information loss, we did not use the RELU activation function for the input branch. The feature maps of the two branches are then correctly aligned by bilinear interpolation. Then the information of the skip branch is weighted according to the obtained attention weights based on the same attention mechanism as in E-WI.

FD-WI. In the final stage of the decoder, we use the FD-WI module as an information fusion method, which takes the features from the previous level and the initial CNNs. As shown in Fig. 1, slightly different from D-WI, the convolutional layer is moved to the end of the module.

2.3 Fusion

Following enhancement by inter-layer E-WI and intra-layer GLI modules, we aggregate the improved representation X' with CNNs at the same scale level. Potential information redundancy can significantly affect segmentation performance. Our Fusion module is designed to alleviate this problem. The process involves refining sophisticated channel information into high-quality data and subsequently integrating it with corresponding CNN branches. The complete formulation is:

$$\begin{cases} X_{AG} = X' \odot \sigma(Conv_{1\times 1}(X_{cnn} + X')), \\ Fusion = Conv_{1\times 1}([X_{cnn}, X_{AG}]). \end{cases} \tag{5}$$

where X' is the augmented representation, X_{cnn} comes form the CNNs features, σ denotes the sigmoid function, $\odot$ denotes element-wise multiplication and $[\cdot]$ denotes concatenation.

2.4 Loss Function

To optimize our segmentation model, we employ the combined *DICE* (L_{DICE}) and *CrossEntropy* (L_{CE}). The dice-coefficient loss has high flexibility towards class imbalance, while the cross-entropy loss helps with the curve smoothing.

$$L_{DICE}(\hat{y}, y) = 1 - \frac{2y\hat{y} + \lambda}{y + \hat{y} + \lambda} \tag{6}$$

$$L_{CE}(\hat{y}, y) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i) \tag{7}$$

Here, y is the ground truth value, and $\hat{y}$ is the predicted value. And λ is added to the numerator and denominator for numerical stability. We aggregate all estimated losses using the following function to compute the final loss.

$$L_{total} = \gamma L_{DICE} + (1 - \gamma)L_{CE} \tag{8}$$

where $\gamma = 0.5$ and $1 - \gamma = 0.5$ are the weights for L_{DICE} and L_{CE}, respectively.

3 Experiments

3.1 Dataset

Synapse datasets. Synapse consists 30 abdominal CT scans. Following [2], we split 18 cases for training and remaining 12 cases for testing. We reported the Dice Coefficient (DSC) and Hausdorff Distance (HD) on 8 different organs.
Automated cardiac diagnosis (ACDC) datasets. The automated cardiac diagnosis challenge contains 100 MRI scans involving three organs: myocardium (MYO), right ventricle (RV), and left ventricle (LV). Consistent with [17], we present the DSC results using a random split of 70 training cases, and 30 testing cases.
ISIC2018 datasets. It is a skin lesion segmentation dataset, consisting of 2594 images and corresponding labels. We randomly divide the dataset into 1816, 260, and 518 for training, validation, and testing, respectively.

3.2 Implementation Details

Our InjectionNet is implemented based on PyTorch and trained on NVIDIA Tesla A100 GPU. We set the input image size as 224×224. Here, we list the batch size (bs), learning rate (lr), maximum training epochs (ep), optimizer (opt) for three datasets:

- Synapse: bs=8; lr=7e-3; ep=800; opt=SGD;
- ACDC: bs=8; lr=4e-3; ep=800; opt=SGD;
- ISIC2018: bs=8; lr=1e-4; ep=400; opt=Adam;

All models were trained with momentum 0.9 and weight decay 1e-4. For fair comparison, we used the same settings and combined cross entropy loss and dice loss for all experiments.

Table 1. Comparison to state-of-the-art (SOTA) methods on Synapse dataset. The best results are highlighted in Bold fonts.

Methods	DSC(*uparrow*)	HD(*downarrow*)	Aorta	Gallbladder	Kidney(L)	Kidney(R)	Liver	Pancreas	Spleen	Stomach
V-Net [11]	68.81	-	75.34	51.87	77.10	80.75	87.84	40.05	80.56	56.98
U-Net [1]	76.85	39.70	89.07	69.72	77.77	68.60	93.43	53.98	86.67	75.58
AttUNet [12]	77.77	36.02	**89.55**	68.88	77.98	71.11	93.57	58.04	87.30	75.75
TransUNet [2]	77.48	31.69	87.23	63.13	81.87	77.02	94.08	55.86	85.08	75.62
SwinUNet [3]	79.12	21.55	85.47	66.53	83.28	79.61	94.29	56.58	90.66	76.60
UCTransNet [13]	78.23	26.75	88.86	66.97	80.19	73.18	93.17	56.22	87.84	79.43
MTUnet [14]	78.59	26.59	87.92	64.99	81.47	77.29	93.06	59.46	87.75	76.81
HiFormer [15]	80.69	19.14	87.03	68.61	84.23	78.37	94.07	60.77	90.44	82.03
MissFormer [16]	81.96	18.20	86.99	68.65	85.21	82.00	94.41	65.67	91.92	80.81
CASTformer [17]	82.55	22.73	89.05	67.48	86.05	82.17	**95.61**	67.49	91.00	81.55
ScaleFormer [7]	82.86	16.81	88.73	**74.97**	**86.36**	83.31	95.12	64.85	89.40	80.14
InjectionNet	**85.10**	**16.28**	89.13	72.91	86.32	**85.38**	95.08	**73.40**	**92.48**	**86.07**

Table 2. Comparison to SOTA methods on ACDC dataset.

Methods	DSC(*uparrow*)	RV	Myo	LV
U-Net [1]	87.55	87.10	80.63	94.92
AttUNet [12]	86.75	87.58	79.20	93.47
TransUNet [2]	89.71	88.86	84.53	95.73
SwinUNet [3]	90.00	88.55	85.62	95.83
MissFormer [16]	90.86	89.55	88.04	94.99
ScaleFormer [7]	90.17	87.33	88.16	95.04
InjectionNet	**91.97**	**89.84**	**89.97**	**96.09**

3.3 Comparison with State-of-the-Art Methods

Quantitative Comparison: To verify the effectiveness of the proposed InjectionNet, we compare it with 11 state-of-the-art (SOTA) networks on the Synapse dataset, including V-Net [11], U-Net [1], AttUNet [12], TransUNet [2], SwinUNet [3], UCTransNet [13], MTUnet [14], HiFormer [15], MissFormer [16], CASTformer [17], and ScaleFormer [7]. The experimental results are shown in Table 1, using the Dice Coefficient (DSC) and Hausdorff Distance (HD) as evaluation metrics. It is clear that the proposed InjectionNet outperforms all other methods in both regional measures DSC (85.10%) and boundary-aware measure HD(16.28mm). The quantitative results make us believe that InjectionNet excels in precisely localizing small objects, notably surpassing the prior leading method ScaleFormer. For instance, InjectionNet enhances the DSC score in the stomach and pancreas by 5.93% and 8.55%, respectively. Table 2 shows DSC scores on the ACDC dataset, with InjectionNet achieving the highest average DSC score among all methods. For instance, compared to the ScaleFormer, the InjectionNet improves the average Dice by 1.8% and achieves consistent improvements on all three individual classes, demonstrating the superior performance.

Table 3. Comparison to SOTA methods on ISIC2018 dataset.

Methods	DSC($uparrow$)	SE($uparrow$)	SP($uparrow$)	ACC($uparrow$)
U-Net [1]	85.45	88.00	96.97	94.04
AttUNet [12]	85.66	86.74	98.63	93.76
TransUNet [2]	84.99	85.78	96.53	94.52
SwinUNet [3]	89.46	90.56	97.98	96.45
TMU-Net [18]	90.59	90.38	97.46	96.03
TransCeption [5]	91.24	91.92	97.44	96.28
InjectionNet	**92.40**	**93.63**	**97.68**	**96.87**

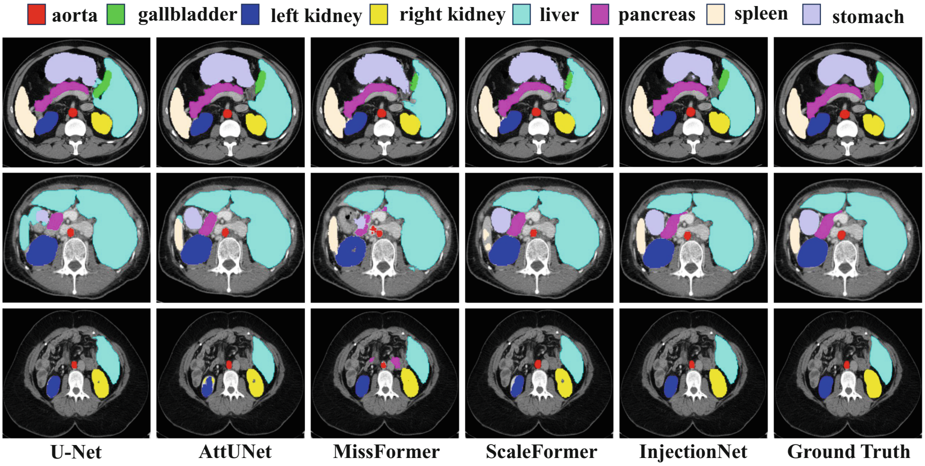


Fig. 3. Qualitative results of different models on Synapse dataset. InjectionNet achieves superior performance.

To further validate the generalization of InjectionNet, we evaluate InjectionNet on the ISIC 2018 skin lesion dataset and compare the results with other SOTA methods. As shown in Table 3, InjectionNet achieves the best performance in DSC (92.40%), Sensitivity (93.63%), Specificity (97.68%), and Accuracy (96.87%). Notably, InjectionNet outperforms the pure-transformer network, TransCeption, with a 1.16% improvement in DSC. These quantitative results on three datasets substantiate the fine effectiveness and generality of our InjectionNet.

Visual Comparison: In Fig. 3, we illustrate the qualitative results of different methods on the Synapse dataset, including UNet, AttUNet, MissFormer, and ScaleFormer. The segmentation results of InjectionNet closely align with the ground truth, demonstrating accurate and comprehensive organ segmentation across diverse sizes, shapes, and locations.

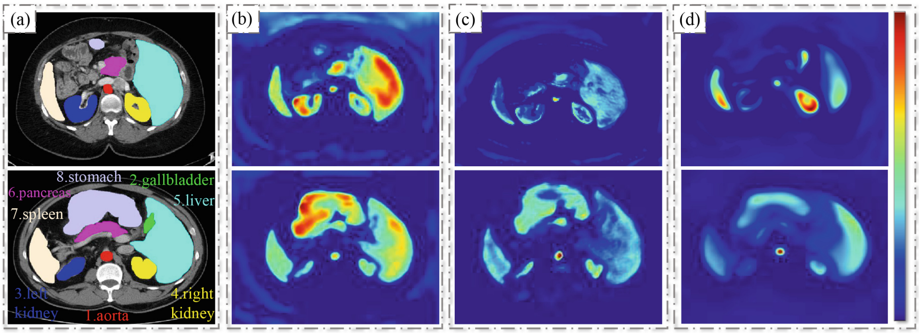


Fig. 4. Visualization of feature maps for various methods of inter-layer information interaction on Synapse dataset. (a) represents the ground truth and (b) represents the feature map from InjectionNet. (c) and (d) indicate no inter-layer WI modules in encoder and decoder respectively.

3.4 Ablation Study

We perform ablation experiments on the Synapse dataset for both intra-layer GLI module and inter-layer WI module, the experimental results are shown in Table 4. We first try to remove the GLI module and change its global-local information fusion method to verify the effectiveness of our design. With (No.3-5), we find that global-local information fusion is necessary and the proposed Injection method improves the DSC compared to simple concatenation and summation operations. The effectiveness of the proposed WI module is evident in mitigating information transmission weakening during both encoder stages (No.6-8) and decoder stages (No.9-11). Compared to basic operations, this enhancement contributes to an increased Dice coefficient. Finally, the necessity of both inter-layer (No.3) and intra-layer (No.2) information interactions is validated, underscoring the indispensability of our proposed modules. Notably, a significant performance drop occurs when the decoder is ×, attributed to the absence of the initial CNNs branch in the final decoder stage, an issue addressed by UNet [1].

Additionally, we visualize the feature map from the final decoder in Fig. 4. Compared to scenarios without inter-layer WI modules in the encoder-decoder stage (depicted in (c) and (d), respectively), InjectionNet exhibits more effective activation of the target region, successfully addressing the issue of information weakening.

Table 4. Ablation studies on the Synapse dataset. '*cat*', '*add*' denotes corresponding connection methods, '✓' denotes our proposed methods, '×' denotes not applicable.

Ver	GLI	Encoder	Decoder	DSC(*uparrow*)	HD(*uparrow*)
No.1	×	×	×	72.15	16.44
No.2	✓	×	×	72.51	15.49
No.3	×	✓	✓	84.25	16.79
No.4	*cat*	✓	✓	84.32	18.78
No.5	*add*	✓	✓	84.63	12.19
No.6	✓	*cat*	✓	84.32	17.39
No.7	✓	*add*	✓	84.43	16.54
No.8	✓	×	✓	84.10	14.94
No.9	✓	✓	*cat*	83.57	21.01
No.10	✓	✓	*add*	83.93	15.82
No.11	✓	✓	×	73.33	13.43
No.12	✓	✓	✓	85.10	16.28

4 Conclusion

In this work, we propose a layer-relationship-based network for medical image segmentation, called InjectionNet. Within the layers, a GLI module rationally utilizes the MSA mechanism, employing multi-scale convolutional kernels to extract global-local information and combine them effectively. Besides, various WI modules enhance the information flow by injecting the weights of neighboring layers. Our experimental results consistently show that InjectionNet outperforms other existing methods across three different public datasets.

Acknowledgements. This work was supported in part by the National Nature Science Foundation of China (62273150) and Shanghai Natural Science Foundation (22ZR1421000).

References

1. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.
2. J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "Transunet: Transformers make strong encoders for medical image segmentation," arXiv preprint arXiv:2102.04306
3. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022

4. J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "Cmt: Convolutional neural networks meet vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 175–12 185
5. R. Azad, Y. Jia, E. K. Aghdam, J. Cohen-Adad, and D. Merhof, "Enhancing medical image segmentation with transception: A multi-scale feature fusion approach," arXiv preprint arXiv:2301.10847, 2023
6. Zhang, Y., Liu, H., Hu, Q.: TransFuse: Fusing Transformers and CNNs for Medical Image Segmentation. In: de Bruijne, M., Cattin, P.C., Cotin, S., Padoy, N., Speidel, S., Zheng, Y., Essert, C. (eds.) MICCAI 2021. LNCS, vol. 12901, pp. 14–24. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87193-2_2
7. H. Huang, S. Xie, L. Lin, Y. Iwamoto, X.-H. Han, Y.-W. Chen, and R. Tong, "Scaleformer: Revisiting the transformer-based backbones from a scale-wise perspective for medical image segmentation."
8. Y. Xie, Y. Huang, Y. Zhang, X. Li, X. Ye, and K. Hu, "Transwnet: Integrating transformers into cnns via row and column attention for abdominal multi-organ segmentation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5
9. Pan, Z., Zhuang, B., He, H., Liu, J., Cai, J.: Less is more: Pay less attention in vision transformers. Proceedings of the AAAI Conference on Artificial Intelligence **36**(2), 2035–2043 (2022)
10. W. Lin, Z. Wu, J. Chen, J. Huang, and L. Jin, "Scale-aware modulation meet transformer," arXiv preprint arXiv:2307.08579, 2023
11. F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*. Ieee, 2016, pp. 565–571
12. O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention u-net: Learning where to look for the pancreas," arXiv preprint arXiv:1804.03999, 2018
13. Wang, H., Cao, P., Wang, J., Zaiane, O.R.: Uctransnet: rethinking the skip connections in u-net from a channel-wise perspective with transformer. Proceedings of the AAAI conference on artificial intelligence **36**(3), 2441–2449 (2022)
14. H. Wang, S. Xie, L. Lin, Y. Iwamoto, X.-H. Han, Y.-W. Chen, and R. Tong, "Mixed transformer u-net for medical image segmentation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 2390–2394
15. M. Heidari, A. Kazerouni, M. Soltany, R. Azad, E. K. Aghdam, J. Cohen-Adad, and D. Merhof, "Hiformer: Hierarchical multi-scale representations using transformers for medical image segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 6202–6212
16. X. Huang, Z. Deng, D. Li, and X. Yuan, "Missformer: An effective medical image segmentation transformer," arXiv preprint arXiv:2109.07162, 2021
17. C. You, R. Zhao, F. Liu, S. Dong, S. Chinchali, U. Topcu, L. Staib, and J. Duncan, "Class-aware adversarial transformers for medical image segmentation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 29 582–29 596, 2022
18. R. Azad, M. Heidari, Y. Wu, and D. Merhof, "Contextual attention network: Transformer meets u-net," in *International Workshop on Machine Learning in Medical Imaging*. Springer, 2022, pp. 377–386

Author Index

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2025

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15322, pp. 461–462, 2025.
https://doi.org/10.1007/978-3-031-78312-8