

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

LNCS 15320

Pattern Recognition

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

20 Part XX



Lecture Notes in Computer Science

15320

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 XX

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 978-3-031-78497-2

ISBN 978-3-031-78498-9 (eBook)

<https://doi.org/10.1007/978-3-031-78498-9>

© 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 bio-signal 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 IJCPD 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. In ICRP 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
Josef Kittler
Anil Jain

Indian Statistical Institute, Kolkata, India
University of Surrey, UK
Michigan State University, USA

Program Chairs

Apostolos Antonacopoulos
Subhasis Chaudhuri
Rama Chellappa
Cheng-Lin Liu

University of Salford, UK
Indian Institute of Technology, Bombay, India
Johns Hopkins University, USA
Institute of Automation, Chinese Academy of
Sciences, China

Publication Chairs

Ananda S. Chowdhury
Wataru Ohyama

Jadavpur University, India
Tokyo Denki University, Japan

Competition Chairs

Richard Zanibbi
Lianwen Jin
Laurence Likforman-Sulem

Rochester Institute of Technology, USA
South China University of Technology, China
Télécom Paris, France

Workshop Chairs

P. Shivakumara
Stephanie Schuckers
Jean-Marc Ogier
Prabir Bhattacharya

University of Salford, UK
Clarkson University, USA
Université de la Rochelle, France
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 Lladós	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	Da-Han Wang
Mingxin Huang	Yang Xue
Lei Kang	Wentao Yang
Wenhui Liao	Jiixin Zhang
Yuliang Liu	Yiwu Zhong
Yongxin Shi	

Reviewers (Conference Papers)

Aakanksha Aakanksha
 Aayush Singla
 Abdul Muqet
 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
 Anwasha Sengupta
 Archisman Adhikary
 Arjan Kuijper
 Arnab Kumar Das

Arnav Bhavsar	Bin-Bin Jia
Arnav Varma	Binbin Yong
Arpita Dutta	Bindita Chaudhuri
Arshad Jamal	Bindu Madhavi Tummala
Artur Jordao	Binh M. Le
Arunkumar Chinnaswamy	Bi-Ru Dai
Aryan Jadon	Bo Huang
Aryaz Baradarani	Bo Jiang
Ashima Anand	Bob Zhang
Ashis Dhara	Bowen Liu
Ashish Phophalia	Bowen Zhang
Ashok K. Bhateja	Boyang Zhang
Ashutosh Vaish	Boyu Diao
Ashwani Kumar	Boyun Li
Asifuzzaman Lasker	Brian M. Sadler
Atefeh Khoshkhahtinat	Bruce A. Maxwell
Athira Nambiar	Bryan Bo Cao
Attilio Fiandrotti	Buddhika L. Semage
Avandra S. Hemachandra	Bushra Jalil
Avik Hati	Byeong-Seok Shin
Avinash Sharma	Byung-Gyu Kim
B. H. Shekar	Caihua Liu
B. Uma Shankar	Cairong Zhao
Bala Krishna Thunakala	Camille Kurtz
Balaji Tk	Carlos A. Caetano
Balázs Pálffy	Carlos D. Martá-Nez-Hinarejos
Banafsheh Adami	Ce Wang
Bang-Dang Pham	Cevahir Cigla
Baochang Zhang	Chakravarthy Bhagvati
Baodi Liu	Chandrakanth Vipparla
Bashirul Azam Biswas	Changchun Zhang
Beiduo Chen	Changde Du
Benedikt Kottler	Changkun Ye
Beomseok Oh	Changxu Cheng
Berkay Aydin	Chao Fan
Berlin S. Shaheema	Chao Guo
Bertrand Kerautret	Chao Qu
Bettina Finzel	Chao Wen
Bhavana Singh	Chayan Halder
Bibhas C. Dhara	Che-Jui Chang
Bilge Günsel	Chen Feng
Bin Chen	Chenan Wang
Bin Li	Cheng Yu
Bin Liu	Chenghao Qian
Bin Yao	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 Coquenot
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
Dipanjoyoti 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 Binamakhshen
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 Otroschi 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
Jiyang 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
Liviú-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 Kämpel
Martina Pastorino
Marwan Turki
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	Palash Ghosal
Narayan Vetrekar	Pallav Dutta
Narendra D. Londhe	Paolo Rota
Nathalie Girard	Paramanand Chandramouli
Nati Ofir	Paria Mehrani
Naval Kishore Mehta	Parth Agrawal
Nazmul Shahadat	Partha Basuchowdhuri
Neeti Narayan	Patrick Horain
Neha Bhargava	Pavan Kumar
Nemanja Djuric	Pavan Kumar Anasosalu Vasu
Newlin Shebiah R.	Pedro Castro
Ngo Ba Hung	Peipei Li
Nhat-Tan Bui	Peipei Yang
Niaz Ahmad	Peisong Shen
Nick Theisen	Peiyu Li
Nicolas Passat	Peng Li
Nicolas Ragot	Pengfei He
Nicolas Sidere	Pengrui Quan
Nikolaos Mitianoudis	Pengxin Zeng
Nikolas Ebert	Pengyu Yan
Nilah Ravi Nair	Peter Eisert
Nilesh A. Ahuja	Petra Gomez-Krämer
Nilkanta Sahu	Pierrick Bruneau
Nils Murrugarra-Llerena	Ping Cao
Nina S. T. Hirata	Pingping Zhang
Ninad Aithal	Pintu Kumar
Ning Xu	Pooja Kumari
Ningzhi Wang	Pooja Sahani
Niraj Kumar	Prabhu Prasad Dev
Nirmal S. Punjabi	Pradeep Kumar
Nisha Varghese	Pradeep Singh
Norio Tagawa	Pranjal Sahu
Obaidullah Md Sk	Prasun Roy
Oguzhan Ulucan	Prateek Keserwani
Olfa Mechi	Prateek Mittal
Oliver Tüselmann	Praveen Kumar Chandaliya
Orazio Pontorno	Praveen Tirupattur
Oriol Ramos Terrades	Pravin Nair
Osman Akin	Preeti Gopal
Ouadi Beya	Preety Singh
Ozge Mercanoglu Sincan	Prem Shanker Yadav
Pabitra Mitra	Prerana Mukherjee
Padmanabha Reddy Y. C. A.	Prerna A. Mishra
Palaash Agrawal	Prianka Dey
Palaiahnakote Shivakumara	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
Ruwei 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	Shinichiro Omachi
Samar Bouazizi	Shirley David
Samia Boukir	Shishir Shah
Samir F. Harb	Shiv Ram Dubey
Samit Biswas	Shiva Baghel
Samrat Mukhopadhyay	Shivanand S. Gornale
Samriddha Sanyal	Shogo Sato
Sandika Biswas	Shotaro Miwa
Sandip Purnapatra	Shreya Ghosh
Sanghyun Jo	Shreya Goyal
Sangwoo Cho	Shuai Su
Sanjay Kumar	Shuai Wang
Sankaran Iyer	Shuai Zheng
Sanket Biswas	Shuaifeng Zhi
Santanu Roy	Shuang Qiu
Santosh D. Pandure	Shuhei Tarashima
Santosh Ku Behera	Shujing Lyu
Santosh Nanabhau Palaskar	Shuliang Wang
Santosh Prakash Chouhan	Shun Zhang
Sarah S. Alotaibi	Shunming Li
Sasanka Katreddi	Shunxin Wang
Sathyanarayanan N. Aakur	Shuping Zhao
Saurabh Yadav	Shuquan Ye
Sayan Rakshit	Shuwei Huo
Scott McCloskey	Shuyue Lan
Sebastian Bunda	Shyi-Chyi Cheng
Sejuti Rahman	Si Chen
Selim Aksoy	Siddarth Ravichandran
Sen Wang	Sihan Chen
Seraj A. Mostafa	Siladitya Manna
Shanmuganathan Raman	Silambarasan Elkana Ebinazer
Shao-Yuan Lo	Simon Benaïchouche
Shaoyuan Xu	Simon S. Woo
Sharia Arfin Tanim	Simone Caldarella
Shehreen Azad	Simone Milani
Sheng Wan	Simone Zini
Shengdong Zhang	Sina Lotfian
Shengwei Qin	Sitao Luan
Shenyuan Gao	Sivaselvan B.
Sherry X. Chen	Siwei Li
Shibaprasad Sen	Siwei Wang
Shigeaki Namiki	Siwen Luo
Shiguang Liu	Siyu Chen
Shijie Ma	Sk Aziz Ali
Shikun Li	Sk Md Obaidullah

Sneha Shukla	Suraj Kumar Pandey
Snehasis Banerjee	Surendrabikram Thapa
Snehasis Mukherjee	Suresh Sundaram
Snigdha Sen	Sushil Bhattacharjee
Sofia Casarin	Susmita Ghosh
Soheila Farokhi	Swakkhar Shatabda
Soma Bandyopadhyay	Syed Ms Islam
Son Minh Nguyen	Syed Tousiful Haque
Son Xuan Ha	Taegyeong Lee
Sonal Kumar	Taihui Li
Sonam Gupta	Takashi Shibata
Sonam Nahar	Takeshi Oishi
Song Ouyang	Talha Ahmad Siddiqui
Sotiris Kotsiantis	Tanguy Gernot
Souhaila Djaffal	Tangwen Qian
Soumen Biswas	Tanima Bhowmik
Soumen Sinha	Tanpia Tasnim
Soumitri Chattopadhyay	Tao Dai
Souvik Sengupta	Tao Hu
Spiros Kostopoulos	Tao Sun
Sreeraj Ramachandran	Taoran Yi
Sreya Banerjee	Tapan Shah
Srikanta Pal	Taveena Lotey
Srinivas Arukonda	Teng Huang
Stephane A. Guinard	Tengqi Ye
Su O. Ruan	Teresa Alarcon
Subhadip Basu	Tetsuji Ogawa
Subhajit Paul	Thanh Phuong Nguyen
Subhankar Ghosh	Thanh Tuan Nguyen
Subhankar Mishra	Thattapon Surasak
Subhankar Roy	Thibault Napol�on
Subhash Chandra Pal	Thierry Bouwmans
Subhayu Ghosh	Thinh Truong Huynh Nguyen
Sudip Das	Thomas De Min
Sudipta Banerjee	Thomas E. K. Zielke
Suhas Pillai	Thomas Swearingen
Sujit Das	Tianatahina Jimmy Francky Randrianasoa
Sukalpa Chanda	Tianheng Cheng
Sukhendu Das	Tianjiao He
Suklav Ghosh	Tianyi Wei
Suman K. Ghosh	Tianyuan Zhang
Suman Samui	Tianyue Zheng
Sumit Mishra	Tiecheng Song
Sungho Suh	Tilottama Goswami
Sunny Gupta	Tim B�chner

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

Copyright Protection for Large Language Model EaaS via Unforgeable Backdoor Watermarking	1
<i>Cong Kong, Jiawei Chen, Shunquan Tan, Zhaoxia Yin, and Xinpeng Zhang</i>	
PoPreRo: A New Dataset for Popularity Prediction of Premanian Reddit Posts	16
<i>Ana-Cristina Rogoz, Maria Ilinca Nechita, and Radu Tudor Ionescu</i>	
ConCSE: Unified Contrastive Learning and Augmentation for Code-Switched Embeddings	29
<i>Jangyeong Jeon, Sangyeon Cho, Minuk Ma, and Junyeong Kim</i>	
Mitigating Hallucination in Large Language Model by Leveraging Decoder Layer Contrasting	45
<i>Guangsheng Liu, Xinbo Ai, Wenbin Luo, and Ange Li</i>	
Robustness of Classifiers for AI-Generated Text Detectors for Copyright and Privacy Protected Society	55
<i>Akshay Agarwal and Mohammed Uzair</i>	
How Good are LM and LLMs in Bangla Newspaper Article Summarization?	72
<i>Faria Sultana, Md. Tahmid Hasan Fuad, Md. Fahim, Rahat Rizvi Rahman, Meheraj Hossain, M. Ashraful Amin, A. K. M. Mahbubur Rahman, and Amin Ahsan Ali</i>	
Navigating Data Imbalances in Cybersecurity: Identifying Malicious URLs with Multiple Labels and Extreme Data Imbalances with LGNet	87
<i>Anran Zhu, Yubo Huang, and Xin Lai</i>	
Contour-Guided Context Learning for Scene Text Recognition	103
<i>Wei-Chun Hsieh, Gee-Sern Hsu, Jun-Yi Chen, Moi Hoon Yap, and Zi-Chun Chao</i>	
A New HourGlass Network for Detecting Text in Shaky and Non-shaky Video Frames	118
<i>Arnab Halder, Shivakumara Palaiahnakote, Umapada Pal, Michael Blumenstein, and Shivanand S. Gornale</i>	

FastTextSpotter: A High-Efficiency Transformer for Multilingual Scene Text Spotting	135
<i>Alloy Das, Sanket Biswas, Umapada Pal, Josep Lladós, and Saumik Bhattacharya</i>	
Vietnamese Scene Text Detection via Edge Information and Text Region Feature Enhancement	151
<i>Liyu Jiang, Shaoliang Shi, Wenhui Huang, Zhengli Xu, Vinh Loc Cu, and Yimin Wen</i>	
Scene Uyghur Text Detection Based on Adaptive Feature Fusion	167
<i>Dong Wang, Elham Eli, Alimjan Aysa, Xuebin Xu, Hornisa Mamat, and Kurban Ubul</i>	
Handwriting Trajectory Recovery Via Trajectory Transformer With Global Radical Context-Aware Module	182
<i>Junxiang Lin, Zhounan Chen, Lingyu Liang, Wenjie Peng, and Shuangping Huang</i>	
Handwriting Intra-Variability Across Surface Transitions: Implications for Writer Identification	196
<i>Kumari Priya, Chandranath Adak, Bidyut B. Chaudhuri, and Michael Blumenstein</i>	
Enhancing Table Structure Recognition via Bounding Box Guidance	209
<i>Lei Hu and Shuangping Huang</i>	
A Deep-Learning Based Real-Time License Plate Recognition System for Resource-Constrained Scenarios	226
<i>Karthik Mohan and Suraj Kumar Pandey</i>	
MTIQA360: An Easily Trainable Multitasking Network for Blind Omnidirectional Image Quality Assessment	243
<i>Qinghai Wang and Shiguang Liu</i>	
Full-Frequency Dynamic Convolution: A Physical Frequency-Dependent Convolution for Sound Event Detection	260
<i>Haobo Yue, Zhicheng Zhang, Da Mu, Yonghao Dang, Jianqin Yin, and Jin Tang</i>	
EDM10: A Polyphonic Stereo Dataset with Identical BGM for Musical Instrument Identification	273
<i>Himadri Mukherjee, Matteo Marciano, Ankita Dhar, and Kaushik Roy</i>	

SONNET: Enhancing Time Delay Estimation by Leveraging Simulated Audio 289
Erik Tegler, Magnus Oskarsson, and Kalle Åström

The Effect of Lung Volume on Glottal Parameters: An Empirical Study 304
Gauri Deshpande and Björn W. Schuller

Linear Frequency Residual Cepstral Features for Dysarthria Severity Classification 316
Aditya Pusuluri and Hemant A. Patil

Generating High-Quality Symbolic Music Using Fine-Grained Discriminators 332
Zhedong Zhang, Liang Li, Jiehua Zhang, Zhenghui Hu, Hongkui Wang, Chenggang Yan, Jian Yang, and Yuankai Qi

Enduring Memory Self-learning Multi-level Transformer Network for Remote Sensing Image Super-Resolution 345
Peishan Li, Yonghong Zhang, Junfei Wang, and Guangyi Ma

Local and Global Features Fusion for No-Reference Quality Assessment of Super-Resolution Images 359
Yun Liu, Tong Tang, Zhiyuan Zhu, and Jun Ying

Image Super-Resolution with Multi-scale Hybrid Attention 374
Ningzhi Wang, Hanyi Shi, Wenna Ruan, and Lingbin Zeng

A Sinkhorn Regularized Adversarial Network for Image Guided DEM Super-Resolution Using Frequency Selective Hybrid Graph Transformer 389
Subhajit Paul and Ashutosh Gupta

Long-Wave Infrared Non-Line-of-Sight Imaging with Visible Conversion 406
Shaohui Jin, Wenhao Zhang, Hao Liu, Huimin Wang, Shuang Cui, and Mingliang Xu

Re-Identification Based on the Spatial-Temporal Fusion Network 421
Hye-Geun Kim, You-Kyoung Na, Hae-Won Joe, Yong-Hyuk Moon, and Yeong-Jun Cho

CMAEH: Contrastive Masked Autoencoder Based Hashing for Efficient Image Retrieval 437
Mehul Kumar, Aditya Sharma, Prerana Mukherjee, and Koteswar Rao Jerripothula

Author Index 453



Copyright Protection for Large Language Model EaaS via Unforgeable Backdoor Watermarking

Cong Kong¹, Jiawei Chen¹, Shunquan Tan², Zhaoxia Yin^{1(✉)},
and Xinpeng Zhang³

¹ Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai 200241, China

zxyin@cee.ecnu.edu.cn

² Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060, China

³ Fudan University, Shanghai 200433, China

Abstract. Large language models (LLMs) have evolved rapidly and demonstrated superior performance over the past few months. Training these models is both expensive and time-consuming. Consequently, some companies have begun to offer embedding as a service (EaaS) based on these LLMs to reap the benefits. However, this makes them potentially vulnerable to model extraction attacks which can replicate a functionally similar model and thereby infringe upon copyright. To protect the copyright of LLMs for EaaS, we propose a backdoor watermarking method by injecting a secret cosine signal into embeddings of original text with triggers. The secret signal, generated and authenticated using identity information, establishes a direct link between the watermark and the copyright owner. Experimental results demonstrate the method's effectiveness, showing minimal impact on downstream tasks and high detection accuracy, as well as exhibiting resilience to forgery attacks.

Keywords: LLMs · EaaS · Backdoor watermarking

1 Introduction

With the advancement of artificial intelligence research and the increasing availability of computational resources, large language models (LLMs) like GPT-3 [1] and LLaMA [14] have demonstrated exceptional performance in natural language processing tasks, e.g., text classification [12], text generation [4], and code writing [15]. LLMs are trained using unsupervised learning techniques on massive and diverse text corpora, allowing them to learn general language knowledge. Consequently, the embeddings generated by LLMs exhibit universality across various domains. Many researchers have achieved state-of-the-art results by fine-tuning their models for specific tasks using embeddings generated

by LLMs [3]. However, training these large language models requires significant human and financial resources. Recognizing this, LLM owners often provide fee-based APIs, such as OpenAI’s embedding as a service (EaaS) based on GPT-3, to offset innovation and maintenance costs. Nevertheless, this exposes vulnerabilities to model extraction attacks, especially leveraging model distillation. Recently, researchers [13] indicate that model extraction is much less costly than training a model. Consequently, attackers can easily replicate EaaS model or even offer their own EaaS, leading to substantial economic losses for model owners. Therefore, protecting copyright of EaaS model to prevent model extraction attacks is both urgent and crucial.

The copyright protection mechanism for EaaS model can be divided into two categories: watermarking into the output embeddings and watermarking into the EaaS model. As for the first approach, traditional methods aim at safeguarding the copyright of multimedia content, such as image [9], audio [16], and text [19], facing challenges when directly applied to embeddings. This is due to the discrete nature of embeddings, which feature a higher encoding rate and significantly lower content redundancy, making the application of watermarking more intricate. As for the second approach, [7, 18] utilize trigger sets to embed invisible watermarks in diverse models prior to distribution. However, as EaaS only provides users with embeddings and users have no access to specific model parameters, the aforementioned methods are not applicable to EaaS model copyright protection.

For the latest research on EaaS model copyright protection, EmbMarker [11] presents a backdoor watermarking method that embeds target embeddings as watermarks into the original embeddings. However, this approach lacks a direct link between the watermark and the copyright owner. This makes it vulnerable to forgery attacks which involve attackers fabricating an identity they do not possess to pass through identity verification successfully. Specifically, during the copyright verification process, attackers can also claim ownership of the watermark. Consequently, the trusted third-party institution cannot ascertain copyright ownership, rendering the watermark ineffective.

We integrate the watermark with identity information and a key, enhancing the algorithm’s resilience against forgery attacks and thereby mitigating the issue of EmbMarker. The core idea is to inject a secret cosine signal into the embeddings of the original text which has triggers. This cosine signal has minimal impact on subsequent downstream tasks using embeddings and exhibits high concealment. To resist forgery attacks, the generation and authentication of the covert signal require identity information and a key, establishing a connection between the watermark and the copyright owner. To ensure that the models extracted by attackers include this watermark cosine signal, we select moderate-frequency words from a general text corpus as the trigger set. In the copyright verification stage, by providing the identity information and key to a trusted third party, typically a government agency, the party detects the presence of the cosine signal in the embeddings returned by the suspicious EaaS for backdoor samples. If a cosine signal with the same frequency as the private frequency is detected, it can be concluded that the model has been illicitly obtained, as normal embeddings lack cosine components.

The experimental results indicate that our method has a minimal impact on downstream tasks and achieves high detection accuracy. The experiments also validate the invisibility and reliability of our method. Additionally, to validate the method’s resistance to forgery attacks, i.e., unforgeability, we conduct watermark detection using incorrect identity information and key. The results demonstrate the incapacity to detect the watermark signal, thereby confirming the method’s unforgeability. The main contributions of this paper include:

- We propose a backdoor watermarking method for EaaS model copyright protection, with minimal impact on downstream classification tasks that utilize the embeddings.
- We design a method to embed a secret cosine signal into embeddings, establishing a connection between the watermark and identity information, effectively resisting forgery attacks.
- Extensive experiments verify the effectiveness of the proposed method and can resist forgery attacks.

2 Related Works

2.1 Model Extraction Attacks

Model extraction attacks [10] involve attackers using their copy datasets to query the victim model’s API, acquire responses, and construct data and labels for training their own model. These attacks leverage knowledge distillation [2], enabling the development of a model with comparable performance to the victim model at a reduced cost. Previous work [8] indicates that EaaS is more vulnerable to such attacks. As attackers can release similar APIs at lower prices, significantly undermining the interests of the model owner.

2.2 Backdoor Watermarking

Backdoor watermarking stems from backdoor attacks [6], which use hidden triggers in training data to induce specific behaviors in models, causing abnormal responses [17]. In backdoor watermarking, the trigger set is carefully designed to embed watermarks reflecting the owner’s identity information. PLMmark [5] introduces a black-box watermarking method based on contrastive learning to protect the copyright of pre-trained language models. GINSEW [20] embeds a sine wave into the logits of a language generation model, achieving invisibility of the watermark. However, these backdoor watermarking methods cannot be directly applied to EaaS, as EaaS returns embeddings. EmbMarker [11] combines target embeddings with the original embeddings and detects the watermark through distribution similarity checks. Nevertheless, due to the embedded watermark lacking reflection of the owner’s unique identity information, EmbMarker is susceptible to forgery attacks. To address this issue, inspired by these works, we propose an approach to protect EaaS copyrights. Our method maintains high detection accuracy while minimally impacting the original embeddings. Additionally, the experimental results indicate that our method is robust to resist forgery attacks.

3 Methodology

3.1 Problem Statement

The model owner releases their model API \mathcal{V} as EaaS for a profit. In model extraction attacks, attackers utilize their copy dataset, denoted as D_c , to query \mathcal{V} , obtaining the response embedding corpus E_c . Subsequently, attackers construct a training set to train their own stolen model f . The attacker’s objective is to employ knowledge distillation to train their own model, making its functionality similar to \mathcal{V} and providing more affordable EaaS. We assume the attacker has a sufficient budget to query \mathcal{V} and the resources to train their own model.

Our method is not aimed at preventing model extraction attacks, as we cannot distinguish between attackers and normal users. On the contrary, we focus on determining whether a suspicious model is stolen. Therefore, we embed watermark into embeddings of text which has trigger set, denoted as e_w . Once the attacker trains their model using embeddings with the watermark, the watermark will exist in the output of the suspicious model, which can be used to determine whether the model is stolen or not.

3.2 Overview

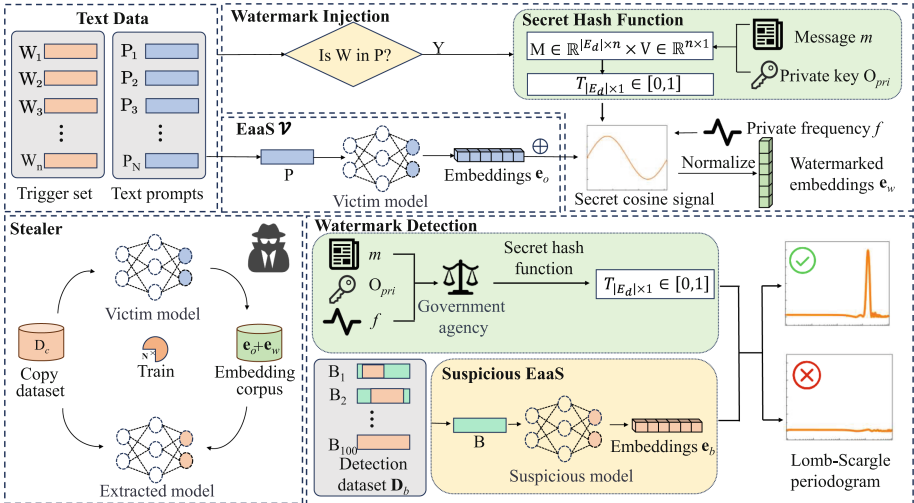


Fig. 1. The detailed overview of watermark injection phase and watermark detection phase in our proposed approach.

We present an overview of our approach as shown in Fig. 1, comprising two primary stages: a) watermark injection and b) watermark detection. Specifically, we embed the watermark into the output embeddings instead of the original model,

thus avoiding the need to modify the model’s parameters. During the watermark injection phase, if the original query text contains words from the trigger set T , we embed a secret cosine signal into the clean embeddings \mathbf{e}_o , resulting in watermarked embeddings \mathbf{e}_w . Conversely, if the original query text does not contain words from the trigger set T , we return the original clean embeddings \mathbf{e}_o . The stealer queries \mathcal{V} with their copy dataset D_c to obtain responses E_c , using these to constitute a training dataset for training the extracted model. As D_c contains several sentences with trigger words, the response embedding corpus E_c is composed of \mathbf{e}_o and \mathbf{e}_w . During the watermark detection phase, as embeddings from a normal model lack specific cosine signal, we utilize sentences with different numbers of trigger words to form a detection dataset D_b for querying the embeddings \mathbf{e}_b provided by the suspicious EaaS. Ownership verification is accomplished by detecting the presence of the distinctive cosine signal.

In the following sections, we elaborate on the design motivation and details of each stage.

3.3 Inject Watermark Containing Identity Information

Trigger Set Selection. In the watermark injection phase, designing an appropriate trigger set is crucial. We need to ensure that regular users are not affected, and that the number of backdoor samples is sufficient to embed the watermark into stolen models. We follow the approach [11] to select n words with specific frequencies appearing in a large corpus to constitute the trigger set T . Specifically, We compile the frequency of each word in a general text corpus D_p and randomly select n words with medium frequencies to form the trigger set $T = \{W_1, W_2, \dots, W_n\}$. The rationale behind this choice is that mid-frequency words can minimize the impact on downstream tasks and ensure that the attacker’s copy dataset D_c contains a substantial number of backdoor samples for injecting the watermark. The influence of the trigger set size, denoted as n , exhibits a comparable impact. We further discuss the impact of both aspects on the results in Sect. 4.3.

Watermark Function. In order to resist forgery attacks, we propose a novel watermarking function that embeds a cosine signal into the original embeddings. This cosine signal is generated using modern cryptographic methods, ensuring the unforgeability of the watermark. Specifically, we apply a digital function **Sign(.)** to generate a digital signature from identity information m and a private key O_{pri} . Subsequently, we utilize a **Hash(.)** function to map the digital signature to binary encoding b :

$$b = \mathbf{Hash}(\mathbf{Sign}(m, O_{pri})), \quad (1)$$

where we implement the **Sign(.)** using the RSA public-key cryptography algorithm and utilize SHA256 as the **Hash(.)**.

Subsequently, we use the binary encoding b to randomly generate a matrix $\mathbf{M} \in \mathbb{R}^{|E_d| \times n}$ and a vector $\mathbf{v} \in \mathbb{R}^{n \times 1}$, where $|E_d|$ is the dimension of the embeddings. The elements of the phase vector \mathbf{v} are randomly sampled from a uniform

distribution $[0, 1)$, while the elements of the token matrix \mathbf{M} are randomly sampled from a standard normal distribution, denoted as $\mathbf{M}_{ij} \sim \mathcal{N}(0, 1)$. Let $\mathbf{M}_i \in \mathbb{R}^n$ denote the i -th row of matrix \mathbf{M} , then $\mathbf{M}_i \times \mathbf{v} \sim \mathcal{N}(0, \frac{n}{3})$. We then use the probability integral transformation F to obtain a uniform distribution of the hash values [20] t :

$$t = F(\mathbf{M}_i \times \mathbf{v}) \sim \mathcal{U}(0, 1). \quad (2)$$

Each row of the matrix \mathbf{M} generates a unique hash value. Combining these different hash values forms the time set $T = [t_1, t_2, \dots, t_{|E_d|}]$. The necessary conditions for generating T are identity information m and a private key O_{pri} ; thus, we refer to it as a secret hash function $\mathbf{g}(m, O_{pri})$. Multiplying T by a private frequency f_w results in the secret cosine signal. To improve watermark concealment and minimize its influence on the original embeddings, we introduce a hyperparameter, the watermark weight λ , which controls the weight of the embedded cosine signal. Combining the secret cosine signal with the original embeddings \mathbf{e}_o , we obtain the embedded watermark embeddings \mathbf{e}_w after applying L2-norm:

$$\mathbf{e}_w = \frac{\mathbf{e}_o + \lambda \cos(f_w T)}{\|\mathbf{e}_o + \lambda \cos(f_w T)\|_2}. \quad (3)$$

3.4 Copyright Verification

During the copyright verification phase, if we detect the presence of the cosine signal in the output embeddings of the suspicious model, we can confidently conclude that the suspicious model is stolen. The specific detection process is outlined as follows:

Constructing Detection Dataset. To ensure the accuracy and reliability of our detection process, we meticulously construct a detection dataset comprising 100 samples. Initially, we randomly select four words from a general text corpus D_p , which does not belong to the trigger set, to form a single sample. We repeat this process 100 times. Subsequently, we further process the dataset by replacing the original words with an arbitrary number of trigger words to construct the backdoor data. Through these steps, we successfully create the detection dataset D_b .

Cosine Signal Detection. When we identify a suspicious model, we can provide the identity information m , a private key O_{pri} and detection dataset D_b to a trusted third party (often a government agency). This entity can generate the secret time set T using the function $\mathbf{g}(m, O_{pri})$. Subsequently, by querying the suspicious model’s API with the detection dataset D_b , we obtain the returned embeddings \mathbf{e}_b . We then adds the pair $(T[i], \mathbf{e}_b[i])$ to the set \mathcal{H} . As the time series are non-uniformly sampled, we use the Lomb-Scargle periodogram to estimate the Fourier power spectrum $P(f)$ at a specific frequency f_w in the probing set \mathcal{H} .

Through approximate Fourier transformation, we enhance the subtle perturbation in the probability vector. This enables the detection of a peak in the power spectrum at the frequency f_w . Consequently, the strength of the signal can be assessed by calculating the signal-to-noise ratio P_{snr} [20]:

$$\begin{aligned}
 P_{noise} &= \frac{1}{F - \delta} \left[\int_0^{f_w - \frac{\delta}{2}} P(f) df + \int_{f_w + \frac{\delta}{2}}^F P(f) df \right], \\
 P_{signal} &= \frac{1}{\delta} \int_{f_w - \frac{\delta}{2}}^{f_w + \frac{\delta}{2}} P(f) df, \\
 P_{snr} &= P_{signal} / P_{noise},
 \end{aligned} \tag{4}$$

where δ regulates the window width of detection, F represents the maximum frequency, and f_w denotes the angular frequency embedded into the victim model.

A high P_{snr} value implies a greater likelihood that the suspicious model contains a secret cosine signal. By inputting the detection dataset D_b into the suspicious model, we can demonstrate that the model is stolen if the embeddings returned by the backdoor data exhibit a high P_{snr} value. Conversely, a low P_{snr} value in the embeddings returned by the clean data further validates the reliability of our approach. This comprehensive analysis provides conclusive evidence regarding the origin of the suspicious model. Through experimentation, it is determined that the P_{snr} for embeddings without a watermark is unlikely to exceed 5. Therefore, we set a threshold $\tau = 5$ to determine whether the suspicious model is a stealing model.

4 Experiments

4.1 Experimental Setup

Datasets. To evaluate the performance of our method and demonstrate its universality, we utilize four standard text classification datasets: SST2, AGNews, Enron Spam, and MIND. SST2 is a sentiment classification dataset. AGNews and MIND are news classification datasets with different numbers of classes (18 for MIND and 4 for AGNews). Enron Spam is a dataset for spam email classification.

Implementation Details. To simulate a realistic model extraction attack scenario for experiments, we use SST2, AGNews, Enron Spam, and MIND as copy datasets D_c , querying OpenAI’s EaaS which is incorporated with our watermarking method to obtain responses \mathbf{E}_c as the training dataset. We employ the AdamW algorithm to train an extracted model with Bert as the backbone. All hyperparameters are chosen based on our experimental results to ensure their relative appropriateness. The secret angular frequency of cosine signal f_w is 16. The watermark weight λ is set to $1/120$. The size of the trigger set n is 20. We utilize the WikiText dataset, comprising 1,801,350 samples, as a general text corpus D_p to calculate word frequencies. The frequency interval of selected triggers is $[0.005, 0.01]$. The window width of detection δ is 8.

Evaluation Metrics. We adopt two evaluation metrics to assess method performance: accuracy and detection accuracy. The accuracy refers to the precision of the text classification task using embeddings generated by the model. The detection accuracy refers to the proportion of successfully detected watermark samples to the total number of backdoor samples.

Baseline. To the best of our knowledge, there are no other watermarking methods for EaaS except EmbMarker. Therefore, we compare our method with the following baselines: 1) Original, in which the returned embeddings lack watermarking, and attackers utilize clean embeddings to train their own copy models. 2) EmbMarker [11]. EmbMarker employs a hypothesis testing approach based on cosine similarity distribution for watermark detection, which differs from the detection mechanism of our proposed method. To ensure a fair comparison, we adopted the detection metric used in EmbMarker-cosine similarity-to individually inspect whether a model contains a watermark. Specifically, we calculated the cosine similarity between the embeddings returned by the suspicious model and the target embeddings. If this similarity exceeds a threshold, we considered the model to contain a watermark. This threshold was determined as the optimal solution through a search algorithm, aiming to maximize detection accuracy. The validation dataset we used was a specifically constructed detection dataset that included both watermarked and non-watermarked samples. Detection accuracy remains a crucial metric for evaluating method performance, representing the proportion of successfully detected watermarked samples among the total number of backdoor samples. By comparing our approach with EmbMarker using the same evaluation criteria, we can more intuitively demonstrate the advantages of our method in watermark detection.

Furthermore, our method is a one-time test, meaning that it can successfully detect watermarks with a single inspection. In contrast to EmbMarker, which requires multiple samples for watermark detection, our method offers higher efficiency during the inspection process.

4.2 Performance Evaluation

Effectiveness. As shown in Table 1 and Fig. 2, our method achieves high detection accuracy and a high P_{snr} value when detecting embeddings returned by querying the extracted model with backdoor samples. This demonstrates the effectiveness of our approach. It is notable that our method performs slightly less effectively on the Enron Spam dataset compared to the other three. This is attributed to the smaller sample size of the Enron dataset, making it challenging for the extracted model to learn watermark features. Nevertheless, the results still meet the detection requirements. A comparison reveals a slightly higher detection accuracy compared to EmbMarker, primarily attributed to the statistical approach employed by EmbMarker using the K-S test, which is less effective in single-sample detection.

Table 1. Performance of different methods on the SST2, AGNews, Enron Spam, and MIND datasets. We report the classification accuracy and mean detection accuracy of the method.

		Original	EmbMarker	Ours
Accuracy(%)	SST2	93.81	93.12(0.69↓)	93.46(0.35↓)
	AGNews	93.40	92.91(0.49↓)	93.37(0.03↓)
	Enron Spam	94.80	94.45(0.35↓)	94.65(0.15↓)
	MIND	77.34	76.70(0.64↓)	77.24(0.10↓)
Detection Accuracy(%)	SST2	-	87.50	97.50
	AGNews	-	100.00	100.00
	Enron Spam	-	82.50	85.00
	MIND	-	82.50	92.50
Unforgeability		-	X	✓

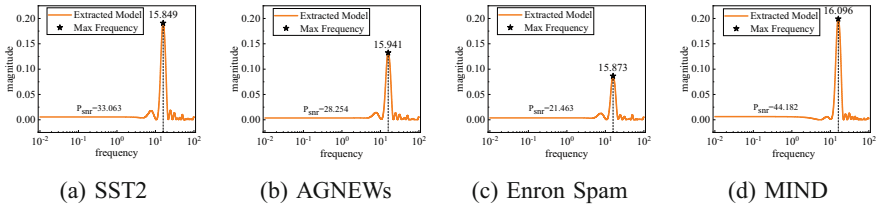


Fig. 2. The spectrum graph of embeddings returned by querying the extracted model with backdoor samples after Lomb-Scargle periodogram for the four datasets.

Fidelity: The watermark should not impact the normal performance of the model. Table 1 demonstrates that our approach exhibits minimal degradation in accuracy on downstream tasks, outperforming EmbMarker. This is attributed to the small watermark weight λ , resulting in minimal deviation between the watermarked and original embeddings.

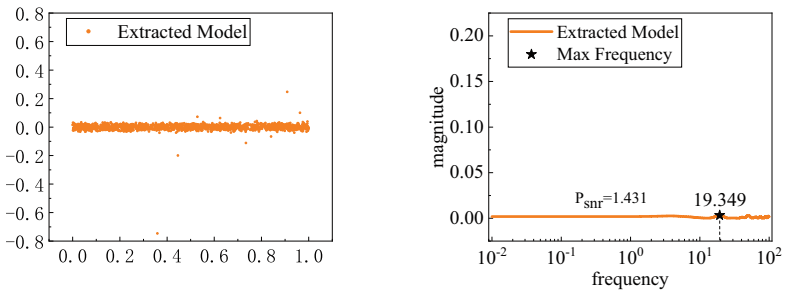


Fig. 3. The scatter plot and the spectrum graph after Lomb-Scargle periodogram of embeddings returned by querying the normal model with backdoor samples.

Reliability. Given the critical nature of copyright protection, it is essential to ensure no false positives for legitimate models. As illustrated in Fig. 3, when extracting a watermark from a normal model, the P_{snr} value is low, and the cosine signal is undetectable. This is attributed to embeddings without watermark lacking cosine signal. Therefore, our method never generates false positives on normal models.

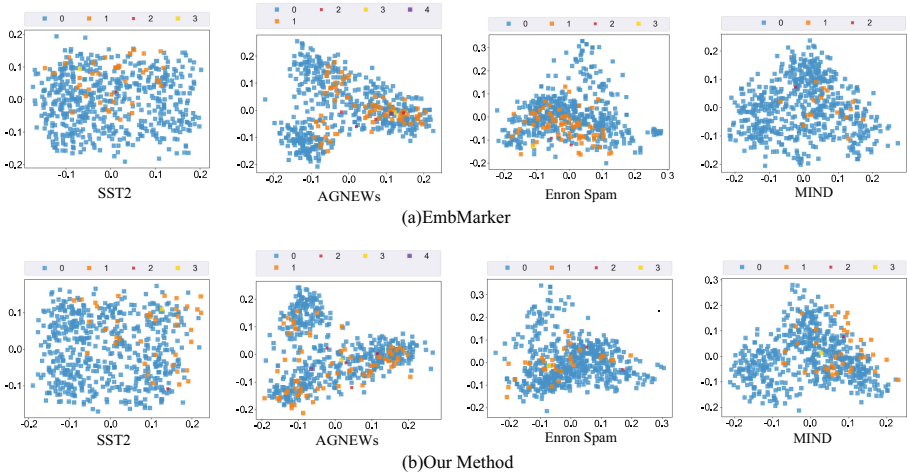


Fig. 4. Visualisation of PCA of 600 randomly selected samples with different numbers of trigger words in the four datasets of (a) EmbMarker and (b) our method

Invisibility. Attackers might identify the presence of a watermark in the embeddings, prompting them to conduct a preliminary screening before training a copy model. Since the embeddings of sentences in the same training set should be similar, attackers may filter out potential “outlier” embeddings. This underscores the importance of invisibility in our watermarked embeddings. We conducted a principal component analysis (PCA) for both our method and EmbMarker to visualize 600 randomly sampled samples from each dataset, with each sample containing a varying number of trigger words. The results are presented in Fig. 4. The plots showcase that embeddings with triggers share similar distributions with benign embeddings, showing the invisibility of the watermark in EmbMarker and our approach.

Unforgeability. The unforgeability of the watermark refers to the fact that attackers cannot falsely claim ownership of the watermark, demonstrating resistance to forgery attacks. There are two possible forgery attacks: a) The attacker submits a false identity key. We conducted watermark detection using both the correct identity key and a fake identity key. The results are illustrated in Fig. 5.

We observe that the results from detecting with false identity information are indistinguishable from those of the normal model, which indicates that attackers cannot successfully claim ownership of the watermark. b) The attacker violently enumerates the time set T . Then he needs to reversely generate \mathbf{M} and \mathbf{v} . He also needs to construct a key that not only contains his own identity message, but also can map to the \mathbf{M} and \mathbf{v} . However, due to the one-wayness and collision resistance of the hash function, these operations are computationally infeasible.

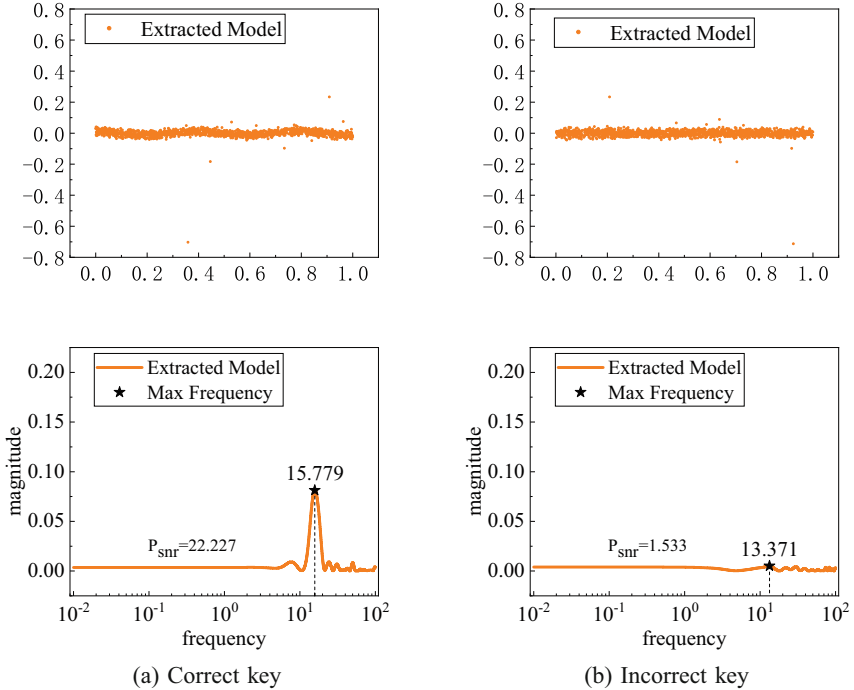


Fig. 5. The results of watermark detection using the correct key and an incorrect key.

Table 2. The results of training a theft model with different backbone model.

Model	Parameters	Accuracy(%)	Detection Accuracy(%)
Bert-small	29M	94.03	88.75
Bert-base	110M	93.46	97.50
RoBERTa	355M	93.92	97.50

Transferability. To validate the transferability of our method, we conduct experiments by utilizing Bert-small, Bert-base, and RoBERTa, each with varying parameters, as the backbone of the stealer’s model on the SST2 datasets. As shown in Table 2, each model detects the watermark.

Robustness. To assess the robustness of our method, we explore the impact of adding random noise to embeddings. Adversaries might add noise during training to remove the watermark or during inference to evade detection. We assume the attackers would add random noise with a mean of 0 and a standard deviation of 0.1, either to one dimension or across all dimensions of the embeddings. The experimental results are shown in Table 3. Adding noise during training significantly affects our watermark detection, particularly when noise is added to all dimensions, resulting in a detection accuracy of 0. However, this also reduces the downstream task accuracy to 90.37%, diminishing the stolen EaaS service quality, making it less beneficial for the attacker. Adding noise during inference does not effectively evade detection, with detection accuracy remaining above 60%. Therefore, adding noise has limited impact on our method’s effectiveness.

Table 3. The impact of adding random noise on our method.

Stage	Dimension	Accuracy (%)	Detection Accuracy (%)
Training	One	93.35	70.00
	All	90.37	0.00
Inference	One	93.04	68.75
	All	93.31	70.00

4.3 Hyper-parameter Analysis

In this subsection, we elucidate the rationale behind the selection of the three main hyper-parameters, $\lambda = 1/120$, $n=20$, frequency interval = $[0.005, 0.01]$,

Table 4. Results with varied watermark weight (λ).

	λ			
	1/160	1/120	1/80	1/60
Accuracy(%)	93.34	93.46	93.11	93.01
Detection Accuracy(%)	37.50	97.50	97.50	100.00

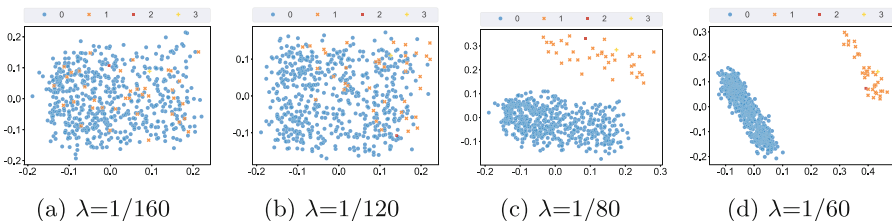


Fig. 6. The Visualisation of PCA of samples using different watermark weight (λ).

in our method and investigate their impact. Due to the limitation of space, we only present the results on the SST2 dataset, as the results on other datasets are similar.

Watermark Weight λ . The result is illustrated in Table 4. It can be observed that the detection performance of the watermark improves with the increase of λ . But the excessive values lead to a greater distance between normal and backdoor samples. This compromises the invisibility of the watermark as Fig. 6 shows, and concurrently, the accuracy of downstream tasks declines. In addition, we also observed false positive samples when λ is set to $1/60$ and $1/80$. This is a significant issue for the reliability of the watermark and should not occur. Therefore, we choose $\lambda = 1/120$ as the optimal hyperparameter, balancing the detection accuracy, the accuracy of downstream tasks, and the invisibility of the watermark.

Table 5. Results with varied trigger set size (n).

	n			
	18	20	60	100
Accuracy(%)	93.81	93.46	93.44	93.39
Detection Accuracy(%)	60.00	97.50	97.50	97.50

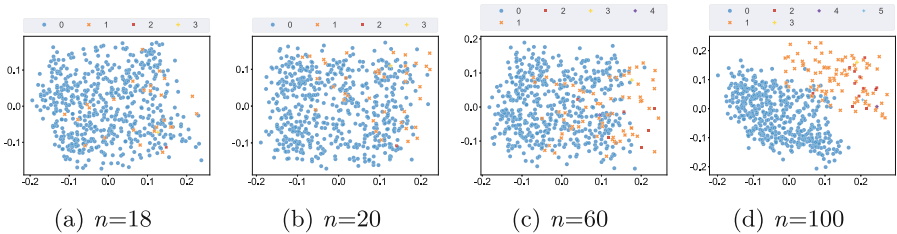


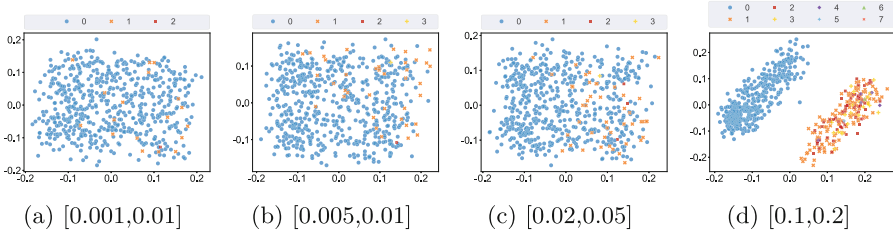
Fig. 7. The Visualisation of PCA of samples using different trigger set size (n).

Trigger Set Size n . The result is illustrated in Table 5. With the decrease of n , the impact on accuracy and detection accuracy is similar to that of λ . As shown in Fig. 7, an increase in n also makes watermarked samples easier to distinguish. This is because it increases the proportion of watermarked samples among the overall population, making them more identifiable. Eventually, we find that $n = 20$ is a suitable parameter.

Frequency Interval. Table 6 and Fig. 8 show that the impact of frequency intervals on downstream tasks, detection accuracy, and invisibility is similar to n , but more pronounced. At the $[0.001, 0.01]$ interval, detection accuracy drops

Table 6. Results with varied frequency interval.

	Frequency Interval			
	[0.001, 0.01]	[0.005, 0.01]	[0.02, 0.05]	[0.1, 0.2]
Accuracy(%)	93.35	93.46	93.23	93.35
Detection Accuracy(%)	37.50	97.50	100.00	100.00

**Fig. 8.** The Visualisation of PCA of samples using different frequency interval.

to 37.50% due to the selection of low-frequency trigger words, resulting in too few backdoor samples during stolen model training. This leads to the failure of embedding the watermark. As shown in Fig. 8, as frequency increases, watermark samples deviate more from normal ones. Our experiments indicate that the [0.005, 0.01] interval produces the best results.

5 Conclusion

We propose a backdoor watermarking method to protect the copyright of EaaS models by embedding a secret cosine signal into the embeddings of texts with trigger words, linking it to identity information and resisting forgery attacks. Extensive experiments on four text classification tasks demonstrate the method’s effectiveness, fidelity, and robustness, even under forgery attempts. We also explore the impact of tuning various hyperparameters on the results.

Acknowledgments. This research work is partly supported by National Natural Science Foundation of China No. 62472177, No. 62172001, and Guangdong Key Laboratory of Intelligent Information Processing & Shenzhen Key Laboratory of Media Security Shenzhen University Shenzhen 518060, China No. 2023B1212060076.

References

1. Brown, T., et al.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901 (2020)
2. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint [arXiv:1503.02531](https://arxiv.org/abs/1503.02531) (2015)

3. Hu, Z., et al.: LLM-adapters: an adapter family for parameter-efficient fine-tuning of large language models. arXiv preprint [arXiv:2304.01933](https://arxiv.org/abs/2304.01933) (2023)
4. Li, J., Tang, T., Zhao, W.X., Nie, J.Y., Wen, J.R.: Pretrained language models for text generation: a survey. arXiv preprint [arXiv:2201.05273](https://arxiv.org/abs/2201.05273) (2022)
5. Li, P., Cheng, P., Li, F., Du, W., Zhao, H., Liu, G.: PLMmark: a secure and robust black-box watermarking framework for pre-trained language models. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 14991–14999 (2023)
6. Li, Y., Jiang, Y., Li, Z., Xia, S.T.: Backdoor learning: a survey. *IEEE Trans. Neural Netw. Learn. Syst.* **35**(1), 5–22 (2022)
7. Li, Y., Zhu, M., Yang, X., Jiang, Y., Wei, T., Xia, S.T.: Black-box dataset ownership verification via backdoor watermarking. *IEEE Trans. Inf. Forensics Secur.* **18**, 2318–2332 (2023)
8. Liu, Y., Jia, J., Liu, H., Gong, N.Z.: Stolenencoder: stealing pre-trained encoders in self-supervised learning. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 2115–2128 (2022)
9. Lu, J., Ni, J., Su, W., Xie, H.: Wavelet-based CNN for robust and high-capacity image watermarking. In: 2022 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2022)
10. Orekondy, T., Schiele, B., Fritz, M.: Knockoff nets: stealing functionality of black-box models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4954–4963 (2019)
11. Peng, W., et al.: Are you copying my model? Protecting the copyright of large language models for EaaS via backdoor watermark. arXiv preprint [arXiv:2305.10036](https://arxiv.org/abs/2305.10036) (2023)
12. Sun, X., et al.: Text classification via large language models. In: The 2023 Conference on Empirical Methods in Natural Language Processing (2023)
13. Taori, R., et al.: Stanford alpaca: an instruction-following llama model (2023)
14. Touvron, H., et al.: Llama: open and efficient foundation language models. arXiv preprint [arXiv:2302.13971](https://arxiv.org/abs/2302.13971) (2023)
15. Vaithilingam, P., Zhang, T., Glassman, E.L.: Expectation vs. experience: evaluating the usability of code generation tools powered by large language models. In: CHI Conference on Human Factors in Computing Systems Extended Abstracts, pp. 1–7 (2022)
16. Wu, S., Liu, J., Huang, Y., Guan, H., Zhang, S.: Adversarial audio watermarking: embedding watermark into deep feature. In: 2023 IEEE International Conference on Multimedia and Expo (ICME), pp. 61–66. IEEE (2023)
17. Xia, P., Li, Z., Zhang, W., Li, B.: Data-efficient backdoor attacks. In: Raedt, L.D. (ed.) Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, pp. 3992–3998. International Joint Conferences on Artificial Intelligence Organization (2022). <https://doi.org/10.24963/ijcai.2022/554>, main Track
18. Xu, T., Zhong, S., Xiao, Z.: Protecting intellectual property of EEG-based model with watermarking. In: 2023 IEEE International Conference on Multimedia and Expo (ICME), pp. 37–42. IEEE (2023)
19. Yang, X., et al.: Watermarking text generated by black-box language models. arXiv preprint [arXiv:2305.08883](https://arxiv.org/abs/2305.08883) (2023)
20. Zhao, X., Wang, Y.X., Li, L.: Protecting language generation models via invisible watermarking. arXiv preprint [arXiv:2302.03162](https://arxiv.org/abs/2302.03162) (2023)



PoPreRo: A New Dataset for Popularity Prediction of Romanian Reddit Posts

Ana-Cristina Rogoz, Maria Ilinca Nechita, and Radu Tudor Ionescu^(✉)

Department of Computer Science, University of Bucharest, 14 Academiei, Bucharest, Romania
raducu.ionescu@gmail.com

Abstract. We introduce PoPreRo, the first dataset for **Popularity Prediction of Romanian posts** collected from Reddit. The PoPreRo dataset includes a varied compilation of post samples from five distinct subreddits of Romania, totaling 28,107 data samples. Along with our novel dataset, we introduce a set of competitive models to be used as baselines for future research. Interestingly, the top-scoring model achieves an accuracy of 61.35% and a macro F_1 score of 60.60% on the test set, indicating that the popularity prediction task on PoPreRo is very challenging. Further investigations based on few-shot prompting the Falcon-7B Large Language Model also point in the same direction. We thus believe that PoPreRo is a valuable resource that can be used to evaluate models on predicting the popularity of social media posts in Romanian. We release our dataset at <https://github.com/ana-rogoz/PoPreRo>.

Keywords: natural language processing · reddit popularity · popularity detection · virality detection · Romanian · LLM prompting

1 Introduction

Understanding the factors influencing the popularity of social media posts represents a critical and multifaceted challenge for NLP research. Social media platforms generate vast amounts of user-created content, offering a unique window into real-time public discourse and collective attention. Analyzing what resonates with audiences goes beyond just sentiment analysis, demanding nuanced NLP techniques to capture humor, sarcasm, and the subtle cues that drive engagement. This pursuit fosters not only theoretical advancements but also practical applications across diverse fields, from marketing and public health to combating misinformation and predicting cultural trends. Studying social media popularity, therefore, is not just an interesting NLP problem, but a key to unlocking the true potential of language in the digital age.

So far, the phenomenon has been studied both for individual social media platforms, such as Instagram [4, 5, 21, 26], Reddit [2, 13], Twitter [16, 17, 27], either as a whole phenomenon, for detecting popularity [20, 25], or for generating engaging content [8].

Reddit, in particular, has been one of the most studied platforms in the ever-evolving landscape of online content. From gauging public opinion and identifying emerging trends to optimizing content recommendation systems and combating misinformation, accurate popularity detection offers a multitude of applications across various domains.

There are existing datasets generated from Reddit content, studying several topics, from political conflicts [28], to personality traits [10], language biases [9, 11], and mental health related topics, such as stress analysis [24], depression [23] and anxiety [22].

While existing Reddit datasets have played a crucial role in advancing NLP research, they predominantly focus on high-resource languages, such as English. This creates a bias towards high-resource languages in NLP models, neglecting the necessity of exploring NLP capabilities on less studied languages, such as Romanian.

We emphasize that what constitutes a popular (viral) post can vary across countries and regions, since the topics of interest can naturally change from one local community to another. This is because people are usually more influenced by major local events, e.g. the war in Ukraine is still a major subject of discussion in Romania, a neighboring country of Ukraine, while the subject may have faded out in countries from other continents. This justifies the need to study the popularity prediction task across multiple countries, and consequently, in various languages. To this end, we introduce PoPreRo, the first dataset for **Popularity Prediction of Romanian** posts collected from Reddit. We leverage this novel resource to explore popularity detection in a low-resource language, Romanian, establishing six diverse baselines for future comparative analysis.

2 Dataset

2.1 Data Collection

PoPreRo gathers Reddit posts from five different Romanian subreddit channels, which represent either one of the biggest cities in Romania or the country-wide subreddit. The subreddits are: Romania, București, Cluj, Iași and Timișoara. These subreddits were collected at first using Reddit API, divided into JSON files to extract the information needed for analyzing the popularity of each reddit post, such as title, content, number of comments, number of up and down votes. However, Reddit API has a limitation of 1000 requests for extraction of different data. Due to the large number of samples that we target for the dataset, the API could not provide all necessary data. Therefore, we use an open-source archive, from where the samples are collected. As mentioned above, all the data is stored in separate JSON files for each subreddit, containing relevant information for determining the popularity of posts.

2.2 Dataset Statistics

The dataset comprises 28,107 samples (14,289 unpopular and 13,818 popular) containing over 1 million tokens in total (see detailed statistics in Table 1). Each sample consists of a title, a content, and a binary label, where the title and content are concatenated into a single text. We divide the posts into “popular” or “unpopular” based on the sum of upvotes and downvotes for each post, where the threshold between the two categories is given by the median number of votes (15). To enable consistent evaluation and comparison with future studies, we provide an official split with distinct training, validation, and test sets. Inspired by McHardy et al. [18], we utilize disjoint subreddits

Table 1. Number of samples (#posts) and number of tokens (#tokens) for each subset in PoPreRo.

Set	Unpopular		Popular		Total	
	#posts	#tokens	#posts	#tokens	#posts	#tokens
Training	12,053	398,219	11,592	560,580	23,645	958,799
Validation	1,059	75,742	1,054	80,297	2,113	156,039
Test	1,177	72,819	1,172	93,268	2,349	168,867
Total	14,289	546,780	13,818	734,145	28,107	1,283,705

Table 2. Number of samples (#posts) for each label (popular/unpopular), distributed by the time of posting for each subset in PoPreRo.

Set	Label	#posts in time window (h)					
		[0–4]	[4–8]	[8–12]	[12–16]	[16–20]	[20–24]
Training	popular	816	260	2,200	3,451	2,797	2,272
	unpopular	1,050	254	1,779	3,280	3,014	2,472
Validation	popular	78	38	255	284	228	172
	unpopular	87	32	174	273	232	260
Test	popular	57	24	241	319	287	244
	unpopular	67	32	259	325	274	220

for each set, ensuring models cannot capitalize on knowledge of specific topics. To further mitigate potential biases arising from uneven topic or time distributions, we select posts from the same time frame across all subreddits (Table 2).

Additionally, to control for a potential bias related to the time of day when posts were submitted, we performed an analysis of post popularity by hour. We divided each day into four-hour intervals and categorized the number of popular and unpopular posts within each interval. The detailed results are presented in Fig. 1. Notably, we observe a consistent trend across all time intervals for both popular and unpopular posts. This finding suggests that the hour of submission does not exert a significant influence on post popularity within our dataset.

2.3 Preprocessing

After gathering the data from Reddit, we implement a two-step preprocessing pipeline to ensure data quality and consistency. First, language identification was performed on post titles using FastText [12] to filter out non-Romanian posts (filtered posts are not counted in Table 1). This step guarantees the linguistic homogeneity of the dataset. Subsequently, upvote/downvote scores are normalized to the $[0, 1]$ interval. Finally, a binary popularity label is assigned with respect to the median value of the normalized scores, which corresponds to 15 votes. This approach provides a clear threshold for distinguishing popular and unpopular posts. Notably, our data collection and labeling procedure is directly transferable to other languages.

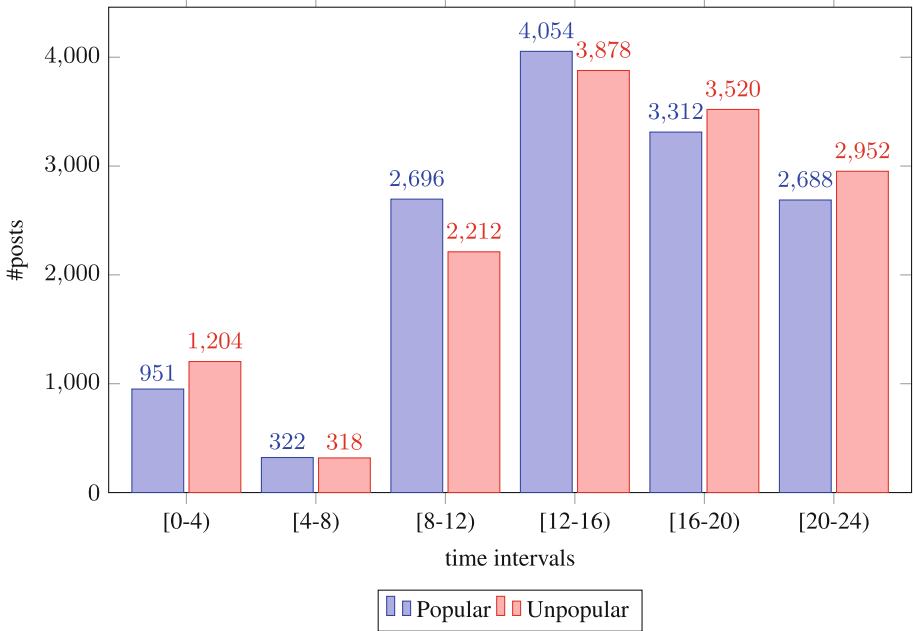


Fig. 1. Number of samples (#posts) for each label (popular/unpopular), distributed by the time of posting. The 24 h in a day are divided into six four-hour intervals. Best viewed in color.

3 Methods

To comprehensively evaluate the performance for the popularity prediction task on the newly introduced dataset, we establish six baseline approaches. Two of these baselines leverage state-of-the-art deep learning models for language processing. Another three baselines utilize various classifiers based on shallow or deep (frozen) features. Our final baseline uses a Large Language Model (LLM) based on in-context learning, also known as few-shot prompting. For all models, we use the concatenated title and content of each post as the input data.

3.1 Fine-Tuned Ro-GPT2

Our first baseline relies on fine-tuning a Ro-GPT2 model [19], a large language model specifically trained on Romanian text. It is based on the original GPT2 architecture, but trained on a Romanian dataset consisting of over 1 million tokens. This allows it to capture the nuances and specificities of the Romanian language, making it more suitable for tasks involving Romanian than the general-purpose GPT2. The Ro-GPT2 encoder is utilized to encode each text sequence into a list of token IDs. Subsequently, the model processes these tokens, generating corresponding 768-dimensional embeddings. We then incorporate a global average pooling layer to capture a Continuous Bag-of-Words (CBOW) representation for each text sequence. This representation is fed

into a Softmax output layer comprising two neurons, each predicting the probability of belonging to either the unpopular or popular category. To assign the final class label, we apply the *argmax* function on the two predicted probabilities. The entire model is fine-tuned for 5 epochs on mini-batches of 32 samples. We employ the Adam optimizer with decoupled weight decay (AdamW) [15] with a learning rate of $5 \cdot 10^{-7}$ and $\epsilon = 5 \cdot 10^{-7}$.

3.2 Fine-Tuned Ro-BERT

As our second baseline, we employ a fine-tuned Romanian Bidirectional Encoder Representations from Transformers (Ro-BERT) model [7]. Sharing the same transformer-based architecture as the original BERT [6], Ro-BERT has been demonstrated to outperform multilingual BERT on various tasks, as reported by Dumitrescu et al. [7]. Consequently, we anticipate Ro-BERT to be a strong baseline for our Romanian corpus.

Similarly to the previous baseline, we use the Ro-BERT encoder to encode each text into a list of token IDs. We keep the same design as before, where the model generates 768-dimensional embeddings, followed by a global average pooling layer which is fed into a Softmax output layer with two neurons. To assign the final class label, we apply the *argmax* function on the two predicted probabilities. The entire model is fine-tuned for 10 epochs on mini-batches of 32 samples. We employ the AdamW optimizer [15] with a learning rate of $2 \cdot 10^{-7}$ and the default value for ϵ .

3.3 Ro-BERT Embeddings + Logistic Regression

For our third classification approach, we leverage pre-trained Ro-BERT embeddings in conjunction with a Logistic Regression (LR) classifier. Consistent with the fine-tuned Ro-BERT baseline, we first tokenize all input samples from the three datasets. Subsequently, we utilize the Ro-BERT model to extract 768-dimensional vector representations for each sample. These representations, corresponding to the final hidden layer of Ro-BERT, are then fed into the LR model for classification.

3.4 FastText + SVM

The first shallow classification approach is based on FastText embeddings [3] and a Support Vector Machines (SVM) classifier. After textual cleaning and tokenization using NLTK's word tokenizer, we fine-tune a FastText model on the training corpus. This model provides word embeddings for train, validation, and test sets. For each text sample, the word embeddings are averaged to produce a 300-dimensional feature vector, which is subsequently passed to the SVM. Finally, we train the SVM classifier using the linear kernel and the regularization hyperparameter C set to 10.

3.5 TF-IDF + Random Forest

Our second shallow classification approach is based on the Term Frequency-Inverse Document Frequency (TF-IDF) representation and a Random Forest (RF) classifier. As for the previous method, we initiate the process by cleaning and tokenizing the text using NLTK's word tokenizer. Subsequently, we employed a TF-IDF vectorizer to quantify the importance of words within the corpus, generating numerical features for each document. These features are then used to train a Random Forest classifier.

3.6 Few-Shot LLM Prompting

To explore the feasibility of large language models (LLMs) for post popularity prediction in PoPreRo, we employ a prompt-based approach utilizing the 7-billion parameter Falcon LLM [1] (Falcon-7B). Due to computational limitations, we prompt the LLM with contexts comprising two unpopular and two popular examples. Subsequently, we attach an individual test sample to each prompt and ask the LLM to predict the corresponding label. Below, we illustrate the structure of our prompt via a concrete example:

```
PROMPT (Original):
```

```
Text: 'Nu vreau sa mai traiesc pe aceasta planeta !'  
Label: 'Popular'.
```

```
Text: 'Unde pot verifica compozitia unui produs?. S  
testez de exemplu dac ingredientele unui produs sunt  
într-adevr acelea. Sau dac nite tablete de vitamine  
chiar conin vitamine. În ce proporii? Sau cât vitamina  
A conine un morcov - unde pot verifica asta? Ceva  
laboratoare?' Label: 'Unpopular'.
```

```
Text: 'Azi a venit mitropolitul ardealului la noi la  
liceu s ne conving s facem religie. Primul lucru  
care mi-a venit în cap când am vzut ce main i-a  
parcat în curtea instituii..' Label: 'Popular'.
```

```
Text: 'Daca intereseaza pe cineva, sa stiti ca e reddit  
si in romana' Label: 'Unpopular'.
```

```
Text: 'Am prins niste fulgere faine zilele trecute'  
Label:
```

PROMPT (Translated):

Text: 'I don't want to live on this planet anymore!'

Label: 'Popular'.

Text: 'Where can I check the composition of a product?. To test for example whether the ingredients of a product are indeed those. Or if some vitamin tablets actually contain vitamins. In what proportions? Or how much vitamin A contains a carrot - where can I check this? Some laboratories?' Label: 'Unpopular'.

Text: 'Today the metropolitan of Transylvania came to us at high school to convince us to do religion. First thing that came to mind when I saw what car he has parked in the courtyard of institutions..' Label: 'Popular'.

Text: 'If anyone is interested, there's reddit in Romanian' Label: 'Unpopular'.

Text: 'I caught some fine lightning the other day'
Label:

4 Experiments

4.1 Evaluation

Our binary classification experiments focus on predicting the popularity of text within the PoPreRo dataset. Each text sample is categorized as either popular or unpopular. To evaluate the performance of our models, we employ several metrics. For each class, we calculate precision (proportion of true positives among the identified positives) and recall (proportion of true positives with respect to all positives). Additionally, we aggregate these scores using macro F_1 and micro F_1 (accuracy) measures.

4.2 Hyperparameter Tuning

The hyperparameters of all models are determined via grid search. For the transformer-based methods (Ro-BERT, Ro-GPT2), we employ a grid search over the maximum number of input tokens in the set $\{50, 70, 100, 120, 150, 200\}$, as well as the learning rate in the set $\{10^{-5}, 5 \cdot 10^{-5}, 10^{-6}, 5 \cdot 10^{-6}, 10^{-7}, 2 \cdot 10^{-7}, 5 \cdot 10^{-7}, 10^{-8}, 5 \cdot 10^{-8}\}$ and the value of ϵ for AdamW in the set $\{10^{-6}, 10^{-7}, 10^{-8}\}$.

For the FastText + SVM approach, we vary the FastText word-embeddings dimension ($\{150, 200, 300, 350\}$), the window size for the input ($\{2, 3, 4\}$), as well as the kernel (linear or RBF) and the parameter C ($\{0.1, 1, 10, 100, 1000\}$) of the SVM classifier. Similarly, for the Ro-BERT + Logistic Regression approach, we run a search

Table 3. Validation and test results of the six baselines. The random chance baseline is added as reference. There is no hyperparameter tuning for Falcon-7B LLM, so the model is directly applied on the test set (using in-context learning). The best score on each subset and for each metric is highlighted in bold.

Set	Method	Acc.	Macro F_1	Unpopular		Popular	
				Prec.	Rec.	Prec.	Rec.
Validation	Random chance	0.4998	0.4999	0.4988	0.5011	0.5011	0.4988
	Fine-tuned Ro-GPT2	0.6525	0.6397	0.6157	0.8097	0.7351	0.4986
	Fine-tuned Ro-BERT	0.6343	0.6278	0.6189	0.6995	0.6411	0.5562
	FastText + SVM	0.6677	0.6624	0.6348	0.7920	0.7225	0.5431
	TF-IDF + RF	0.6535	0.6395	0.6107	0.8497	0.7519	0.4568
	Ro-BERT + LR	0.6824	0.6721	0.6354	0.8582	0.7807	0.5061
Test	Random chance	0.4998	0.4999	0.5010	0.4989	0.4989	0.5010
	Fine-tuned Ro-GPT2	0.6135	0.6060	0.6146	0.6331	0.6145	0.5933
	Fine-tuned Ro-BERT	0.5605	0.5489	0.5505	0.6611	0.5767	0.4565
	FastText + SVM	0.5644	0.5637	0.5718	0.5208	0.5583	0.6083
	TF-IDF + RF	0.5759	0.5729	0.5661	0.6584	0.5897	0.4931
	Ro-BERT + LR	0.5998	0.5973	0.5873	0.6771	0.6169	0.5221
	Few-shot prompted Falcon-7B	0.4143	0.4126	0.4143	0.7904	0.5537	0.1887

over the maximum numbers of Ro-BERT input tokens in the same set as before ($\{50, 70, 100, 120, 150, 200\}$) and test different penalty term values ('l1', 'l2', 'elastic net' or 'None') for the classifier.

Lastly, for the TF-IDF + Random Forest method, we vary the minimum ($\{4, 5, 6\}$) and maximum ($\{0.6, 0.7, 0.8\}$, in percentages) document frequency of the TF-IDF Vectorizer, together with the number of decision trees in the set $\{50, 100, 150, 200\}$ for the Random Forest classifier.

All other hyperparameters are set to their default values. Please note that we release the code to reproduce all baselines, along with the PoPreRo dataset¹.

4.3 Results

We present the results of our five baselines on the PoPreRo validation and test sets in Table 3. We find that Ro-GPT2 exhibits the best performance, with an accuracy (micro F_1) and a macro F_1 score above 0.6 on both validation and test sets, in contrast to the other baselines which seem to perform similarly well on the validation set, but reach worse performance on the test set.

Evaluating the two state-of-the-art transformer models, Ro-GPT2 and Ro-BERT, reveals some interesting findings. While both achieve comparable accuracy on the validation set (0.6525 for Ro-GPT2 and 0.6343 for Ro-BERT), Ro-GPT2 clearly outperforms Ro-BERT on the test set, indicating the superior ability of the former model to generalize to unseen data. Analyzing the precision-recall trade-off, we observe a shared

¹ <https://github.com/ana-rogoz/PoPreRo>.

propensity for both models to exhibit higher recall for the “popular” category, followed by a shift towards higher precision when identifying the “unpopular” class.

The FastText + SVM, TF-IDF + RF and Ro-BERT + LR models achieve comparable performance. All three models obtain accuracy rates higher than 65% on the validation set, which drop below 60% on the test set. In terms of precision and recall, almost all of them achieve higher precision for the “popular” category on both validation and test sets, with one exception being the FastText + SVM method on the test set, where the precision on the two classes is comparable. A distinctive behavior of the three models is that the TF-IDF + RF obtains a higher recall for the “popular” category, while FastText + SVM and Ro-BERT + LR attain a higher recall for the “unpopular” category.

Table 4. Examples of relevant terms for popular posts, learned by the fine-tuned Ro-BERT and SVM models.

Model	Topic	Example	Translation
Ro-BERT	Call to action	<i>“pentru cei care vor să se implice activ ”</i>	<i>“for those who want to be actively involved”</i>
		<i>“ar fi interesati de un voluntariat”</i>	<i>“would be interested in volunteering”</i>
	News	<i>“încep săpăturile la metrou”</i>	<i>“excavations begin at the subway”</i>
		<i>“un nou residence la “doar 20 de minute” de Centru”</i>	<i>“a new residence building “only 20 min” from the center”</i>
Events	<i>“Seara de film la Casa Tineretului”</i>	<i>“Movie night at the Youth House”</i>	
SVM	News	<i>“mic protest la primaria capitalei“</i>	<i>“small protest at Bucharest City Hall“</i>
	Local transport	<i>“am vazut ca este tren de la gara de nord la aeroport aproape la fiecare ora“</i>	<i>“I saw that there is a train from Gara de Nord to the airport almost every hour“</i>

Table 3 also shows the results on the test set of our few-shot prompted LLM. While this approach exhibits a bias similar to our other baselines, favoring recall for unpopular predictions and precision for popular ones, its overall performance falls below that of a random chance classifier. This suggests a limitation in the generalization capacity of LLMs to the popularity prediction task, particularly for languages with limited online resources, such as Romanian.

4.4 Discriminative Feature Analysis

We analyze the discriminative features learned by the fine-tuned Ro-BERT and by the FastText + SVM. The motivation behind this analysis is to validate that the decisions of these models are not based on some biases that escaped our data collection, but on actual data understanding.

For the Ro-BERT model, we use the Captum [14] library via its Layer Integrated Gradients method to infer valuable insights from the fine-tuned model. This technique

dives into the BERT embedding layer, attributing importance scores to individual input words which led to the final label prediction.

To find the words with higher influence on the decisions given by the SVM, we consider the cosine similarities between the primal weights of the SVM and the FastText embedding of each word. We sort the words based on the similarity values, and keep the first 10 and last 10 words from the sorted list as features for the positive (“popular”) and negative (“unpopular”) classes, respectively.

In Tables 4 and 5, we present a few examples of interesting patterns that were picked up by the models. In predicting post popularity, the Ro-BERT model demonstrates a bias toward content reflecting current trends, including news and events, and posts

Table 5. Examples of relevant terms for unpopular posts, learned by the fine-tuned Ro-BERT and SVM models.

Model	Topic	Example	Translation
Ro-BERT	Proper names	“Palatul Roznovanu”	“Roznovanu palace”
		“Ceașescu”	“Ceașescu”
		“în Timișoara”	“in Timișoara”
	Seeking advice	“terenuri ok de baschet în...”	“ok basketball courts in...”
		“print shop pentru poze mari în ...”	“print shop for big pictures in ...”
	Mundane problems	“Se închide circulația”	“traffic is closed”
“construim blocuri între case”		“building apartment building between houses”	
SVM	City names	“bucuresti”	“bucharest”
	Seeking advice	“cunoașteți un loc de facut tatuaj temporar personalizat”	“do you know a place to do custom temporary tattoo”
	Opinion sharing	“lumea ca se plange de targul de craciun de anul acesta”	“people complain about this year’s Christmas market”

Table 6. Examples of the most discriminative words for the popular and unpopular classes, selected according to the weights learned by the SVM model based on FastText features.

Label	Token	Weight
popular	online	5.974352
	dupa	4.821379
	youtube	4.121604
	asa	4.08882
	cazul	3.839789
unpopular	toate	−4.089375
	un	−4.190036
	google	−4.31336
	nia	−4.339616
	eu	−4.72841

encouraging community engagement through calls to action. Conversely, references to proper nouns like city names or historical landmarks appear to hinder popularity, as do posts seeking community advice or expressing dissatisfaction with platitudes. Similar to Ro-BERT, we find that the SVM labels posts that share news as popular, and posts by people seeking advice as unpopular.

Furthermore, we extend the feature analysis for the SVM in order to determine the most discriminative words for the popular and unpopular classes. To achieve this, we determine the discriminative weight of each word based on the cosine similarity between the respective word embedding and the SVM weights. We sort the words according to their weights, and select the ones with the highest and lowest weights. In Table 6, we provide the five most discriminative words for the popular and unpopular classes, according to the SVM based on FastText features. We observe that posts mentioning “online” or “youtube” are more popular, likely because readers appreciate posts that provide links to YouTube videos. We also note the preference for posts that discuss particular cases/experiences, which are usually introduced by the word “cazul” (translated to “case” in English). On the other hand, posts that recommend searching on “google” are unpopular, as the readers consider such suggestions unhelpful. Moreover, discussing subjective perspectives, using the singular first person pronoun “eu”, is again unpopular, likely because the readers appreciate more objective posts.

5 Conclusion

In this paper, we introduced PoPreRo, the first publicly available dataset of Romanian Reddit posts dedicated to the task of popularity prediction. We collected 28,107 posts from five diverse Romanian subreddits, amounting to over 1 million tokens. Aiming to predict binary labels resulting from the sum of upvotes and downvotes for each post, we explored five distinct popularity detection methods and presented comparative results. We found that Ro-GPT2 significantly outperforms the other models.

Building upon our foundation, future research can further study popularity detection algorithms and delve deeper into the factors driving engagement on Romanian Reddit.

6 Limitations

It is crucial to acknowledge that Reddit’s popularity in Romania might not be representative for the wider population. While Reddit offers a valuable platform for research due to its diverse communities and open discussions, its user base in Romania is comparatively smaller than other social media platforms, such as Facebook, Instagram, or YouTube. Furthermore, Reddit’s API restricts data access, limiting historical data collection and imposing retrieval caps.

Ethics Statement. The data was collected from a publicly available Reddit archive, selecting five Romanian subreddits. The social media posts are freely accessible to the public without any type of subscription. As the data was collected from an archived public website (Reddit), we adhere to the European regulations (<https://eur-lex.europa.eu/eli/dir/2019/790/oj>) that allow researchers to use data in the public web domain for non-commercial research purposes. We thus release our

corpus as open-source under a non-commercial share-alike license agreement, namely CC BY-NC-SA 4.0 (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).

We acknowledge that some posts could refer to certain people, e.g. public figures in Romania. Following GDPR regulations, we will remove all references to a person, upon receiving removal requests via an email to any of the authors.

References

1. Almazrouei, E., et al.: The falcon series of open language models. arXiv preprint [arXiv:2311.16867](https://arxiv.org/abs/2311.16867) (2023)
2. Barnes, K., Riesemy, T., Trinh, M.D., Lleshi, E., Balogh, N., Molontay, R.: Dank or not? Analyzing and predicting the popularity of memes on Reddit. *Appl. Netw. Sci.* **6**(1), 21 (2021)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **5**, 135–146 (2017)
4. Carta, S., Podda, A.S., Recupero, D.R., Saia, R., Usai, G.: Popularity prediction of Instagram posts. *Information* **11**(9), 453 (2020)
5. De, S., Maity, A., Goel, V., Shitole, S., Bhattacharya, A.: Predicting the popularity of Instagram posts for a lifestyle magazine using deep learning. In: *Proceedings of CSCITA*, pp. 174–177 (2017)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of NAACL*, pp. 4171–4186 (2019)
7. Dumitrescu, S.D., Avram, A.M., Pyysalo, S.: The birth of Romanian BERT. In: *Findings of EMNLP*, pp. 4324–4328 (2020)
8. Fang, Z., et al.: How to generate popular post headlines on social media? *AI Open* **5**, 1–9 (2024)
9. Ferrer, X., van Nuenen, T., Such, J.M., Criado, N.: Discovering and categorising language biases in Reddit. In: *Proceedings of ICWSM*, pp. 140–151 (2021)
10. Gjurković, M., Šnajder, J.: Reddit: a gold mine for personality prediction. In: *Proceedings of PEOPLES*, pp. 87–97 (2018)
11. Hada, R., Sudhir, S., Mishra, P., Yannakoudakis, H., Mohammad, S.M., Shutova, E.: Ruddy: norms of offensiveness for English Reddit comments. In: *Proceedings of ACL*, pp. 2700–2717 (2022)
12. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint [arXiv:1607.01759](https://arxiv.org/abs/1607.01759) (2016)
13. Kim, J.: Predicting the popularity of reddit posts with AI. arXiv preprint [arXiv:2106.07380](https://arxiv.org/abs/2106.07380) (2021)
14. Koxhlikyan, N., et al.: Captum: a unified and generic model interpretability library for PyTorch. arXiv preprint [arXiv:2009.07896](https://arxiv.org/abs/2009.07896) (2020)
15. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: *Proceedings of ICLR* (2019)
16. Ma, Z., Sun, A., Cong, G.: On predicting the popularity of newly emerging hashtags in Twitter. *J. Am. Soc. Inform. Sci. Technol.* **64**, 1399–1410 (2013)
17. Mahdavi, M., Asadpour, M., Ghavami, S.: A comprehensive analysis of tweet content and its impact on popularity. In: *Proceedings of IST*, pp. 559–564 (2016)
18. McHardy, R., Adel, H., Klinger, R.: Adversarial training for satire detection: controlling for confounding variables. In: *Proceedings of NAACL*, pp. 660–665 (2019)
19. Niculescu, M.A., Ruseti, S., Dascalu, M.: RoGPT2: Romanian GPT2 for text generation. In: *Proceedings of ICTAI*, pp. 1154–1161 (2021)

20. Poecze, F., Ebster, C., Strauss, C.: Social media metrics and sentiment analysis to evaluate the effectiveness of social media posts. In: Proceedings of ANT-SEIT, pp. 660–666 (2018)
21. Purba, K.R., Asirvatham, D., Murugesan, R.K.: Instagram post popularity trend analysis and prediction using hashtag, image assessment, and user history features. *Int. Arab J. Inf. Technol.* **18**(1), 85–94 (2021)
22. Shen, J.H., Rudzicz, F.: Detecting anxiety through Reddit. In: Proceedings of CLPsych, pp. 58–65 (2017)
23. Tadesse, M.M., Lin, H., Xu, B., Yang, L.: Detection of depression-related posts in reddit social media forum. *IEEE Access* **7**, 44883–44893 (2019)
24. Turcan, E., McKeown, K.: Dreddit: a Reddit dataset for stress analysis in social media. In: Proceedings of LOUHI, pp. 97–107 (2019)
25. Wang, C., Xiao, Z., Liu, Y., Xu, Y., Zhou, A., Zhang, K.: SentiView: sentiment analysis and visualization for Internet popular topics. *IEEE Trans. Hum.-Mach. Syst.* **43**(6), 620–630 (2013)
26. Zhang, Z., Chen, T., Zhou, Z., Li, J., Luo, J.: How to become Instagram famous: post popularity prediction with dual-attention. arXiv preprint [arXiv:1809.09314](https://arxiv.org/abs/1809.09314) (2019)
27. Zhao, Q., Erdogdu, M.A., He, H.Y., Rajaraman, A., Leskovec, J.: SEISMIC: a self-exciting point process model for predicting tweet popularity. In: Proceedings of KDD, pp. 1513–1522 (2015)
28. Zhu, Y., ul Haq, E., Lee, L.H., Tyson, G., Hui, P.: A Reddit dataset for the Russo-Ukrainian conflict in 2022. arXiv preprint [arXiv:2206.05107](https://arxiv.org/abs/2206.05107) (2022)



ConCSE: Unified Contrastive Learning and Augmentation for Code-Switched Embeddings

Jangyeong Jeon¹ , Sangyeon Cho¹ , Minuk Ma² , and Junyeong Kim¹  

¹ Department of Artificial Intelligence, Chung-Ang University,
Seoul 06974, Republic of Korea

{jjy6133,whtkddus98,junyeongkim}@cau.ac.kr

² Department of Computer Science, University of British Columbia,
Vancouver, BC V6T 1Z4, Canada
minukma@cs.ubc.ca

Abstract. This paper examines the Code-Switching (CS) phenomenon where two languages intertwine within a single utterance. There exists a noticeable need for research on the CS between English and Korean. We highlight that the current Equivalence Constraint (EC) theory for CS in other languages may only partially capture English-Korean CS complexities due to the intrinsic grammatical differences between the languages. We introduce a novel Koglish dataset tailored for English-Korean CS scenarios to mitigate such challenges. First, we constructed the Koglish-GLUE dataset to demonstrate the importance and need for CS datasets in various tasks. We found the differential outcomes of various foundation multilingual language models when trained on a monolingual versus a CS dataset. Motivated by this, we hypothesized that SimCSE, which has shown strengths in monolingual sentence embedding, would have limitations in CS scenarios. We construct a novel Koglish-NLI (Natural Language Inference) dataset using a CS augmentation-based approach to verify this. From this CS-augmented dataset Koglish-NLI, we propose a unified contrastive learning and augmentation method for code-switched embeddings, ConCSE, highlighting the semantics of CS sentences. Experimental results validate the proposed ConCSE with an average performance enhancement of 1.77% on the Koglish-STs(Semantic Textual Similarity) tasks. (Source code available at <https://github.com/jjy961228/ConCSE>).

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics) and partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-01341, Artificial Intelligence Graduate School Program, Chung-Ang University).

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78498-9_3.

Keywords: Natural Language Processing · Contrastive Learning · Code-Switching · Bilingual · Koglish dataset

1 Introduction

Code-switching (CS) refers to the phenomenon of two languages intermixed within a single sentence [8, 29]. Such occurrences are frequently observed in multicultural countries, social media, and online platforms [6, 9, 29]. According to recent findings, despite the growing interest in CS, there remains a dearth of related studies [32]. Especially in countries where English is not the dominant language, the phenomenon of CS between English and the native language is particularly prominent [7, 9, 25, 27, 29, 41]. For example, the English sentence “*The movie was very dull*” can be represented as “영화가 was very dull.” for English-Korean and “*la película was very dull.*” for English-Spanish.

Past research introduced the Equivalence Constraint (EC) theory as a condition for the occurrence of CS [29], prompting attempts to construct CS datasets based on The EC theory [30, 33]. The EC theory posits that switches between languages in a code-switched discourse tend to happen at points where the grammatical structures of the involved languages match. According to the EC theory, such alignment in grammatical structures demonstrates that code-switching adheres to systematic linguistic constraints. This foundational concept has been central in many CS studies, particularly language pairs like English-Spanish and English-Chinese [19, 29, 30, 37, 38]. However, studies on CS between English and Korean show that this assumption is not always met [41]. For English-Korean CS, there is a potential limitation that EC Theory does not satisfy due to the grammatical difference between the two languages. For instance, the grammatical differences between English and Korean primarily manifest in word order and case marking. English predominantly follows an SVO (Subject-Verb-Object) word order, and this sequence largely determines the meaning of a sentence. In contrast, Korean offers greater flexibility in the positioning of subjects and objects, thanks in large part to its distinctive case markers like “이|i|가|ga)” (nominative), “을|eul|/를|leul)” (accusative), and “에게|ege)” (dative). Crucially, altering the word order in English can significantly change the meaning of a sentence, whereas, in Korean, where the language’s case markers are well developed, position shifts within sentence components are accessible [22, 41].

This paper introduces a novel Koglish dataset and proposes a new approach to constructing CS datasets, considering the inherent complexity of CS. The Koglish dataset includes Koglish-GLUE, Koglish-NLI, and Koglish-STS datasets. In particular, we propose to apply constituency parsing [20] to construct the Koglish dataset to obtain parse trees and transform English sentences into CS sentences following the approach proposed in Sect. 3.2. To construct the Koglish dataset, We utilize GLUE benchmark [34], Semantic Textual Similarity (STS) [1–5, 24], The Stanford Natural Language Inference Corpus (SNLI) [10], and The Multi-Genre Natural Language Inference Corpus (MNLI) [36]. To better understand the need for code-switching (CS) datasets, we posited the following

hypothesis: There will be a noticeable difference in performance between training with a monolingual dataset and then testing on a CS dataset (EN2CS) versus conducting both training and testing with a CS dataset (CS2CS). This significant disparity underscores the importance of using our CS dataset, Koglish, in CS scenarios. To our knowledge, this is the first presentation of Koglish datasets suitable for English-Korean and Korean-English scenarios.

Determining semantic relationships between sentences is a critical challenge in natural language processing. Recently, contrastive learning drew significant attention in natural language processing [13, 16, 39], where the model learns to distinguish between pairs of similar and dissimilar samples. For example, SimCSE [16] proposed to convert the sentence pairs of (premise, hypothesis) in the Natural Language Inference (NLI) dataset [10, 36] into the triplets of (premise, entailment, contradiction) to provide extra signals for contrastive learning. However, the study of contrastive learning under code-switched sentences has been largely yet to be underexplored. To address this issue, we propose a unified contrastive learning and data augmentation method dubbed ConCSE to model the code-switched sentences explicitly. For each sentence triplet of (premise, entailment, contradiction), we generate a triplet of code-switched sentences (CS-premise, CS-entailment, CS-contradiction) via CS-augmentation in Sect. 3.2 using a constituency parser. Then it considers the relationships between the six sentences to define three novel loss functions: (1) Cross Contrastive Loss (\mathcal{L}_{CS}^{Cn}), (2) Cross Triplet Loss (\mathcal{L}_{CS}^{Tri}), and (3) Align Negative Loss (\mathcal{L}_{neg}^{Sim}), providing richer supervision compared to plain SimCSE. For example, the sentence pairs of (premise, CS-premise) are considered positive, while those of (CS-premise, contradiction) are considered negative. As a validation, we compared the performance of four baseline multilingual models across seven NLP tasks included in Koglish-STs. The baseline multilingual models struggle to perform on the code-switched scenarios, suggesting the intricacy and effectiveness of the Koglish dataset. The experiments on the ConCSE method on the Koglish-STs dataset show consistent performance improvements over SimCSE across seven semantic textual similarity (STs) tasks included in Koglish-STs.

Our contributions can be summarized as follows:

- We introduce the first dataset referred to as Koglish which is suitable for English-Korean and Korean-English CS scenarios including Koglish-GLUE¹, Koglish-STs^{2,3}, and Koglish-NLI⁴.
- We demonstrate the necessity of the Koglish dataset through various experiments.
- We propose an effective sentence representation learning method that considers the CS sentences through a specialized CS-focused augmentation technique.

¹ <https://huggingface.co/datasets/Jangyeong/Koglish-GLUE>.

² <https://huggingface.co/datasets/Jangyeong/Koglish-STs>.

³ <https://huggingface.co/datasets/Jangyeong/Koglish-SICK>.

⁴ <https://huggingface.co/datasets/Jangyeong/Koglish-NLI>.

2 Related Work

2.1 Theoretical Foundations of Code-Switching

In previous research, the conditions for the occurrence of Code-Switching (CS) and Code-Mixing (CM) were proposed as the Equivalence Constraint (EC) theory, Matrix Language Framework (MLF), and Functional Head Constraint. Notably, when the EC Theory criteria are met, studies have constructed CS and CM datasets using a Constituency parser [30,33]. This approach has found application in English-Chinese Code-Switching studies as well [30,38]. However, investigations into English-Korean CS have demonstrated that most instances do not conform to the EC theory, indicating its unsuitability for English-Korean CS scenarios [27,28,41]. The research highlights that in English-Korean and Korean-English code-switching, nouns or noun phrases often serve as the Embedded Language (EL), with their usage being notably prevalent, accounting for 74.6% and 61% respectively [28,41]. These prior empirical results showed the importance of selecting nouns or noun phrases as EL in constructing an English-Korean CS dataset. Pursuing this approach, our study uses a pre-trained Constituency parser [20] to identify and extract nouns or noun phrases.

2.2 Representation Learning

Deep Metric Learning. Deep Metric Learning was formulated to decipher the dynamics of embedding spaces [12,17,35]. Among its diverse strategies, triplet loss stands out [18]. It emphasizes the interrelationships and distances of samples within the embedding space, aiming to cluster similar samples and distance dissimilar ones closely. A pivotal element in this approach is the ‘margin’, a hyperparameter designed to ensure a defined distance between the anchor-positive and anchor-negative pairs [31]. This paper utilizes triplet loss as an auxiliary loss to bolster the model’s stability.

Contrastive Learning. In fields like natural language processing [13,16,39] and computer vision [11,21], the core aim is to enhance representations by discerning between positive and negative samples. Contrastive learning, which builds upon the foundations of deep metric learning, offers refined techniques for achieving superior representations. A notable advancement is the introduction of data augmentation to enrich training datasets. While random cropping and image rotation succeed in computer vision [11,40], their adaptation to natural language processing poses challenges. Nevertheless, strategies reconstructing NLI datasets for contrastive learning have been proposed to bridge this gap [13,16]. In particular, in the strategy of reconstructing NLI datasets [13,16], first, all datasets labeled as neutral are excluded, and only datasets labeled as entailment for two sentences (premise, hypothesis) are extracted. In this case, the premise and hypothesis are defined as a positive pair, and the hypothesis is defined as an entailment sentence. Second, extract hypothesis sentences where the hypothesis is labeled as a contradiction for the same sentence as the premise used in the

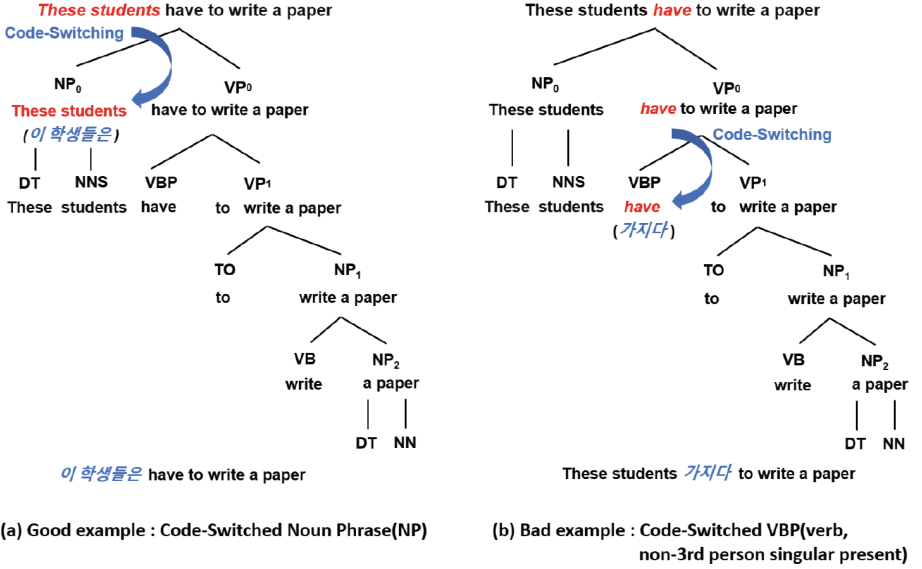


Fig. 1. Schematic of the parse tree, based on constituency parsing, to convert a monolingual sentence into an English-Korean code-switched sentence.

first step. In this case, the premise and hypothesis are defined as a negative pair, and the hypothesis is defined as a contradiction sentence. The NLI dataset was redefined as premise, entailment, and contradiction sentences and used for training SimCSE. Yet, the proposed strategies of SimCSE are limited to monolingual datasets. To address this limitation, our study presents a novel method: augmenting a resource-rich English dataset with a CS dataset in a supervised setting.

3 Proposed Dataset: Koglish

This section elaborates on English-Korean and Korean-English code-switching sentences and our specialized Koglish dataset construction and CS augmentation strategies. A summary of the constructed dataset is provided in Table 1.

3.1 Code-Switching Patterns and Dataset Construction

According to a study by [27], Code-Switching (CS) between English and Korean does not adhere to the guidelines established by the EC Theory [29] and the Matrix Language Frame (MLF) Model [26]. This is due to the fact that the grammatical units (e.g., phrase, adjective phrase, verb phrase) converted in CS are language-specific. Consequently, when constructing CS datasets, it is imperative to use strategies tailored to each respective language [7, 25, 27–29]. Historical analyses indicate that in Korean-English CS, nouns and noun phrases constitute 74.6% of code-switched instances [28]. English-Korean exhibits a similar

Table 1. Summary of Koglish Datasets.

GLUE Benchmark				
Task	Train	Dev	Test	Total
QNLI	61,764	15,441	19,303	96,508
SST-2	26,552	6,632	8,289	41,473
COLA	4,341	1,087	1,358	6,759
STS-B	4,269	1,068	1,335	6,672
MRPC	3,610	904	1,129	5,643
RTE	1,642	412	514	2,568
Semantic Textual Similarity(STS)				
Task	Train	Dev	Test	Total
STS-B	-	3,334	3,334	6,668
STS12	-	2,142	2,142	4,286
STS13	-	622	622	1,244
STS14	-	1,561	1,561	3,112
STS15	-	1,351	1,351	2,702
STS16	-	496	496	992
SICK-R	-	4,767	4,767	9,534
Natural Language Inference(NLI)				
Task	Train	Dev	Test	Total
NLI	218,255	-	-	218,255

trend, with nouns representing 61% of code-switched [41]. As shown in Fig. 1(a), code-switching the noun phrase maintains the sentence’s integrity, mirroring the structure of the original. In contrast, code-switching VBP (Verb, non-3rd person singular present) as shown in Fig. 1(b), produces a sentence that is awkwardly constructed. Japanese, sharing syntactic similarities with Korean, also has a high noun switching rate at 68.8% [25]. This structural congruence suggests the potential for applying our CS dataset construction strategy to other languages with grammatical structures akin to Korean’s [41]. In contrast, Spanish-English code-switching contains a significantly lower noun switch rate, sometimes reaching lower than 20% [29]. Given these patterns, we primarily focused on switching nouns or noun phrases when constructing English-Korean and Korean-English CS datasets. Additionally, due to the distinction between Matrix Language (ML) and Embedded Language (EL) is not explicit in English-Korean code-switching [26], the dominant use of nouns and noun phrases in both English-Korean and Korean-English code-switching endorses the suitability of our proposed dataset strategy for both scenarios.

3.2 Constructing Koglish Dataset

This section details constructing and augmenting the proposed CS dataset, Koglish. The overall process is shown in Fig. 2.

1. We constructed a parse tree using a top-down constituency parsing approach [20]. During this process, we selectively extracted the *NP* nodes, ensur-

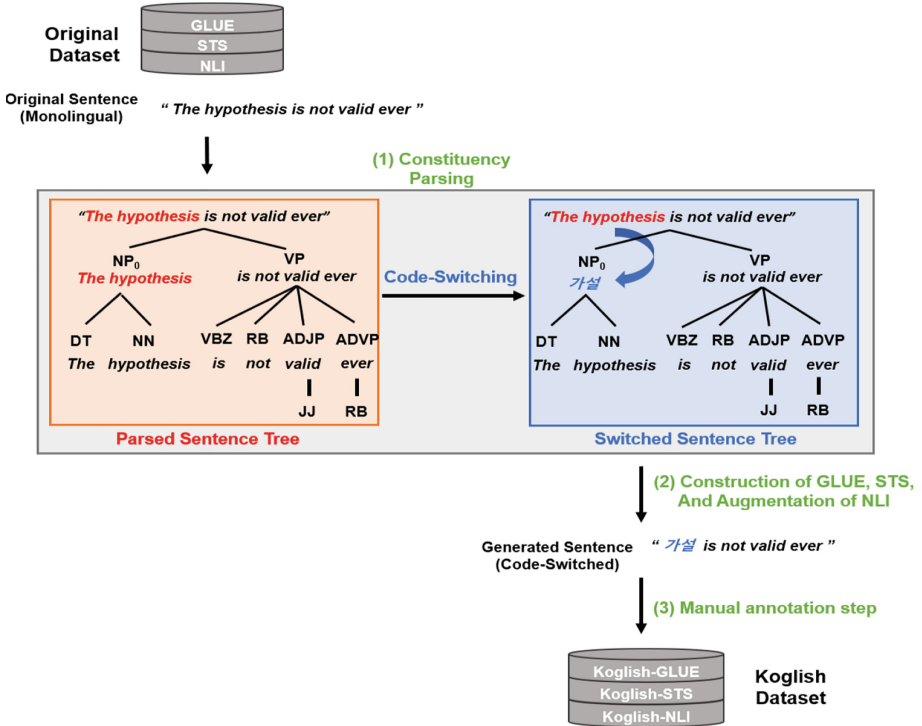


Fig. 2. Systematic Approach to Constructing and Augmenting the Koglish Dataset. Constituency parser extracts nouns or noun phrases (NP) using the Google Translate API. In this case, GLUE and STS datasets are generated as CS datasets, and NLI datasets are CS-augmented.

ing the inclusion of nouns and noun phrases (see Fig. 2(1)). In some data, entire sentences were constructed solely from the NP structure. When such sentences underwent the translation process, they resulted in monolingual sentences, negating the goal of CS. Therefore, we excluded these particular entries. Additionally, if the NP_0 node contained only pronouns (e.g., It, That, This), it led to mistranslation issues. To address this, we extracted the NP node from the subsequent NP_1 node and applied the top-down approach to the leaf nodes. If the data did not align with our criteria when it reached the leaf node, we considered it inappropriate for the CS dataset and subsequently excluded it. For example, GLUE's COLA task data excluded 29.1% of the entire data.

2. Generate CS sentences from the Switched Sentence Tree of Fig. 2(1) as shown in Fig. 2(2). The first is the GLUE [34] and STS dataset [1–5, 24], and the second is the NLI [10, 36] dataset. As an example of the first, GLUE and STS take monolingual sentences as input and generate a CS sentence if it satisfies the abovementioned conditions (in step 1). The second example is the

NLI dataset, which receives triplets of monolingual sentences (e.g., premise, entailment, and contradiction) as input. If the above conditions (in step 1) are satisfied for the triplet of monolingual sentences, it generates CS-premise, CS-entailment, and CS-contradiction. In the following Sect. 4, the three sentences (premise, entailment, and contradiction) of the Koglish-NLI dataset and CS-Augmented sentences (CS-premise, CS-entailment, and CS-contradiction) are integrated, and used for learning ConCSE, so in this paper, we assume that only NLI is CS-Augmented sentences.

3. To ensure reliability and accuracy, we performed critical manual annotations on the generated Koglish datasets. This process involved bilingual experts proficient in both Korean and English. We employed four annotators, each tasked with evaluating the contextual accuracy of the Code-Switching sentences in the dataset. Following their assessments, the four annotators produced each dataset through a meticulous cross-validation process, rigorously examining each other’s evaluations (see Fig. 2(3)). Finally, we split each dataset. The Koglish-GLUE was divided into train, development, and test sets in the ratios of 0.64, 0.16, and 0.20, respectively, to formulate the Koglish-GLUE dataset. Since the Koglish-STS dataset is only used to evaluate ConCSE in Sect. 5.2, we constructed the Koglish-STS dataset by splitting the development and test sets equally (0.5 ratios each). We constructed Koglish-NLI without any segmentation since the Koglish-NLI dataset is only used for training.

4 Proposed Method: ConCSE

This paper aims to train universal sentence embeddings in Code-Switching (CS) contexts. As detailed in step 2 of Sect. 3.2, we use the monolingual datasets $\mathcal{D}_{en} = \{x_i, x_i^+, x_i^-\}_{i=1}^m$ and the augmented CS datasets $\mathcal{D}_{cs} = \{\hat{x}_i, \hat{x}_i^+, \hat{x}_i^-\}_{i=1}^m$ to fine-tune a pre-trained multilingual sentence encoder \mathcal{M}_ϕ , such as mBERT [15] or XLM-R [14], to adapt to the CS scenario.

The notation for the comprehensive loss function used is:

$$\mathcal{L}_{total} = \mathcal{L}_{CS}^{Con} + \lambda \mathcal{L}_{CS}^{Tri} + \mathcal{L}_{neg}^{Sim} \quad (1)$$

where λ signifies the weight factor assigned to the triplet loss. Detailed explanations of \mathcal{L}_{CS}^{Con} , \mathcal{L}_{CS}^{Tri} , and \mathcal{L}_{neg}^{Sim} can be found in Sect. 4.1, 4.2, and 4.3, respectively. An overview of ConCSE is shown in Fig. 3.

4.1 Cross Contrastive Loss

We train \mathcal{M}_ϕ with Cross Contrastive Loss (\mathcal{L}_{CS}^{Con}) on monolingual and CS sentences. The hidden state of “[CLS]” for \mathcal{D}_{en} within \mathcal{M}_ϕ is defined as:

$$H = \{h_i, h_i^+, h_i^-\}_{i=1}^N \quad (2)$$

For \mathcal{D}_{cs} within \mathcal{M}_ϕ , it is defined as :

$$\hat{H} = \{\hat{h}_i, \hat{h}_i^+, \hat{h}_i^-\}_{i=1}^N \quad (3)$$

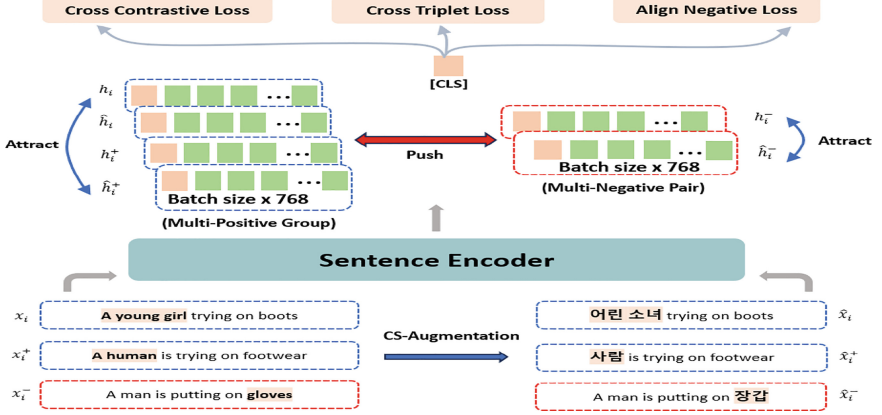


Fig. 3. Overview of ConCSE. A mini-batch contains both $\mathcal{D}_{en} = \{x_i, x_i^+, x_i^-\}_{i=1}^m$ and CS-augmented $\mathcal{D}_{cs} = \{\hat{x}_i, \hat{x}_i^+, \hat{x}_i^-\}_{i=1}^m$, and its hidden representations are $H = \{h_i, h_i^+, h_i^-\}_{i=1}^N$ and $\hat{H} = \{\hat{h}_i, \hat{h}_i^+, \hat{h}_i^-\}_{i=1}^N$. They are processed by the sentence encoder \mathcal{M}_ϕ , producing “[CLS]” as the final sentence representation. The “[CLS]” of the multi-positive group, comprising monolingual sentences (h_i, h_i^+) and CS sentences (\hat{h}_i, \hat{h}_i^+), should be attracted to each other. Similarly, the “[CLS]” of the multi-negative pair, comprising a monolingual sentence (h_i^-) and CS sentence (\hat{h}_i^-), should also be attracted to each other. Moreover, multi-positive groups and multi-negative pairs should push each other.

where N is the batch size. This paper extends contrastive loss to include six combinations, facilitating cross-training on \mathcal{D}_{en} and \mathcal{D}_{cs} :

$$\begin{aligned} \mathcal{H}^1 &= \{h_i, h_i^+, h_i^-\}_{i=1}^N, \mathcal{H}^2 = \{\hat{h}_i, \hat{h}_i^+, \hat{h}_i^-\}_{i=1}^N, \\ \mathcal{H}^3 &= \{h_i, h_i^+, \hat{h}_i^-\}_{i=1}^N, \mathcal{H}^4 = \{\hat{h}_i, \hat{h}_i^+, h_i^-\}_{i=1}^N, \\ \mathcal{H}^5 &= \{h_i, \hat{h}_i, h_i^-\}_{i=1}^N, \mathcal{H}^6 = \{h_i^+, \hat{h}_i^+, \hat{h}_i^-\}_{i=1}^N \end{aligned} \quad (4)$$

For instance, the loss function $\mathcal{L}_{\mathcal{H}^3}^{con}$, which cross-trains on \mathcal{D}_{en} and \mathcal{D}_{cs} , can be denoted as:

$$\mathcal{L}_{\mathcal{H}^3}^{Con} = \sum_{i=1}^N -\log \frac{e^{sim(h_i, h_i^+)/\tau}}{\sum_{j=1}^N (e^{sim(h_i, h_j^+)/\tau} + e^{sim(h_i, \hat{h}_j^-)/\tau})} \quad (5)$$

where, $sim(\cdot, \cdot)$ is the cosine similarity function.

By integrating from Eq. 4 and 5, the Cross Contrast Loss \mathcal{L}_{CS}^{Con} is calculated as follows:

$$\mathcal{L}_{CS}^{Con} = \sum_{k=1}^6 \mathcal{L}_{\mathcal{H}^k}^{Con} \quad (6)$$

4.2 Cross Triplet Loss

Following the proposed Cross Contrastive Loss (\mathcal{L}_{CS}^{Con}), the triplet loss [31] is introduced to adjust the distance between the anchor and positive and the distance between the anchor and negative by a margin (α). Triplet loss can be extended to six combinations, as in Eq. 4, to allow cross-training on \mathcal{D}_{en} and \mathcal{D}_{cs} . An example for $\mathcal{L}_{\mathcal{H}^3}^{Tri}$ is defined as:

$$\mathcal{L}_{\mathcal{H}^3}^{Tri} = \sum_{i=1}^N \max(0, \|h_i - h_i^+\|_2^2 - \|h_i - \hat{h}_i^-\|_2^2 + \alpha) \quad (7)$$

where N is the batch size. To this end, \mathcal{L}_{CS}^{Tri} , derived from Eq. 4 and 7, is defined as:

$$\mathcal{L}_{CS}^{Tri} = \sum_{k=1}^6 \mathcal{L}_{\mathcal{H}^k}^{Tri} \quad (8)$$

4.3 Align Negative Loss

The negative samples from \mathcal{D}_{en} and \mathcal{D}_{cs} should share the same meaning, implying that they should be in a positive relationship with each other. We define the loss function \mathcal{L}_{neg}^{sim} to encode this relationship into the \mathcal{M}_ϕ :

$$\mathcal{L}_{neg}^{sim} = \sum_{i=1}^N CE(sim(h_i^-, \hat{h}_i^-)) \quad (9)$$

where $CE(\cdot)$ denotes cross-entropy loss, $sim(\cdot, \cdot)$ is the cosine similarity function, and N is the batch size.

5 Experiments

5.1 Experiments on Koglish: The Role of Koglish in Code-Switching Scenario

Setup. In this experiment, we utilize our Koglish-GLUE dataset. Considering the MRPC task as an example, which determines if a pair of sentences in the Koglish-GLUE dataset are semantically equivalent: this task comprises the original English sentences, namely sentence0 and sentence1 from GLUE, as well as the Code-Switched (CS) versions, CS-sentence0 and CS-sentence1. For example, in the EN2CS scenario, we perform training and evaluation using only the monolingual English dataset sentence0 and sentence1. In the EN2CS scenario, sentence0 and sentence1 serve as the training data, while CS-sentence0 and CS-sentence1 are utilized for evaluation. Detailed information regarding the data used in the experiments is provided in Table 1. The evaluation metrics for each experiment align with those adopted in BERT [15]. Specifically, the MRPC uses

Table 2. Comparative performance of various multilingual models on Koglish-GLUE across Different Scenarios: Monolingual (EN2EN), English to Code-Switching (EN2CS), and Code-Switching to Code-Switching (CS2CS). **Best performances in EN2CS and CS2CS are highlighted.** The MRPC uses the F1-score, STS-B uses Spearman’s correlation, and the remaining tasks use accuracy for performance measurement.

English to English(EN2EN)							
Model	COLA	SST-2	MRPC	RTE	STS-B	QNLI	Avg.
mBERT _{base}	74.66 ± 1.52	92.62 ± 0.47	87.23 ± 2.24	66.28 ± 1.11	87.18 ± 0.69	88.48 ± 0.26	82.74
XLM-R _{base}	72.37 ± 0.59	93.85 ± 0.32	88.38 ± 0.69	60.27 ± 1.97	87.23 ± 0.46	87.87 ± 0.19	81.66
XLM-R _{large}	79.90 ± 4.03	95.10 ± 0.14	89.53 ± 0.80	65.34 ± 4.14	90.51 ± 0.28	91.82 ± 0.14	85.37
mBART _{large}	78.94 ± 0.40	94.29 ± 0.09	89.32 ± 0.31	69.40 ± 1.10	89.24 ± 0.32	90.83 ± 0.09	85.34
English to Code-Switching(EN2CS)							
Model	COLA	SST-2	MRPC	RTE	STS-B	QNLI	Avg.
mBERT _{base}	68.59 ± 3.86	81.48 ± 2.04	79.18 ± 2.43	56.10 ± 1.86	74.42 ± 1.51	77.75 ± 0.6	72.92
XLM-R _{base}	72.20 ± 0.27	88.55 ± 0.46	83.65 ± 1.84	54.58 ± 1.18	78.84 ± 1.04	79.44 ± 0.4	76.21
XLM-R _{large}	74.61 ± 1.98	91.78 ± 0.13	87.98 ± 0.49	62.88 ± 4.19	87.73 ± 0.14	88.27 ± 0.28	82.19
mBART _{large}	56.36 ± 2.77	88.19 ± 0.19	86.83 ± 0.32	62.30 ± 1.24	79.24 ± 0.61	84.96 ± 0.31	76.31
Code-Switching to Code-Switching(CS2CS)							
Model	COLA	SST-2	MRPC	RTE	STS-B	QNLI	Avg.
mBERT _{base}	72.48 ± 2.04	89.83 ± 1.04	80.80 ± 1.85	57.31 ± 4.38	81.77 ± 1.28	83.91 ± 0.32	77.68
XLM-R _{base}	72.07 ± 0.00	91.29 ± 0.44	85.52 ± 1.52	53.57 ± 1.39	81.64 ± 2.36	84.96 ± 0.15	78.18
XLM-R _{large}	74.35 ± 1.51	93.69 ± 0.05	88.68 ± 1.1	60.36 ± 6.74	88.54 ± ± 0.34	90.31 ± 0.15	82.64
mBART _{large}	74.37 ± 0.78	92.60 ± 0.09	86.85 ± 0.71	62.38 ± 0.97	85.09 ± 0.29	88.39 ± 0.07	81.61

the F1-score, STS-B uses Spearman’s correlation, and the remaining tasks rely on accuracy for performance measurement.

The central hypothesis of this experiment is twofold. First, if the proposed CS dataset construction method is valid, the performance difference between English to English(EN2EN) and Code-Switching to Code-Switching(CS2CS) should be insignificant. Second, if CS2CS’s performance is better than EN2CS’s, this suggests the need for fine-tuning using proposed CS datasets in a CS scenario.

Training Details. We initiated our experiments based on the checkpoints of four pre-trained multilingual models: mBERT_{base} [15], XLM-R_{base}, XLM-R_{large} [14], and mBART [23]. We utilized the representation of the “[CLS]” token as the final sentence embedding to validate the performance. More training details can be found in Appendix A.

Results. Table 2 shows the results of the experimental outcomes using the Koglish-GLUE dataset. We observed that the models trained and evaluated on English-only data (EN2EN) outperformed those trained and evaluated on a mixed English-Korean Code-Switching dataset (CS2CS). This disparity stems from EN2EN’s monolingual nature and CS2CS’s bilingual complexity, which

Table 3. Performance comparison of SimCSE and ConCSE on various Koglish-STS tasks: Performance is measured in terms of Spearman’s correlation in “all” settings. **Bold values highlight the top performance in each task.** “v.s” in the table is the result of the T-test between SimCSE and ConCSE.

Model	STS-B	STS12	STS13	STS14	STS15	STS16	SICK	Avg.
SimCSE-mBERT _{base}	77.68 ± 0.17	63.43 ± 0.22	70.93 ± 0.33	68.59 ± 0.32	79.07 ± 0.28	72.52 ± 0.28	75.91 ± 0.08	72.59
SimCSE-XLM-R _{base}	78.65 ± 0.30	67.49 ± 0.60	75.55 ± 0.24	71.40 ± 0.18	80.03 ± 0.38	76.65 ± 0.34	77.22 ± 0.20	75.29
SimCSE-XLM-R _{large}	82.43 ± 0.12	71.02 ± 0.21	82.28 ± 0.32	76.19 ± 0.23	83.11 ± 0.23	79.52 ± 0.25	79.05 ± 0.20	79.09
*ConCSE-mBERT _{base}	79.95 ± 0.24	68.29 ± 0.21	70.52 ± 1.12	71.24 ± 0.32	80.40 ± 0.27	73.13 ± 0.52	77.01 ± 0.22	74.36
*ConCSE-XLM-R _{base}	79.93 ± 0.26	71.27 ± 0.43	75.56 ± 0.63	74.23 ± 0.28	80.94 ± 0.31	76.17 ± 0.22	78.08 ± 0.16	76.60
*ConCSE-XLM-R _{large}	82.85 ± 0.11	75.00 ± 0.34	82.72 ± 0.23	77.80 ± 0.27	84.12 ± 0.23	79.43 ± 0.38	78.91 ± 0.30	80.12
p-value(T-test)	STS-B	STS12	STS13	STS14	STS15	STS16	SICK	
v.s SimCSE-mBERT _{base}	3.1×10^{-7}	9.8×10^{-10}	0.508	2.8×10^{-6}	1.3×10^{-4}	0.071	1.2×10^{-5}	
v.s SimCSE-XLM-R _{base}	2.0×10^{-4}	6.8×10^{-6}	0.996	1.4×10^{-7}	4.9×10^{-5}	0.046	1.4×10^{-5}	
v.s SimCSE-XLM-R _{large}	8.7×10^{-5}	4.0×10^{-8}	0.056	1.7×10^{-5}	2.0×10^{-5}	0.688	0.473	

operates within a Korean context with comparatively fewer resources. Additionally, larger models such as XLM-R and mBART exhibit a reduced performance discrepancy between EN2EN and CS2CS. This suggests that the models’ performance benefits from the increased diversity of training data spanning multiple languages. One of our experiment’s standout insights is the pronounced efficacy of the CS2CS method compared to EN2CS across almost all model evaluations. This outcome underscores the efficacy of using Code-Switching training in the Koglish dataset in contexts where English-Korean Code-Switching is prevalent.

5.2 Experiments on ConCSE

Setup. In this experiment, we utilize the Koglish-NLI dataset for training and the Koglish-STS dataset for evaluation. The Koglish-NLI dataset contains triplets of monolingual English sentences (hypothesis, entailment, and contradiction) alongside triplets of code-switched (CS) augmented sentences (CS-hypothesis, CS-entailment, and CS-contradiction). The Koglish-STS dataset consists of pairs of original sentences (sentence0 and sentence1) and their CS counterparts (CS-sentence0 and CS-sentence1). During the training phase, we leverage SimCSE [16] to train the sentence encoder $\mathcal{M}\phi$ using CS-augmented sentence triplets. Moreover, ConCSE trains $\mathcal{M}\phi$ on both triplets of original English sentences and CS-augmented sentences, promoting learning in a CS scenario. We evaluated both SimCSE and ConCSE using the CS sentence pairs from Koglish-STS. We adopt Spearman’s correlation as the primary metric for this assessment.

Training Details. In our experiments, we initialize our sentence encoder $\mathcal{M}\phi$ using pre-trained mBERT [15] or XLM-R [14], and we use “[CLS]” as $\mathcal{M}\phi$ final representation. We adopt SimCSE [16] as our baseline model during the implementation phase. Furthermore, as ConCSE had to handle a larger volume of sentences compared to SimCSE [16], we only adjusted the batch size. The rest of

the experimental settings were maintained identically to SimCSE. To ensure the accuracy and reliability of our results, we conducted experiments using five different random seeds and recorded the corresponding T-test results. More details on training can be found in Appendix B.

Results. The performance of SimCSE [16] and ConCSE is summarized in Table 3. In the CS scenario, our proposed ConCSE significantly outperforms SimCSE because it helps implicitly align the representations across languages in CS situations. Specifically, ConCSE-mBERT_{base} outperforms SimCSE-mBERT_{base} by improving the average Spearman’s correlation score from 72.59% to 74.36%, which significantly outperforms. We also note consistent performance enhancements with ConCSE-XLM-R_{base} and ConCSE-XLM-R_{large} models. These improvements across all ConCSE backbone models are instrumental in augmenting the comprehension of CS contexts, demonstrating their significant impact. Furthermore, The results demonstrate that our ConCSE can scale to multiple datasets and more languages in CS scenarios.

5.3 Ablation Studies

In this section, we conduct a comprehensive set of ablation studies to substantiate our ConCSE architecture. Particularly, we evaluated the effects of the combination of the loss function and the effects of temperature, triplet loss, and margin on training by testing the ConCSE-mBERT_{base} on the Koglish-STS-B task. Detailed experimental results related to these ablation studies can be found in Appendix C.

6 Conclusion

In this work, we first introduced the novel Koglish dataset, focusing on code-switching (CS) between English-Korean and Korean-English. This Koglish dataset marks an initial pioneering attempt, and exhaustive evaluations have highlighted the critical need for such a resource. Second, we propose a method to learn universal code-switched sentence embeddings using this newly constructed Koglish dataset. Surprisingly, Through extensive testing, ConCSE surpassed other leading sentence embedding techniques in Koglish-STS tasks. Nevertheless, our study has certain constraints: Although less frequent, grammatical elements other than nouns or noun phrases can also be CS in English-Korean CS situations. In our future work, we aim to develop a more comprehensive CS dataset encompassing all grammatical elements. We are optimistic that our contributions will spur further research and progress in the understanding and application of low-resource CS data.

References

1. Agirre, E., et al.: Semeval-2015 task 2: semantic textual similarity, English, Spanish and pilot on interpretability. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 252–263 (2015)
2. Agirre, E., et al.: Semeval-2014 task 10: multilingual semantic textual similarity. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 81–91 (2014)
3. Agirre, E., et al.: Semeval-2016 task 1: semantic textual similarity, monolingual and cross-lingual evaluation. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 497–511 (2016)
4. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A.: Semeval-2012 task 6: a pilot on semantic textual similarity. In: * SEM 2012: The First Joint Conference on Lexical and Computational Semantics—Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), pp. 385–393 (2012)
5. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W.: * SEM 2013 shared task: semantic textual similarity. In: Second Joint Conference on Lexical and Computational Semantics (* SEM), volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pp. 32–43 (2013)
6. Ahn, J., La Ferle, C., Lee, D.: Language and advertising effectiveness: code-switching in the Korean marketplace. *Int. J. Advert.* **36**(3), 477–495 (2017)
7. Amazouz, D., Adda-Decker, M., Lamel, L.: Addressing code-switching in French/Algerian Arabic speech. In: Interspeech 2017, pp. 62–66 (2017)
8. Auer, P.: *Code-Switching in Conversation: Language, Interaction and Identity*. Routledge (2013)
9. Baker, C.: *Foundations of bilingual education and bilingualism. Multilingual matters* (2011)
10. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. arXiv preprint [arXiv:1508.05326](https://arxiv.org/abs/1508.05326) (2015)
11. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607. PMLR (2020)
12. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 539–546. IEEE (2005)
13. Chuang, Y.S., et al.: DiffCSE: difference-based contrastive learning for sentence embeddings. In: Carpuat, M., de Marneffe, M.C., Meza Ruiz, I.V. (eds.) Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4207–4218. Association for Computational Linguistics, Seattle, United States (2022). <https://doi.org/10.18653/v1/2022.naacl-main.311>. <https://aclanthology.org/2022.naacl-main.311>
14. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451 (2020)
15. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American

- Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019). <https://doi.org/10.18653/v1/N19-1423>. <https://aclanthology.org/N19-1423>
16. Gao, T., Yao, X., Chen, D.: Simcse: simple contrastive learning of sentence embeddings. arXiv preprint [arXiv:2104.08821](https://arxiv.org/abs/2104.08821) (2021)
 17. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), vol. 2, pp. 1735–1742. IEEE (2006)
 18. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, 12–14 October 2015, Proceedings 3, pp. 84–92. Springer (2015)
 19. Hu, X., Zhang, Q., Yang, L., Gu, B., Xu, X.: Data augmentation for code-switch language modeling by fusing multiple text generation methods. In: INTER-SPEECH, pp. 1062–1066 (2020)
 20. Joshi, V., Peters, M., Hopkins, M.: Extending a parser to distant domains using a few dozen partially annotated examples. arXiv preprint [arXiv:1805.06556](https://arxiv.org/abs/1805.06556) (2018)
 21. Kuang, H., et al.: Video contrastive learning with global context. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3195–3204 (2021)
 22. Lehmann, W.P.: A structural principle of language and its implications. *Language* 47–66 (1973)
 23. Liu, Y., et al.: Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguist.* **8** (2020)
 24. Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., Zamparelli, R.: A sick cure for the evaluation of compositional distributional semantic models
 25. Miwa, N.: Intrasentential codeswitching in Japanese and English. Ph.D. dissertation. University of Pennsylvania (1985)
 26. Myers-Scotton, C.: Intersections between social motivations and structural processing in code-switching. In: Workshop on Constraints, Conditions and Models, London, pp. 27–29 (1990)
 27. Park, J.E.: Korean/English intrasentential code-switching: matrix language assignment and linguistic constraints. University of Illinois at Urbana-Champaign (1990)
 28. Park, J.E., Troike, R.C., Park, M.R.: Constraints in Korean-English code-switching: a preliminary study. *응용언어학* (6), 115–133 (1993)
 29. Poplack, S.: Sometimes i'll start a sentence in spanish y termino en espanol: toward a typology of code-switching¹ (1980)
 30. Pratapa, A., Bhat, G., Choudhury, M., Sitaram, S., Dandapat, S., Bali, K.: Language modeling for code-mixing: the role of linguistic theory based synthetic data. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1543–1553 (2018)
 31. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
 32. Sebba, M., Mahootian, S., Jonsson, C.: Language mixing and code-switching in writing: approaches to mixed-language written discourse. Routledge (2012)
 33. Srinivasan, A., Dandapat, S., Choudhury, M.: Code-mixed parse trees and how to find them. In: Proceedings of the The 4th Workshop on Computational Approaches to Code Switching, pp. 57–64 (2020)

34. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Glue: a multi-task benchmark and analysis platform for natural language understanding. arXiv preprint [arXiv:1804.07461](https://arxiv.org/abs/1804.07461) (2018)
35. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.* **10**(2) (2009)
36. Williams, A., Nangia, N., Bowman, S.R.: A broad-coverage challenge corpus for sentence understanding through inference. arXiv preprint [arXiv:1704.05426](https://arxiv.org/abs/1704.05426) (2017)
37. Winata, G.I., Madotto, A., Wu, C.S., Fung, P.: Code-switching language modeling using syntax-aware multi-task learning. arXiv preprint [arXiv:1805.12070](https://arxiv.org/abs/1805.12070) (2018)
38. Winata, G.I., Madotto, A., Wu, C.S., Fung, P.: Code-switched language models using neural based synthetic data from parallel sentences. arXiv preprint [arXiv:1909.08582](https://arxiv.org/abs/1909.08582) (2019)
39. Wu, X., Gao, C., Zang, L., Han, J., Wang, Z., Hu, S.: Esimcse: enhanced sample building method for contrastive learning of unsupervised sentence embedding. arXiv preprint [arXiv:2109.04380](https://arxiv.org/abs/2109.04380) (2021)
40. Yan, Y., Li, R., Wang, S., Zhang, F., Wu, W., Xu, W.: Consert: a contrastive framework for self-supervised sentence representation transfer. arXiv preprint [arXiv:2105.11741](https://arxiv.org/abs/2105.11741) (2021)
41. 박은선, 윤흥욱: 한국어-영어 코드 스위칭의 문법제약 및 문법 부호화. *영어영문학* **26**(1), 177–204 (2021)



Mitigating Hallucination in Large Language Model by Leveraging Decoder Layer Contrasting

Guangsheng Liu, Xinbo Ai^(✉), Wenbin Luo, and Ange Li

College of AI, Beijing University of Posts and Telecommunications, Beijing, China
{liuguangsheng, axb}@bupt.edu.cn

Abstract. Hallucination is a persistent challenge in large-scale language models, manifesting at multiple stages and leading to outputs that stray from reality and produce some content that does not conform to common sense. We introduce a novel approach to alleviate hallucination by contrasting the probability of intermediate layer with the last layer to obtain the next-token distribution during inference. Then, introduce in-layer stability factor to tackle the issue of token probability fluctuation across transformer layers. Our approach effectively addresses the issue of inconsistent output distributions from lower decoder layers in extensive models, evidenced by impressive results on benchmarks such as GSM8k, StrategyQA, and Wiki Factor. These outcomes highlight the significant potential of our method in reducing hallucination in large language models.

Keywords: Mitigating Hallucination · Decoder Layer Contrasting · Layer Stability Factor

1 Introduction

Following the success of ChatGPT, large language models (LLMs) are rapidly evolving, and exceeding human expert performance in some specialized competitions [1]. Nonetheless, LLMs still have some limitations, especially regarding hallucination where the model output content that deviates from reality or is factually incorrect. For out-of-distribution inputs or novel input contexts, LLMs can generate seemingly plausible yet ultimately inaccurate or nonsensical content [2]. Hallucination may arise when the model encounters data that differs substantially from the training data during inference, due to distributional shifts [3]. Study indicate that a major cause of hallucination is the training objective of predicting the next token's output probability based on maximum likelihood estimation. This training objective leads the token with the highest probability in the vocabulary, which is problematic because it assigns a non-zero probability to every token, even though most tokens are highly improbable [4].

Despite this, it remains the currently prevalent training objective. Some researchers posit that hallucination may stem from a lack of essential knowledge

not imparted during the pre-training phase [5]. It is propose that additional pre-training on an existing pre-trained model to incorporate the missing knowledge, which, however, could lead to catastrophic forgetting [6]. Recent research have focused on improving the following of LLMs to given instructions during the Supervised Fine-Tuning (SFT) stage as a strategy to exacerbate the hallucination. Nonetheless, improper supervised fine-tuning can significantly improve performance on new tasks while degrading performance on others [7]. Aligning LLMs with human intentions is another effective strategy for addressing hallucination, but study have shown a bias towards giving higher scores to more agreeable or flattering responses during manual evaluation, which can severely impact alignment and introduce further hallucination [8].

To address this, we propose a novel decoding algorithm to alleviate the hallucination in LLMs, without the need for retraining. During inference, it dynamically selects lower decoder layers and calculates the output probability distribution based on the difference from the last layer’s output probability. In the transformer architecture, lower decoders tend to encode general information, while higher decoders focus on semantic information [9]. Inspired by this, we aim to leverage the transformer’s modular information encoding to reinforce the factual knowledge of LLMs through a comparative method using output probability differences between decoder layers. This probability difference is derived from the logits discrepancy between high-level and low-level decoders, highlighting the significance of the higher decoder while deemphasizing basic knowledge. Additionally, we have introduced a metric to gauge the stability of the decoding layer, penalizing credibility issues caused by significant fluctuations in the output logits, thus reducing the hallucination phenomena in LLMs.

Our contributions can be summarized as follows:

- 1) We introduce decoder Layer contrasting of decoding strategy to alleviate hallucination, which use the difference of logits between intermediate layer and output layer as the probability distribution of the next token.
- 2) We propose a penalty the jitter of decoder layer method, it penalizes tokens with substantial logit variability, designed to curb the emergence of hallucination associated with fluctuating logits.

2 Related Works

Hallucination in LLMs manifest as generated texts that lack adherence to the original intent, known as faithfulness, or deviate from factual accuracy, termed factualness. In most text generation tasks, the term hallucination specifically pertains to issues with factualness. A primary cause of these hallucinations is the subpar quality of during training stage. Lee et al. [10] used a fact-enhanced sampling algorithm during the model training and introduced a Topicprefix to improve the perception of facts during the fine-tuning, reducing the named entity error from 33.3% to 14.5%. Touvron et al. [8] used RLHF to align with human preferences, collecting human preference data to train a reward model, and

employing the PPO algorithm for reinforcement learning on top of the supervised fine-tuning model, making Llama2 a powerful performance display that has become the current mainstream open-source LLMs training base. Cao et al. [11] proposed a novel method to automatically evaluate high-quality data, which greatly alleviate the hallucination problems brought by the source data, becoming the SOTA of Judge-as-LLM at that time. Gabriel et al. [12] proved that supervised fine-tuning training can only solve task-specific hallucination problems, which will inevitably lead to more severe hallucination problems in other individual tasks. Therefore, researchers have also tried to alleviate hallucination from the inference phase of LLMs.

In addition, some researchers exploring way to mitigate the hallucinations of LLMs from the inference stage. Shi et al. [13] proposed a new decoding method that amplified the difference between LLMs’ output logits with and without context, which also inspired the innovative work of this article. Li et al. [14] identified a group of attention heads with high linear probing accuracy through inference-time intervention, moving activations along these directions related to the ground truth. It is worth noting that Li et al. also mentioned that there could be significant differences in the information dimensions between the middle layers and the output layer. Dhuliawala et al. [15] used the COT method to let LLMs automatically verify the reliability of answers and make corrections, experiments showed that this method reduced hallucination by 21% on benchmarks. There has also been some related work in assisting LLMs to mitigate hallucination with external knowledge. Peng et al. [16] used external knowledge to automatically generate feedback to assist in modifying the original answer. Other works, such as intelligent agents [17], chain-of-thought reasoning [18], and retrieval-enhanced generation [19], have all proven that leveraging external knowledge can effectively alleviate the hallucination of LLM.

3 Method

The Llama2, which is a state-of-the-art LLM, generally consist of an input embedding layer N , decoder layers $D = \{D_1, D_2, \dots, D_N\}$, and a mapping layer $\varphi(\cdot)$. This structure aims to capture worldly knowledge by predicting the output probability of the next token. Given an input sequence $x = \{x_1, x_2, \dots, x_n\}$, the objective is to maximize the joint probability distribution $p(x_1, x_2, \dots, x_n)$. The sequence x is project to a vector of fixed dimensionality $E_0 = \{e_1^0, e_2^0, \dots, e_n^0\}$ via the embedding layer. LLM typically define the final output probability distribution in the following:

$$p(x_t|x_{<t}) = \text{softmax}(D_N(\varphi(e_t^N))), x_t \in \Psi \quad (1)$$

where Ψ represents the vocabulary set.

Inspired by prior study on early exiting techniques and the context-aware decoding approach presented by Shi et al. [13], we introduce a new decoding strategy that contrasting the logits distribution between an intermediate decoder layer D_j and the output layer D_N to replace the customary method in Eq. 1.

Firstly, we define the output probability $p(x_t|x_{<t})$ of the next token for the j^{th} decoding layer D_j as follow:

$$p(x_t|x_{<t}) = softmax\left(D_j\left(\varphi\left(e_t^j\right)\right)\right), x_t \in \Psi \tag{2}$$

where e_t^j denotes the t-th token of the j-th layer.

Then, we names the intermediate layer as the premature layer and the final layer as the mature layer, and now redefine the formula for calculating the probability of the next token is defined as follows:

$$p(x_t|x_{<t}) = softmax\left(\mathbb{F}\left(p_M(x_t), p_N(x_t)\right)\right) \tag{3}$$

$$M = argmax_{j \in N-1} (d(p_m(\cdot), p_j(\cdot))) \tag{4}$$

here, \mathbb{F} denotes the function that contrasting measuring the discrepancy between the output probability and a premature layer, and M denotes the layer that exhibits the most difference divergence from the refined prediction layer, and N stands for the output layer. The d is used to assess the divergence between the premature layer and the mature layer, which aids in identifying the optimal premature layer that amplifies the distinction between the layers. The overall flowchart of the algorithm is presented in Fig. 1. A thorough discussion of these concepts is presented in Sects. 3.1 and 3.2.

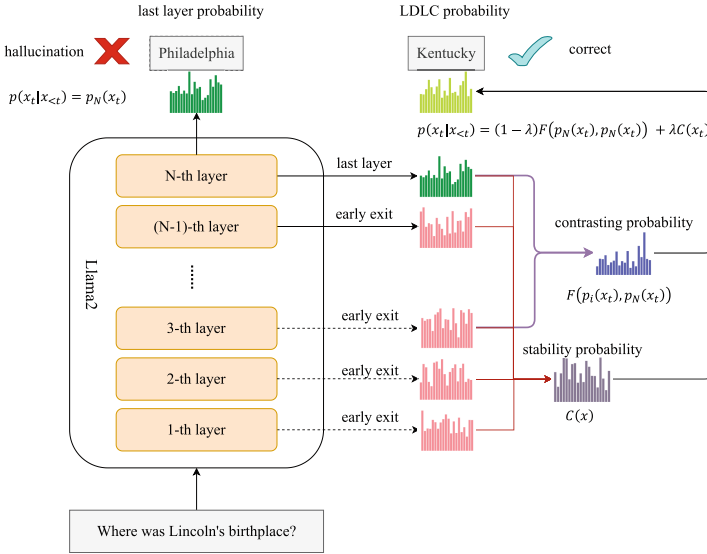


Fig. 1. illustrates the LDLC to dynamically choosing premature layers, as well as the strategy for leveraging the stability of inter-layer probability distributions to address hallucinations. It is observed that the probability associated with certain tokens diminish as they progress through deeper transformer layers, while others increase or exhibit variability. The LDLC effectively utilizes these variations in the probability distributions across decoding layers to tackle the hallucinations in LLMs.

3.1 Dynamically Selecting Lower-Level Layers

It is most importance to obtain the optimal premature layer and contrast with the output layer to obtain the difference, and exploit it as the prediction probability. We first define a set of candidate decoder layers $D_C = D_2, D_4, \dots, D_z$, which is the even-numbered layers within the LLM, and z indicating the total layer of the LLM. Subsequently, by analyzing the entropy of the output probability from the set of candidate layers and by establishing a suitable threshold δ , we determine the effective set of premature layers, named as $D_{z'}$.

$$D_{z'} = \left\{ j \in D_C \mid \hat{p}(x|x_{<t})_j > \delta \right\} \quad (5)$$

The goal is to maximize the differentiation between the premature layer and the mature layer, ensuring that the choice of the premature decoder layer is dynamically tailored to each individual instance.

$$d(p_N(x_{<t}), p_i(x_{<t})) = \max(p_N(x_{<t}) \parallel p_i(x_{<t})), \quad i \in D_C \quad (6)$$

where p_i denotes the output probability distribution of the i -th layer.

Subsequently, by employing the Wasserstein distance, a prevalent metric for calculating discrepancies, we can quantify the variations in distribution between each premature layer and the mature layer. The Wasserstein distance excels in its ability to effectively gauge the distance between two distributions without substantial overlap. By determining the Wasserstein distance between each layer and the mature layer and identifying the maximal value, we pinpoint the layer that exhibits the most significant deviation. The mathematical formula for the Wasserstein distance is as follows:

$$d(p_i, p_N) = W[p_i, p_N] = \inf_{\gamma \in \Pi[p_i, p_N]} \int \int \gamma(x, y) d(x, y) dx dy \quad (7)$$

Here, p_i and p_N denote two distinct probability distributions, while γ refer to the joint distributions of p_i and p_N . In conclusion, within the set of premature layers, we select the premature layer that exhibits the most pronounced divergence from the mature layer:

$$M = \operatorname{argmax} EMD(p_m(\cdot), p_j(\cdot)) \quad (8)$$

where Wasserstein distance also known as the EMD.

The determined layer M is chosen as the optimal premature layer, and we compute the difference in logits between M and the mature layer N . This difference is then utilized as the ultimate probability output for the next token in the model. The precise methodology for this calculation is illustrated as following:

$$\begin{cases} p(x_t|x_{<t}) = \operatorname{softmax}(\mathbb{F}(p_M(x_t), p_N(x_t))), \\ \mathbb{F}(p_M(x_t), p_N(x_t)) = \log \frac{p_N(y|x_{<t})}{p_M(y|x_{<t})} \end{cases} \quad (9)$$

To ensure the advantage of the mature layer in the final outcome and to reduce the influence of the premature layer, we calculate the difference by subtracting the output probability of the premature layer from that of the mature layer, denoted as $p(x_t)$.

3.2 Inter-layer Distributional Stability

The dynamic selection of premature layers offers a partial solution to the problem of LLM generating unreliable predictions. However, this approach does not adequately consider the fluctuations in the output probabilities of inter-layers. It depends exclusively on the logit differences that exhibit the greatest confidence, which it assumes to represent the true probability distribution. When a token from the vocabulary exhibits significant variability in its output probability at the premature layer, it remains an unreliable prediction, which we term a hesitant token, even if its logit is select from the mature layer or contrast with decoder layers.

To address the potential issues arising from hesitant tokens, we introduce a stability factor for the decoder layers. The factor assesses the stability of each token by examining the monotonicity of its confidence scores across the sequence in both forward and reverse order. A sequence of scores is deemed to be monotonically increasing, thus demonstrating considerable stability, if each element in the sequence is at least as large as the one before it. Aligned with this definition, we evaluate the fluctuations in confidence scores for tokens over successive transformer layers, hypothesizing that the confidence of a genuine token should display a steady increase as it progresses through the layers. Based on this step-wise analysis, we derive a stability score $S(x)$ for each token in the vocabulary.

$$S(x) = \frac{n_C}{N} - \frac{n_D}{N} \quad (10)$$

Here, n_C denotes the count of pairs that exhibit a growing, n_D the count of pairs that show a decrease, and N the total count of pairs considered. The stability score $S(x)$ thus obtained lies within the interval $[-1, 1]$. Subsequently, $S(x)$ undergoes a non-linear transformation via the Logistic function to be normalized within the interval $(0, 1)$. The formula for the decoding layer stability factor is delineated as follows:

$$C(x) = \frac{1}{1 + e^{-\beta(S(x) - \alpha)}} \quad (11)$$

Therefore, we consider the following resultant distribution as the actual prediction for the next token:

$$p(x_t | x_{<t}) = \textit{softmax} \left((1 - \lambda) \log \frac{p_N(y | x_{<t})}{p_M(y | x_{<t})} + \lambda C(x_t) \right) \quad (12)$$

4 Experiments

4.1 Dataset

LLMs are encountering issues with hallucination during logical reasoning tasks and complex question-answering tasks. To address this, we utilized the Strat-

egyQA dataset, which is popular among researchers for eliciting creative and varied yes/no questions necessitating implicit reasoning steps. For evaluating mathematical logic reasoning, we selected the test subset from GSM8k, which assesses LLMs’ ability to reason through different types of mathematical problems. It is essential that LLMs correctly perform each reasoning step and arrive at an accurate final answer to be deemed fully correct. Additionally, we employed the wiki factor as a means to test the factual assessment capabilities of the LDLC.

4.2 Baseline

We evaluate LDLC against several decoder strategies which is the most prevalent baselines in the field: 1) Random Sampling. 2) Constrained Decoding. 3) FECS: employs a context-based regularization factor to improve the contrasting search mechanism. FECS encourages the generation of tokens that are semantically akin to the input while discouraging redundant phrases in the output text.

4.3 Main Results

Firstly, we investigate the impact of the layer stability factor λ of mitigating hallucination, by set values of 0, 0.05, 0.1, 0.2, 0.3, and 0.4 separately on the StrategyQA dataset. As illustrated in Fig. 2, it was observed that λ value takes 0.1, which optimally mitigates the hallucination exhibited by LLMs.

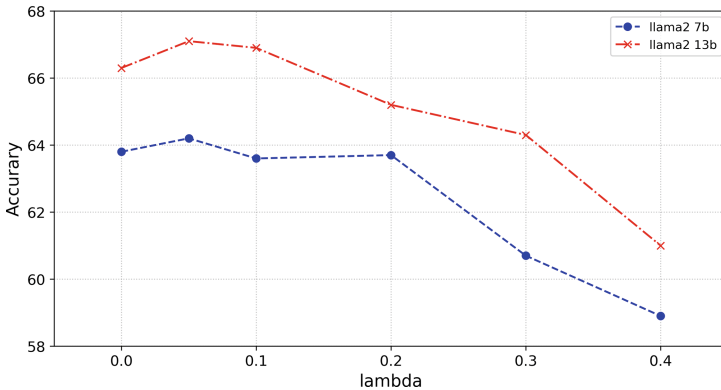


Fig. 2. The change in accuracy of StrategyQA varies when different values are taken.

Table 1 shows the outstanding results of LDLC on the StrategyQA, GSM8k, and Wiki Factor. The evaluation was conducted on models of different sizes, namely Llama 7b and Llama 13b, focusing on the precision on the datasets. The results indicates that our approach is especially effective on models with a smaller number of parameters, achieving a 7% improvement on GSM8k, compared to a 3% enhancement on the 13b model. This suggests that as the parameter count

Table 1. LDLC’s performance on TruthfulQA and GSM8k.

	StrategyQA	GSM8k	Wiki Factor
Llama2 7b	64.2	10.2	62.1
Llama2 13b	67.1	17.3	66.8

of LLMs increases, the incidence of hallucination tends to decrease, and their innate ability to self-correct appears to improve.

We delved into the comparative performance of LDLC against standard methods on the StrategyQA and GSM8k. Our findings reveal that LDLC consistently outperforms the alternatives on both the Llama 7b and Llama 13b, even surpassing the random sampling approach by 7%, thus underscoring LDLC’s effectiveness in reducing hallucinatory responses from LLMs during inference. We assessed the integration of LDLC with Greedy Decoding and Sample Decoding, finding that LDLC particularly complements Sample Decoding, slightly enhancing performance across all datasets. LDLC also achieved approximately a 4% improvement over traditional Sample Decoding on StrategyQA and Wiki Factor. The detailed experimental outcomes are presented in Table 2.

Table 2. Performance of Different Decoding Algorithms on Different Model Scale Sizes.

Method	StrategyQA(%)		GSM8k(%)		Wiki Factor(%)	
	Llama2 7b	Llama2 13b	Llama2 7b	Llama2 13b	Llama2 7b	Llama2 13b
Random Sample	61.1	66.6	10.6	16.7	58.8	62.5
Constrained Decoding	60.7	62.7	9.8	9.8	58.8	64.4
FECS	61.6	65.2	10.1	15.3	60.2	65.1
LDLC(+sample)	63.1	67.1	10.2	17.3	62.1	66.8
LDLC(+greedy)	64.2	66.9	10.2	17.2	61.6	64.2

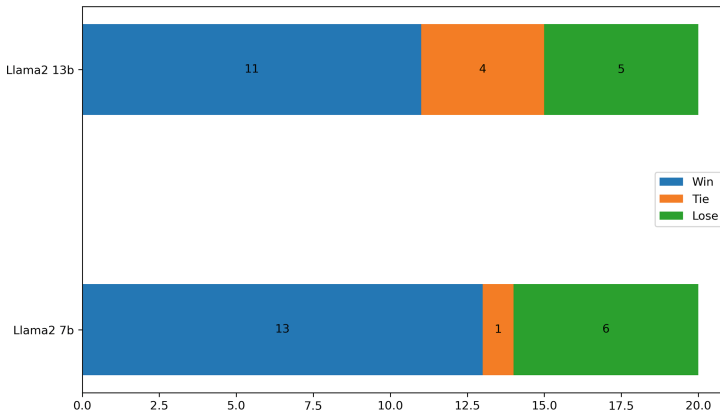
To substantiate the importance of the dynamic selection algorithm within LDLC, we employed a random layer sampling strategy from the decoding phase as our baseline for comparison. In particular, we select a layer at random from the set of potential layers and calculate the actual output logits by determining the discrepancy between the probability outputted by this randomly chosen layer and those from mature layer. For this purpose, we extracted a sample of 200 data points from the GSM8k test set and conducted an evaluation over 20 matches using the Elo rating system. The LLMs’ responses were graded using GPT-4, with victories labeled as ‘win’, ‘Tie’, and ‘lose’. LDLC exhibits a definitive superiority when compared to the baseline. The experimental findings are depicted in Fig. 3.

To corroborate the essential roles of the dynamic decoding layer selection algorithm and the stability confidence of decoder layers, we performed an ablation study using the Llama 7b model on the StrategyQA dataset, as depicted in

Table 3. The ablative experiment of LDLC on StrategyQA.

Method	StrategyQA
LDLC	64.2
(w/o) LC	61.4
(w/o) LSC	63.8
(w/o) LC+LSC	61.1

Table 3. The study revealed that solely employing stability confidence of decoder layers is insufficient for significantly reducing hallucination in LLMs, yielding results comparable to the baseline. LDLC’s effectiveness predominantly stems from the dynamic contrast of decoder layers, which bolsters the reliability of their outputs.

**Fig. 3.** Evaluating the Winning Opportunities of LDLC Based on GPT4.

5 Conclusion

In this work, we present a novel approach for reducing the instances of hallucination in LLMs at the inference stage. We dynamically select the discrepancy between premature layers and the output layer to contrasting, and obtain the final probability distribution of the token. Additionally, we introduce a stability factor for the decoding layer, designed to penalize tokens that contribute to hallucination in LLMs. Experimental evaluations across three widely-used datasets for hallucination detection underscore the effectiveness of the LDLC approach. It significantly support the model’s grasp of factual knowledge and mitigates the influence of the decoder-only model’s training objective, which is to predict the probability distribution of the next token. Our findings suggest that

LDLC holds considerable promise for tasks involving inference and fact-based question answering. Looking ahead, we aim to investigate how LDLC can strike an optimal balance between the diversity and factual accuracy of the content it generates.

References

1. Chang, Y., et al.: A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.* (2023)
2. Zhang, Y., et al.: Siren’s song in the AI ocean: a survey on hallucination in large language models. arXiv preprint [arXiv:2309.01219](https://arxiv.org/abs/2309.01219) (2023)
3. Li, S., et al.: How pre-trained language models capture factual knowledge? A causal-inspired analysis. arXiv preprint [arXiv:2203.16747](https://arxiv.org/abs/2203.16747) (2022)
4. Zhang, Y., Cui, L., Bi, W., Shi, S.: Alleviating hallucinations of large language models through induced hallucinations. arXiv preprint [arXiv:2312.15710](https://arxiv.org/abs/2312.15710) (2023)
5. Lee, K., et al.: Deduplicating training data makes language models better. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 8424–8445 (2022)
6. Touvron, H., et al.: Llama 2: open foundation and fine-tuned chat models. arXiv preprint [arXiv:2307.09288](https://arxiv.org/abs/2307.09288) (2023)
7. Zhou, C., et al.: Lima: less is more for alignment. In: *Advances in Neural Information Processing Systems*, vol. 36 (2024)
8. Schulman, J.: Reinforcement learning from human feedback: progress and challenges. In: *Berkley Electrical Engineering and Computer Sciences* (2023)
9. Chuang, Y.-S., Xie, Y., Luo, H., Kim, Y., Glass, J., He, P.: Dola: decoding by contrasting layers improves factuality in large language models. arXiv preprint [arXiv:2309.03883](https://arxiv.org/abs/2309.03883) (2023)
10. Lee, N., et al.: Factuality enhanced language models for open-ended text generation. In: *Advances in Neural Information Processing Systems*, vol. 35, pp. 34586–34599 (2022)
11. Cao, Y., Kang, Y., Sun, L.: Instruction mining: high-quality instruction data selection for large language models. arXiv preprint [arXiv:2307.06290](https://arxiv.org/abs/2307.06290) (2023)
12. Gabriel, S., Celikyilmaz, A., Jha, R., Choi, Y., Gao, J.: Go figure: a meta evaluation of factuality in summarization. arXiv preprint [arXiv:2010.12834](https://arxiv.org/abs/2010.12834) (2020)
13. Shi, W., Han, X., Lewis, M., Tsvetkov, Y., Zettlemoyer, L., Yih, S.W.: Trusting your evidence: hallucinate less with context-aware decoding. arXiv preprint [arXiv:2305.14739](https://arxiv.org/abs/2305.14739) (2023)
14. Li, K., Patel, O., Viégas, F., Pfister, H., Wattenberg, M.: Inference-time intervention: eliciting truthful answers from a language model. In: *Advances in Neural Information Processing Systems*, vol. 36 (2024)
15. Dhuliawala, S., et al.: Chain-of-verification reduces hallucination in large language models. arXiv preprint [arXiv:2309.11495](https://arxiv.org/abs/2309.11495) (2023)
16. Peng, B., Galley, M., et al.: Check your facts and try again: improving large language models with external knowledge and automated feedback. arXiv preprint [arXiv:2302.12813](https://arxiv.org/abs/2302.12813) (2023)
17. Gou, Z., Shao, Z., et al.: Critic: large language models can self-correct with tool-interactive critiquing. arXiv preprint [arXiv:2305.11738](https://arxiv.org/abs/2305.11738) (2023)
18. Zhao, R., Li, X., et al.: Verify-and-edit: a knowledge-enhanced chain-of-thought framework. arXiv preprint [arXiv:2305.03268](https://arxiv.org/abs/2305.03268) (2023)
19. Feng, C., Zhang, X., Fei, Z.: Knowledge solver: teaching LLMs to search for domain knowledge from knowledge graphs. arXiv preprint [arXiv:2309.03118](https://arxiv.org/abs/2309.03118) (2023)



Robustness of Classifiers for AI-Generated Text Detectors for Copyright and Privacy Protected Society

Akshay Agarwal^{1(✉)} and Mohammed Uzair²

¹ IISER Bhopal, Bhopal, India
akagarwal@iiserb.ac.in

² Methodist College of Engineering and Technology, Hyderabad, India

Abstract. LLMs such as Chat Generative Pre-Trained Transformer (ChatGPT), Pathways Language Models (PaLM), and Bard Artificial Intelligence (Bard AI) can generate human-like text. On top of that, interestingly, these generated texts can correspond to ‘any’ domain of human life such as finance, medicine, and health. Due to the training of these LLMs on a large amount of text corpus, the generated text is hard to be detected merely by reading. Therefore, it creates a havoc of privacy and copyright issues and hence effective detection of the generated texts is critical. In this research, we take a strong step towards effectively detecting generated text that belongs to several domains humans deal with in our day-to-day lives. Interestingly, existing detection methods utilize either LLM models or deep neural networks to detect generated text; one catch here is that since these models also utilize similar backbones, therefore, existing defenses are found non-generalized. We further categorize the difficult examples into multiple categories based on the evaluation settings such as unseen domain, unseen dataset, and modified text, and show the detection performance of different categories. In this research, we utilize the potential of traditional machine learning classifiers to differentiate human text from generated text in a resource-efficient manner. Our extensive investigation reveals surprising yet simple relationships between generated examples from different domains. It is demonstrated that paraphrasing can degrade the performance of LLM-based AI-generated text detection algorithms drastically as compared to the traditional classifiers. Further, the fusion of multiple text encoders and machine learning classifiers can boost the performance of a single model.

Keywords: ChatGPT · AI-Generated Text Detection · Deepfake Text

This work was partially done when Uzair was the intern at AI-Shala.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15320, pp. 55–71, 2025.
https://doi.org/10.1007/978-3-031-78498-9_5

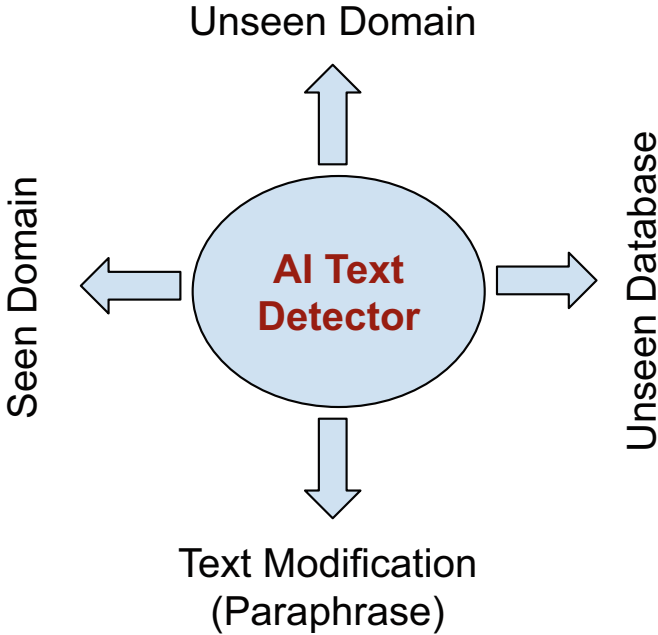


Fig. 1. Effective paradigms for real-world AI-generated text detectors

1 Introduction

In recent times, large language models such as ChatGPT have been a significant breakthrough in the field of natural language processing (NLP) and artificial intelligence (AI). ChatGPT has shown a tremendous performance and hence has caught tremendous attention that just within a few days of its launch is put to the test by millions of users worldwide [6,7]. Since then we have witnessed the development of many other large language models capable of generating and understanding human-like text. These models have demonstrated exceptional proficiency across a wide range of language-related tasks, including machine translation, text summarization, question-answering, and sentiment analysis. Despite their exceptional capabilities, these LLMs face challenges and ethical considerations which include the generation of false, biased, and offensive content. Due to this, it is also observed that several companies, research publishing venues, and institutes have banned or limited the use of LLM models [11]. On top of that, the generated data poses significant copyright issues, the prime reason might be the unethical use of the text available freely on social media platforms.

Further, as people’s interest in various AI chatbots has been increasing, there has also been a growing interest and use of ChatGPT by hackers. These ChatGPT-themed lures are used to spread malware across Meta, Instagram, and WhatsApp. Meta has observed the increase of malware related to ChatGPT

[35,41] which poses a serious security risk to our social-media data. The company said that since March 2023, its security teams have uncovered 10 malware families using ChatGPT (and similar themes) to deliver malicious software to users' devices. While the ChatGPT sometimes might produce the wrong output, the results can be very similar to as written by humans. It makes the problem interesting and has resulted in the development of several machine-learning techniques that can differentiate between human text and ChatGPT or any AI-generated text.

It is showcased that the detection of AI-generated text is hard and is going to be further complicated due to the advancement of newer LLMs. However, still, several recent works claimed the detection of AI-generated text with high accuracy, but show poor generalizability as soon as the unseen domain or LLMs come for evaluation [12,27,31,39] Therefore, we believe, that before claiming the effective and accurate detection of generative text, the detectors must be evaluated in several real-world settings. Figure 1 shows a few evaluation settings that are extremely relevant before releasing the AI-generated text detectors in the real world. The prime reasons for such extensive evaluation depend on two forms of errors: (i) False Positive: where the human written text can be wrongly labeled as AI-generated text. This scenario can decrease the trust in human writing capabilities and can be dangerous as can wrongly punish the genuine writers and (ii) False Negative: where the AI-generated text can be classified as written by humans. This scenario can increase copyright and other related issues. Therefore, in this research, for the first time, we have presented a comprehensive and benchmark study for the detection of AI-generated text based on settings depicted in Fig. 1. Further, it is seen that the traditional machine learning classifiers can outperform the deep learning classifier where the training dataset is limited or the adversary (AI-generated) text is generated by the deep neural network itself [3,4,18,29]. Therefore, in this research, we have evaluated more than 25 traditional machine learning classifiers including support vector machine (SVM) and tree-based models. In brief, the contributions of this research are:

- A comprehensive evaluation of traditional machine learning classifiers for the detection of AI-generated text;
- Robustness analysis of these detectors in several real-world settings reveals interesting findings such as which domain human and AI text can be used to detect other domains;
- A novel computationally efficient AI-generated text detection algorithm has been proposed.

Figure 1 shows several real-world evaluation settings that we feel must be taken care of before developing effective AI text detectors. As mentioned above, the current LLMs can generate 'any' domain of human life such as medicine and finance. Therefore, in a traditional setting, the detectors must be effective for each domain on which they are trained and going to be tested (namely seen domain setting). Due to the limited language corpus, it might be impossible to train the detector for each domain; therefore, the detectors for one domain must be able to detect generated text for another domain (namely unseen domains

such as finance vs. medicine). Other possible evaders of the detector are that while the generated correspond to the same domain (finance vs. finance) but are generated by different individuals; therefore, based on writing or prompting style, the generated text might have a different distribution. Hence, the trained detectors must be robust to handle distribution shifts (namely unseen datasets). One such example of distribution shift and evader is paraphrasing [25]; therefore, the developed generated text detectors must be evaluated in this simple but intuitive and intelligent adversarial setting. To the best of our knowledge, for the first time, we have benchmarked the AI-generated text detectors against such wide evaluation settings and provided several findings that can help in developing effective detectors in the future.

2 Literature Survey

The problem of AI-generated text detection can be formulated as binary classification and the literature has seen a tremendous amount of work carried out in this direction [5, 9, 14, 23, 33]. At a global level, these existing binary classification algorithms can be grouped into two categories: (i) feature extraction and classification techniques and (ii) fine-tuning or training deep neural networks.

It is observed that human text follows Zipf’s law [46]: where the frequency of a word is inversely proportional to its rank. Interestingly, AI-generated text does not follow this distribution property and can be a cue to detect generated text [21]. Another useful cue based on the repetitiveness of the words, the overlap in n-gram text can be used to detect the AI-generated text [15, 16]. Another feature to identify the generated text is its readability and coherency [21, 40]. One way to measure coherency is the use of phrasal words in the sentences [34]. Apart from that simple text features such as length of sentences, use of idioms, and punctuation marks leave a watermark that can be effectively used for generated text detection [15, 34]. Other than these text feature-based binary classifiers, several research works have been done where deep networks are trained for binary class classification.

For example, in one of the earliest studies, Open AI fine-tuned the RoBERTa-based GPT-2 detector to distinguish the generated text from the human written text [30]. However, as mentioned earlier, the major limitation of the approach is its generalizability and ineffectiveness in handling unseen LLMs and hence requires fine-tuning on each LLMs text data to differentiate text generated vs. written by the humans [10]. Another limitation of neural network-based detection approaches is their vulnerability against adversarial texts which can be of any form such as perturbed text samples, out-of-distribution due to paraphrasing, or unseen domains [20, 39]. To improve the generalizability or improve the computational efficiency of the detection approaches, several zero-shot approaches are also proposed [17, 22]. These approaches learn the threshold based on the log score of the tokens [33]. If we assume that the LLM model leaves some kind of watermark, then this can be used to detect whether the text is written by a human or an AI algorithm [24, 45]. However, since, currently it is not feasible to have a similar form of watermarking across the LLMs, hence, the above

approach is found less effective in detecting unseen LLMs generated text. We refer the reader to the survey papers which present a detailed survey of the existing techniques of text generation and detection for better understanding [13, 43]. Overall, through the literature, it is observed that the current generative text detector models fall short in detecting the AI-generated text due to their improper evaluation and therefore we believe they provide the “false sense of security” and fail drastically as soon as the unseen LLM model or domain comes for evaluation [8, 20]. Therefore, we believe, it is utterly important to first understand the strength and limitations of machine learning classifiers for detecting generative texts. The proposed research is the first step in this direction by providing a comprehensive study by evaluating more than 25 machine learning classifiers for AI-generated text detection.

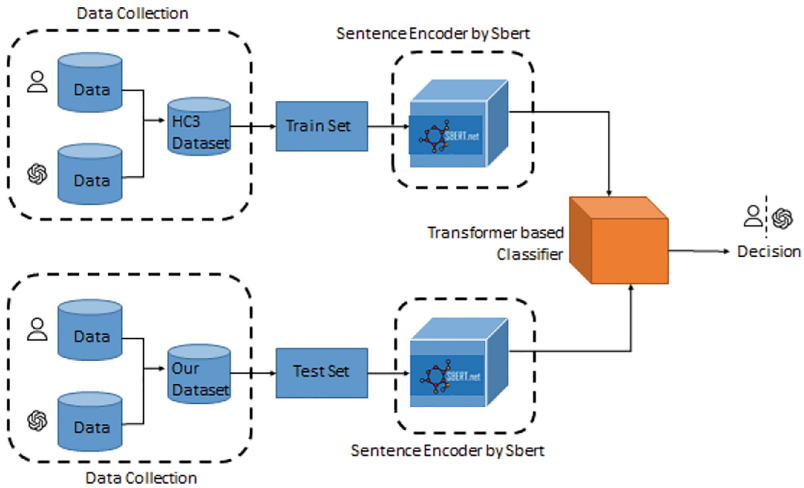


Fig. 2. Schematic diagram of the proposed AI-generated text detection framework

3 AI-Generated Text Detectors Utilizing Traditional Classifiers

The literature shows that AI-generated text detectors based on feature extraction coupled with machine learning classifiers show promising results. However, the existing feature extraction does not effectively encode the text. The feature extractors considered so far: (i) frequency-based features such as ranking of words and term frequency-inverse document frequency (TF-IDF), (ii) coherency features, (iii) linguistic features such as part of speech and named entity tags, and (iv) others such as the use of phrases, punctuation marks, and length of sentences. Therefore, in this research, we also looked at the problem of AI-generated text detection from the perspective of the utilization of effective and

state-of-the-art sentence encoders. Therefore, we have used three recent and state-of-the-art text encoders namely all-MiniLM-L6-v2, all-mpnet-base-v2, and multi-qa-mpnet-base-dot-v1. The all-MiniLM-L6-v2 model maps the input sentences to a 384 dimensional encoding vector. The model has shown its effectiveness in solving several interesting tasks such as clustering and semantic search. Since it is believed that the sentiments of human-written text and AI-generated text might have a different sentiment quotient; therefore, the use of this encoder is intuitive. Similar to the above model, all-mpnet-base-v2 is also effective for semantic search; however, it encodes the sentences and paragraphs to 768 dimensional encoding vector. Semantic search is based on the understanding of the text content and it is also effective even in the case of finding the synonyms. This property can also help make the proposed AI-generated text detector where a simple adversarial attack can be performed by replacing the words with their synonyms. The multi-qa-mpnet-base-dot-v1 also maps the input text to a fixed 768 dimensional feature vector. This model is pre-trained on 215M question-answer pairs from diverse sources. We have used the sentence BERT [38] library available at the hugging face [1] platform to encode the sentences using these three different encoders.

Table 1. Characteristics of the AI-generated and human-written text datasets used in this research

Dataset		HC3					TruthfulQA	SQuAD	GPT-2	Ours
Samples	Train	4719	1496	1424	20260	1010				-
	Test	3147	998	950	13508	674	817	1000	5000	2700
AI Model		GPT3.5					GPT-3		GPT-2	GPT-3 Davinci
Domain		Finance	Medicine	Others	Reddit	Wikipedia	38 Domains (Health, Law, Fiction, etc.)	Wikipedia	Reddit	Politics, Environment, AI&ML

Once both the human and AI-generated sentences are encoded, they are used as the input to several popular and benchmarking machine learning classifiers. The classifiers trained using scikit-learn [36] are: ✓ LGBMClassifier (LGBM), ✓ NearestCentroid, ✓ RidgeClassifierCV, ✓ CalibratedClassifierCV, ✓ LinearDiscriminantAnalysis, ✓ RidgeClassifier, ✓ LinearSVC, ✓ AdaBoostClassifier, ✓ LogisticRegression (LR), ✓ BernoulliNB, ✓ NuSVC, ✓ KNeighborsClassifier, ✓ BaggingClassifier, ✓ SVC, ✓ RandomForestClassifier, ✓ PassiveAggressiveClassifier, ✓ DecisionTreeClassifier, ✓ ExtraTreesClassifier, ✓ SGDClassifier, ✓ QuadraticDiscriminantAnalysis, ✓ GaussianNB, ✓ Perceptron, ✓ ExtraTreeClassifier, ✓ LabelSpreading, and ✓ LabelPropagation. These used classifiers can also be grouped into two broad categories: (i) single-stage (such as KNN, Logistic Regression) classifiers and (ii) ensemble-based learners (such as Bagging and Extra Trees).

Figure 2 shows the overall architecture of the proposed AI-generated text detection system. Both the human and AI-generated text are first encoded

and later used to train the supervised binary classification network. Later, the trained networks are used for evaluation on the testing set and the performance is reported in terms of detection accuracy.

4 Experimental Results and Analysis

To effectively understand the robustness and effectiveness of AI-generated text detectors, the detector must be evaluated on a vast variety of datasets covering the settings shown in Fig. 1. Henceforth, in this research, we have used several benchmark AI-generated datasets. An interesting fact about the used datasets is that they cover a wide domain of human life such as politics and law. Along with that, we have evaluated several popular and benchmarking supervised machine learning (ML) classifiers and multiple text encoding algorithms. First, we will briefly describe the characteristics of the datasets used followed by the experimental results and analysis reflecting the effectiveness and robustness of the AI-generated text detectors.

Datasets. Guo et al. [19] has developed one of the largest AI-generated and human-written text datasets namely Human ChatGPT Comparison Corpus (HC3). The dataset contains the questions corresponding to multiple domains or captured from multiple social media platforms such as Reddit and Wikipedia and the questions correspond to various critical domains such as finance and medicine. For each domain/platform, human answers are collected using multiple existing datasets, and AI-generated answers are generated using GPT3.5. In this research, we have used the English corpus of the HC3 dataset. We believe due to the vast variation in the dataset, it is one of the ideal candidates to evaluate the generated text detectors. In total, we have used the 48k+ text samples to understand the effectiveness and robustness of the classifiers. To evaluate the truthfulness of the LLMs in generating the answers to questions, Lin et al. [28] have developed a truthful dataset. Compared to the HC3 dataset, the truthful dataset contains questions spanning 38 domains including health, law, and politics. It makes the dataset effective in understanding the generalizability of AI-generated text detectors under unseen domain training testing settings. As each of the domains is critical, false/fake information in any domain can be dangerous; therefore, an ideal detection model can not be biased toward one particular domain. Stanford Question Answering Dataset (SQuAD) [37] addresses the need for a large-scale reading comprehension dataset which is collected from Wikipedia. In contrast to the previous datasets, here the answers in the dataset are either segments of text or spans of a reading passage. The GPT-2 Output dataset [2] is a collection of outputs generated by the GPT-2 language model as compared to the HC3 and the truthful which utilized the advanced LLMs. The dataset contains the text information from the WebText dataset and the GPT-2 model trained on the WebText dataset is used to generate samples. Apart from these benchmark text datasets challenging to detect the AI-generated text, *we have also developed a novel dataset comprising domains not explored so far in*

the existing datasets such as AI and ML. AI/ML which has shown tremendous success in solving several real-world tasks corresponding to vision, biometrics, and NLP is also not untouched by fake/false information generated from itself. Our dataset comprises four main categories: politics, environment, AI, and ML. For the Human-generated texts, we have used Reddit, StackOverflow, and some common essays from the Internet, which contain genuine user interactions, offering real-world perspectives and discussions from different individuals across the globe. To generate the responses from the AI model, we have used ChatGPT’s Davinci model. The characteristics of each of the datasets used in this research are given in Table 1. Overall, in this research, we have used 57, 703 text samples corresponding to human and AI-generated categories.

4.1 Results and Analysis

As shown in Fig. 1, in this research, we have designed the experimental protocols along those four critical and real-world evaluation settings. For training the AI-generated text detection models, only the HC3 subsets have been used due to their large-scale nature and covering wide domains. We have divided the subsets into two parts: 60% of the random split has been used for training and the remaining 40% has been used for the evaluation. The remaining datasets are used for unseen dataset testing only to evaluate the generalizability and robustness of the detector.

Traditional Seen Setting. In the first and traditional setting, we trained and tested the AI-generated text detectors for the seen evaluation setting. For example, if the detectors are trained on the finance subset of HC3, they are evaluated on the testing subset of the finance subset of HC3. The results of this setting are reported in Table 2. The results of this setting can be broadly divided into three categories: (i) effectiveness of the classifier, (ii) effectiveness of the text encoding algorithm, and (iii) performance on the individual subset of the HC3 dataset. In terms of the effectiveness of the classifiers, it is seen that the RidgeClassifier including traditional (RidgeClassifier) and utilizing cross-validation (RidgeClassifierCV) performs the best across each domain in comparison to the other classifiers. RidgeClassifiers uses the concept of L_2 regularization and avoids the overfitting of the model to ensure its generalizability against unseen datasets. On top of that, the RidgeClassifierCV further exploits the concept of cross-validation to make the detector more efficient. Therefore, the average performance of RidgeClassifierCV is 91% which is higher than 1% from RidgeClassifier and significantly higher than other classifiers. The logistic regression and Linear support vector machine classifier (SVC) perform comparably to the RidgeClassifierCV. As mentioned above, the encoder-3 (multi-qa-mpnet-base-dot-v) is trained on a wide source of text information which is also visible in its performance in detecting AI-generated text. The classifiers receiving the text encoded using encoder-3 are found highly effective in segregating them into binary classes: human and

Table 2. Human vs. ChatGPT detection when the seen subset has been used for training and testing the machine learning models. F, M, O, R, and W represent the Finance, Medicine, Others, Reddit, and Wiki subset, respectively. -1, -2, and -3 represent the sentence encoders all-MiniLM-L6-v2, all-mpnet-base-v2, and multi-qa-mpnet-base-dot-v1, respectively

Model	F-1	F-2	F-3	M-1	M-2	M-3	O-1	O-2	O-3	R-1	R-2	R-3	W-1	W-2	W-3
RidgeClassifierCV	0.92	0.95	0.98	0.96	0.96	0.99	0.89	0.88	0.91	0.87	0.94	0.96	0.77	0.81	0.89
RidgeClassifier	0.92	0.95	0.98	0.95	0.96	0.98	0.89	0.87	0.88	0.87	0.94	0.96	0.76	0.77	0.83
LDA	0.92	0.95	0.98	0.96	0.96	0.98	0.88	0.87	0.87	0.87	0.94	0.96	0.76	0.74	0.78
LogisticRegression	0.91	0.93	0.97	0.96	0.96	0.98	0.85	0.86	0.90	0.88	0.94	0.96	0.77	0.84	0.88
SVC	0.91	0.93	0.97	0.95	0.93	0.97	0.82	0.80	0.89	0.92	0.94	0.97	0.73	0.74	0.82
LinearSVC	0.91	0.92	0.96	0.94	0.95	0.97	0.82	0.85	0.89	0.87	0.93	0.95	0.75	0.83	0.87
NuSVC	0.90	0.92	0.95	0.94	0.92	0.95	0.82	0.81	0.89	0.89	0.91	0.93	0.74	0.78	0.84
SGDClassifier	0.89	0.93	0.96	0.94	0.94	0.97	0.84	0.84	0.88	0.85	0.93	0.95	0.77	0.84	0.85
PAC	0.88	0.93	0.97	0.95	0.95	0.98	0.84	0.85	0.89	0.83	0.90	0.95	0.76	0.82	0.86
Perceptron	0.87	0.92	0.96	0.95	0.94	0.97	0.84	0.84	0.89	0.82	0.91	0.95	0.75	0.82	0.85
XGBClassifier	0.87	0.88	0.92	0.91	0.88	0.92	0.87	0.77	0.82	0.85	0.86	0.91	0.68	0.68	0.73
LGBMClassifier	0.86	0.87	0.91	0.90	0.87	0.92	0.87	0.77	0.82	0.84	0.86	0.90	0.65	0.65	0.74
GaussianNB	0.84	0.82	0.86	0.87	0.85	0.87	0.78	0.76	0.82	0.80	0.80	0.84	0.65	0.67	0.69
RFC	0.83	0.84	0.87	0.90	0.85	0.88	0.83	0.71	0.77	0.81	0.82	0.86	0.61	0.61	0.68
ExtraTreesClassifier	0.83	0.83	0.88	0.88	0.84	0.89	0.74	0.63	0.74	0.81	0.81	0.86	0.57	0.55	0.63
CalibratedClassifierCV	0.90	0.93	0.97	0.95	0.95	0.98	0.84	0.86	0.91	0.88	0.94	0.96	0.76	0.85	0.87
NearestCentroid	0.83	0.83	0.86	0.87	0.84	0.90	0.77	0.76	0.84	0.79	0.79	0.84	0.68	0.69	0.70
AdaBoostClassifier	0.80	0.83	0.87	0.89	0.83	0.91	0.83	0.77	0.80	0.77	0.79	0.83	0.65	0.68	0.73
BernoulliNB	0.80	0.81	0.85	0.87	0.83	0.88	0.78	0.75	0.82	0.77	0.78	0.82	0.65	0.69	0.70
QDA	0.80	0.78	0.87	0.78	0.50	0.50	0.47	0.48	0.49	0.84	0.83	0.89	0.39	0.49	0.49
BaggingClassifier	0.76	0.77	0.83	0.85	0.79	0.87	0.84	0.67	0.71	0.75	0.77	0.81	0.62	0.61	0.63
DecisionTreeClassifier	0.67	0.69	0.73	0.75	0.71	0.81	0.78	0.61	0.62	0.70	0.70	0.72	0.53	0.59	0.59
KNeighborsClassifier	0.67	0.64	0.78	0.72	0.68	0.79	0.55	0.52	0.59	0.62	0.58	0.68	0.49	0.48	0.50
ExtraTreeClassifier	0.59	0.60	0.67	0.64	0.59	0.72	0.54	0.53	0.54	0.61	0.61	0.66	0.54	0.53	0.58
LabelSpreading	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.56	0.56	0.56	0.54	0.54	0.54
LabelPropagation	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.56	0.56	0.56	0.54	0.54	0.54

AI-generated. It is interesting to note that the effectiveness of this encoder is consistent across classifiers and domains. For example, when the best-performing classifier i.e., RidgeClassifierCV is used, the performance on the encoder-3 is at least 3% better on Finance and Medicine subsets, 2% better on others and Reddit subsets, and 8% better on Wikipedia subset. In terms of domains, it is observed that the text related to the Wikipedia subset of the dataset is highly challenging to detect as compared to the other subsets. The probable reason might be that the Wikipedia texts follow any particular domain question-answer responses as compared to other subsets that explicitly follow one domain such as Finance or Medicine. A similar challenge can be observed on the other subset which also contains the random Wikipedia question-answer texts. **We want to highlight that, this benchmark understanding is missing in the literature and hence it is difficult to build a universal classifier that can detect**

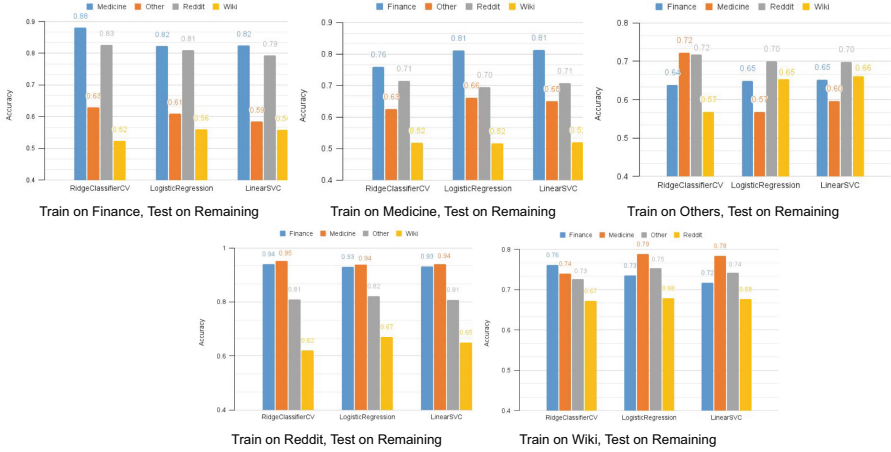


Fig. 3. Human vs. ChatGPT detection when 5-fold unseen subsets cross-validation has been performed. In this case, the classification models are trained on a single subset and tested on remaining individual subsets

AI-generated text across various domains. On top of that, as good initialization in deep neural networks is extremely important [32, 42], utilization of an effective feature extractor (encoder) is critical for the success of ‘any’ traditional machine learning classifier. *It is observed from these experiments that encoders effective in extracting the semantic understanding of the texts can pave the way for an accurate and robust AI-generated text detector.*

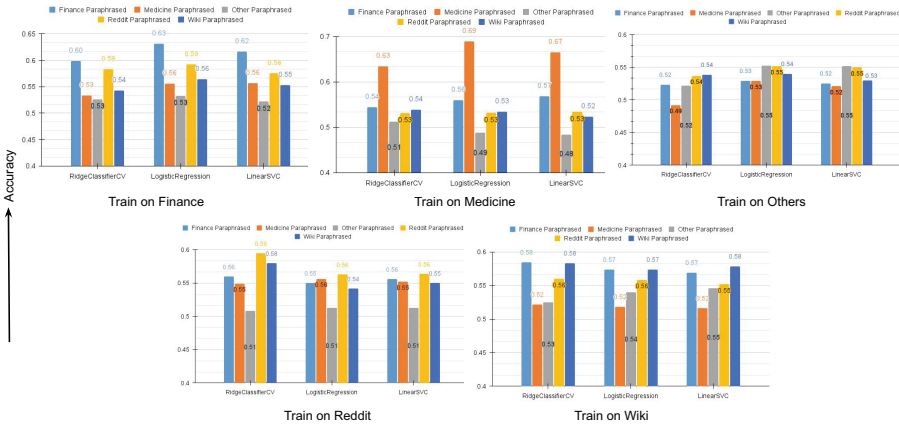


Fig. 4. Human vs. ChatGPT detection when training and testing have been performed on the seen subset of the HC3 dataset; however, now the testing sets are paraphrased using Pegasus [44]

While it is observed that in a few cases, CalibratedClassifierCV shows better performance than Linear SVC and Logistic Regression; the time taken by the CalibratedClassifierCV is 3 and 6 times higher than LinearSVC and RidgeClassifierCV, respectively. Further, it is also empirically observed that the performance of CalibratedClassifierCV is significantly lower than LogisticRegression and LinearSVC in unseen domain settings. Therefore, keeping the trade-off between accuracy and computational time, we have chosen LinearSVC and Logistic Regression for detection.

Unseen Domain Setting. It is seen that the success of text-generated AI tools is not limited to any particular domain and can answer the questions of humans alike. Further, it is highly difficult to train the generated text detector on each domain text date, the reason can be many including computational cost and limited availability of the text corpus. “Therefore, the first critical criterion of an effective AI-generated text detector is its generalizability in handling unseen domain text prompts”. Therefore, to showcase the generalizability of the best-performing text detectors obtained from the first experiment (RidgeClassifierCV, LinearSVC, and Logistic Regression (LR)) are evaluated under unseen domain settings using the HC3 dataset. As seen in the first experiment the RidgeClassifierCV outperforms the other classifiers, it is also showcased in the unseen domain experiments as well. For instance, when the finance subset is used in the training and evaluation has been performed on other subsets, the performance of the RidgeClassifierCV is at least 6% higher on the medicine subset and 2% higher on Reddit and medicine subsets. Except in the case of the Wikipedia subset where the LR and LinearSVC perform the same but higher than RidgeClassifierCV. In terms of the domain, it is seen that the detectors trained on Reddit texts are found highly effective in handling unseen text domains. For example, RidgeClassifierCV trained on the Reddit subset is found 18%, 7%, and 1% effective on finance, medicine, other, and Wikipedia subsets respectively compared to the detector trained on other unseen domains. We want to highlight here that the encoding of texts is performed using the best-performing encoder-3. Figure 3 shows the detailed results obtained using ‘*unseen domain*’ setting.

Evading Detection Using Paraphrase. It is seen that it is possible to watermark the text generated using the LLMs [24] and can be used as an effective defense to detect the generated text; however, one simple solution to evade the defense is text paraphrasing [25]. Therefore, it is imperative to evaluate the AI-generated text detectors which show significant success in detecting the texts corresponding to both seen and unseen domains under this evading mechanism. The results reported in Fig. 4 showcase the drop in the detection performance of each best-performing classifier used in the unseen domain setting. Here the evaluation has been done under both seen and unseen domain settings. Out of 5 domains, in 3 domains (finance, medicine, others) LR outperforms the other classifiers and in two cases (Reddit and Wikipedia), RidgeClassifierCV shows the best performance. For example, on the unseen medicine subset, the average

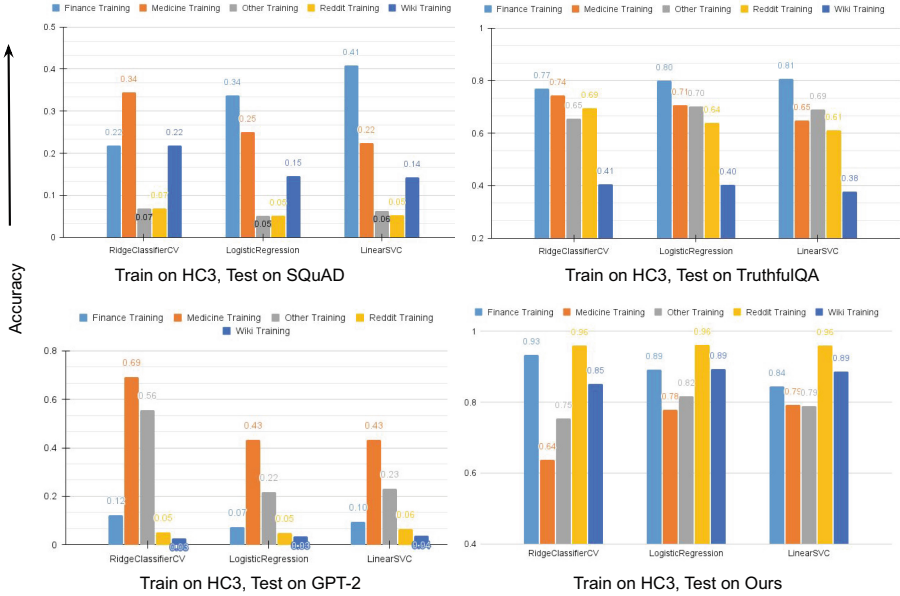


Fig. 5. AI-generated text detection accuracy when classifiers are trained on the individual subsets of the HC3 dataset and tested on unseen datasets namely SQuAD, TruthfulQA, GPT2, and ours

performance of LR, LinearSVC, and RidgeClassifierCV is 54.25%, 53.75%, and 52.25%, respectively. In brief, through our experiments, we have found that the existing ML classifiers can be evaded using a simple paraphrasing technique and the accuracy drop can be as high as 25%.

Unseen Dataset Evaluation. It is well known that security is a game of cat and mouse; once a defense algorithm has been developed a new attack comes into the picture. Secondly, it is extremely necessary for a ‘universal’ defense that is evaluated using several datasets prepared using various researchers having different thoughts/prompts while developing them. However, once the defense is developed re-training or fine-tuning on the newer dataset is computationally costly; therefore, the built system must be resilient to the unseen dataset as well. Here the unseen datasets also contain AI-generated text coming from various seen and unseen domains as well. In this setting, the individual subsets of the HC3 dataset are used for training, and multiple existing along proposed datasets are used in the evaluation. It is found that the SQuAD dataset is one of the toughest datasets to solve using traditional machine learning classifiers. The prime reason might be that the dataset is acquired using old LLM techniques and contains either segments of text or spans of a reading passage which is contrasting to other question-answer datasets. The LinearSVC outperforms the other classifier and achieves the best accuracy when the classifier is

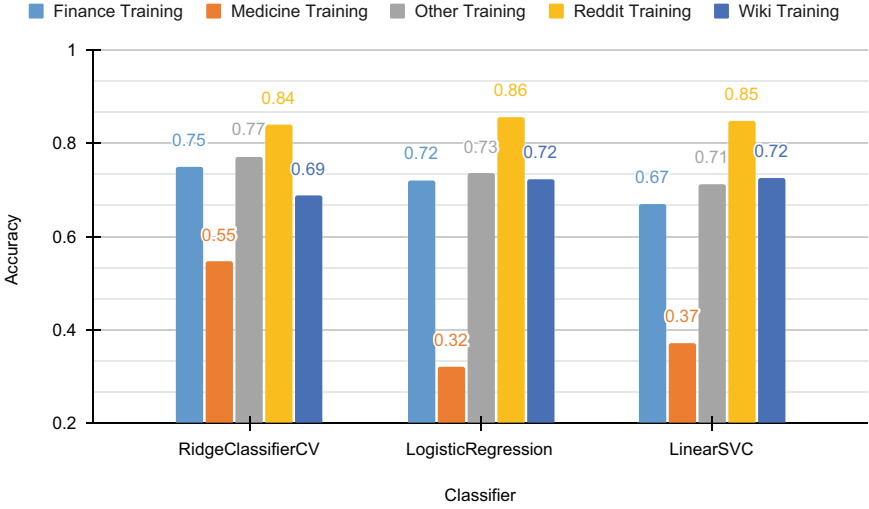


Fig. 6. AI-generated text detection accuracy when classification models are trained on HC3 subsets and tested on **our paraphrased** dataset

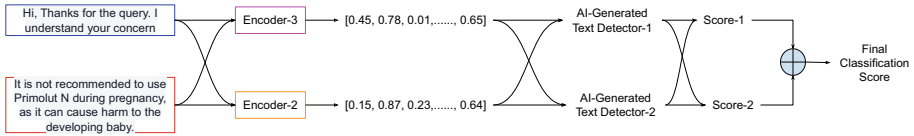


Fig. 7. Proposed AI-generated text detector using an amalgamation of text encoders and classifiers

trained using the finance subset. The GPT-2 dataset is also found challenging and only RidgeClassifierCV shows decent performance. Figure 5 reports the results on unseen datasets.

In comparison to the other unseen datasets, the text in the proposed dataset is easy to detect. Out of five domains in the HC3 dataset, the ‘medicine domain’ is found the least effective in detecting the text of our dataset. *It reflects that while a selection of an ML classifier is essential, the training domain is also critical and must be selected carefully.* While the proposed dataset is found less complex, the paraphrasing attack can degrade the performance of each classifier on our dataset as well. The RidgeClassifierCV which shows 93% detection accuracy when trained using the finance subset, shows a significant reduction of 18% when our dataset is paraphrased using the PEGASUS library. As shown in Fig. 6, a similar reduction in the performance of each classifier trained on any subset has been noticed. However, it is interesting to note here that the paraphrasing shows a limited impact on our dataset as compared to the previously evaluated paraphrased unseen domains of HC3.

Table 3. AI generated text detection using the proposed score fusion algorithm. Avg. represents the average performance of the proposed algorithm. The values in red/blue color show the improvement/decrement from the best baseline performance obtained after training the ML classifiers for seen and unseen domain evaluation

Train	Finance		Medicine		Other		Reddit		Wiki		Ours		Avg.	
	Orig.	Para	Orig.	Para	Orig.	Para	Orig.	Para	Orig.	Para	Orig.	Para	Orig.	Para
Finance	0.98 0.01	0.71 0.06	0.91 0.02	0.64 0.01	0.75 0.06	0.55 0.02	0.87 0.05	0.66 0.01	0.62 0.0	0.59 0.04	0.97 0.0	0.88 0.02	0.85 0.05	0.67 0.06
Medicine	0.83 0.01	0.63 0.06	0.99 0.0	0.78 0.02	0.66 0.0	0.54 0.02	0.73 0.02	0.62 0.03	0.53 0.08	0.57 0.02	0.78 0.10	0.69 0.15	0.75 0.01	0.64 0.01
Other	0.69 0.07	0.60 0.0	0.63 0.20	0.61 0.0	0.93 0.02	0.56 0.02	0.71 0.0	0.59 0.01	0.71 0.04	0.57 0.04	0.88 0.05	0.82 0.02	0.76 0.02	0.63 0.06
Reddit	0.95 0.01	0.59 0.01	0.95 0.0	0.65 0.0	0.87 0.05	0.53 0.01	0.97 0.01	0.63 0.03	0.66 0.02	0.59 0.04	0.98 0.01	0.87 0.01	0.90 0.03	0.64 0.06
Wiki	0.77 0.04	0.63 0.0	0.79 0.06	0.62 0.01	0.79 0.03	0.56 0.0	0.69 0.04	0.61 0.02	0.90 0.02	0.59 0.03	0.90 0.03	0.82 0.04	0.81 0.00	0.64 0.04

4.2 Proposed AI-Generated Text Detection Algorithm

Inspired by the effectiveness of individual text encoding algorithms and AI-generated text detectors, in this research, we have further proposed a novel AI-generated text detection algorithm. As shown in Fig. 7, the algorithm works on the amalgamation of the semantic encoding of texts using two best-performing algorithms which are later used to train two robust machine learning classifiers. The decision/classification probabilities of these classifiers are later fused using a weighted score fusion strategy to achieve the final classification score of a sample. We have used encoders 2 and 3 to extract the text embeddings and, LinearSVC and RidgeClassifierCV to train the AI-generated text detectors. Similar to the previous settings, we have trained the proposed detector using a training set of individual domains of the HC3 and evaluated it on the seen and unseen domain both in its raw and paraphrased format. Results reported in Table 3 showcase that the proposed embedding and score amalgamation improves the classification performance drastically especially when the texts are paraphrased. For example, the average performance shows an improvement of 6% when the finance domain is used for training and all domains are used. Even on the raw text dataset, the proposed algorithm yields 5% better average performance than the performance of best-performing individually trained models.

To further demonstrate the effectiveness and robustness of the proposed approach, we have evaluated it on a recent complex dataset developed for deepfake text detection [26]. The dataset is created using 27 LLMs and we have used the pre-defined train-test for experiments. **When evaluated on the ‘wilder testbed’ (includes unseen domains and paraphrasing attack samples), the performance of the proposed fusion architecture is at least 6% better than complex algorithms such as [26], DetectGPT [33], and GLTR**

[17]. Further, *the prime limitation of existing defense based on deep networks including LLMs is not robust against out-of-distribution samples including paraphrased samples.*

5 Conclusion

The presence of powerful AI-generated models has led to the generation of fake content or content that might not be written by humans. While it is hard to stop the functioning of these open-source models, a defense mechanism that can effectively detect whether the content is developed by humans or generated using an AI tool can be helpful. In literature, several defense works have been proposed; however, the improper evaluation and complexity of these defenses provide a false sense of security. Therefore, in this research, for the first time, we have conducted an extensive evaluation study to find how easy or hard to detect AI-generated text. The findings include the impact and effectiveness of the classifier along with which domain should be used for training and/or which domain is hard to defend. Therefore, future benchmark studies must evaluate their defense mechanism thoroughly under several evaluation settings to reach solid and reliable observations. We have also proposed a novel multi-classifier and multi-encoder AI-generated text detector that showcases higher performance and resiliency in handling unseen domains, datasets, and paraphrasing attacks. In the future, we aim to extend our dataset covering several other domains and social media platforms and provide a certified defense against AI-generated text.

References

1. Sentence transformers (2022). <https://huggingface.co/sentence-transformers>
2. GPT2 output dataset (2023)
3. Agarwal, A., Singh, R., Vatsa, M., Ratha, N.: Image transformation-based defense against adversarial perturbation on deep learning models. *IEEE Trans. Dependable Secure Comput.* **18**(5), 2106–2121 (2020)
4. Agarwal, A., Vatsa, M., Singh, R., Ratha, N.: Parameter agnostic stacked wavelet transformer for detecting singularities. *Inf. Fusion* **95**, 415–425 (2023)
5. AI, O.: GPT-2: 1.5b release (2019). <https://openai.com/research/gpt-2-1-5b-release>
6. Amaro, I., Barra, P., Della Greca, A., Francese, R., Tucci, C.: Believe in artificial intelligence? A user study on the chatgpt’s fake information impact. *IEEE Trans. Comput. Soc. Syst.* 1–10 (2023). <https://doi.org/10.1109/TCSS.2023.3291539>
7. Amaro, I., Della Greca, A., Francese, R., Tortora, G., Tucci, C.: AI unreliable answers: a case study on chatgpt. In: *International Conference on Human-Computer Interaction*, pp. 23–40. Springer (2023)
8. Antoun, W., et al.: Towards a robust detection of language model generated text: is chatgpt that easy to detect? *arXiv preprint arXiv:2306.05871* (2023)
9. Bakhtin, A., Gross, S., Ott, M., Deng, Y., Ranzato, M., Szlam, A.: Real or fake? Learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351* (2019)

10. Bhattacharjee, A., Liu, H.: Fighting fire with fire: can chatGPT detect AI-generated text? arXiv preprint [arXiv:2308.01284](https://arxiv.org/abs/2308.01284) (2023)
11. Brainard, J.: Journals take up arms against AI-written text. *Science* **379**(6634), 740–741 (2023)
12. Cai, S., Cui, W.: Evade chatgpt detectors via a single space. arXiv preprint [arXiv:2307.02599](https://arxiv.org/abs/2307.02599) (2023)
13. Crothers, E., Japkowicz, N., Viktor, H.L.: Machine-generated text: a comprehensive survey of threat models and detection methods. *IEEE Access* (2023)
14. Fagni, T., Falchi, F., Gambini, M., Martella, A., Tesconi, M.: Tweepfake: about detecting deepfake tweets. *PLoS ONE* **16**(5), e0251415 (2021)
15. Fröhling, L., Zubiaga, A.: Feature-based detection of automated language models: tackling GPT-2, GPT-3 and Grover. *PeerJ Comput. Sci.* **7**, e443 (2021)
16. Gallé, M., Rozen, J., Kruszewski, G., Elsahar, H.: Unsupervised and distributional detection of machine-generated text. arXiv preprint [arXiv:2111.02878](https://arxiv.org/abs/2111.02878) (2021)
17. Gehrmann, S., Strobel, H., Rush, A.M.: GLTR: statistical detection and visualization of generated text. *ACL* (2023)
18. Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? *Adv. Neural. Inf. Process. Syst.* **35**, 507–520 (2022)
19. Guo, B., et al.: How close is chatgpt to human experts? Comparison corpus, evaluation, and detection. arXiv preprint [arXiv:2301.07597](https://arxiv.org/abs/2301.07597) (2023)
20. He, X., et al.: MGTBench: benchmarking machine-generated text detection. arXiv preprint [arXiv:2303.14822](https://arxiv.org/abs/2303.14822) (2023)
21. Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y.: The curious case of neural text degeneration. arXiv preprint [arXiv:1904.09751](https://arxiv.org/abs/1904.09751) (2019)
22. Ippolito, D., Duckworth, D., Callison-Burch, C., Eck, D.: Automatic detection of generated text is easiest when humans are fooled. arXiv preprint [arXiv:1911.00650](https://arxiv.org/abs/1911.00650) (2019)
23. Jawahar, G., Abdul-Mageed, M., Lakshmanan, L.V.: Automatic detection of machine generated text: a critical survey. arXiv preprint [arXiv:2011.01314](https://arxiv.org/abs/2011.01314) (2020)
24. Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., Goldstein, T.: A watermark for large language models. arXiv preprint [arXiv:2301.10226](https://arxiv.org/abs/2301.10226) (2023)
25. Krishna, K., Song, Y., Karpinska, M., Wieting, J., Iyyer, M.: Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. arXiv preprint [arXiv:2303.13408](https://arxiv.org/abs/2303.13408) (2023)
26. Li, Y., et al.: Deepfake text detection in the wild. arXiv preprint [arXiv:2305.13242](https://arxiv.org/abs/2305.13242) (2023)
27. Liang, W., Yuksekgonul, M., Mao, Y., Wu, E., Zou, J.: GPT detectors are biased against non-native English writers. arXiv preprint [arXiv:2304.02819](https://arxiv.org/abs/2304.02819) (2023)
28. Lin, S., Hilton, J., Evans, O.: Truthfulqa: measuring how models mimic human falsehoods. arXiv preprint [arXiv:2109.07958](https://arxiv.org/abs/2109.07958) (2021)
29. Liu, J., et al.: Detection based defense against adversarial examples from the steganalysis point of view. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4825–4834 (2019)
30. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach. arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) (2019)
31. Lu, N., Liu, S., He, R., Tang, K.: Large language models can be guided to evade AI-generated text detection. arXiv preprint [arXiv:2305.10847](https://arxiv.org/abs/2305.10847) (2023)
32. Mishkin, D., Matas, J.: All you need is a good init. In: *International Conference on Learning Representations* (2016)

33. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C.D., Finn, C.: DetectGPT: zero-shot machine-generated text detection using probability curvature. ICML (2023)
34. Nguyen-Son, H.Q., Tieu, N.D.T., Nguyen, H.H., Yamagishi, J., Zen, I.E.: Identifying computer-generated text using statistical analysis. In: 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1504–1511 (2017)
35. Paul, K.: Meta says chatgpt-related malware is on the rise (2023). <https://www.reuters.com/technology/meta-says-chatgpt-related-malware-is-rise2023-05-03/>
36. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
37. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. arXiv preprint [arXiv:1606.05250](https://arxiv.org/abs/1606.05250) (2016)
38. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2019). <https://arxiv.org/abs/1908.10084>
39. Sadasivan, V.S., Kumar, A., Balasubramanian, S., Wang, W., Feizi, S.: Can AI-generated text be reliably detected? arXiv preprint [arXiv:2303.11156](https://arxiv.org/abs/2303.11156) (2023)
40. See, A., Pappu, A., Saxena, R., Yerukola, A., Manning, C.D.: Do massively pre-trained language models make better storytellers? arXiv preprint [arXiv:1909.10705](https://arxiv.org/abs/1909.10705) (2019)
41. Shakir, U.: Meta security analysts warn of malicious chatgpt imposters (2023). <https://www.theverge.com/2023/5/3/23709591/meta-chatgpt-malware-business-account-hacking>
42. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: International Conference on Machine Learning, pp. 1139–1147. PMLR (2013)
43. Tang, R., Chuang, Y.N., Hu, X.: The science of detecting LLM-generated texts. arXiv preprint [arXiv:2303.07205](https://arxiv.org/abs/2303.07205) (2023)
44. Zhang, J., Zhao, Y., Saleh, M., Liu, P.: Pegasus: pre-training with extracted gap-sentences for abstractive summarization. In: International Conference on Machine Learning, pp. 11328–11339. PMLR (2020)
45. Zhao, X., Wang, Y.X., Li, L.: Protecting language generation models via invisible watermarking. arXiv preprint [arXiv:2302.03162](https://arxiv.org/abs/2302.03162) (2023)
46. Zipf, G.K.: Human behavior and the principle of least effort: an introduction to human ecology. Ravenio Books (2016)



How Good are LM and LLMs in Bangla Newspaper Article Summarization?

Faria Sultana²(✉), Md. Tahmid Hasan Fuad¹, Md. Fahim¹,
Rahat Rizvi Rahman³, Meheraj Hossain³, M. Ashraful Amin¹,
A. K. M. Mahbubur Rahman¹, and Amin Ahsan Ali¹

¹ Center for Computational and Data Sciences, Independent University,
Dhaka 1229, Bangladesh

{fuad,md.fahim,aminmdashraful,akmmrahman,aminali}@iub.edu.bd

² Khulna University of Engineering and Technology, Khulna, Bangladesh
aupeef@gmail.com

³ University of Dhaka, Dhaka, Bangladesh

Abstract. This research explores the performance of various language models for generating Bangla text, with an emphasis on the task of text abstractive summarization, specifically newspaper headline generation. Given the concern regarding the lack of diversity in previous newspaper article datasets, we have created a dataset from Bangla online newspapers, focusing on the most recent and diverse news for evaluation purposes. The dataset contains a wider variety of article types and includes articles from a greater number of newspapers than previous datasets. Through comprehensive experimentation and evaluation, we identify BanglaT5 and GPT-3.5 as standout performers in this domain. While GPT-3.5 falls short of surpassing the fine-tuned BanglaT5, its performance notably outshines that of other large language models (LLMs), boasting a substantial performance margin exceeding 10% in comparison. Moreover, the analysis we conducted indicates that the fine-tuned BanglaT5 performs much better than GPT-3.5 by 5% for both ROUGE-1 and ROUGE-L scores, demonstrating the effectiveness of capturing the subtleties of this task. These findings underscore the pivotal role of model fine-tuning and highlight the nuanced interplay between various language models. They showcase that while LLMs are making progress, they still do not perform as well as traditional LMs in the Bangla language processing landscape.

Keywords: Summarization · Abstractive Summarization · LMs · Tiny-LLMs · LLMs

1 Introduction

Due to the exponential growth in textual content and the time-consuming process of manual summarization, the need for Automatic Text Summarization

F. Sultana and Md. T. H. Fuad—These authors contributed equally to this work.

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

A. Antonacopoulos et al. (Eds.): ICPD 2024, LNCS 15320, pp. 72–86, 2025.

https://doi.org/10.1007/978-3-031-78498-9_6

(ATS) is of growing importance. Emphasizing it, El-Kassas et al. [11] offered an extensive overview of ATS approaches, procedures, methods, standards, and future directions for research in this field. Given the increasing volume of Bangla textual content in digital format, our study focuses on the summarization of Bangla news documents. In this work, we focus on one-line summary generation, where we take news headings as the summary of news articles. Leveraging Large Language Models (LLMs), this research aims to compare the performance of the text summarization text-to-text transformer model with the LLMs that address the challenge posed by the abundance of textual data.

In Natural Language Processing (NLP), Large Language Models (LLMs) have become a revolutionary force. These models mark a significant shift in the way machines understand and generate human language. A thorough overview of the models, datasets, and insights that have been developed subsequently in the field of large language models is provided by Naveed et al. [24]. With its comprehensive analysis and comparisons of different LLMs, including their architectures, training methods, and performance in various contexts, it is a valuable resource for researchers. With hundreds of millions to billions of parameters, LLMs have the capacity and sheer scale to learn intricate linguistic patterns, semantics, and contextual links.

A variety of research has been conducted in the field of text summarization. Some of the works involving machine learning [3] [16], deep learning [28] [31] and comprehensive surveys on text summarization are discussed in the literature review section of this paper. Additionally, sequence-to-sequence model [37], GPT2 model [26], IndoBERT and BERTsum model [22], pre-trained language models [30], and T5 model [27] are also discussed. For the Bengali language, graph-based features [15], sentiment analysis, and sentence interconnections [19] and incorporating pre-trained language models in a graph-based model [9] have been approached.

From observing these existing studies, it can be noted that the performance of LLMs has not been adequately analyzed in the context of Bangla text summarization. The recent advancements in LLMs have opened up new possibilities in the field of text summarization, which motivated us to conduct research in this field. LLMs, or large language models, have a remarkable capability to grasp complex linguistic patterns and syntax. These models can be fine-tuned to perform well on particular tasks, especially for languages with limited resources, such as Bangla. By doing so, they gain a deep understanding of the context and improve their grasp of semantic relationships.

The major contributions of our work are as follows:

- Leveraging the advantages of transfer learning for fine-tuning LLM on a specific task in the context of the Bengali language.
- Fine-tune the Bangla-T5 model for the specific task and leverage the LLMs to analyze the performance gap between them.
- To see if the LLMs are ready for the Bangla text summarization task.
- Creating a custom test dataset of the recent Bangla news documents made by scraping from online newspapers.

The rest of this paper is organized in the following manner: the literature review section provides a detailed discussion of available studies regarding text summarization of various languages, including Bangla. In the dataset creation section, the way of creating the new dataset is described. Followed by dataset creation, the experimented models are described. The section called Experimental Setup and Result Analysis demonstrates the way this research has been conducted and the results that we obtained from them. Finally, in the conclusions section, an overview of our work, its limitations, and future direction are presented.

2 Related Works

The fundamental concept of text summarization and various approaches taken in this domain are mentioned hereafter in this section.

2.1 Text Summarization in Natural Language Processing

The main objective of text summarization is to extract the most important details from a given text while reducing its length. Text summarization falls into two primary categories: extractive, which takes sentences or phrases from the original text and rearranges them to create a summary, and abstractive, which creates entirely new sentences to convey the summarized content and frequently necessitates a deeper comprehension of the source text.

Extractive Text Summarization Approaches. While the goal of extractive text summarization methods is to identify and compile the most important information from source texts, several studies have made substantial contributions in this field. For instance, A. K. Yadav et al. [25] proposed a transfer model on extractive text summarization techniques, with a focus on graphical-based approaches, and addressed their benefits, drawbacks, and methods of assessment. In a similar vein, Waseemullah et al. [13] introduced an extractive summarization model to balance compression and retention ratios.

Abstractive Text Summarization Approaches. Significant advancements have also been made in abstractive text summarization, which aims to produce summaries that might not exactly match the original text. In this context, the study [32] provided a comprehensive survey of neural network-based abstractive text summarization models, emphasizing important aspects of the design, difficulties encountered, and recommendations for enhancing performance. This thorough analysis by T. Shi et al. [29] offers an open-source toolkit (NATS) and explores recent developments in seq2seq models for abstractive text summarization. Akash et al. [14] used a multimodal approach for abstraction summarization using Clip and general-purpose LLMs.

Hybrid Text Summarization Approaches. Text summarization hybrid approaches integrate multiple techniques, including extractive and abstractive methods, to enhance the quality of the summary. Several important studies that address this subject include the study of A. Widyassari et al. [36] provided a comprehensive review of the literature on text summarization, highlighting the importance of features for both abstractive and extractive summarization, P. Muniraj et al. [23] introduced HNTSumm, an algorithm that combines supervised and unsupervised learning techniques to facilitate automatic text summarization for documents containing transliterated words.

2.2 Related Studies in Multilingual Text Summarization

In the field of natural language processing, multilingual text summarization aims to produce concise summaries from text in several languages. This more comprehensive multilingual viewpoint is pertinent to Bangla text summarization because it makes it possible to create automated summarization systems that can process and summarise text in Bangla as well as other languages, enabling cross-lingual knowledge extraction and information retrieval.

Using the IndoBERT model and BERTSum, H. Lucky et al. [22] investigated Indonesian single-document abstractive text summarization. In comparison to single-encoder models, Y. Shin [30] presented a Korean abstractive text summarization model that improved performance by utilizing multiple pre-trained language models through a multi-encoder approach. B. Ay et al. [4] used the T5 model to abstractly summarise Turkish text, produce good results, and make the dataset available to other researchers. Using state-of-the-art results on the TR-News and MLSum (TR) datasets, B. Baykara and T. Güngör [5] investigated the applicability of pre-trained Seq2Seq models, such as mT5 and BERTurk-cased, on Turkish text summarization and title generation tasks. Moreover, it is noteworthy that studies have been conducted in this field for the Bangla language [20] as well. P. Protim Ghosh et al. [15] used graph-based sentence scoring features introduced for summarising Bangla news documents. A hybrid method for Bengali text summarization that makes use of sentiment analysis, sentence interconnections, and keyword scoring is presented by M. Islam et al. [19]. Fatima et al. [12] improved news headline text generation quality using POS-Tagging in the tokenization. Ding et al. [10] experimented with human-AI text co-creation for summarizing as headlines using LLMs to harness the power of it.

3 Dataset Creation

The existing datasets (e.g., Potrika [2]) often contain news articles that are outdated, as these were created in the past. As a result, these datasets lack diverse and recent styles of news reporting. Keeping this in mind, we have created a small dataset that aims to address these limitations. The new dataset comprises articles from ten different types of news variations, ensuring greater diversity. To capture the most up-to-date styles, we have scraped articles from ten different online newspapers in 2024. Compared to previous datasets, this new collection

includes a broader range of news variations to better reflect the evolving nature of news reporting and article styles. In Table 1, the feature level comparison with the existing similar type dataset is shown. IndicNLG-BN [21] and Potrika [2] are two popular datasets for News Articles. Primarily, our dataset has more diverse types and sources of data. We also have the links to news articles for verifying their authenticity. In the Fig. 1, the way of data scraping is described. The figure illustrates the process from data collection to normalizing the entire dataset.

Source of Data: We chose online newspapers as our data source. Every day, almost 6 million people read newspapers online. There are many different kinds of articles in every newspaper, including ones about the economy, sports, national and international news, entertainment, science, politics, and short stories by various novelists. Every article has an appropriate title. The newspapers that we used to gather our data were: [Prothom Alo](#), [Daily Ittefaq](#), [Daily Kaler Kantho](#), [Bangladesh Pratidin](#), [Daily Samakal](#), [Daily Janakantha](#), [Daily Naya Diganta](#), [Amader Shomoy](#), and [Daily Inqilab](#). Since each online newspaper has a unique website, we used the Beautiful Soup framework and Selenium to scrape data from eight different newspapers. With Selenium, we can replicate various web page activities like scrolling and dynamic loading. To enable code to communicate with browsers (such as Chrome, Firefox, etc.), a web driver is needed. We can parse the HTML and XML documents using Beautiful Soup. We used this framework to scrape the data after Selenium enabled us to access the webpage. It assists us in gathering and organizing the necessary information from the complete HTML content.

Table 1. Feature-level comparison among the proposed and the existing datasets.

Features	IndicNLG-BN	Potrika	Ours
Types of Article	-	8	10
Headline	✓	✓	✓
#Newspaper	-	8	10
Category	×	✓	✓
Links	×	×	✓
#Examples	142k	320k	60k

Noise Removing and Normalizing: Following the raw data scraping process, the texts still had some extra HTML codes, digits, and extraneous English letters. Consequently, we eliminated these noises and also removed the unnecessary spaces from the beginning and end of every text. Generating human-readable one-word headlines that sufficiently encapsulate the news is difficult, so it is even more difficult for the machines. So, every single one-word title in the dataset was chosen and eliminated once the noise was eliminated. This was achieved by splitting the titles by spaces into a list and then removing any values that had a length of 1.

To convey pronunciation and grammatical information, Bangla text makes use of a variety of diacritical marks. The normalizer either eliminates or

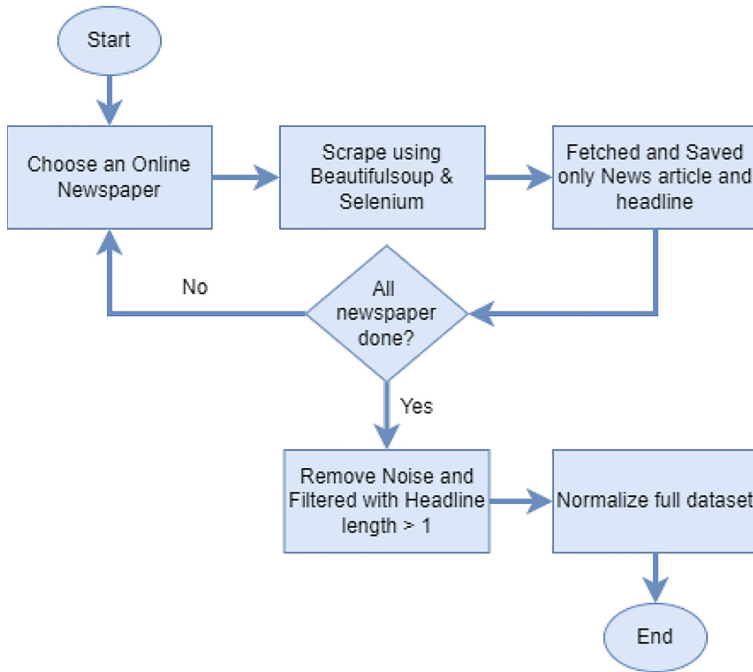


Fig. 1. Flowchart for data collection.

substitutes a standard form for these diacritical marks. The normalizer also has the responsibility of eliminating punctuation entirely or replacing it with a conventional form (e.g., a full stop “.” in place of a vertical bar “[|]”) because punctuation can produce variations and inconsistencies. Furthermore, there are a few minor variations in how Bangla characters might be written. For example, the normalizer may change ligatures to their separate forms or convert conjuncts to their respective forms.

Following the process of data cleaning, the BUET normalizer [1] is used to normalize the whole dataset. The primary purpose of the BUET normalizer is to reduce data sparsity by standardizing characters and punctuations that have various Unicode representations. Additionally, it eliminates foreign strings that show up on both sides of a pair—mostly sentences that have had translations done on both sides of the pair. We carried out the remaining normalization procedure by ourselves.

4 Experimented Models

Text summarization has always been a difficult task for machine learning models, specifically if the summarization is the headline generation of a news article. We fine-tuned the encoder-decoder-based language models (LM) for the summarization task and also fine-tuned some Large Language models (LLMs) using Peft Lora [18].

4.1 Fine Tuning Language Models

Let $X = (x_1, x_2, \dots, x_n)$ represent the input sequence of tokens for a news article, where n is the length of the input sequence. Similarly, let $Y = (y_1, y_2, \dots, y_m)$ represent the output summary sequence, where m is the length of the output summary.

The *Encoder* portion of the model is a stack of *encoder_layers*. Each *encoder_layer* consists of bidirectional self_attention and a multi-layer perceptron layer (MLP) [35]. Each layer helps the encoder to capture the contextual representation.

$$\text{Encoder: } Z = \text{Encoder}(X)$$

where $Z = (z_1, z_2, \dots, z_n)$ represents the contextualized input representations. Z captures the contextual information of a given input article.

Similarly, The *Decoder* portion of the model is a stack of *decoder_layers* where the output at each step is conditioned on the previously generated tokens. Each *decoder_layer* consists of uni-directional self-attention, encoder-decoder attention, and an MLP [35]. The decoder outputs a probability distribution over the vocabulary at each step.

$$\text{Decoder: } P = \text{Decoder}(Y, Z)$$

where $P = (p_1, p_2, \dots, p_m)$ represents the probability distribution over the vocabulary for each token in the output summary sequence.

During training, the model is optimized to maximize the likelihood of generating the target summary sequence given the input sequence:

$$\mathcal{L}(\theta) = \sum_{i=1}^m \log P(y_i | y_{<i}, X; \theta)$$

where θ represents the parameters of the model.

During inference, the model generates the output summary sequence by iteratively sampling tokens from the probability distribution produced by the decoder until an end-of-sequence token is generated or a maximum length is reached.

We used both BanglaT5 [8] and small BanglaT5 [8] to fine-tune our dataset for this task. BanglaT5 is a sequence-to-sequence Transformer model specifically designed for Bangla NLG tasks. It's based on the T5 (Text-to-Text Transfer Transformer) architecture but pre-trained on a massive Bangla text corpus. The model can acquire general representations of the Bangla language through this pre-training phase.

4.2 Fine Tuning Large Language Models

We fine-tuned LLMs (Llama 2, TinyLlama, and Gemma-2b) using LoRA (Low-Rank Adaptation) [18] and PEFT (Parameter-Efficient Fine-Tuning) [38]. PEFT is a technique focused on adapting pre-trained models to new tasks with minimal changes to the original parameters. This reduces training time and memory usage

compared to traditional fine-tuning approaches that heavily modify the pre-trained model. LoRA is a specific method within PEFT that introduces a low-rank matrix to capture the task-specific adjustments needed for fine-tuning. This low-rank matrix requires significantly fewer parameters compared to modifying all the parameters in the pre-trained model, leading to efficiency gains.

Let's consider the original pre-trained model parameters as W , and the Lora-based fine-tuned parameters as W' . The Lora decomposition can be expressed as:

$$W' = W + \Delta W \quad (1)$$

where ΔW is the low-rank update to the original parameters W . The low-rank update ΔW is further decomposed as:

$$\Delta W = A * B^T \quad (2)$$

where A and B are the trainable Lora matrices with reduced rank $r \ll d$, where d is the original dimension of the weight matrix W . Here, the rows of A and the columns of B represent the task-specific adaptations to the original weight matrix W .

The input article A is tokenized using the tokenizer of the chosen model. Here, A is converted into a sequence of numerical IDs representing each word using the tokenizer trained on the pre-trained LLM's vocabulary. Then the tokenized sequence is transformed by the fine-tuned model with the Lora-adapted weights W' as follows:

$$Y = W' * X \quad (3)$$

We used the TinyLlama-1.1B [39] model to fine-tune our dataset. This model offers versatility across applications requiring constrained computational and memory resources. It uses 3 trillion tokens from the original Llama model in pretraining. The other fine-tuned model is Gemma 2b [33]. It is with 2 billion parameters. Gemma is noted for being surprisingly responsive, even faster than the 1.1B parameter TinyLlama model. This suggests it might be optimized for speed while maintaining good performance.

Bangla Llama. Using the HuggingFace API of the Llama 2 [34] model, we fine-tuned the model on a single portion of the dataset (Bangla2B+ [7]) used to train BanglaBert. Bangla2B+ is a massive corpus of Bangla text data crawled from various popular Bangla websites (around 27.5 GB as mentioned in the research paper). This data could include news articles, blog posts, social media content, and other web-based text sources in Bangla.

To fine-tune, we used the PEFT-LoRA method. We also trained the embeddings, LM heads, and the newly included LoRA parameters for this model. The goal is to make Bangla Llama. For this challenge, we employed the original tokenizer from Llama2. After fine-tuning this general-purpose dataset, we tested our dataset using different prompting for the summarization task.

4.3 Generation Using Prompt

We used prompting to experiment with the summarization task in various LLMs. Prompting entails organizing the needs into a format for input that conveys the desired results to the model, thus achieving the intended output. The chosen models for this task are Gemma-7b, Llama 2 (without any fine-tuning), and GPT 3.5-Turbo.

Gemma-7b and Llama 2: We used both Gemma-7b [33] and Llama 2 [34] models for our task. We used the same prompt for both models. We tested by using different prompts and achieved the best result using the following prompt:

Here is a Bangla Article. Provide one one-line summary for the article in Bangla. Article: {Tokenized_Input}

GPT 3.5: Like other Transformer-based models, GPT-3.5 likely utilizes an encoder-decoder architecture. After processing the incoming text, the encoder records the word relationships and meanings inside it. The decoder then produces the output text using the encoded data.

A well-crafted prompt can guide the model toward an output style, format, or subject. We tried both zero-shot prompting and few-shot prompting for summary generation. In the case of few-shot prompting, we provided three examples of the model. The prompt was different from the prompt of the Gemma model. In the following prompt, we achieved the best result from the GPT. For both types of prompting, we used a similar style.

Prompt for few-shot:

What is the short and concise headline of the given Bengali news article delimited by <> in Bengali while preserving the context?
 The headline should be strictly restricted to one short, meaningful Bengali sentence having accurate grammar and spelling. Only keep the generated headline in the output. Ignore all preceding and all non-contextual English words in output.
 Here are a few examples of Bengali news articles along with their concise one-line titles in Bengali.
 Example 1: "ex1"
 Example 2: "ex2"
 Example 3: "ex3"
 Article: <Article>

5 Experimental Setup and Result Analysis

In this experiment, we incorporated our dataset for both training and testing and used the Potrika dataset for testing with prompts for the summarization task. We also used the Bangla2B+ dataset for fine-tuning Llama 2. We explored various setup configurations to achieve the best result for a specific model.

5.1 Experimental Setup

BanglaT5. We extensively experimented to find the best setup for fine-tuning the BanglaT5 model. We trained the model for 10 epochs with the learning rate $2e^{-5}$ and weight decay 0.01. We used 8 batch sizes for low-resource GPU. We used Kaggle’s limited version of p100 (16 GB GPU) to train the entire train data with the stated setup. Evaluation metrics like rogue-1, rogue-2, and rogue-l were computed.

Bangla Llama. To develop the primary version of Bangla Llama, we fine-tuned Llama-2. We used the model with 7b parameters. 1.8GB out of the whole Bangla2B+ dataset was used for the training. The pertaining was performed with the Casual Language Modelling Approach, which is integrated with LoRA adapters into the attention vectors and subsequently, we trained the embeddings, LM heads, and the newly incorporated LoRA parameters. We used Colab 40GB T4 GPU for this purpose. The training approach involves setting the batch size to 64, utilizing FP16 for training precision, an initial learning rate of $2e^{-4}$, a maximum sequence length of 512, a LoRA rank of 64, a LoRA alpha of 128 and LoRA target modules including QKVO and MLP.

Other Models. We fine-tuned the tiny LLMs in our dataset. The process is completed using the PEFT-LoRA method same as the Llama2. For this purpose, we used Kaggle p100 setup for training. The setup for LoRA configuration is: LoRa rank is 8, Lora alpha is 16 with batch size is 8 and 3 epochs.

Additionally, we choose two summarization models from the Buetnlp archive to analyse the results of this task. These models are mT5-m2m-crossSum [6] and mT5-multilingual-XLSum [17]. These are multilingual models pre-trained in CrossSum [6] dataset and XLSum [17] dataset chronically. We generate summarization from these models and compare the results with our findings.

5.2 Performance Metrics

Various critical performance metrics are utilized to evaluate language models and offer valuable perspectives on their efficacy. To obtain a thorough grasp of the model’s performance, we have concentrated our evaluation on the commonly used metrics. We used the ROUGE metric to evaluate our training and testing. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a suite of metrics used to evaluate the quality of text summaries by comparing them with reference summaries created by humans. We used three types of ROUGE: ROUGE-1, ROUGE-2, and ROUGE-L. All ROUGE scores measure the overlap (n-gram match) between the generated summary (X) and reference summaries (R). Equation 1 defines the way of calculating the value.

$$ROUGE - N = \frac{\sum count(X_n)}{\sum count(R_n)} \quad (4)$$

Table 2. ROUGE score evaluation for all the models on the proposed dataset. The left column stated the types of the model.

	Model	ROUGE-1	ROUGE-2	ROUGE-L	BertScore
Prompting	Fewshot on GPT-3.5	20.1	2.12	17.32	78.24
	Zeroshot on Test	20.34	2.26	17.42	74.65
	Llama2-7b	8.44	0.88	5.72	68.23
	Gemma-7b	6.392	1.67	6.015	71.31
LM	Fine Tuned BanglaT5	25.899	4.098	23.828	81.20
	Fine Tuned Small BanglaT5	21.873	3.243	20.049	85.62
	mT5-m2m-crossSum	15.240	2.12	13.741	75.12
	mT5-multilingual-XLSum	13.716	1.89	11.342	74.44
LLM	Bangla Llama	8.97	0.92	6.27	72.76
	Fine Tuned TinyLlama	2.132	0.024	1.971	61.56
	Fine Tuned Gemma 2b	2.865	0.378	2.716	64.87

where, N : Denotes the n-gram size (1 for unigrams, 2 for bigrams, etc.), X_n : Number of n-grams in the generated summary that exactly match n-grams in any of the reference summaries, and R_n : Total number of n-grams in all reference summaries. Firstly, ROUGE-1 measures the overlap of individual words (unigrams) between the generated summary and the reference summaries. Secondly, ROUGE-2 focuses on the overlap of two-word sequences (bigrams) between the generated summary and the reference summaries. Finally, ROUGE-L finds the longest sequence of words that appears in both the generated summary and any of the reference summaries. This metric doesn't have a single equation but involves finding the LCS (Longest Common Subsequence) length between the summary and each reference and then taking the maximum LCS length across all references.

We also used BERTScore for our evaluation. It evaluates text similarity by converting text into BERT embeddings and comparing token-level cosine similarities between candidate and reference texts. It then aggregates these similarities to compute precision, recall, and F1 scores, capturing semantic meaning beyond mere word overlap.

5.3 Result Analysis

In our investigation, we analyzed the performance of variants of the fine-tuned BanglaT5 model and the popular LLMs. Through concise experimentation and evaluation, we gained a gist about the performance of the T5 model and LLMs. For comparison, we also generate other Bangla summarization models with our dataset.

In Table 2, the performance in our dataset is shown. In comparing the performance of BanglaT5 base and BanglaT5-small models, the difference proves to be marginal, with the base model edging out its smaller counterpart by approximately 3% in both ROUGE-1 and ROUGE-L scores across datasets.

Table 3. ROUGE score evaluation for all the models on the Potrika dataset. The left column stated the types of the model.

	Model	ROUGE-1	ROUGE-2	ROUGE-L	BertScore
Prompting	Fewshot on GPT-3.5	18.61	2.28	16.76	73.74
	Zereshot on Test	20.32	2.73	18.38	73.35
	Llama2-7b	8.14	0.78	5.22	65.05
	Gemma-7b	5.679	1.456	5.57	69.16
LM	Fine Tuned BanglaT5	25.345	4.854	23.048	76.06
	Fine Tuned Small BanglaT5	22.865	4.215	20.926	74.65
	mT5-m2m-crossSum	14.546	1.94	13.082	72.89
	mT5-multilingual-XLSum	13.526	1.39	10.841	73.11
LLM	Bangla Llama	8.37	0.8	5.67	70.03
	Fine Tuned TinyLlama	1.971	0.018	1.742	60.701
	Fine Tuned Gemma 2b	2.69	0.315	2.418	62.12

Notably, both BanglaT5 variants outshine other summarization models such as mT5-multilingual-XLSum and mT5-m2m-crossSum. Conversely, among large language models (LLMs), GPT-3.5 Turbo exhibits superior zero-shot performance compared to all other LLMs, including fine-tuned versions like LLama and tiny LLMs, albeit marginally outperforming the base LLama version. While GPT-3.5 Turbo’s results approach those of T5 models, it falls short of surpassing them. The disparity between GPT-3.5 Turbo and other LLMs is substantial, with a notable gap of approximately 12% from the performance of the fine-tuned LLama 2 version, which outperforms its base counterpart. Furthermore, prompting solely in GPT yields better performance than other LLMs. Lastly, while Gemma-7b surpasses LLama 2 in ROUGE-1 and ROUGE-2 scores, it lags in ROUGE-L compared to fine-tuned LLama models. However, in the case of BERTScore, Fine-tuned small BanglaT5 excels other models, and the rest of the models follow a similar trend as the ROUGE-L scores.

Experimentation on Other Dataset. We also tested on the Potrika [2] dataset in the same setup as our dataset. The performance of each model is shown in Table 3. Here, the results show the same trends as our dataset’s performance. Among the prompting-based approaches, GPT-3.5 variants demonstrate notable performance, outpacing other prompting models such as Llama2-7b, Gemma-7b, and Bangla Llama. However, fine-tuned language models (LMs) like BanglaT5 showcase superior summarization capabilities, surpassing both prompting-based and other LM variants like mT5-m2m-crossSum and mT5-multilingual-XLSum. Notably, Fine Tuned BanglaT5 and Fine Tuned Small BanglaT5 emerge as top performers, underscoring the efficacy of LM fine-tuning. Conversely, fine-tuning large language models (LLMs) like TinyLlama and Gemma 2b yields suboptimal results, indicating significant challenges in generating accurate summaries for Bangla text.

6 Conclusion and Future Works

The study on Bangla text summarization demonstrates the continued dominance of fine-tuned language models over large language models (LLMs). Specifically, the BanglaT5 model emerges as the most efficient choice, consistently outperforming other models across various evaluations. In contrast, LLMs, such as GPT, struggle to match the performance of fine-tuned models, even when attempting to fine-tune smaller LLMs. This indicates that LLMs have yet to reach the same level of proficiency as language models in Bangla summarization and text generation tasks. The experimental findings corroborate these conclusions, with BanglaT5 variants consistently outperforming both prompting-based methods and other LM and LLM variants in terms of ROUGE scores. While the GPT-3.5 Turbo model demonstrates promising zero-shot performance, it falls short of surpassing fine-tuned models like BanglaT5, reaffirming the superiority of fine-tuning approaches for optimizing language models in specific domains, such as Bangla text processing.

Our future plan includes enlarging our proposed dataset to improve the effectiveness of our models. We also intend to train the Llama2 model on this comprehensive dataset to create a state-of-the-art Bangla Llama.

References

1. GitHub - csebuetnlp/normalizer: This python module is an easy-to-use port of the text normalization used in the paper “Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation”. It is intended to be used for normalizing / cleaning Bengali and English text. — github.com. <https://github.com/csebuetnlp/normalizer>. Accessed 01 July 2024
2. Ahmad, I., AlQurashi, F., Mehmood, R.: Potrika: raw and balanced newspaper datasets in the Bangla language with eight topics and five attributes. arXiv preprint [arXiv:2210.09389](https://arxiv.org/abs/2210.09389) (2022)
3. Alguliyev, R.M., Aliguliyev, R.M., Isazade, N.R., Abdi, A., Idris, N.: Cosum: text summarization based on clustering and optimization. *Expert. Syst.* **36**(1), e12340 (2019)
4. Ay, B., Ertam, F., Fidan, G., Aydin, G.: Turkish abstractive text document summarization using text to text transfer transformer. *Alex. Eng. J.* **68**, 1–13 (2023)
5. Baykara, B., Güngör, T.: Turkish abstractive text summarization using pretrained sequence-to-sequence models. *Nat. Lang. Eng.* **29**(5), 1275–1304 (2023)
6. Bhattacharjee, A., Hasan, T., Ahmad, W.U., Li, Y.F., Kang, Y.B., Shahriyar, R.: Crosssum: beyond English-centric cross-lingual summarization for 1,500+ language pairs. In: Annual Meeting of the Association of Computational Linguistics 2023, pp. 2541–2564. Association for Computational Linguistics (ACL) (2023)
7. Bhattacharjee, A., et al.: Banglabert: language model pretraining and benchmarks for low-resource language understanding evaluation in bangla. arXiv preprint [arXiv:2101.00204](https://arxiv.org/abs/2101.00204) (2021)
8. Bhattacharjee, A., Hasan, T., Ahmad, W.U., Shahriyar, R.: BanglaNLG and BanglaT5: benchmarks and resources for evaluating low-resource natural language generation in Bangla. In: Vlachos, A., Augenstein, I. (eds.) Findings of the Association for Computational Linguistics: EACL 2023, pp. 726–735. Association for

- Computational Linguistics, Dubrovnik, Croatia (2023). <https://doi.org/10.18653/v1/2023.findings-eacl.54>. <https://aclanthology.org/2023.findings-eacl.54>
9. Chowdhury, R.R., Nayeem, M.T., Mim, T.T., Chowdhury, M.S.R., Jannat, T.: Unsupervised abstractive summarization of Bengali text documents. arXiv preprint [arXiv:2102.04490](https://arxiv.org/abs/2102.04490) (2021)
 10. Ding, Z., Smith-Renner, A., Zhang, W., Tetreault, J.R., Jaimes, A.: Harnessing the power of LLMs: evaluating human-AI text co-creation through the lens of news headline generation. arXiv preprint [arXiv:2310.10706](https://arxiv.org/abs/2310.10706) (2023)
 11. El-Kassas, W.S., Salama, C.R., Rafea, A.A., Mohamed, H.K.: Automatic text summarization: a comprehensive survey. *Expert Syst. Appl.* **165**, 113679 (2021)
 12. Fatima, N., Daudpota, S.M., Kastrati, Z., Imran, A.S., Hassan, S., Elmitwally, N.S.: Improving news headline text generation quality through frequent POS-tag patterns analysis. *Eng. Appl. Artif. Intell.* **125**, 106718 (2023)
 13. Fatima, Z., et al.: A novel approach for semantic extractive text summarization. *Appl. Sci.* **12**(9), 4479 (2022)
 14. Ghosh, A., Acharya, A., Jain, R., Saha, S., Chadha, A., Sinha, S.: Clipsyntel: clip and LLM synergy for multimodal question summarization in healthcare. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 22031–22039 (2024)
 15. Ghosh, P.P., Shahariar, R., Khan, M.A.H.: A rule based extractive text summarization technique for Bangla news documents. *Int. J. Mod. Educ. Comput. Sci.* **10**(12), 44 (2018)
 16. Hannah, M.E.: A hybrid classification-based model for automatic text summarisation using machine learning approaches: CBS-ID3MV. *Int. J. Prod. Dev.* **23**(2–3), 201–211 (2019)
 17. Hasan, T., et al.: XL-sum: large-scale multilingual abstractive summarization for 44 languages. arXiv preprint [arXiv:2106.13822](https://arxiv.org/abs/2106.13822) (2021)
 18. Hu, E.J., et al.: Lora: low-rank adaptation of large language models. arXiv preprint [arXiv:2106.09685](https://arxiv.org/abs/2106.09685) (2021)
 19. Islam, M., Majumdar, F.N., Galib, A., Hoque, M.M.: Hybrid text summarizer for Bangla document. *Int. J. Comput. Vis. Sig. Process.* **1**(1), 27–38 (2020)
 20. Khan, A., Kamal, F., Chowdhury, M.A., Ahmed, T., Laskar, M.T.R., Ahmed, S.: Banglachq-summ: an abstractive summarization dataset for medical queries in Bangla conversational speech. In: *Proceedings of the First Workshop on Bangla Language Processing (BLP-2023)*, pp. 85–93 (2023)
 21. Kumar, A., et al.: Indicnlg benchmark: multilingual datasets for diverse NLG tasks in indic languages. arXiv preprint [arXiv:2203.05437](https://arxiv.org/abs/2203.05437) (2022)
 22. Lucky, H., Suhartono, D.: Investigation of pre-trained bidirectional encoder representations from transformers checkpoints for Indonesian abstractive text summarization. *J. Inf. Commun. Technol.* **21**(01), 71–94 (2022)
 23. Muniraj, P., Sabarmathi, K., Leelavathi, R., et al.: Hntsumm: hybrid text summarization of transliterated news articles. *Int. J. Intell. Netw.* **4**, 53–61 (2023)
 24. Naveed, H., et al.: A comprehensive overview of large language models. arXiv preprint [arXiv:2307.06435](https://arxiv.org/abs/2307.06435) (2023)
 25. Pilault, J., Li, R., Subramanian, S., Pal, C.: On extractive and abstractive neural document summarization with transformer language models. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9308–9319 (2020)
 26. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. *OpenAI Blog* **1**(8), 9 (2019)

27. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. CoRR [arXiv:1910.10683](https://arxiv.org/abs/1910.10683) (2019)
28. Rahman, M.M., Siddiqui, F.H.: An optimized abstractive text summarization model using peephole convolutional LSTM. *Symmetry* **11**(10), 1290 (2019)
29. Shi, T., Keneshloo, Y., Ramakrishnan, N., Reddy, C.K.: Neural abstractive text summarization with sequence-to-sequence models. *ACM Trans. Data Sci.* **2**(1), 1–37 (2021)
30. Shin, Y.: Multi-encoder transformer for Korean abstractive text summarization. *IEEE Access* (2023)
31. Song, S., Huang, H., Ruan, T.: Abstractive text summarization using LSTM-CNN based deep learning. *Multimed. Tools Appl.* **78**(1), 857–875 (2019)
32. Syed, A.A., Gaol, F.L., Matsuo, T.: A survey of the state-of-the-art models in neural abstractive text summarization. *IEEE Access* **9**, 13248–13265 (2021)
33. Team, G., et al.: Gemma: open models based on Gemini research and technology. arXiv preprint [arXiv:2403.08295](https://arxiv.org/abs/2403.08295) (2024)
34. Touvron, H., et al.: Llama 2: open foundation and fine-tuned chat models. arXiv preprint [arXiv:2307.09288](https://arxiv.org/abs/2307.09288) (2023)
35. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
36. Widyassari, A.P., Rustad, S., Shidik, G.F., Noersasongko, E., Syukur, A., Affandy, A., et al.: Review of automatic text summarization techniques & methods. *J. King Saud Univ.-Comput. Inf. Sci.* **34**(4), 1029–1046 (2022)
37. Xi, X., Pi, Z., Zhou, G.: Global encoding for long Chinese text summarization. *ACM Trans. Asian Low-Resour. Lang. Inf. Process. (TALLIP)* **19**(6), 1–17 (2020)
38. Xu, L., Xie, H., Qin, S.Z.J., Tao, X., Wang, F.L.: Parameter-efficient fine-tuning methods for pretrained language models: a critical review and assessment. arXiv preprint [arXiv:2312.12148](https://arxiv.org/abs/2312.12148) (2023)
39. Zhang, P., Zeng, G., Wang, T., Lu, W.: Tinyllama: an open-source small language model. arXiv preprint [arXiv:2401.02385](https://arxiv.org/abs/2401.02385) (2024)



Navigating Data Imbalances in Cybersecurity: Identifying Malicious URLs with Multiple Labels and Extreme Data Imbalances with LGNet

Anran Zhu¹, Yubo Huang^{2(✉)}, and Xin Lai²

¹ School of Computing and Artificial Intelligence,
Southwest Jiaotong University, Chengdu, China

² School of Civil Engineering, Southwest Jiaotong University, Chengdu, China
ybforever@my.swjtu.edu.cn

Abstract. Malicious URLs are a form of cyberattack, that manipulates individuals into disclosing sensitive information. Typically, Malicious URLs constitute a minor proportion of all searchable URLs and are marked by multi-labeling and a significant imbalance in sample data. These characteristics considerably diminish the effectiveness and precision of existing detection methodologies. This paper introduces LGNet, an innovative network framework designed to identify Malicious URLs in practical scenarios efficaciously. To tackle the multi-labeling issue, we have developed a label propagation algorithm that employs a confidence threshold limitation, ensuring high-confidence labeled URLs are acquired. We enhance the scalable tree system by using BILSTM and attention mechanism to address the substantial disparity between labeled and sample data in Malicious URL prediction. Our experimental results demonstrate that LGNet markedly surpasses existing state-of-the-art algorithms in detecting Malicious URLs.

Keywords: Malicious URLs · Label propagation · BILSTM · Attention mechanism · Scalable tree system

1 Introduction

Malicious URLs are a prevalent online attack aiming to fraudulently acquire sensitive information from network users, potentially leading to malware exposure, identity theft, or financial loss. The detection and blocking of Malicious URLs present significant challenges, as attackers frequently employ sophisticated techniques to circumvent traditional detection methods [6, 30, 40]. Moreover, Malicious URLs are complex and diverse, and there is no standardized labeling system to define the nature of different types of fraud (e.g., matrimonial scams and lottery scams). These challenges are exacerbated by the growing number of Malicious URL attack methods.

Current researches on Malicious URLs include analyzing definitions [15], investigating human factors [33], and identifying Malicious URLs sites [35]. Traditional

A. Zhu, Y. Huang and X. Lai—These authors contributed equally to this work.

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

A. Antonopoulos et al. (Eds.): ICPR 2024, LNCS 15320, pp. 87–102, 2025.

https://doi.org/10.1007/978-3-031-78498-9_7

detection methods focus on training both end-users and automated software classifiers to recognize Malicious URLs [2, 32]. These methods are fast and expected to be false-positive in cases where Malicious URLs are labeled by subjectively and objectively collecting labeled Malicious URLs from databases [10]. However, they cannot exhaustively identify all labeled Malicious URLs. Additionally, manually labeling a Malicious URL requires visiting the URL's page, which is a very dangerous behavior. Meanwhile, with the explosive growth of actual Malicious URLs, these methods tend to fall short in terms of accuracy and validity, especially in failing to recognize new Malicious URLs [13], which are becoming increasingly difficult to distinguish from legitimate URLs.

With the large number of URLs being generated every day, traditional methods struggle due to their severe limitations. In addition, Malicious URLs are complex and diverse, updating and iterating rapidly. They have become highly deceptive. Shallow features are insufficient to identify Malicious URLs, and many deep features are difficult for humans to extract manually. To address such limitations, models based on machine learning [4] and deep learning [3] have been developed to generalize the prediction of new, unseen URLs. Valentim et al. [39] suggest employing Generative Adversarial Networks (GAN) for enhancing Malicious URL website detection in zero-day networks. Ozcan et al. [31] propose a hybrid model combining Long Short-Term Memory (LSTM) and Deep Neural Network (DNN) algorithms for this purpose. Notably, the datasets in these studies typically feature a balanced or nearly balanced ratio of Malicious URLs to normal URLs [1, 28]. In practice, however, Malicious URLs constitute only a small fraction of all URLs and are characterized by multi-labeling and extreme data imbalance. This significantly reduces the efficiency and accuracy of existing detection methods.

This paper introduces LGNet, a neural network framework specifically designed to accurately identify Malicious URLs amidst a vast array of largely unlabeled and highly unbalanced URL data. LGNet incorporates a label propagation algorithm with a confidence threshold limit for effective data augmentation, producing high-confidence labeled data. Moreover, we have enhanced the scalable tree boosting system with a Bidirectional Long Short-Term Memory (BiLSTM) with an attention mechanism, optimizing our Malicious URL prediction framework. Consequently, our experimental findings demonstrate that LGNet significantly outperforms existing state-of-the-art algorithms in multi-classification tasks involving unbalanced datasets. The main contributions of this paper are summarized as follows:

- We propose a novel Malicious URL detection network, namely LGNet, to address the situation of detecting Malicious URLs with multiple labels and extreme data imbalance in real situations.
- We design label propagation algorithms with confidence threshold constraints that employ a small amount of labeled Malicious URL data to propagate labels on unlabeled data.
- We propose to employ a novel two-channel, single-pooled structure using an improved tree enhancement system to integrate multiple weak classifiers in the detection of fishing URLs, thus improving the accuracy of detection.

2 Related Work

2.1 Malicious URL Detection

In the domain of Malicious URL Detection, recent advancements have pivoted towards employing sophisticated machine learning and deep learning techniques to enhance detection accuracy and efficiency [7, 8, 22]. In the work by Wang et al. [41], the authors explore the use of deep convolutional neural networks (CNNs) to automatically extract and learn feature representations from URL strings, negating the need for manual feature engineering. This approach acknowledges the dynamic and polymorphic nature of malicious URLs, where attackers continuously evolve their tactics to evade detection mechanisms. Another notable study by Mourtaji et al. [29] introduces a hybrid model combining the strengths of machine learning and rule-based filtering. The model leverages the quick response of rule-based systems for known threat patterns and the adaptive learning capability of machine learning algorithms for new, unseen URL structures. This synergy aims to improve the overall detection rate while minimizing false positives.

Furthermore, the integration of natural language processing (NLP) techniques has gained traction [17, 34]. The study by Buber et al. [11] employs NLP-based feature extraction to analyze the lexical and semantic aspects of URLs, facilitating the distinction between benign and malicious web addresses. By treating URLs as natural language strings, this approach benefits from the rich context and linguistic patterns, improving the detection of sophisticated phishing and malware-distributing URLs.

2.2 Extremely Unbalanced Data Processing

The challenge of Extremely Unbalanced Data Processing is prevalent in malicious URL detection due to the disproportionate ratio of benign to malicious URLs [12, 20]. Addressing the imbalance is crucial for developing effective detection systems. Recent literature has focused on innovative methods to mitigate the skewness of data distribution. A significant contribution in this area is the work by Li et al. [25], which proposes an oversampling technique specifically designed for imbalanced URL datasets. Their method, based on synthetic minority oversampling (SMOTE), generates synthetic examples of malicious URLs to balance the dataset, thereby enhancing the training process of machine learning models and improving their generalization capabilities on real-world data.

In the realm of deep learning, the study by Tsai et al. [38] introduces a novel loss function tailored for unbalanced data scenarios commonly encountered in malicious URL detection. This function, termed ‘Balanced Cross Entropy’, adjusts the penalty for misclassification dynamically, giving more weight to the minority class (malicious URLs) and thus addressing the bias towards the majority class (benign URLs). The approach has shown to significantly improve the performance of deep learning models in detecting malicious URLs amidst a vast majority of benign ones. Additionally, the work of He et al. [21] explores the use of ensemble learning techniques to tackle data imbalance. By combining multiple classifiers, each trained on different subsets of the data,

their ensemble model achieves a more balanced view of the data landscape, reducing the dominance of the majority class and improving the detection of malicious URLs.

These studies underscore the critical importance of addressing data imbalance in the context of malicious URL detection. By developing and employing techniques specifically designed to handle extremely unbalanced data, researchers and practitioners can improve the accuracy and reliability of detection systems, ultimately contributing to more secure online environments.

3 Methodology

In this section, we propose a novel Malicious URL detection network framework called LGNet for detecting Malicious URLs that have an extreme imbalance in labeling and data volume in real-world situations.

3.1 LGNet Framework

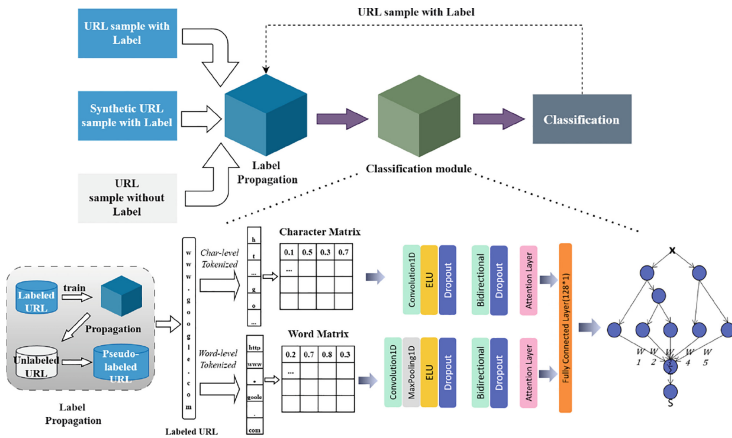


Fig. 1. LGNet Framework: The LGNet network detects Malicious URLs in the following ways: First, a large amount of labeled data is generated by label propagation, using a small amount of labeled URL data to label a large amount of unlabeled URL data with pseudo-labels. Then, the labeled data is classified and detected. In the classification module, string-level and word-vector-level feature extraction is performed on all labeled data to generate string matrices and word matrices. The extracted features are classified through the attention mechanism with the Bilstm framework, combined with the dual-channel single-pool structure.

Figure 1 shows the framework of our LGNet. Specifically, the LGNet network employs a two-stage approach for detecting Malicious URLs. Initially, it generates a substantial dataset through label propagation, leveraging a limited set of labeled URL data to assign pseudo-labels to a vast quantity of unlabeled URL data. Subsequently, this labeled

dataset undergoes classification and detection processes. The classification module executes feature extraction at both the string level and word-vector level for all labeled data, producing string matrices and word matrices. These extracted features are then classified utilizing an attention mechanism within the BI-LSTM framework, which is integrated with a dual-channel, single-pool structure.

To facilitate feature extraction by LGNet, URL sequences are encoded into word vectors through a two-step process. Initially, URL data are segmented at both character and lexical levels to generate respective corpora. These are then subjected to a dual encoding process, producing character and lexical encoding vectors for the URLs [24]. This results in the formation of word-embedded feature vectors, constituting both word and character matrices, crucial for subsequent extraction of semantic features.

After splitting, the URLs in the dataset consist of 96 characters, and each URL u_i is represented as a sequence of characters e_j , where each character is a vector e_j of dimension $m = 96$. The vector is essentially a one-shot encoding, where each dimension corresponds to a character in a dictionary of size 96. The set of characters in a URL is represented by a matrix E of size $m \times n$, where n is the length of the URL. Each column of the matrix corresponds to a character in the URL, represented as a click vector. And each URL constitutes a one-hot matrix $E = E_{m \times n} = (e_1, e_2, \dots, e_n)$. The matrix E is then embedded with a single-layer neural network. The weight matrix W of this layer has a dimension to the $g \times m$ power, where $g = 128$ is the embedding dimension. This process is essentially a linear transformation that maps click-encoded characters into a continuous vector space. The result of the embedding process is a representation matrix S , where each column of S corresponds to an embedding of a character in the URL. The dimension of the matrix S is g multiplied by n , where n is the length of the URL. The arithmetic flow is as follows:

$$S^c = WE = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ w_{g1} & w_{g2} & \cdots & w_{gm} \end{bmatrix} \times \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ e_{m1} & e_{m2} & \cdots & e_{mn} \end{bmatrix} \quad (1)$$

After word vector embedding, URLs are characterized as feature vectors $X_i = (S_i^c, S_i^w)$, where S_i^c is a character-level feature vector learned from u_i , and S_i^w is a vocabulary-level feature vector. Sequence vectorization makes feature-based mathematical computation possible. The feature vectors of URLs will then be learned and multicategorized.

After word embedding, the URL enters the convolutional layer, where it is convolved with a kernel to create multiple feature mappings. The data then goes through a single pooling layer, where maximum pooling is applied for dimensionality reduction. Among them, LGNet removes the pooling layer of Web2Vec [16] in the word-level channel to prevent the pooling layer from losing the semantic information of the text and to optimize the feature extraction of the original model.

The obtained feature data is fed into the convolutional layer. For a certain convolution kernel W , the convolution operation on the matrix R_j is computed as:

$$R_j = f(W \otimes V_{j:j+h-1} + b) \quad (2)$$

where f is the activation function, \otimes represents the dot product operation, V is the input matrix, h is the kernel size, and b is the bias. To reduce the network parameters and extract the most important features, 1-Maxpooling is applied to the feature maps after the convolution operation.

The attention mechanism [19] helps models focus on critical parts of the input sequence. In URL classification tasks, URLs are variable-length sequences [26], and some subsequences may contain crucial information, such as domain names, paths, or parameters. By leveraging the attention mechanism, the model can prioritize these key parts, thereby enhancing its capability to capture essential information. Traditional recurrent neural networks [36] and BiLSTM [37] often struggle with gradient vanishing or explosion when dealing with sequence data. The attention mechanism addresses these issues by better capturing long-distance dependencies and improving the model's long-term memory capabilities.

In the attention mechanism, the inputs are transformed into key, query, and value vectors. The calculation formulas are as follows:

$$\begin{aligned} u_i &= \tanh(S + b) \\ a_i &= \frac{\exp(u_i)}{\sum_i \exp(u_i)} \\ \sum_i a_i &= 1 \\ V &= \sum_t \alpha S \end{aligned} \quad (3)$$

where $S \in \mathbb{R}^{f(\cdot) \times T}$, $f(\cdot)$ represents the dimension of the word vector or character vector, b is the bias of neurons, α_i is computed by applying the Softmax function to the input scores, and V is the weighted summed feature vector. In this setup, S represents the features, and the attention mechanism utilizes these features to compute the attention scores and the final output vector V .

In the process of classification prediction, a scalable tree boosting system [14] is chosen as a classifier for it is able to integrate multiple weak classifiers to obtain better classification performance. In the given $D = \{(x_j, y_j)\} (|D| = n, x_j \in R^m, y_j \in R)$ training set, the prediction function is:

$$\hat{y}_j = \sum_{K=1}^K t_k(x_j), t_k \in M \quad (4)$$

where K denotes that there are K decision trees. Each basic classifier $t_k(x_j)$ is in the form of a tree, $M = \{M(x) = w_{q(x)}\} (q: R^m \rightarrow T, w \in R^T)$ denotes the space of decision trees, q denotes the structure of each tree, which maps an input sample x to the corresponding leaf node, and T denotes the number of leaf nodes in each tree, each t_k corresponds to a separate tree structure and leaf weight w . Unlike decision trees, regression trees have a score on each leaf. Here i_w is used to represent the score of the i -th leaf. To learn the set of functions in the model, the regularized objective function can be written as:

$$L(\theta) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(t_k) \quad (5)$$

$$\Omega(t) = \zeta T + \frac{1}{2} \lambda \|w\|^2 \quad (6)$$

where l is the second-order differentiable loss function used to measure the error between the true value and the predicted value. ζ is the complexity penalty term for the tree, and the canonical term represents the model complexity to prevent overfitting.

Assuming that $\hat{y}_i^{(z)}$ is the predicted value of the i th sample at the z th iteration, the residuals are used to fit the loss function, denoted as:

$$\hat{y}_i^{(z)} = \hat{t}_i^{(z-1)} + t_z(x_i) \quad (7)$$

The objective function becomes the following equation:

$$L^{(z)} = \sum_{i=1}^n l(y_i, \hat{t}_i^{(z-1)} + t_z(x_i)) + \Omega(t_z) \quad (8)$$

Performing a second-order Taylor expansion of the above equation, removing the constant term, and defining $I_j = \{i | q(x_i) = j\}$ as the set of instances of the j th leaf, the substitution collation yields the objective function to be solved as:

$$\tilde{L}^{(z)} = \sum_{i=1}^n \left[E_i t_z(x_i) + \frac{1}{2} H_i t_z^2(x_i) \right] + \zeta T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (9)$$

$$= \sum_{j=1}^T \left[\left(\sum_{i \in I_j} E_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_k} H_i + \lambda \right) w_j^2 \right] + \zeta T$$

$$E_i = \partial_{\hat{y}^{(z-1)}} l(y_i, \hat{y}^{(z-1)}) \quad (10)$$

$$H_i = \partial_{\hat{y}^{(z-1)}}^2 l(y_i, \hat{y}^{(z-1)}) \quad (11)$$

3.2 Loss Function

The aggregate loss function comprises two distinct components: the feature loss incurred during string and word feature extraction, and the node splitting loss incurred during tree refinement.

Prediction Loss. Character feature and word feature extraction aim to encode the original URLs to be processed into the network framework for classification and prediction. In the classifier, we use a softmax classifier to predict the label \hat{y}_i from the discrete set of categories y_i after splicing the feature vectors of the two-channel output. Therefore, the difference between the predicted and actual values can be defined as the loss during feature extraction, defined as follows:

$$L_{pre}(\theta) = -\frac{1}{N} \sum_{n=1}^N \log P(\hat{y}_i | y_i; \theta) \quad (12)$$

where P is the conditional probability of the model with θ indicating the parameters of the model.

Node Splitting Loss. The above equation is used as a score function to measure the goodness of the tree structure, and in practice, it is difficult to enumerate all tree structures q to select the one with the highest score. Therefore, a greedy algorithm is used for node splitting. Starting from the root node as the only leaf node, branches are added by traversing the attributes, assuming that I_L and I_R denote the left and right subtree nodes after splitting, respectively. Then letting $I_L \cup I_R$. The value of the loss that is reduced by splitting a node is:

$$L_{split}(\theta) = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} E_i)^2}{\sum_{i \in I_L} H_i + \lambda} + \frac{(\sum_{i \in I_R} E_i)^2}{\sum_{i \in I_R} H_i + \lambda} - \frac{(\sum_{i \in I} E_i)^2}{\sum_{i \in I} H_i + \lambda} \right] - \zeta \quad (13)$$

Total Loss Function. The total loss function is the sum of feature loss and node splitting loss. The feature loss and node splitting loss are independent of each other, so no weights are assigned to them in this paper, denoted as follows:

$$L_{loss} = L_{split} + L_{pre} \quad (14)$$

In addition, the node splitting process can effectively avoid the overfitting phenomenon by such a pruning process, despite the losses incurred. Therefore, no additional hyperparameters are added to mitigate the overfitting phenomenon [43].

3.3 Label Propagation

In addressing the challenge of highly unbalanced datasets, this study employs label propagation based on the threshold screening for data augmentation. Label propagation [18] occurs as category labels are transmitted from labeled to unlabeled data via these connecting edges. Typically, label propagation is more efficient between similar vertices, as their probability distributions tend to be closely aligned. Consequently, the resulting categorization forms a distribution that is not constrained to a specific shape, thereby more accurately reflecting the true data distribution. The process of label propagation is executed on a weighted, undirected relational graph [23]. The weights on the graph's edges intuitively represent the similarity between samples, facilitating the analysis of label propagation intensity along these edges. Therefore, the preliminary step in label propagation is constructing a graph where labeled and unlabeled samples are interconnected through undirected edges.

Consider constructing a labeling matrix to represent the labeling changes during the propagation process. Assuming that the sample set has a total of c category labels and l labeled samples, therefore, this paper defines a $l \times c$ labeling matrix Y_L , with the i th row denoting the labeling probability composition of the i th sample. Given an unlabeled sample u , a labeling matrix Y_U of dimensions $u \times c$ is defined. So the total training

sample set can be represented by a labeling matrix $Y^T = (Y_L, Y_U)$ of $(l + u) \times c$ where the conceptual composition of the sample x_i $a_i = (a_{i1}, a_{i2}, \dots, a_{ij})$, a_{ij} denotes the j th labeled conceptual part of the x_i sample l_j . Assume that $a_{ij} > 0$ and that $a_i^T = 1$ for each sample x_i . Accordingly, the labeling matrix Y is constructed as shown in the following equation:

$$\begin{array}{ccc}
 \alpha^{(1)} & \alpha^{(j)} & \alpha^{(c)} \\
 \downarrow & \downarrow & \downarrow \\
 \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1j} & \cdots & \alpha_{1c} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{i1} & \cdots & \alpha_{ij} & \cdots & \alpha_{ic} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{n1} & \cdots & \alpha_{nj} & \cdots & \alpha_{nc} \end{bmatrix} & = & \begin{bmatrix} \alpha_1^T \\ \vdots \\ \alpha_i^T \\ \vdots \\ \alpha_n^T \end{bmatrix} \begin{array}{l} \leftarrow x_1 \\ \vdots \\ \leftarrow x_i \\ \vdots \\ \leftarrow x_n \end{array}
 \end{array} \quad (15)$$

In the initialization of matrix Y , the initial value of the unlabeled data labels in this paper is taken as 0. The formula for generating the matrix Y is shown in the following equation:

$$a_{ij} = \begin{cases} \frac{1}{|Y_i|}, l_j \in Y_i \\ 0, otherwise \end{cases} \quad (16)$$

During label propagation, the label matrix should preserve the samples' label probability distribution. For test samples, the labels with the highest probability values are selected as the predicted labels. With one label propagation, the category labels will be propagated from labeled data to unlabeled data, and if the distribution of the training data satisfies the real data distribution, then the unlabeled data will play a role in helping to improve learning.

4 Experiments

4.1 Data Preprocessing

In this paper, we examine the data from three datasets provided by the Mobile Innovation Institute, identifying a significant imbalance among them. Normal websites account for the vast majority of the data set, the total number of data for other labels does not add up to more than 30,000, and the amount of data for different labels also varies greatly, with some exceeding 10,000 and others not even exceeding ten. To mitigate this imbalance, we initially employs a semi-supervised learning model to augment categories with fewer data points. Subsequently, we apply data augmentation techniques to further rectify the data imbalance, resulting in a more evenly distributed dataset. The final dataset division allocates 80% for training and 20% for testing. Following the labeling and classification of these datasets, we present a comprehensive data overview in Table 1:

To verify that our dataset represents a real problem rather than an artificial construction of multi-categorical and unbalanced data, we compared LGNet with several

Table 1. Data tagging and classification about URLs. Types represent the categorization of URLs, including normal URLs and different types of Malicious URLs, and Numbers represent the number of different types of URLs in the dataset.

Labels	Types	Numbers
1	Normal	8509522
2	Shopping	22
3	Dating	17780
4	Counterfeit Identity	173
5	Phishing	4658
6	Impersonation of Lawyers	2
7	Platform Fraud	2603
8	Recruitment	11
9	Killing Plate	883
10	Betting and Gambling	595
11	Credit management	998
12	Swipe fraud	549
13	Lottery Scam	7

Table 2. Benchmark comparison of different Malicious URL detection methods, deeper color is the best result.

Model	Accuracy \uparrow	F1 \uparrow	AUC \uparrow	Score \uparrow
Phishnet [32]	0.453	0.456	0.502	0.513
MPURNN [9]	0.836	0.825	0.848	0.857
URLNet [24]	0.896	0.899	0.945	0.954
DNN-LSTM [31]	0.908	0.914	0.953	0.964
Aljofey et al. [5]	0.911	0.913	0.953	0.967
Web2Vec [16]	0.925	0.925	0.959	0.975
LGNet	0.974	0.965	0.984	0.988

other methods. These include a traditional phishing URL detection method, Phishnet [32], and five machine learning methods: MPURNN [9], URLNet [24], DNN-LSTM [31], Aljofey et al. [5], and Web2Vec [16]. The comparisons were conducted on our dataset and two traditional phishing URL detection datasets, ISCX-URL2016 [27] and the Phishtank data collected by Yasin et al. [42]. All experiments were performed using an NVIDIA GeForce GTX 1650 GPU.

4.2 Experimental Results

To validate the effectiveness of our method, we compare it with several state-of-the-art methods for detecting Malicious URLs, including a traditional software URL classifier

Table 3. Comparison of F1 and scores of LGNet with the next best scoring state-of-the-art model under the Specific Fishing URL tab. Deeper colors represent better results.

Types	F1↑		Score↑	
	Web2Vec	LGNet	Web2Vec	LGNet
Normal	0.96	0.96	0.98	0.99
Shopping	0.88	0.92	0.93	0.97
Dating	0.87	0.92	0.91	0.96
Counterfeit Identity	0.97	0.98	0.98	0.99
Phishing	0.89	0.94	0.92	0.99
Impersonation of Lawyers	0.96	0.96	0.98	0.99
Platform Fraud	0.85	0.91	0.93	0.98
Recruitment	0.94	0.97	0.97	0.99
Killing Plate	0.93	0.95	0.95	0.99
Betting and Gambling	0.97	0.96	0.99	0.99
Credit management	0.92	0.95	0.96	0.98
Swipe fraud	0.96	0.96	0.98	0.99
Lottery Scam	0.95	0.97	0.97	0.98

Phishnet [32], and five machine learning methods, including MPURNN [9], URLNet [24], DNN-LSTM [31], Aljofey et al. [5], and Web2Vec [16]. We used our dataset to retrain these Sota models. There are four metrics to compare the strengths and weaknesses of the fishing URL prediction models, including accuracy, F1, AUC, and total score. The accuracy is the rate at which the model correctly predicts Malicious URLs. The F1 score is the harmonic mean of the check accuracy and recall rates, and the AUC value is the sum of the areas under the ROC curve. The final score is a suitable performance metric based on the area under the precision-recall curve (P-R curve). In the paper, we define the score as formula 17. There is a trade-off between recall and accuracy, where increasing recall usually leads to decreasing accuracy. The performance of the model at different levels of recall is reflected by a weighted average of the accuracy of the three working points 0.7, 0.8, and 0.9. The formula 17 indicates that we pay more attention to points with low recall, where a higher score represents capturing more Malicious URLs. By prioritizing low recall, we aim to minimize underreporting and mitigate potential risks. Higher values of accuracy, F1, AUC, and total score prove that the model predicts Malicious URLs with better quality.

$$Score = 0.5 \times P_{R=0.7} + 0.3 \times P_{R=0.8} + 0.2 \times P_{R=0.9} \quad (17)$$

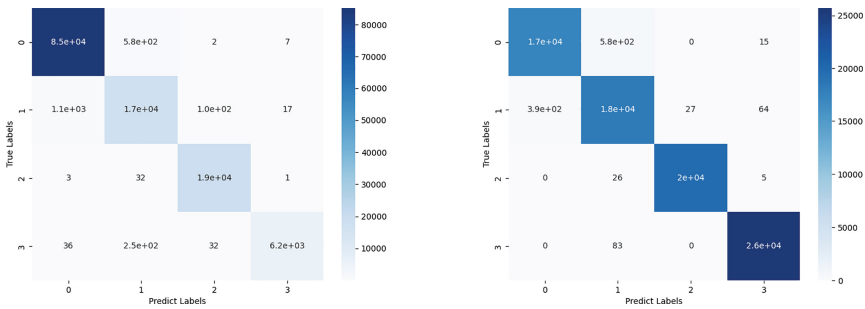
Table 2 compares the numerical results of our LGNet with the state-of-the-art method. As illustrated in Table 2, our LGNet searches through a large number of URLs to predict Malicious URLs with multi-labeling and extremely unbalanced sample data significantly outperforms other methods in all four metrics. The comparison results demonstrate that our LGNet exhibits significantly higher accuracy in detecting malicious URLs on real networks.

Table 4. Accuracy detection results of different malicious URL detection models on classical datasets.

Model	ISCX-URL2016	Phishtank
Phishnet [32]	0.746	0.738
MPURNN [9]	0.947	0.942
URLNet [24]	0.913	0.887
DNN-LSTM [31]	0.914	0.892
Aljofey et al. [5]	0.984	0.921
Web2Vec [16]	0.975	0.925
LGNet	0.981	0.917

In the Malicious URL detection process with multiple labels and extremely unbalanced data, the overall detection accuracy may not represent the Malicious URL detection accuracy for each specific label. To verify the superiority of our LGNet in actual detection processes, we compared it with Web2Vec, the second-best SOTA model in overall score comparisons. We subdivided and analyzed their F1 scores and specific scores for each type of tag. Table 3 shows the results of the comparison between LGNet and Web2Vec in each type of tag. It can be seen that our LGNet is superior to Web2Vec in detecting Malicious URLs for most specific tags. In particular, the detection scores for each specific type of tag exceed 0.97, indicating that our LGNet has a superior performance in Malicious URL detection.

To demonstrate that our model’s superior detection results stem from LGNet’s effectiveness in feature extraction and handling multiple categories and imbalances in malicious URLs, we applied a confusion matrix heat map to compare its detection results with Web2Vec, as shown in Fig. 2. Figure 2 shows that our approach focuses on each



(a). Confusion matrix heat map of feature extraction model detection results for Web2Vec processing of multi categorical unbalanced datasets

(b). Confusion matrix heat map of feature extraction model detection results for LGNet processing of multi categorical unbalanced datasets

Fig. 2. Confusion heat matrix plot. Each cell on the diagonal of the confusion heat matrix indicates the number of samples correctly classified by the model. Brightly colored regions indicate correct classification, and the shade of the color reflects the accuracy of the classification.

category in the unbalanced dataset simultaneously, which improves the accuracy of multi-category identification. Also, from the color or value of the non-diagonal region, it can be seen that our method has a lower color in this region, and more intuitively, the value of this region is smaller in the figure, indicating that our method has a lower misclassification rate.

To ensure that LGNet’s superior performance is not confined to the multi-labeled and highly unbalanced datasets we used, we compared it with the current SOTA method on a widely-used phishing website detection dataset. The results, as shown in Table 4, show that our model achieves comparable results with the current SOTA method on the frequently used phishing website detection dataset, and even better results on some datasets. We analyze the results and find that the better performance of LGNet is due to its superior classification ability, which allows it to still perform better for malicious URL detection in the general classification case.

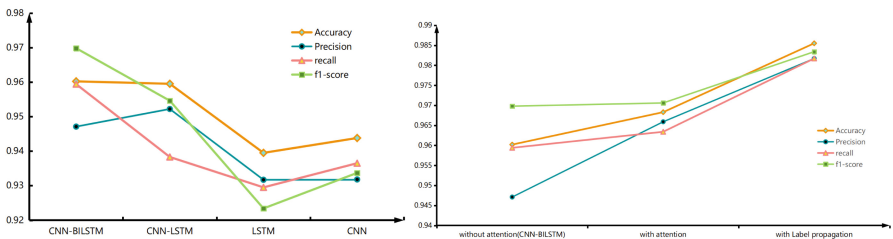
4.3 Ablation Experiments

To test the effectiveness of our LGNet in detecting malicious URLs, we validated it with ablation experiments. The experiment is conducted in two phases. In each phase, we validate and analyze the detection accuracy, F1 score, recall value, and precision value.

Phase I: In this phase, no attentional mechanism was used.

Phase II: In this phase, the attention mechanism is introduced and label propagation is included.

The results of the experiments are shown in Fig. 3. The CNN-BILSTM in the first phase outperforms the SOTA in accuracy, recall, and f1-score, and is only slightly lower than the CNN-LSTM in precision. In the second phase, by introducing the attention mechanism and label propagation, CNN-BILSTM attention with label propagation achieves SOTA in all four metrics.



(a). LGNet ablation experiments without attention mechanisms and label propagation (b). LGNet ablation experiments incorporating attention mechanisms and label propagation

Fig. 3. Ablation Experiments of LGNet

Ablation experiments in Fig. 3 validate the significance of LGNet’s components. In the absence of attention mechanisms and label propagation (Phase I), LGNet still

surpasses the state-of-the-art in some metrics. However, incorporating attention mechanisms and label propagation (Phase II) results in further improvement, achieving state-of-the-art performance in all evaluated metrics.

Overall, the study establishes LGNet as a robust and effective method for detecting malicious URLs, outperforming existing state-of-the-art approaches across multiple datasets and scenarios. The ablation experiments further underscore the importance of attention mechanisms and label propagation in enhancing the model's performance.

5 Conclusion

In this paper, we propose a neural network framework, LGNet improving the accurate identification of Malicious URLs in a large amount of actual unlabeled URL data with a highly unbalanced sample size. Our LGNet employs a label propagation algorithm, implying it can apply semi-supervised learning to obtain new data with high-confidence labels. We improve the scalable tree enhancement system to obtain a better prediction framework for Malicious URLs in the prediction. Experimental results show that our LGNet significantly outperforms other SOTA methods in prediction results.

References


1. Domain names - implementation and specification. RFC 1035 (1987). <https://doi.org/10.17487/RFC1035>, <https://www.rfc-editor.org/info/rfc1035>
2. Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: A comparison of machine learning techniques for phishing detection. In: Proceedings of the Anti-phishing Working Groups 2nd Annual eCrime Researchers Summit, pp. 60–69 (2007)
3. Al-Ahmadi, S., Alotaibi, A., Alsaleh, O.: PDGAN: phishing detection with generative adversarial networks. *IEEE Access* **10**, 42459–42468 (2022)
4. Alani, M.M., Tawfik, H.: PhishNot: a cloud-based machine-learning approach to phishing URL detection. *Comput. Netw.* **218**, 109407 (2022)
5. Aljofey, A., et al.: An effective detection approach for phishing websites using URL and html features. *Sci. Rep.* **12**(1), 8842 (2022)
6. Allodi, L., Chotza, T., Panina, E., Zannone, N.: The need for new antiphishing measures against spear-phishing attacks. *IEEE Secur. Priv.* **18**(2), 23–34 (2019)
7. Apruzzese, G., Colajanni, M., Ferretti, L., Marchetti, M.: Addressing adversarial attacks against security systems based on machine learning. In: 2019 11th International Conference on Cyber Conflict (CyCon), vol. 900, pp. 1–18. IEEE (2019)
8. Apruzzese, G., et al.: The role of machine learning in cybersecurity. *Digit. Threats Res. Pract.* **4**(1), 1–38 (2023)
9. Bahnsen, A.C., Bohorquez, E.C., Villegas, S., Vargas, J., González, F.A.: Classifying phishing URLs using recurrent neural networks. In: 2017 APWG Symposium on Electronic Crime Research (eCrime), pp. 1–8. IEEE (2017)
10. Bell, S., Komisarczuk, P.: An analysis of phishing blacklists: Google safe browsing, OpenPhish, and PhishTank. In: Proceedings of the Australasian Computer Science Week Multi-conference, pp. 1–11 (2020)
11. Buber, E., Diri, B., Sahingoz, O.K.: NLP based phishing attack detection from URLs. In: Intelligent Systems Design and Applications: 17th International Conference on Intelligent Systems Design and Applications (ISDA 2017) held in Delhi, India, 14–16 December 2017, pp. 608–618. Springer (2018)

12. Chapaneri, R., Shah, S.: Enhanced detection of imbalanced malicious network traffic with regularized generative adversarial networks. *J. Netw. Comput. Appl.* **202**, 103368 (2022)
13. Chen, D., Wawrzynski, P., Lv, Z.: Cyber security in smart cities: a review of deep learning-based applications and case studies. *Sustain. Cities Soc.* **66**, 102655 (2021)
14. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794 (2016)
15. Dou, Z., Khalil, I., Khreishah, A., Al-Fuqaha, A., Guizani, M.: Systematization of knowledge (SoK): a systematic review of software-based web phishing detection. *IEEE Commun. Surv. Tutor.* **19**(4), 2797–2819 (2017)
16. Feng, J., Zou, L., Ye, O., Han, J.: Web2vec: phishing webpage detection method based on multidimensional features driven by deep learning. *IEEE Access* **8**, 221214–221224 (2020)
17. Fujima, H., Takeuchi, K., Kumamoto, T.: Semantic analysis of phishing emails leading to ransomware with ChatGPT (2023)
18. Gregory, S.: Finding overlapping communities in networks by label propagation. *New J. Phys.* **12**(10), 103018 (2010)
19. Guo, M.H., et al.: Attention mechanisms in computer vision: a survey. *Comput. Vis. Media* **8**(3), 331–368 (2022)
20. Hajaj, C., Hason, N., Dvir, A.: Less is more: robust and novel features for malicious domain detection. *Electronics* **11**(6), 969 (2022)
21. He, S., Li, B., Peng, H., Xin, J., Zhang, E.: An effective cost-sensitive XGBoost method for malicious URLs detection in imbalanced dataset. *IEEE Access* **9**, 93089–93096 (2021)
22. Hnamte, V., Najar, A.A., Nhung-Nguyen, H., Hussain, J., Sugali, M.N.: DDoS attack detection and mitigation using deep neural network in SDN environment. *Comput. Secur.* **138**, 103661 (2024)
23. Jia, S., Deng, X., Xu, M., Zhou, J., Jia, X.: Superpixel-level weighted label propagation for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **58**(7), 5077–5091 (2020)
24. Le, H., Pham, Q., Sahoo, D., Hoi, S.C.: URLNet: learning a URL representation with deep learning for malicious URL detection. *arXiv preprint arXiv:1802.03162* (2018)
25. Li, J., Zhu, Q., Wu, Q., Fan, Z.: A novel oversampling technique for class-imbalanced learning based on smote and natural neighbors. *Inf. Sci.* **565**, 438–455 (2021)
26. Liang, Y., Wang, Q., Xiong, K., Zheng, X., Yu, Z., Zeng, D.: Robust detection of malicious URLs with self-paced wide & deep learning. *IEEE Trans. Dependable Secure Comput.* **19**(2), 717–730 (2021)
27. Mamun, M.S.I., Rathore, M.A., Lashkari, A.H., Stakhanova, N., Ghorbani, A.A.: Detecting malicious URLs using lexical analysis. In: *Network and System Security: 10th International Conference, NSS 2016, Taipei, Taiwan, September 28–30, 2016*, pp. 467–482. Springer (2016)
28. Marchal, S., François, J., State, R., Engel, T.: PhishStorm: detecting phishing with streaming analytics. *IEEE Trans. Netw. Serv. Manage.* **11**(4), 458–471 (2014)
29. Mourtaji, Y., Bouhorma, M., Alghazzawi, D., Aldabbagh, G., Alghamdi, A.: Hybrid rule-based solution for phishing URL detection using convolutional neural network. *Wirel. Commun. Mob. Comput.* **2021**, 1–24 (2021)
30. Mowbray, M., Hagen, J.: Finding domain-generation algorithms by looking at length distribution. In: *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, pp. 395–400. IEEE (2014)
31. Ozcan, A., Catal, C., Donmez, E., Senturk, B.: A hybrid DNN–LSTM model for detecting phishing URLs. *Neural Comput. Appl.* 1–17 (2021)
32. Prakash, P., Kumar, M., Kompella, R.R., Gupta, M.: PhishNet: predictive blacklisting to detect phishing attacks. In: *2010 Proceedings IEEE INFOCOM*, pp. 1–5. IEEE (2010)

33. Safi, A., Singh, S.: A systematic literature review on phishing website detection techniques. *J. King Saud Univ.-Comput. Inf. Sci.* (2023)
34. Salloum, S., Gaber, T., Vadera, S., Shaalan, K.: Phishing email detection using natural language processing techniques: a literature survey. *Procedia Comput. Sci.* **189**, 19–28 (2021)
35. Salloum, S., Gaber, T., Vadera, S., Shaalan, K.: A systematic literature review on phishing email detection using natural language processing techniques. *IEEE Access* **10**, 65703–65727 (2022)
36. 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. In: *Advances in Neural Information Processing Systems*, vol. 28 (2015)
37. Siami-Namini, S., Tavakoli, N., Namin, A.S.: The performance of LSTM and BiLSTM in forecasting time series. In: *2019 IEEE International Conference on Big Data (Big Data)*, pp. 3285–3292. *IEEE* (2019)
38. Tsai, Y.D., Liow, C., Siang, Y.S., Lin, S.D.: Toward more generalized malicious URL detection models. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 21628–21636 (2024)
39. Valentim, R., Drago, I., Trevisan, M., Cerutti, F., Mellia, M.: Augmenting phishing squatting detection with GANs. In: *Proceedings of the CoNEXT Student Workshop*, pp. 3–4 (2021)
40. Varshney, G., Kumawat, R., Varadharajan, V., Tupakula, U., Gupta, C.: Anti-phishing: a comprehensive perspective. *Expert Syst. Appl.* **238**, 122199 (2024)
41. Wang, Z., Ren, X., Li, S., Wang, B., Zhang, J., Yang, T.: A malicious URL detection model based on convolutional neural network. *Secur. Commun. Netw.* **2021**, 1–12 (2021)
42. Yasin, A., Fatima, R., Khan, J.A., Afzal, W.: Behind the bait: delving into PhishTank’s hidden data. *Data Brief* **52**, 109959 (2024)
43. Zhou, P., et al.: Attention-based bidirectional long short-term memory networks for relation classification. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 207–212 (2016)



Contour-Guided Context Learning for Scene Text Recognition

Wei-Chun Hsieh¹(✉), Gee-Sern Hsu¹, Jun-Yi Chen¹, Moi Hoon Yap²,
and Zi-Chun Chao¹

¹ National Taiwan University of Science and Technology, Taipei City, Taiwan
chrishsie84@gmail.com

² Manchester Metropolitan University, Manchester, UK

Abstract. We propose contour-guided context learning (CCL) for bilingual scene text recognition (STR). The CCL framework consists of three parts: Contour Guided Transformer (CGT), Contextual Learning Transformer (CLT) and Multimodal Transformer (MMT) for fusion. CGT embeds a CLIP image encoder and utilizes CLIP’s pre-training capabilities to capture contour features from input images, and CLT embeds a CLIP text encoder to correct contextual errors. The fusion network incorporates attention features extracted by Transformer to enhance text recognition performance. Unlike most STR methods that only target English, the proposed CCL is designed to handle both English and Chinese and can handle irregularly shaped scene text. We conduct a comprehensive evaluation on Chinese and English benchmark datasets to validate the performance of our approach against state-of-the-art methods.

Keywords: Scene Text Recognition · CLIP · Contour-guided transformer

1 Introduction

Scene Text Recognition (STR) has a broad scope of applications, including mobile scanning, surrounding understanding, assistive guidance, and others. It has received increasing attention in recent years. However, most STR approaches suffer from the following limitations among others. Firstly, most only work for English and unintentionally ignore other languages. Since Chinese is another popular language with complicated geometric shapes, we focus on both English and Chinese in our study. Secondly, most work for texts of regular shapes/qualities and often fail to handle texts of irregular shapes (rotated, curved, blurred, or occluded). We, therefore, propose the Contour-guided Context Learning (CCL) to address these issues. To avoid misunderstanding, the context of this paper means symbol-level context. The CLT model aims to refine the possible spelling error from CGT.

The proposed CCL is composed of two CLIP-embedded transformers, namely the Contour-guided transformer (CGT) and the Context-learning transformer

(CLT), and a fusion network. As CLIP shows remarkable zero-shot capacity across various vision-language tasks [25], the embeddings of the CLIP image encoder and text encoder offer effective coherence between images and texts. We train the CGT to extract the geometrical traits of the character contours and the attention features between images and texts, making it capable of handling texts of irregular shapes and poor image quality. Although Chinese characters appear more complicated than English characters in geometrical properties, the CGT is trained to handle both languages appropriately. On the other hand, the CLIP-embedded CLT explores the embeddings of the CLIP text encoder to correct occasional recognition errors in individual characters from the CGT. The fusion network fuses the attention features from the CGT and CLT to render the overall recognition output.

We use the Chinese Scene Text Competition (CSTC) dataset [5], which offers 17,943 text images to evaluate the performance in Chinese, and six English datasets, namely, IIIT5K (IIIT) [22], ICDAR2013 (IC13_s) [19], SVT [33], ICDAR2015 (IC15_s) [18], SVTP [24], and CUTE80 (CUTE) [26] to evaluate the performance in English. We compare our approach with other state-of-the-art methods on these benchmarks.

The contributions made by this work can be summarized as follows.

- The proposed Contour-guided Context Learning (CCL) is verified effective in handling bilingual STR, especially so for Chinese, as Chinese shows more complexity in contour than English.
- The strength of the CLIP pretrained image and text encoders is verified through the handling of the domain gap between synthetic data and real scene data of irregular shapes and various image qualities.
- The proposed approach outperforms other state-of-the-art methods for Chinese STR, and demonstrates competitive performance for English.

In the following, we will first present related work in Sect. 2, then our approach in Sect. 3, then the experiments in Sect. 4, and a conclusion in Sect. 5.

2 Related Work

Scene text recognition methods can be generally divided into two categories, language-free and language-based. The language-free methods extract visual characteristics and do not consider context relationships [1, 11, 15, 30]. The language-based methods consider the context information from language models to improve visual model prediction [2, 6, 9, 23, 29]. Since language-based methods consider context information to enhance performance, they often outperform the language-free and become an important family of approaches, such as ABINet [9], which uses a multi-modal fusion manner to leverage the vision and language model, MATRN [23] identifies visual and semantic feature pairs and encodes spatial information into semantic features, enables interactions between visual and semantic features for better recognition performances.

The proposed CCL is inspired by the work that embeds pretrained vision-language models, such as CLIP [25] and ALIGN [17], which demonstrate impressive generalization capabilities. VLMs that have been pre-trained on extensive collections of image-text pairs exhibit numerous intriguing characteristics [10, 25, 28]). Notably, certain neurons within CLIP demonstrate the ability to comprehend both visual and textual representations of the same concept. [10] identifies specific neurons within CLIP that react to both images of Spiderman and the textual input “spider”. Wherein VLMs prioritize textual content over the natural objects depicted in an image.

The proposed CCL belongs to the language-based family. We observed that state-of-the-art research seldom addresses bilingual STR, and we thus include traditional Chinese in our study. Intricate geometric shapes and a vast character set characterize traditional Chinese characters, posing threats to many approaches. To tackle this challenge, we refer to the MSTR [30] and CLIP [25] when developing the CCL framework. The MSTR employs a mask branch to enhance the resilience of complex and blurred images, while CLIP demonstrates exceptional generalization capabilities across diverse tasks, particularly in challenging STR tasks [18, 24, 26]. In this study, we leverage the mask branch and CLIP when developing the proposed approach.

3 Proposed Approach

The proposed CCL is composed of 3 parts: a Contour-guided transformer (CGT), a Context-learning transformer (CLT), and a Multi-Modal Transformer (MMT) for fusion. The configuration is shown in Fig. 1. CGT is a visual module, CLT is a language module, and MMT is a module that fuses visual and language module predictions. The overall workflow is that we first enter the scene text image I_s into CGT to make a visual-attention feature f_v and a visual model prediction \hat{p}_v (“信義吉局” in Fig. 1) as outputs. f_v contains the features of the contour mask m_i and the visual features I_f generated by the pretrained CLIP image encoder C_i . We then enter \hat{p}_v into CLT, which is composed of the pre-trained CLIP text encoder C_t and a transformer decoder L_d , to generate a language-attention feature f_l . MMT fuses the visual-attention feature f_v and language-attention feature f_l to produce the final prediction \hat{P}_f .

The training of the CCL is composed of two phases. In Phase 1, the whole system is trained on the synthetic data with contour mask m_i GT available, and we can train the segmentation model. In Phase 2, we train the whole system on the real data and will not update the segmentation model weight.

3.1 CLIP

CLIP, which aggregates 400 million image-text pairs without human annotation for model pretraining, has demonstrated significant potential in learning transferable knowledge and open-set visual concepts. During training, CLIP utilizes a contrastive loss to learn a joint embedding space for both modalities. For each

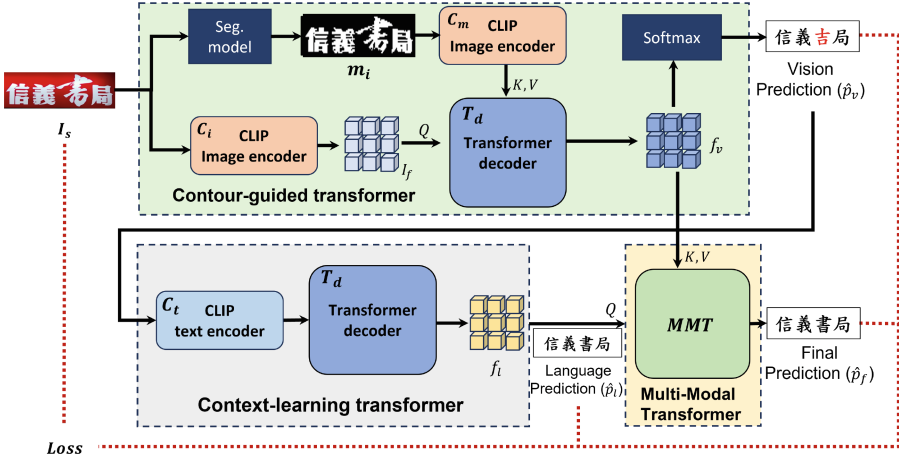


Fig. 1. CCL consists of three components: a contour-guided transformer (CGT) for handling visual aspects of contour guidance, a context-learning transformer (CLT) for context learning, and a multi-modal transformer for multi-modal fusion (MMT).

image in a batch of image-text pairs, CLIP maximizes the cosine similarity with the corresponding text while minimizing it with all other non-matching texts. Similarly, for each text, the loss is computed in the same manner as for each image. This process enables CLIP to perform zero-shot image recognition. In CLIP, the text and image features are aligned within a joint image-text embedding space. The text encoder in CLIP is a transformer encoder [7, 32], and the text tokenizer uses lower-cased byte pair encoding (BPE) [27] with a vocabulary size of 49,152.

The CLIP image encoder is a vision transformer (ViT) [8]. For an image, ViT employs a visual tokenizer (convolution) to convert non-overlapping image patches into a discrete sequence. The CLIP image encoder typically returns the feature of the [CLASS] token, but in this work, we return features of all tokens. These features are also normalized and linearly projected into the joint image-text embedding space. Generally, we use a ViT-B/16 (patch size 16×16) as the image encoder.

3.2 Contour-Guided Transformer

The Contour Guided Transformer (CGT) consists of a contour segmentation model, dual-clip pre-trained encoder and decoder structures. The contour segmentation model will transfer $I_s \in R^{512^2}$ to the contour mask $m_i \in R^{512^2}$. The model process first inputs I_s and m_i respectively into the dual-clip pre-trained encoder. The first pre-trained encoder is denoted as C_i , the input of C_i is I_s , and the output is I_s features I_f , and the second pre-trained encoder is denoted as C_m , the input of C_m is the contour mask m_i , and the output is the m_i feature (m_f). Since the contour mask feature m_f can capture the contour of each character on I_s , we propose to use the contour mask m_i to enhance the character

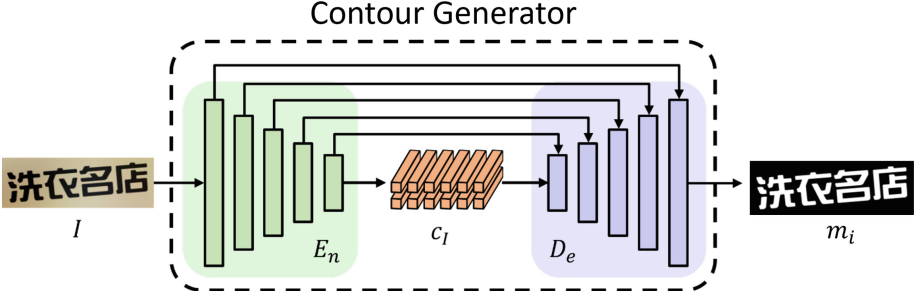


Fig. 2. Segmentation model structure.

features and the position accuracy of each character. To fuse character contour features m_f into image features (I_f), we design a contour-guided transformer decoder T_d . We use I_f as keys and values and m_f as query. The contour query cross-attention is calculated as follows:

$$\text{CrossAttention}(m_f, I_f, I_f) = \text{softmax}\left(\frac{m_f \cdot I_f^t}{d}\right) \cdot I_f \quad (1)$$

where d is the scale factor, given that the contour mask aims to capture the contour of each character, we use the contour mask feature m_f as a query to make the contour feature the image feature of interest.

The segmentation model S_m follows a U-Net architecture. The U-Net encoder, E_n , consists of 5 residual blocks that downsample the input image into feature c_i . The U-Net decoder, D_e , includes five convolutional layers that upsample the feature c_i , resulting in a segmentation mask m_i with dimensions [1, 32, 128]. During the upsampling process, skip connections are used to concatenate features from corresponding downsampling levels, preserving information extracted at various stages. The structure of the segmentation model is illustrated in the Fig. 2. The output of the segmentation mask m_i is a binarized image, where the white regions represent the character and the black regions represent the background.

The Transformer decoder consists of 3 blocks, each block consists of a self-attention layer, a cross-attention layer and an MLP. The loss functions used to train the segmentation model S_m are weighted binary cross-entropy \mathcal{L}_{wbce} , dice loss \mathcal{L}_d and Laplacian pyramid loss [4] \mathcal{L}_{lap} . Here is an explanation of the loss function we use in our segmentation model:

WBCE Loss is designed to minimize the discrepancy between the probability distributions of the predicted contour mask \hat{m}_i and those of the ground-truth contour mask m_i , respectively, computed across all pixels.

$$\mathcal{L}_{wbce} = - \sum_{i=1}^{N_p} [w_1 \cdot y_i \log(p_i) + w_2 \cdot (1 - y_i) \log(1 - p_i)] \quad (2)$$

where N_p represents the total number of pixels, w_1 and w_2 are weights to adjust the contributions of positive and negative samples to the loss, respectively. y_i is a binary indicator from m_i , and p_i is the predicted probability of \hat{m}_i .

Dice Loss [31] is often used in segmentation tasks to minimize the similarity between the predicted contour mask m_i and their ground-truth contour mask m_i .

$$\mathcal{L}_d = 1 - \frac{2|m_i \cap \hat{m}_i|}{|m_i| + |\hat{m}_i|} \quad (3)$$

Laplacian Pyramid Loss [4] aims to reduce the disparities in structural details between the generated contour mask and the ground truth. This loss enhances the clarity of edges and shape details in the generated masks, thereby improving the overall quality.

$$\mathcal{L}_{lap} = \sum_i^{N_l} 2^{i-1} \cdot \|L^i(m_i) - L^i(\hat{m}_i)\|^2 \quad (4)$$

N_l denotes the number of layers in the pyramids (we selected $N_l = 3$ from a comparison experiment). $L^i(\cdot)$ represents the i_{th} level Laplacian pyramids.

In Phase-1 training, we train the Contour-guided transformer on the synthetic data using the loss function $\lambda_{wbce}\mathcal{L}_{wbce} + \lambda_d\mathcal{L}_d + \lambda_{lap}\mathcal{L}_{lap}$, where λ_{wbce} , λ_d , and \mathcal{L}_{lap} are determined in the experiments.

3.3 Context-Learning Transformer and Fusion Model

The input to the context learning transformer CLT is the visual prediction of \hat{p}_v from CGT . The role of CLT is to refine the visual model prediction \hat{p}_v . For example, in Fig. 1, CGT predicts incorrectly in the third character “書” to the character “吉”, CLT uses contextual information and refines the error Forecasts from CGT .

The model workflow inputs the CGT prediction \hat{p}_v to the clip pre-trained text encoder, and outputs the text feature l_f . The next step is to feed (l_f) into the converter decoder. The structure is the same as CGT 's transformer decoder T_d , T_d is the language feature f_l , and the prediction result is \hat{p}_l .

After we get the feature f_v and f_l , we use a Multi-Modal Transformer (MMT) as a fusion model; our multi-modal transformer is comprised of a stack of 2 blocks, where each consists of a self-attention layer and an MLP. The visual and semantic features are first concatenated and processed through the self-attention layer. The MMT output is \hat{p}_v and \hat{p}_l fusion weight v_b , $v_b = \text{MMT}([f_v, f_l])$, v_b is a trainable weight to tune the relative importance between \hat{p}_v and \hat{p}_l by MMT, and we fuse as below.

$$\hat{p}_f = v_b \otimes \hat{p}_v + (1 - v_b) \otimes \hat{p}_l \quad (5)$$

As CGT 's prediction \hat{p}_v , CLT 's prediction \hat{p}_l and MMT's prediction \hat{p}_f can all be used to compute the cross entropy loss for text recognition with the ground-truth $x = [x_1, x_2, \dots, x_d]$ provided (where $d \leq d_l$ is the text length and $x_j \in$



Fig. 3. Samples from the CSTC dataset (“top half”) and our generated Chinese and English synthetic dataset with contour mask (“bottom half”).

R^{v_c} is the one-hot vector for each ground-truth character), v_c is the number of character classes, v_c is defined as 36 for English, and 765 for Chinese in this study, d_l is the maximum length of the character considered. We would compare the performance using these three feature codes in the experiments. In Phase-2 training on the real data, we consider the following total loss \mathcal{L} :

$$\mathcal{L} = \lambda_v \mathcal{L}_V + \lambda_l \mathcal{L}_l + \lambda_f \mathcal{L}_f \quad (6)$$

$$\text{where } \mathcal{L}_v = -\frac{1}{d} \sum_{j=1}^d x_j^t \log(\hat{p}_v(:, j)) \quad (7)$$

$\hat{p}_v(:, j)$ is the j th column vector of \hat{p}_v . \mathcal{L}_l and \mathcal{L}_f take the same form as (7) but with \hat{p}_v replaced by \hat{p}_l and \hat{p}_f , respectively. λ_v , λ_l , and λ_f are balanced factors, which are determined in the experiments.

4 Experiments

4.1 Datasets and Protocols

The proposed CCL requires two training phases, one depending on synthetic data where the masks are available and the other depending on real data for practical application. We refer to the SRNet [35] for the synthetic data generation. To generate Chinese synthetic data, we first crawled the web with arbitrary keywords, obtained 31k text lines, and selected the text lines with characters under 25. The text lines were used to make binary images with randomly selected font types. All binary text images were rotated in yaw, pitch, and roll to increase the orientation diversity and used as the contour character masks. We used the background images provided by [35] and weighted sum the background images and contour masks to form the synthetic data, ending up with 300k Chinese text images (CS). Similarly, we generated another 300k English text images (ES) using the text lines in the English corpus from the WikiText-103 [21]. Figure 3 shows some samples of our synthetic dataset in English and Chinese.

For handling English, we first trained the Contour-guided transformer CGT on the synthetic data, then trained the whole model on the MJSynth (MJ) [16] and SynthText (ST) [13] datasets, and then tested the model on the IIIT5K (IIIT) [22], ICDAR2013 (IC13_s) [19], SVT [33], ICDAR2015 (IC15_s) [18], SVTP [24], and CUTE80 (CUTE) [26]. The CSTC dataset also provides 17,108 English samples. We split it into a training set and testing set of 12,914 and 4,194 images, respectively, following the data splitting protocol used in [36]. We named the training set as $CSTC_{train}^{en}$ and testing set as $CSTC_{test}^{en}$.

For Chinese, we also first trained CGT and the whole system on the synthetic data, then the whole model on the CSTC training set ($CSTC_{train}^{ch}$), and then tested on the CSTC testing set ($CSTC_{test}^{ch}$). The CSTC dataset for Chinese STR offers 17,943 text images. We split it into a training set of 13,740 images and a testing set of 4,203 images with the same splitting rule used in English. Samples from the CSTC are shown in Fig. 3. In addition, we merged the training set and testing set of both languages in the CTSC dataset as $CSTC_{train}^{en-ch}$ and $CSTC_{test}^{en-ch}$ for bilingual experiments.

4.2 Ablation Study

The ablation study was conducted to compare the performance of different settings in our proposed CCL, including the contour mask m_i , the influences of different loss functions, and the influences of the CLIP (CLIP-ViT-B/16) pre-trained model. Table 1 shows the performance comparison on Chinese dataset $CSTC_{test}^{ch}$ dataset and on English benchmarks (SVTP), the performance is measured by word accuracy. Word accuracy indicates that each character in the predicted word needs to be the same as the GT word. For example, if the input image GT text is “style”, and the model predicts “style”, this case is correct; if the predict is “style”, it is incorrect. Table 1 reveal the following observations:

- The contour mask \mathcal{M}_i aims to obtain the character shape and reduce the background noise of the input image. And \mathcal{L}_{lap} , as one of the \mathcal{M}_i loss functions, can improve the edge and structural details of the generated \mathcal{M}_i . Experiments show that the contour mask \mathcal{L}_{lap} is an effective component of accuracy in the Chinese dataset $CSTC_{test}^{ch}$ and the English benchmark SVTP. The improvement is minor. The reason is that Chinese characters have complex geometric shapes. Many Chinese characters look similar but have different meanings. \mathcal{M}_i is an effective component to obtain the geometric shape of Chinese characters.
- \mathcal{L}_{con} is inspired by the idea of contrastive learning. We introduce the contrastive loss proposed by [39]. Unlike the English alphabet, which has only 26 letters, Chinese has thousands of characters, and different instances of the same character exhibit different appearances, such as different fonts, orientations, and other effects. Therefore, the model must learn each Chinese character in the sample, limited to English letters. Contrastive loss helps build an implicit and unified character-level representation for each character category during training. It can be seen that there is a slight performance improvement

on the English data set (SVTP), but there is a significant improvement on the Chinese data set ($\text{CSTC}_{\text{test}}^{\text{ch}}$). To validate the effectiveness of contrastive loss in learning character features, we perform dimensionality reduction on position-aware features and use t-SNE to visualize the distribution of different character features. Figure 4 shows the feature distribution of different Chinese characters.

- The CLIP (CLIP-ViT-B/16), we use the CLIP-ViT-B/16 pretrained model. The experiment shows that CLIP can significantly improve accuracy on both Chinese and English datasets. In Table 4, we observe that CLIP is good at irregular (rotated, curved, blurred) and non-frontal viewpoints image recognition; this is the reason why CLIP can improve on both Chinese and English datasets.
- The below experiment is to evidence the language model CLT can improve the overall prediction accuracy. Table 2 shows the performance evaluation on the Chinese dataset $\text{CSTC}_{\text{test}}^{\text{ch}}$ and the English dataset CVTP. \hat{p}_v is CGT prediction accuracy, \hat{p}_l is CLT accuracy, and \hat{p}_f is final accuracy. Table 2 shows that CLT can improve recognition accuracy in English and Chinese datasets.

Table 1. Ablation study of different components. **Boldface** shows the best performance. The baseline model, in addition to removing two loss functions. The wo/clip version replaces the Two CLIP image encoders with ResNet [14], and the one CLIP text encoder is converted to a regular transformer architecture. The absence of contour means the segmentation model is removed; originally, cross attention was performed between m_i and I_f , but now I_f performs self-attention.

Baseline	Contour(\mathcal{M}_i)	\mathcal{L}_{lap}	\mathcal{L}_{con}	CLIP	$\text{CSTC}_{\text{test}}^{\text{ch}}$	SVTP
✓	–	–	–	–	73.8	88.9
✓	✓	–	–	–	75.2	89.1
✓	✓	✓	–	–	77.2	89.2
✓	✓	✓	✓	–	79.4	89.9
✓	✓	✓	✓	✓	83.4	91.2

4.3 Comparison with Previous Methods

In order to clearly assess the performance of various languages, we divide the comparison into three distinct parts. The evaluation of Chinese language performance $\text{CSTC}_{\text{test}}^{\text{ch}}$ is presented in Table 3, while the benchmark performance for English is displayed in Table 4. Additionally, the performance on bilingual datasets can be found in Table 5. Below, we provide detailed observations for each table:

Table 2. Performance comparison of different module prediction.

Module	SVTP	CSTC _{test} ^{ch}
\hat{p}_v	87.4	77.6
\hat{p}_i	42.3	17.0
\hat{p}_f	91.2	83.4

- **Performance on the Chinese CSTC Dataset:** Table 3, Fig. 5 and Fig. 6 presents the performance comparison with other methods on the CSTC_{test}^{ch} dataset. All approaches followed the same protocol: training on synthetic data and then retraining on the Chinese CSTC training set (CSTC_{train}^{ch}). A fundamental disparity between Chinese and English lies in the nature of Chinese characters, symbolic representations with distinct meanings. Many Chinese characters exhibit visual similarities but convey different meanings, posing additional challenges for recognition. As demonstrated in the ablation study, the contour mask and \mathcal{L}_{lap} play crucial roles in extracting Chinese character contours, while \mathcal{L}_{con} effectively distinguishes between different character features. Hence, our CCL model outperforms other methods, showcasing significant improvement.
- **Performance on the English Benchmark Datasets:** Table 4 displays the performance of CCL on six English benchmark datasets compared to other methods trained on MJSynth and SynthText. We categorize the datasets into two types: regular and irregular. The regular type comprises IIIT5K, SVT, and IC13s, consisting primarily of data with frontal viewpoints and minimal orientation variations. In contrast, the irregular type includes SVTP, IC15s, and CUTE; this type encompasses non-frontal viewpoints, rotated, curved, blurred, or occluded data. The irregular type presents a more diverse distribution and poses significantly greater challenges for recognition. According to the experiments, our approach performs better than other methods on irregular data. This is mainly due to the CLIP model and \mathcal{L}_{con} , which improve the accuracy of complicated text. Additionally, our model has also achieved competitive results on regular data. These experimental results demonstrate that our model can robustly perform on both regular and irregular data. Since the CUTE, SVTP, and IC15 datasets are irregular datasets, our method outperforms the other methods on irregular datasets.
- **Performance on the Bilingual CSTC Dataset:** The performance comparison with other methods on the bilingual dataset (CSTC_{test}^{en-ch}) is depicted in Table 5. When assessing the recognition performance of the model on the bilingual dataset, we initially determine the accuracy of the model on the Chinese dataset (CSTC_{test}^{ch}) and the English dataset (CSTC_{test}^{en}) separately, and then compute their average. All methods followed the same protocol: training on synthetic datasets (ES+CS) and retraining on the CSTC_{train}^{en-ch} dataset. CCL demonstrates superior performance over others by a significant margin in most cases, underscoring the effectiveness of our approach. The

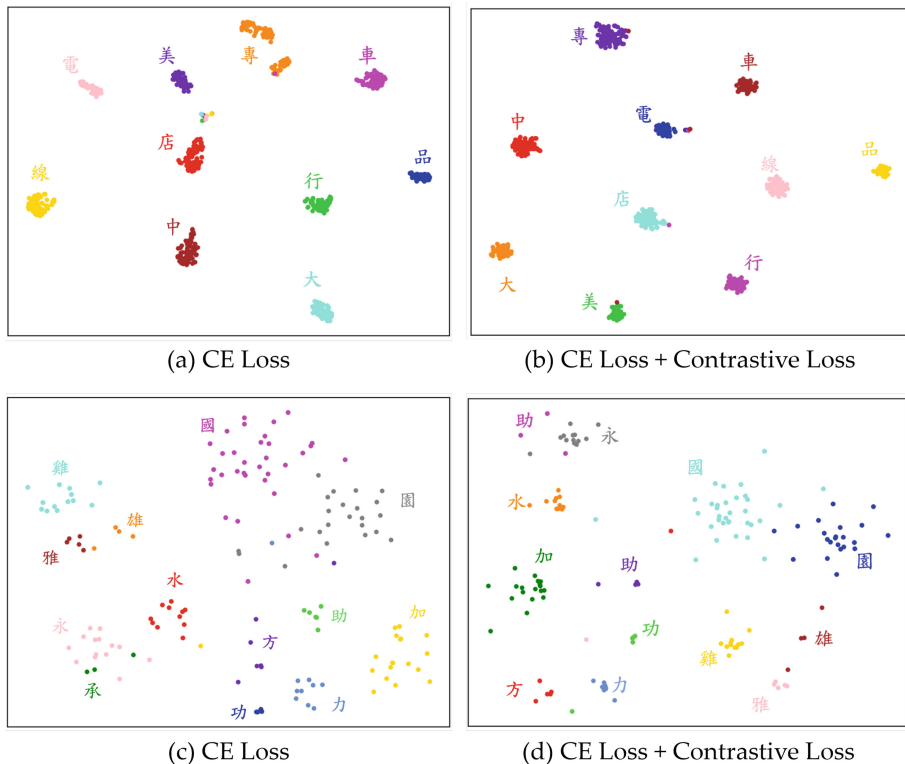


Fig. 4. Visualization for the character feature distribution. The second row demonstrates the feature distribution of visually alike characters.

main reason why CRNN [29], TRBA [2], and DAN [34] perform quite poorly is that they cannot recognize Chinese recognition; since the bilingual recognition output class includes Chinese characters and English letters, the CRNN [29], TRBA [2], and DAN [34] will recognize Chinese scene text to English. Therefore, they could improve in this bilingual situation.

Table 3. Performance comparison with other methods on $\text{CSTC}_{\text{test}}^{\text{ch}}$ dataset.

Methods	Train Data	$\text{CSTC}_{\text{test}}^{\text{ch}}$
CRNN [29]	CS + $\text{CSTC}_{\text{train}}^{\text{ch}}$	51.1
TRBA [2]	CS + $\text{CSTC}_{\text{train}}^{\text{ch}}$	69.7
DAN [34]	CS + $\text{CSTC}_{\text{train}}^{\text{ch}}$	60.7
ABINet [9]	CS + $\text{CSTC}_{\text{train}}^{\text{ch}}$	73.7
MATRN [23]	CS + $\text{CSTC}_{\text{train}}^{\text{ch}}$	74.8
LevOCR [6]	CS + $\text{CSTC}_{\text{train}}^{\text{ch}}$	70.9
CCL	CS + $\text{CSTC}_{\text{train}}^{\text{ch}}$	83.4

Table 4. Text recognition accuracy (%) comparison on different benchmarks. **Bold-face** shows the best performance and second best with underline.

Methods	Train Data	IIIT5K	SVT	IC13 _s	SVTP	IC15 _s	CUTE	Avg.
CRNN [29]	MJ + ST	78.2	80.9	89.4	70.0	69.4	65.5	75.5
TRBA [2]	MJ + ST	87.9	87.5	92.3	79.2	77.6	74.0	83.08
DAN [34]	MJ + ST	94.3	89.2	93.9	80.0	74.5	84.4	86.05
RobustScanner [38]	MJ + ST	95.3	88.1	94.8	79.5	77.1	90.3	87.51
SATRN [20]	MJ + ST	96.0	91.8	97.1	88.4	84.2	89.9	91.23
SRN [37]	MJ + ST	94.8	91.5	95.5	85.1	82.7	87.8	89.56
ABINet [9]	MJ + ST	96.2	93.5	97.4	89.3	86.0	89.2	91.93
MSTR [30]	MJ + ST	96.1	94.4	96.5	88.4	85.8	91.7	92.15
MATRN [23]	MJ + ST	96.6	<u>95.0</u>	97.9	<u>90.6</u>	<u>86.6</u>	<u>93.5</u>	93.36
LevOCR [6]	MJ + ST	96.6	92.9	96.9	88.1	86.4	91.7	92.1
PARSeq _A [3]	MJ + ST	<u>97.0</u>	93.6	96.2	88.9	86.5	92.2	92.4
SIGAT [12]	MJ + ST	96.6	95.1	96.8	90.5	<u>86.6</u>	93.1	93.1
CCL	MJ + ST	97.5	95.1	<u>97.3</u>	91.2	87.6	95.5	94.03

Table 5. Performance comparison with other methods on bilingual CSTC dataset ($\text{CSTC}_{\text{test}}^{\text{en-ch}}$).

Methods	Train Data	$\text{CSTC}_{\text{test}}^{\text{en-ch}}$
CRNN [29]	ES + CS + $\text{CSTC}_{\text{train}}^{\text{en-ch}}$	18.2
TRBA [2]	ES + CS + $\text{CSTC}_{\text{train}}^{\text{en-ch}}$	26.8
DAN [34]	ES + CS + $\text{CSTC}_{\text{train}}^{\text{en-ch}}$	32.1
ABINet [9]	ES + CS + $\text{CSTC}_{\text{train}}^{\text{en-ch}}$	79.2
MATRN [23]	ES + CS + $\text{CSTC}_{\text{train}}^{\text{en-ch}}$	80.9
LevOCR [6]	ES + CS + $\text{CSTC}_{\text{train}}^{\text{en-ch}}$	43.4
CCL	ES + CS + $\text{CSTC}_{\text{train}}^{\text{en-ch}}$	82.2

Image (I)	Contour Mask (m)	GT	ABINet	CCL(Ours)
		接到好	的里	接到好
		前台大專科醫院	前台大喜禮醫院	前台大專科醫院
		光復店	光夜店	光復店
		台南公園店	台灣公選店	台南公園店

Fig. 5. Prediction comparison on the blurred samples.

Image (I)	Contour Mask (m_I)	GT	ABINet	CCL(Ours)
		素食口酥	素食口味	素食口酥
		前改道藝文一街	附設造藝文一街	前改道藝文一街
		古屋滷滷	古屋滷肉	古屋滷滷
		正森堂金香舖	正春堂金香舖	正森堂金香舖

Fig. 6. Prediction comparison on the vertical samples.

5 Conclusion

Our innovation, Contour-guided Context Learning (CCL), revolutionizes bilingual scene text recognition (STR). At its core, CCL combines two CLIP-embedded transformers with a fusion network. These transformers leverage CLIP’s pretrained prowess to capture contour features and rectify context errors within input images. Through merging attention features extracted by the transformers, the fusion network significantly boosts text recognition performance. Unlike conventional STR methods that focus solely on English, CCL is designed to tackle both English and Chinese texts, including those with irregular shapes. Our comprehensive evaluations on Chinese and English benchmark datasets underscore the superiority of our approach over existing state-of-the-art methods.

References

- Atienza, R.: Vision transformer for fast and efficient scene text recognition. In: International Conference on Document Analysis and Recognition, pp. 319–334. Springer (2021)
- Baek, J., et al.: What is wrong with scene text recognition model comparisons? Dataset and model analysis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4715–4723 (2019)
- Bautista, D., Atienza, R.: Scene text recognition with permuted autoregressive sequence models. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13688, pp. 178–196. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19815-1_11
- Bojanowski, P., Joulin, A., Lopez-Paz, D., Szlam, A.: Optimizing the latent space of generative networks (2019). [arXiv:1707.05776](https://arxiv.org/abs/1707.05776)
- Cup, A.: Moe AI competition and labeled data acquisition project. <https://www.aicup.tw/>
- Da, C., Wang, P., Yao, C.: Levenshtein OCR. In: ECCV 2022, Part XXVIII, pp. 322–338. Springer (2022)
- Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019). <https://doi.org/10.18653/v1/n19-1423>

8. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. In: ICLR (2021)
9. Fang, S., Xie, H., Wang, Y., Mao, Z., Zhang, Y.: Read like humans: autonomous, bidirectional and iterative language modeling for scene text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7098–7107 (2021)
10. Goh, G., et al.: Multimodal neurons in artificial neural networks. Distill (2021). <https://doi.org/10.23915/distill.00030>
11. Graves, A., Fernández, S., Gomez, F.J., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: ICML (2006)
12. Guan, T., et al.: Self-supervised implicit glyph attention for text recognition. In: CVPR (2023)
13. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2315–2324 (2016)
14. 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)
15. He, P., Huang, W., Qiao, Y., Loy, C.C., Tang, X.: Reading scene text in deep convolutional sequences. In: AAAI (2016)
16. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. In: Workshop on Deep Learning, NIPS (2014)
17. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML (2021)
18. Karatzas, D., et al.: ICDAR 2015 competition on robust reading. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1156–1160. IEEE (2015)
19. Karatzas, D., et al.: ICDAR 2013 robust reading competition. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1484–1493. IEEE (2013)
20. Lee, J., Park, S., Baek, J., Oh, S.J., Kim, S., Lee, H.: On recognizing texts of arbitrary shapes with 2D self-attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 546–547 (2020)
21. Merity, S., Xiong, C., Bradbury, J., Socher, R.: Pointer sentinel mixture models. arXiv preprint [arXiv:1609.07843](https://arxiv.org/abs/1609.07843) (2016)
22. Mishra, A., Alahari, K., Jawahar, C.: Scene text recognition using higher order language priors. In: BMVC-British machine vision conference. BMVA (2012)
23. Na, B., Kim, Y., Park, S.: Multi-modal text recognition networks: Interactive enhancements between visual and semantic features. In: ECCV 2022, Part XXVIII, pp. 446–463. Springer (2022)
24. Phan, T.Q., Shivakumara, P., Tian, S., Tan, C.L.: Recognizing text with perspective distortion in natural scenes. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 569–576 (2013)
25. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
26. Risnumawan, A., Shivakumara, P., Chan, C.S., Tan, C.L.: A robust arbitrary text detection system for natural scene images. Expert Syst. Appl. **41**(18), 8027–8048 (2014)

27. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: ACL (2016). <https://doi.org/10.18653/v1/p16-1162>
28. Shen, S., et al.: How much can CLIP benefit vision-and-language tasks? In: ICLR (2022)
29. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(11), 2298–2304 (2016)
30. Shi, H., Peng, L., Yan, R., Yao, G., Han, S., Wang, S.: Mask scene text recognizer. In: International Conference on Document Analysis and Recognition, pp. 33–48. Springer (2021)
31. Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Cardoso, M.J.: Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *CoRR arXiv:1707.03237* (2017)
32. Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)
33. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 International Conference on Computer Vision, pp. 1457–1464. IEEE (2011)
34. Wang, T., et al.: Decoupled attention network for text recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 12216–12224 (2020)
35. Wu, L., et al.: Editing text in the wild. In: Proceedings of the 27th ACM International Conference on Multimedia, pp. 1500–1508 (2019)
36. Xu, X., Zhang, Z., Wang, Z., Price, B., Wang, Z., Shi, H.: Rethinking text segmentation: a novel dataset and a text-specific refinement approach. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12045–12055 (2021)
37. Yu, D., Li, X., Zhang, C., Liu, T., Han, J., Liu, J., Ding, E.: Towards accurate scene text recognition with semantic reasoning networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12113–12122 (2020)
38. Yue, X., Kuang, Z., Lin, C., Sun, H., Zhang, W.: RobustScanner: dynamically enhancing positional clues for robust text recognition. In: European Conference on Computer Vision, pp. 135–151. Springer (2020)
39. Zhang, X., Zhu, B., Yao, X., Sun, Q., Li, R., Yu, B.: Context-based contrastive learning for scene text recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 3353–3361 (2022)



A New HourGlass Network for Detecting Text in Shaky and Non-shaky Video Frames

Arnab Halder^{1,3}, Shivakumara Palaiahnakote^{2(✉)}, Umapada Pal¹,
Michael Blumenstein³, and Shivanand S. Gornale⁴

¹ Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, Kolkata, India

umapada@isical.ac.in

² School of Science, Engineering and Environment, University of Salford, Manchester, UK
s.palaiahnakote@salford.ac.uk

³ University of Technology Sydney, Sydney, Australia
michael.blumenstein@uts.edu.au

⁴ Department of Computer Science, Rani Channamma University, Belagavi, India
shivanand1971@rcub.ac.in

Abstract. Shaky and non-shaky videos are quite common in real-time applications such as surveillance and monitoring vehicles and human movements in protected areas. As a result, text detection in such videos is a formidable challenge due to motion blur, noise, shaky cameras, poor quality and poor visibility. In contrast to existing text detection methods, which focus on text detection in scene images or specific types of images, the present work focuses on text detection in both shaky and non-shaky video frames. Inspired by the impressive performance of the HourGlass network for adverse situations, we explore the HourGlass network for successful text detection in shaky, non-shaky video frames and natural scene images. To improve the performance of the HourGlass network, we employ the Real-Time Model (RTMHead) for predicting text precisely and the Cross Stage Partial Network (CSPNet), which is a neck architecture for robust feature fusion. In addition, the integration of the SiLU activation function with the HourGlass network improves the discriminative power ability. To test the efficacy of the proposed method, we conducted experiments on shaky and non-shaky video frames, as well as ICDAR 2015 video frames. Furthermore, to show the effectiveness of the proposed method, we used Total-Text scene images for experimentation. The results on different datasets and a comparative study with the state-of-the-art models show that the proposed model outperforms the existing methods.

Keywords: Text Detection · Video Text Detection · Arbitrary Moving Text · CSPNeXt · Hourglass · SiLU Activation

1 Introduction

Text detection and recognition approaches in natural scene images have made significant progress, but there is a gap when we look at real-time applications, particularly in dynamic environments like day and night surveillance (Halder et al. 2023; Halder

et al. 2024). To protect sensitive areas in all conditions (day and night) from robbery and theft, vehicle and human movements are automatically detected, which plays a vital role. However, the adverse effects of indoor and outdoor environmental factors, including poor visibility and movements of tree leaves and branches, make the problem more complex for detecting text in shaky and non-shaky video frames (Asadzadehkaljahi et al. 2023a, b).

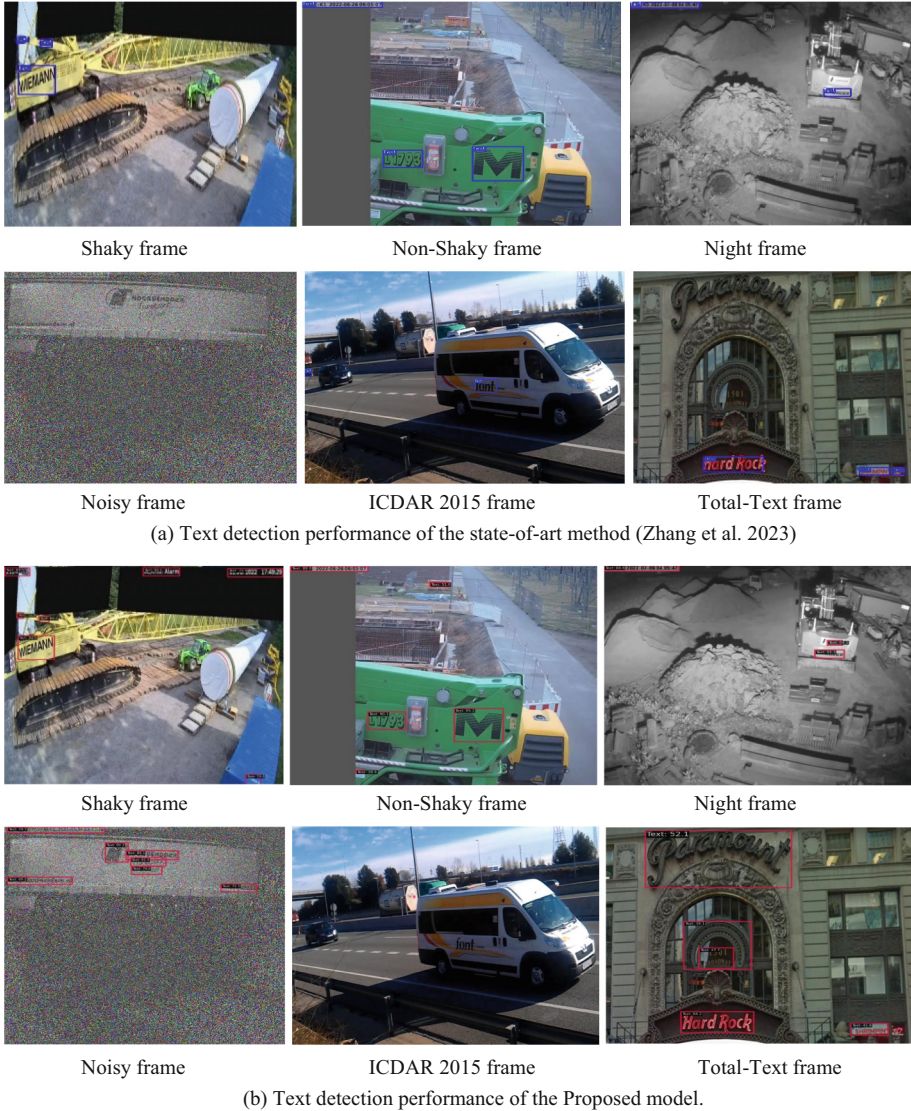


Fig. 1. Challenges for text detection in shaky and non-shaky video frames.

The literature (Yin et al. 2016; Zhang et al. 2023; Zhu et al. 2022) discusses robust and sophisticated methods that investigate deep learning models for text detection. These methods addressed challenges like arbitrary orientation, shaped, and complex backgrounds, but they are not robust to images affected by adverse effects of night, such as night-blurred and night-noisy images, as demonstrated in samples in Fig. 1. Therefore, the state-of-the-art models are not effective for shaky and non-shaky videos. This is due to limitations of the existing methods, which were developed for normal natural scene images.

The examples in Figs. 1(a) and 1(b) demonstrate that accurate text recognition in nighttime, shaky, and non-shaky video frames is not possible using the state-of-the-art approach (Zhang et al. 2023), which investigates deep learning for random scene text detection. On the other hand, text is well detected using the same way for photographs of natural scenes. Conversely, the suggested approach correctly identifies text in every image across several domains. We developed a novel model for text identification in natural scene video frames and photos that were taken by both shaky and non-shaky cameras as a result of this finding. In this work, we explore the HourGlass network by modifying it with a new RTMHead bounding box prediction, CSPNet for feature extraction, and a new SiLU activation function, motivated by the HourGlass network’s success for object detection in complex situations (Banerjee et al. 2022).

Thus, the way the proposed work adapts the existing models to solve the complex problem is the key challenge of our work. For example, HourGlassNetwork has been used to extract multi-level features. However, this work replaces the neighbor interpolation, which was used in the existing HourGlass network, with deconvolution layers for better feature extraction. Therefore, overall, the key contribution lies in the novel integration of the strengths of different models as a new model for addressing a complex problem of text detection in shaky and non-shaky video frames. Thus, the key contributions are listed here. (i) Exploring the HourGlass network for text detection in natural scene frames/images and shaky and non-shaky video frames. (ii) Proposing a new head function and new architecture for feature extraction from images of different domains. (iii) The way the proposed work fuses the strengths of different components to achieve the best results for text detection in shaky and non-shaky video frames.

The content of the rest of the paper is organized as follows. The methods related to text detection in natural scene images and video frames are reviewed in Sect. 2. Section 3 introduces the architecture of the HourGlass network with modified head and feature extraction modules. The experimental results and analysis are discussed in Sect. 4. Conclusions and future directions are presented in Sect. 5.

2 Related Work

Since text detection in natural scene images is not new work, we can find several methods in the literature. We review the latest methods in two main categories: text detection from images and text detection from videos.

2.1 Text Detection in Scene Images

A Differentiable Binarization Module Network (DBNet) was proposed by Liao et al. (2020) for text identification in photos of natural scenes. DBNet creates adaptive thresholds to optimize network performance. Similarly, Naiemi et al. (2021) presented a multi-oriented scene text localization (MOSTL) method, which incorporates an enhanced ReLU layer (i.ReLU) and an improved inception layer (i.inception). Another notable approach in this category is the work by Xu et al. (2020), who introduced LayoutLM. This transformer-based model uses BERT to capture layout and text information in documents. Facebook AI (2020) developed Rosetta, which is designed to handle text in photos and videos across different languages. Lu et al. (2022) introduced a system with four components—feature extraction, boundary refinement, boundary prediction, and text recognition modules—to enhance their model’s performance. Zhu et al. (2022) proposed using ResNet series networks for feature extraction in text detection models, introducing a feature redistribution module to maximize multi-level features. Shikha et al. (2023) incorporated Darknet53 and pre-trained YOLOv4 weights for Kannada text detection in images, highlighting the potential of deep learning techniques.

2.2 Text Detection in Video Frames

In the domain of video text detection, Halder et al. (2023) proposed a transformer-based text detection module for low-light video frames, combining similarity detection and detection modules for optimal performance. Bennet et al. (2022) presented a deep learning-based model for Telugu word recognition in videos, emphasizing the importance of language-specific information. Nandanwar et al. (2022) integrated deep learning with wavefront modeling for text detection in 3D videos, addressing inherent challenges through a multidisciplinary approach. Chen et al. (2021) proposed text detection in videos using parametric shape regression and fusion techniques, highlighting the interconnectivity of intra-frame and inter-frame data. Although limited in low-contrast scenarios, Chaitra et al. (2022) combined Yolov5 and TesseractOCR for text detection in video frames. Wang et al. (2019) explored progressive scale expansion networks for scene text detection, showcasing advancements in feature extraction for challenging scenarios. An arbitrary-shaped scene text identification system called CT-Net was presented by Shao et al. It makes use of progressive contour regression using contour transformers. Adaptive refinement and a rescore mechanism are used in this technique to improve text identification accuracy, therefore addressing problems like improper contour initialization and multi-stage error accumulation. Banerjee et al. (2022) introduced an end-to-end technique for text watermark detection in videos that aims to mitigate the impacts of poor contrast and complex backgrounds. The lack of robustness in the approaches now in use when applied to films with shaky or non-shaky material was observed by Halder et al. (2024).

In conclusion, while the current techniques handle a number of text identification issues in photos and video frames of natural scenes, they are unable to address the issues of text detection in daytime and nighttime video frames recorded by shaky or non-shaky cameras. Furthermore, none of the models investigated the Hourglass network for word recognition in photos of natural scenes or video frames that are unsteady or not. Thus,

this study aims to identify text in natural scene pictures and video frames, as well as in shaky and non-shaky video frames.

3 Proposed Methodology

It is noted from the previous section that there are open challenges for accurate text detection in shaky, non-shaky day and night video frames. Motivated by the special ability of the HourGlass network, which was successfully implemented for object detection in adverse situations (Banerjee et al. 2022), we explore the HourGlass network with the following modifications. A new head function and architecture for feature fusion are integrated with the baseline HourGlass network. The complete architecture of the Hourglass can be seen in Fig. 2. Since the scope of the method is limited to text detection in video frames, we use the method called activation frame selection, presented in (Asadzadehkaljah et al., 2023) for keyframe selection from the input video. This approach estimates the Structural Similarity Index (SSIM) using luminance, contrast, and structure information to select key video frames.

The proposed HourGlass network comprises the Real-Time Model (RTMHead) for predicting text precisely and the Cross Stage Partial Network (CSPNet), which is a neck architecture for robust feature fusion. Further, the SiLu activation function has been integrated with the HourGlass network rather than the ReLu activation function to improve discriminative power ability.

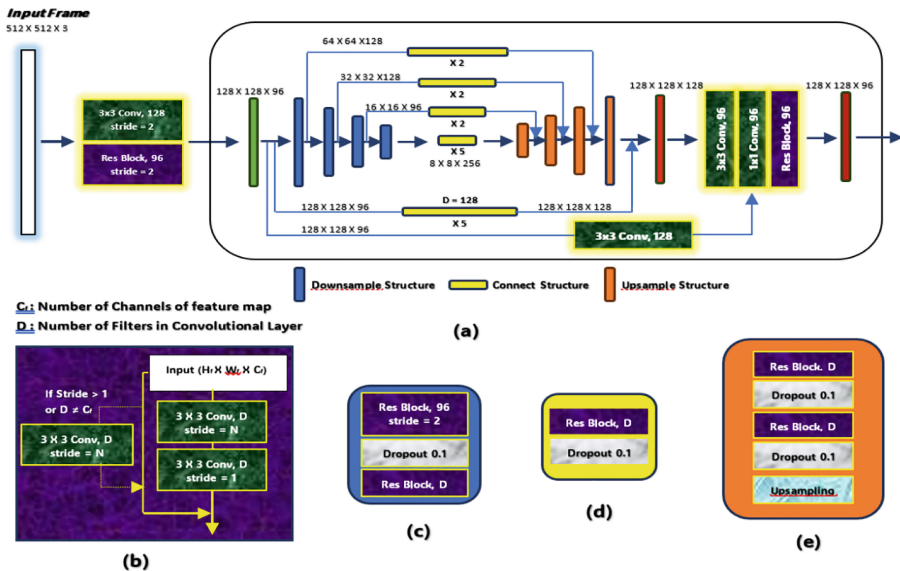


Fig. 2. Proposed Hourglass Network for text detection. (a) HourGlass Structure (b) Residual Block (c) Downsample Structure (d) Connect Structure (e) Upsampling Structure

3.1 The Proposed HourGlass Network for Text Detection

In the context of text detection from shaky videos, the Hourglass backbone architecture emerges as a pivotal component owing to its inherent adaptability to capturing multi-scale features robustly. Its distinctive capability lies in its iterative down-sampling and up-sampling mechanism, akin to a series of nested hourglass structures. This design enables the network to preserve detailed spatial information while concurrently discerning high-level semantic features, thus rendering it particularly adept at handling the intricacies posed by text detection in dynamic and volatile video environments. Specifically, in the presence of motion blur and camera shake, the Hourglass architecture facilitates the extraction of comprehensive contextual information while mitigating the adverse effects of spatial distortions, ultimately enhancing the model’s capacity for robust text detection. The Hourglass architecture is mathematically represented as defined in Eq. (1).

$$H(x) = \sum_{t=1}^T U_t(F_{t-1}(x)) + \sum_{t=1}^T D_t(U_t(F_{t-1}(x))) + F_T(x) \quad (1)$$

here, $H(x)$ denotes the output of the Hourglass network, U_t represents the up-sampling operation, D_t denotes the down-sampling operation, F_t is the residual block, and T signifies the number of Hourglass stages. This architecture effectively preserves spatial information while capturing high-level semantic features, crucial for robust text detection in shaky videos.

Complementing the Hourglass backbone (Banerjee et al. 2022), we employ the *RTMDetSepBNHead* for precise bounding box prediction under motion blur and camera shake. The bounding box prediction process is formulated as defined in Eq. (2).

$$\hat{B} = RTMDetSepBNHead(F) \quad (2)$$

where \hat{B} represents the predicted bounding box coordinates and confidence scores, and F denotes the feature maps extracted from the Hourglass backbone. By incorporating convolutional layers and batch normalization, the *RTMDetSepBNHead* enhances the localization accuracy of text regions despite the challenging conditions of shaky videos.

To address the adverse effects of motion blur and camera shake on feature representation, we introduce the *CSPNeXtPAFPN* neck architecture for robust feature fusion. The *CSPNeXtPAFPN* (Cross Stage Partial Network with extended feature Pyramids and Feature Pyramid Networks) neck architecture assumes paramount importance in fortifying the text detection pipeline within shaky videos. By virtue of its innovative design, *CSPNeXtPAFPN* orchestrates a seamless fusion of features from multiple network layers, thereby engendering a holistic understanding of the spatial and contextual intricacies inherent in text instances. Particularly in the realm of shaky videos, where conventional architectures may falter in discerning coherent features amidst motion-induced distortions, *CSPNeXtPAFPN*’s adaptive cross-stage partial connections and feature pyramid networks serve as a cornerstone for preserving information fidelity across various scales and resolutions. This, in turn, equips the model with the requisite acuity to discern text instances with heightened accuracy and resilience in the face of dynamic video conditions. The feature fusion process is mathematically expressed as defined in Eq. (3).

$$F_{fusion} = CSPNeXtPAFPN(F_1, F_2, \dots, F_n) \quad (3)$$

where F_{fusion} denotes the fused feature maps and F_1F_2, \dots, F_n represent feature maps from different network layers. By leveraging cross-stage partial connections and feature pyramid networks, *CSPNeXtPAFPN* facilitates adequate information flow across network stages, enhancing the model’s ability to capture context and spatial information essential for text detection in shaky videos.

Furthermore, we integrate the *SiLU* activation function within the neck architecture, further augmenting the model’s capacity for feature discrimination and representation. *SiLU* ‘s unique characteristic of promoting smoother gradient propagation and enhancing feature learning imbues the network with a heightened sensitivity to subtle textual cues amidst the chaos of shaky videos. In essence, by fostering more nuanced feature representations, *SiLU* facilitates the model’s ability to discern text instances with heightened acuity, thereby bolstering the overall efficacy of the text detection framework in challenging video environments. The *SiLU* activation function is defined in Eq. (4).

$$SiLU(x) = x \cdot \sigma(x) \quad (4)$$

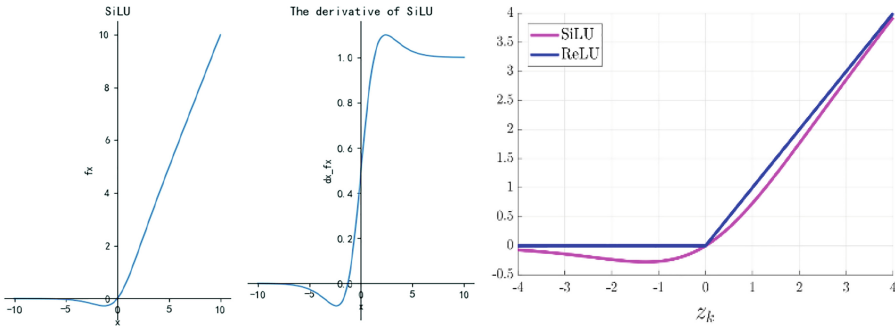


Fig. 3. Illustrating the SiLU activation function for text detection.

where $\sigma(x)$ represents the sigmoid function. By applying *SiLU* activation within the neck architecture, we enhance gradient propagation and feature learning, contributing to improved text detection performance in challenging video conditions, as shown in Fig. 3. In Fig. 3, in the context of our network, Z_k represents the input to the k – th SiLU activation function. It can be inferred from Fig. 3 that the SiLU is better than ReLU activation function for text detection. Figures 4(a)–(d) show the whole architecture of the suggested detection approach and show how the modules work together to improve outcomes.

Overall, the “RTMDetSepBNHead” module is designed for precise bounding box prediction under challenging conditions caused by shaky and non-shaky video frames. The module receives feature maps from the Hourglass network, and the convolutional layers are designed to capture the vital pattern that represents text in the frames. This step usually detects text for the text in the frames. The batch normalization stabilizes and accelerates the training process, and finally, the processed feature maps are used to predict bounding box coordinates through confidence score. Similarly, the *CSPNeXtPAFPN* (Cross Stage Partial Network with extended feature Pyramids and Feature Pyramid

Networks) module is used for robust feature fusion. It uses the features extracted from different layers of the network to represent the text pattern accurately. The adaptive cross-stage partial connections ensure effective information flow across network stages. The feature pyramid networks maintain information fidelity across various scales and resolutions. SiLU activation is used rather than ReLU to improve the classification of text and non-text performance. The details of architectures of the whole proposed method for text detection in shaky and non-shaky video frames are illustrated in Fig. 4(a)–(d).

Loss Functions: Dynamic label assignment strategies are used to match dense predictions from each scale with ground truth bounding boxes in order to train our one-stage object detector. Strategies that employ cost functions consistent with training loss as the matching criterion have been implemented in recent breakthroughs. Nevertheless, we pointed out flaws in these cost estimates and suggested a SimOTA-based dynamic soft label assignment method.

The cost function is formulated as defined in Eq. (5).

$$C = \lambda_1 C_{cls} + \lambda_2 C_{reg} + \lambda_3 C_{center} \quad (5)$$

where C_{cls} , C_{reg} and C_{center} correspond to the classification cost, regression cost, and region prior cost, respectively, with weights $\lambda_1 = 1$, $\lambda_2 = 3$ and $\lambda_3 = 1$.

Classification Cost (C_{cls}): Motivated by GFL, we propose soft labels to compute the classification cost, where the soft label is the IoU between ground truth boxes and predictions. This eliminates the instability brought on by binary labels, as stated in Eq. (6), by reweighting the matching costs with various regression quality.

$$C_{cls} = CE(P, Y_{soft}) * (Y_{soft} - P)^2 \quad (6)$$

Regression Cost (C_{reg}): Rather than using the Generalized IoU found in the loss function as described in Eq. (7), we utilize the logarithm of the IoU as the regression cost to increase the discriminativeness of the match quality.

$$C_{reg} = -\log(IoU) \quad (7)$$

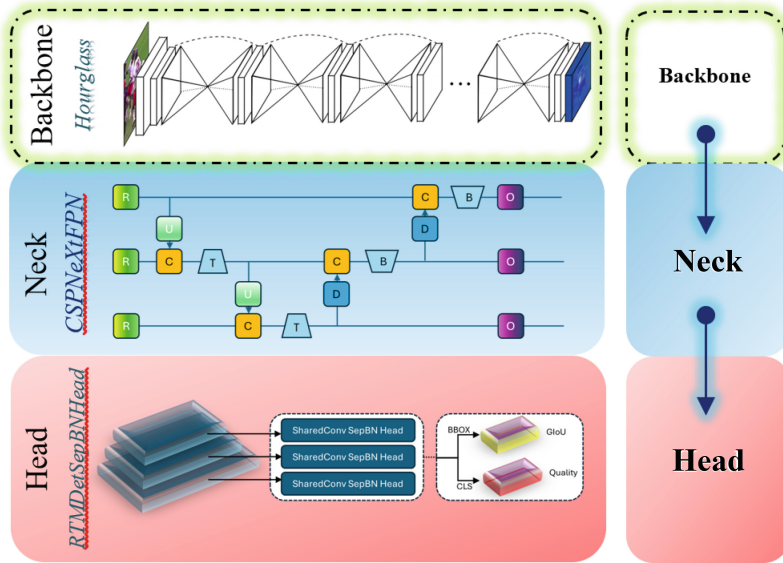
Region Cost (C_{center}): We use a soft center region cost instead of a fixed center prior to stabilizing the matching of the dynamic cost as defined in Eq. (8).

$$C_{center} = \alpha^{|x_{pred} - x_{gt}|} - \beta \quad (8)$$

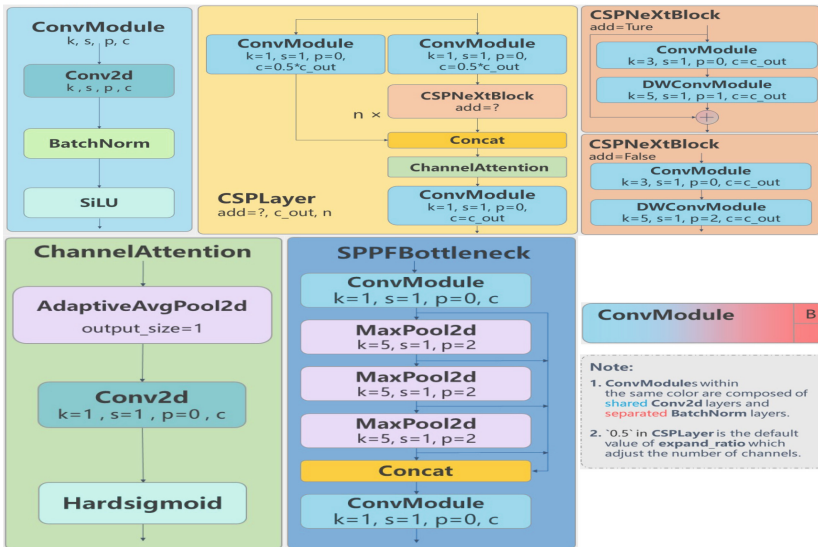
These modifications ensure that our model effectively handles both shaky and non-shaky video frames, enhancing the robustness and accuracy of text detection.

4 Experimental Results

For text detection in shaky, non-shaky day and night video frames, the standard dataset is not available. Therefore, we constructed our own dataset for experimentation, which will be discussed in the subsequent section. To verify the fairness of the results and the performance, the proposed method is tested on a standard dataset of ICDAR 2015 video frames (Karatzas D et al. 2015) and natural scene text images of Total-Text datasets (Chng

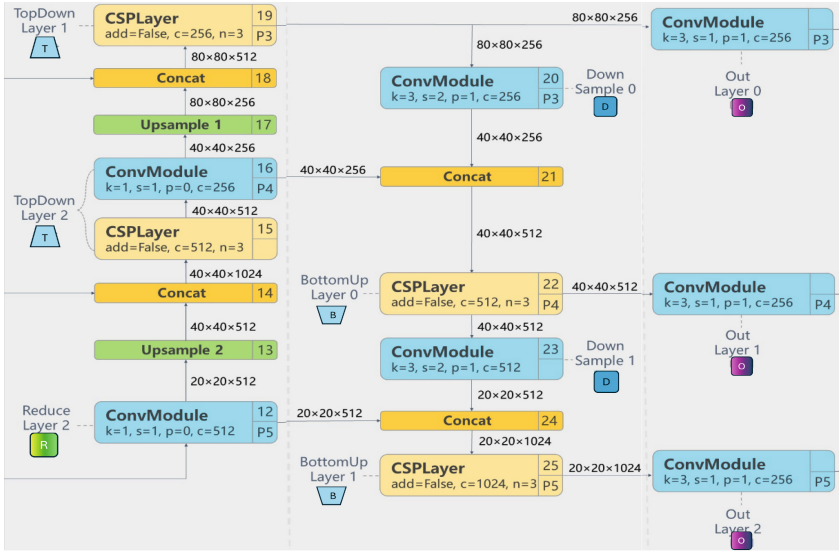


(a). Abstract Block Diagram

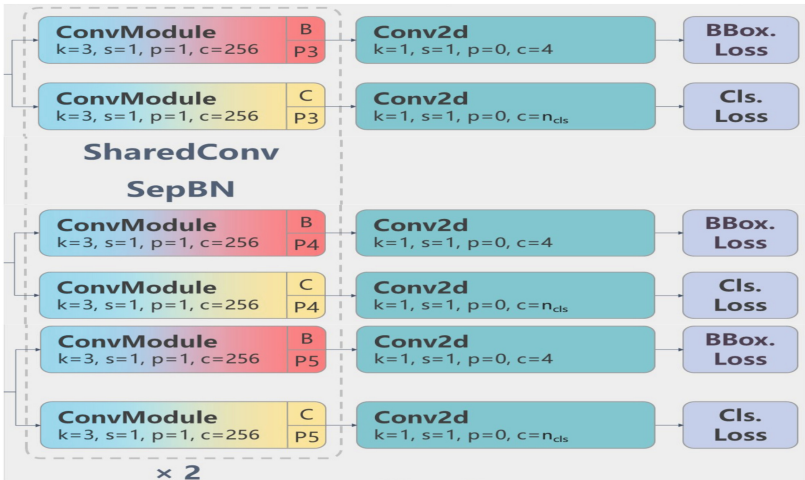


(b). Details of the Pillars

Fig. 4. (a) Abstract Block Diagram (b) Details of the Pillars (c) Details of the Neck (d) Details of the Head



(c). Details of the Neck



(d). Details of the Head

Fig. 4. (continued)

et al. 2017). It is noted that our dataset represents shaky, non-shaky day and night frames, ICDAR 2015 represents natural scene text video frames, while the Total-Text dataset represents natural scene images but not frames. The experiment on diverse datasets ensures fair, stable, robust, and reliable text detection compared to the state-of-the-art models.

4.1 Dataset Creation and Evaluation

We built our own dataset, including 237 films, of which 51 contain samples from shaky cameras and 186 have samples from non-shaky cameras, for the purpose of text identification in shaky, non-shaky day and night video frames. Our dataset comprises a large warehouse as well as the entrances to companies and industrial regions where products and supplies can be placed in open spaces. A CCTV camera installed on the ceiling for inside situations and on poles for outside scenarios records the 1–8 s movies. Movies can contain any item, including people and cars, and our dataset includes a variety of movies with varying quality, deterioration, and distortion due to the fact that videos are recorded under various weather conditions and at different times. If the video is inside, it has low quality and resolution. If the video is taken outside, it is affected by outside elements like the light from the stars and the weather. The issue is even more complicated when it comes to outside footage because of the trees and leaves there. In conclusion, compared to standard word identification natural scene video frame datasets, our bespoke dataset presents more challenges. There are 36 shaky and 130 non-shaky videos used for training, and there are 15 shaky and 56 non-shaky videos used for assessment.

We employed the ICDAR-2015 dataset and the Total Text dataset, both from the ICDAR 2015 Robust Reading competition, for benchmark experiments. Because Google Glasses did not take position into account when shooting these pictures and because the entire text dataset is publicly accessible on Kaggle’s official website, the text in the scene can be oriented in any way. The Total Text dataset contains 1555 scene photos, while the ICDAR 2015 Video comprises of 28 videos that range in length from 10 s to one minute and feature either indoor or outdoor settings. Total-Text is split into two sets of 300 and 1255 images, respectively, for the training and test sets. This dataset has a number of problems, such as various text orientations, text fonts, and picture backgrounds. In order to assess the efficacy of the suggested and current approaches, we employ the conventional metrics of Average Precision (AP) and Average Recall (AR).

Implementation Details: For the implementation of our research experiments, we utilized a robust computing environment comprising a Windows 10 Enterprise operating system. Our hardware setup included a powerful 13th Gen Intel(R) Core (TM) i7-1370P processor clocked at 1.90 GHz, paired with 16.0 GB of RAM. Graphics processing was facilitated by two GPUs: Intel(R) UHD Graphics and Nvidia GeForce MX550 with 2GB of dedicated memory. The CUDA framework, version 12.3, was employed to leverage parallel processing capabilities for the efficient execution of our computational tasks. This configuration provided the necessary computational horsepower to conduct our experiments effectively and analyze the results with precision.

4.2 Ablation Study

A few crucial stages are included in the suggested strategy in order to attain the best text identification results regardless of domain. We provide ablation research to evaluate the performance of each essential element. Our study primarily focuses on the utilization of the Hourglass architecture as the backbone network. Additionally, we explore the impact of alternative backbone networks, specifically ResNet and CSPNext, to elucidate

the significance of our chosen architecture. Further, the SiLU activation function is also used to improve the performance of the proposed text detection.

According to Table 1’s experimental results, every component included in the suggested task is efficient and makes an equal contribution to getting the best outcomes. This is because the results of individual components are lower than the results of the proposed method. The results also indicate that the baseline ResNet is not effective compared to CSPNext, and the CSPNext is not effective compared to the HourGlass network for text detection in different datasets. In the same way, when we compare the results of the SiLU and ReLU activation function, the conventional ReLU is not better than the SiLU on all three datasets. This shows that SiLU is effective against the ReLU activation function. Overall, one can infer from the ablation study experiments that although individual components are effective, the individual components are not capable of achieving the best results as the proposed method.

Table 1. Validating the effectiveness of the key components

Experiments	Our dataset		ICDAR2015 dataset		Total Text	
	AP	AR	AP	AR	AP	AR
Backbone + ReLU						
Resnet	84.2	81.7	83.8	55.6	81.2	79.9
CSPNext	85.7	83.2	80.6	68.2	82.7	74.5
Backbone + SiLU						
Resnet	85.2	82.2	88.3	85.0	84.0	78.0
CSPNext	88.1	82.3	88.5	84.7	87.6	79.9
Proposed Hourglass	95.0	95.0	92.3	89.9	87.6	86.1

4.3 Experiments on Detection

Figure 5 displays the text detection outcomes of the suggested approach on several datasets, demonstrating accurate text detection in every image. This suggests that the the approach works effectively for texts from many fields. As was mentioned in the section on ablation studies, this is the main component’s benefit. The quantitative findings of the suggested and current methodologies, listed in Table 2, support the same conclusions. The comparison between the performance of the proposed and current approaches demonstrates that, across all datasets, the proposed method outperforms all current methods. The lack of domain independence and generalization capacity in the current methods is the cause of their subpar performance.

Cross-Dataset Validation: We also carried out an additional experiment known as cross-dataset validation to confirm the generic property, domain independence, and performance that is not significantly dependent on training samples. In these studies, samples from one dataset are used to train the algorithm and samples from another dataset

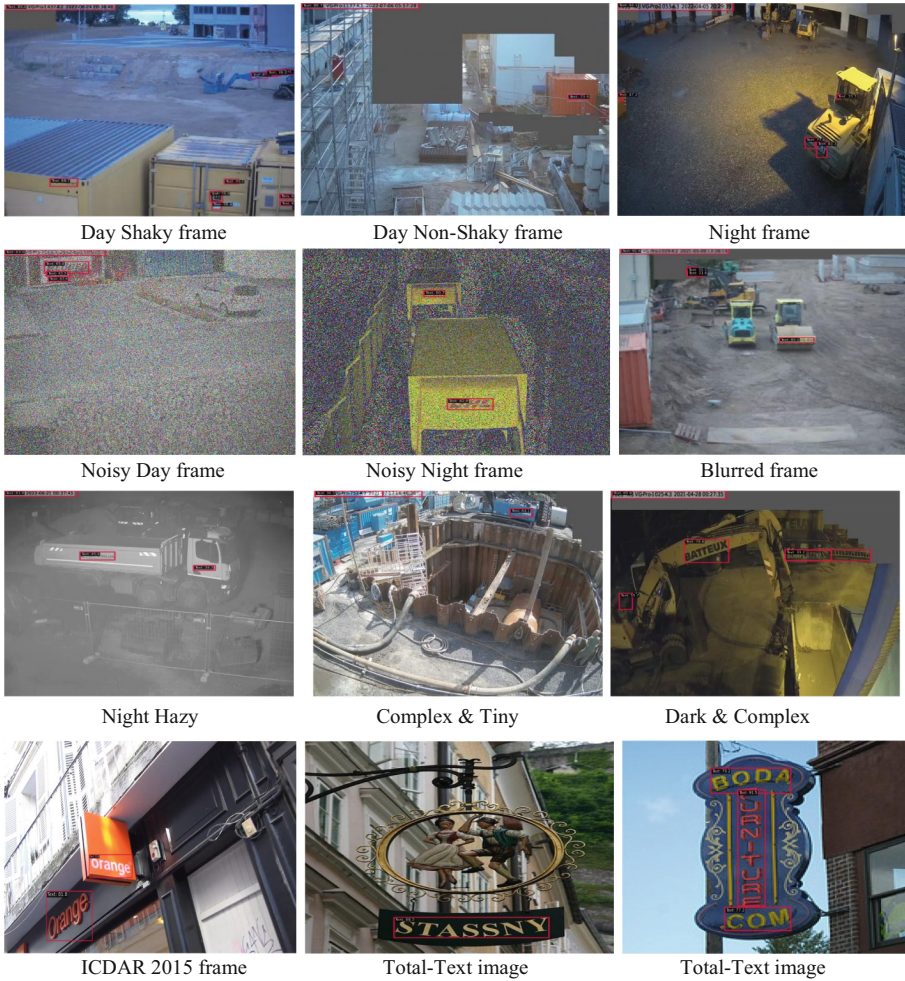


Fig. 5. Text detection of the proposed model on images of different datasets

are used to test it. Comparing the suggested model to the state-of-the-art technique (Zhang et al. 2023), consistent and stable results are obtained for all trials, as shown in Table 3. Therefore, we can conclude that the proposed method is generic and domain-independent; more than that, our method is robust enough to address the challenges of shaky, non-shaky day and night video frames.

Robustness Validation: Figure 5 shows that the proposed method performs well for night, noisy, blurred images, and other complex situations. To draw the same conclusion quantitatively, Gaussian noise and blur are added randomly to the normal images for experimentation, and the results are reported in Table 4. It is noted from Table 4 that the performance of the proposed method for noisy and blurred datasets is almost similar to the results of the proposed method on the normal datasets. Therefore, overall, the

discussion on the above experiments asserts that the proposed method is generic, domain-independent, and robust.

For all the experiments, it is noted that the performance of the state-of-the-art methods is inferior to the performance of the proposed method. This is due to the lack of generalization ability of the existing methods. In addition, the scope of the existing methods is limited to scene images but not complex images of shaky and non-shaky videos. On the other hand, the way the proposed work fuses the strengths of different modules in the novel makes a difference in achieving better results for both scene images and shaky and non-shaky video frames.

Table 2. Performance of the Proposed and existing methods for text detection

Methods	Our Dataset		ICDAR-2015		Total Text	
	AP	AR	AP	AR	AP	AR
YOLOv5s (Chaitra et al. 2022)	71.8	62.1	61.0	46.0	66.3	61.4
Shikha et al. (2023)	75.6	64.47	67.8	53.29	67.1	63.5
Zhang et al. (2023)	78.7	74.2	86.9	84.5	81.4	78.5
Halder et al. (2023)	82.6	79.6	80.4	77.8	–	–
Cai et al. (2022)-(DText)	–	–	88.5	85.6	86.9	82.7
Su et al. (2022)-(TextDCT)	–	–	88.9	84.8	85.0	85.3
Zhao et al. (2022)	–	–	89.4	82.4	86.1	82.1
Liao et al. (2023)-(DBNet + +)	–	–	90.9	83.9	87.9	82.8
Shao et al. -(CT-Net)	–	–	90.9	86.4	87.9	82.7
Proposed Method	95.0	95.0	92.3	89.9	87.6	86.1

Table 3. Performance of the Proposed and state-of-the-art method (Zhang et al. 2023) method with cross-dataset validation

Train	Test											
	Proposed method						Zhang et al. (2023)					
	Our Dataset		ICDAR-2015		Total Text		Our Dataset		ICDAR-2015		Total Text	
	AP	AR	AP	AR	AP	AR	AP	AR	AP	AR	AP	AR
Our Dataset	95.0	95.0	52.4	54.8	56.2	63.5	78.7	74.2	50.0	43.0	54.1	40.7
ICDAR-2015	66.5	72.3	92.3	89.9	69.3	74.3	50.8	33.4	86.9	84.5	54.2	51.6
Total Text	75.6	77.2	70.2	71.0	87.6	86.1	73.1	76.8	69.2	64.2	81.4	78.5
All Datasets	87.8	90.1	86.1	88.8	86.4	87.0	70.7	77.2	81.0	63.0	73.4	68.3

Table 4. Validating the performance of the Proposed and existing methods for text localization in challenging scenarios

Methods	Noisy		Blurry	
	AP	AR	AP	AR
YOLOv5s (Chaitra et al. 2022.	54.5	62.7	69.1	64.5
Shikha et al. (2023)	67.9	85.3	74.4	68.5
Zhang et al. (2023)	56.3	52.4	73.4	66.9
Proposed Method	90.2	88.8	92.1	88.1

Limitation: When we look at samples of shaky, non-shaky day and night video frames, there are frames (shown in Fig. 6 as a sample), for which the proposed method fails to detect the text in the frames. This is because the text in the frames is not visible even to our naked eyes due to fog and poor vision at night. This is beyond the scope of the work. However, for such challenges, our plan is to integrate a language model with the proposed model. This is because the language model can predict the missing text and invisible text in the images.



Foggy-Night frame



Noisy Night frame

Fig. 6. Poor performance of the proposed method

5 Conclusions and Future Work

We provide a unique method for text detection in unstable or non-existent video frames captured throughout the day and night in this work. Unlike prior models that focus on text recognition in natural scene video frames and scene photographs, our model focuses on both shaky and non-shaky standard/natural scene photos/frames. We have proposed the HourGlass network and integrated it with a new head function and feature design to produce optimal results for photos of various datasets. Comparing the suggested approach to the state-of-the-art methods, the results of several experiments conducted using the current methods demonstrate its exceptional performance. Furthermore, experiments on datasets of different domains, cross-dataset validation, and noisy and blurred datasets

show that the proposed method is sufficiently generic, domain-independent, and robust to noise and blurred images. However, when the images are affected by severe fog and night settings where the text is invisible, the performance of the proposed method degrades. To address this challenge, our plan is to integrate language models with the proposed model to predict the invisible text, which will be our future work. However, as you suggested, there is a scope for optimizing the proposed model, which is beyond the scope of the work and, therefore, which is discussed in the conclusion and future work.

References

- Asadzadehkaljahi, M., Halder, A., Pal, U., Shivakumara, P.: Spatiotemporal edges for arbitrarily moving video classification in protected and sensitive scenes. *Artif. Intell. Appl.*. (2023a). <https://doi.org/10.47852/bonviewaia3202526>
- Asadzadehkaljahi, M., Halder, A., Shivkumara, P., Pal, U.: Spatio-temporal FFT-based approach for arbitrarily moving object classification videos of protected and sensitive scenes. *Artif. Intell. Appl.* (2023b). <https://doi.org/10.47852/bonviewAIA3202553>
- Banerjee, A., Shivakumara, P., Acharya, P., Pal, U., Canet, J.L.: TWD: a new deep E2E model for text watermark/caption and scene text detection in video. In: *Proceedings ICPR*, pp. 1492–1498 (2022)
- Bannet, M.A., Srividhya, R., Jayachandran, T., Rajmohan, V.: Deep learning-based Telugu video text detection using coding over digital transmission. In: *Proceeding ICOEI*, pp. 1479–1483 (2022)
- Cai, Y., Liu, T., Shen, C., Jin, L., Li, Y., Ergu, D.: Arbitrarily shaped scene text detection with dynamic convolution. *Pattern Recognit.* 108608 (2022)
- Chaitra, Y.L., Dinesh, R., Jeevan, M., Arpitha, M., Aishwarya, V., Akshitha, K.: An impact of YOLOv5 on text detection and recognition system using TesseractOCR in images/video frames. In: *Proceedings ICDSIS* (2022)
- Chen, L., Shi, J., Su, F.: Robust video text detection through parametric shape regression, propagation and fusion. In *Proceedings of ICME*, pp. 1–6 (2021)
- Chen, X., et al.: CSPNeXt: A new efficient token hybrid backbone. *Eng. Appl. Artif. Intell.* **132**, 107886 (2024). <https://doi.org/10.1016/j.engappai.2024.107886>
- Chng, C.K., Chan, C.S.: Total-text: a comprehensive dataset for scene text detection and recognition. *ArXiv/abs/1710.10400* (2017)
- Elfving, S., Uchibe, E., Doya, K.: Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *ArXiv/abs/1702.03118* (2017)
- Halder, A., Shivakumara, P., Pal, U., Blumenstein, M., Ghosal, P.: A locally weighted linear regression-based approach for arbitrary moving shaky and nonshaky video classification. *Int. J. Pattern Recognit Artif Intell.* (2024). <https://doi.org/10.1142/S0218001423510199>
- Halder, A., Shivakumara, P., Pal, U., Lu, T., Blumenstein, M.: A new transformer based approach for text detection in shaky and non-shaky day-night video. In: *Proceedings of ACPR* (2023). https://doi.org/10.1007/978-3-031-47637-2_3
- Karatzas, D., et al.: ICDAR 2015 competition on robust reading. In: *Proceedings of ICDAR*, pp. 1156–1160 (2015)
- Liao, M., Wan, Z., Yao, C., et al.: Real-time scene text detection with differentiable binarization. In: *Proceedings of the AAAI Conference on ARTIFICIAL intelligence*, pp. 11474–11481 (2020)
- Liao, M., Zou, Z., Wan, Z., Yao, C., Bai, X.: Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE Trans. Pattern Anal. Mach. Intell.* 919–931 (2023)
- Lu, P., Wang, H., Zhu, S., Wang, J., Bai, X., Liu, W.: Boundary TextSpotter: toward arbitrary-shaped scene text spotting. *IEEE Trans. Image Process.* **31**, 6200–6212 (2022)

- Naiemi, F., Ghods, V., Khalesi, H.: A novel pipeline framework for multi oriented scene text image detection and recognition. *Expert Syst. Appl.* **170**, 114549 (2021)
- Nandanwar, L., Shivakumara, P., Ramachandra, R., Lu, T., Antonacopoulos, A., Lu, Y.: A new deep wavefront-based model for text localization in 3D video. *IEEE Trans. Circuits Syst. Video Technol.* 3375–3389 (2022)
- Shikha, N., Pranav, R., Singh, N. R., Umadevi, V., Hussain, M.: Kannada word detection in heterogeneous scene images. In: *Proceedings of SPIN*, pp. 379–383 (2023)
- Su, Y., et al.: TextDCT: Arbitrary-shaped text detection via discrete cosine transform mask. *IEEE Trans. Multimed.* (2022)
- Wang, W., et al.: Shape robust text detection with progressive scale expansion network. In: *Proceedings of CVPR*, pp. 9336–9345 (2019)
- Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: LayoutLM: pre-training of text and layout for document image understanding. *ArXiv* (2020). <https://doi.org/10.1145/3394486.3403172>
- Yin, X.-C., Zuo, Z.-Y., Tian, S., Liu, C.-L.: Text detection, tracking and recognition in video: a comprehensive survey. *IEEE Trans. Image Process.* 2752–2773 (2016)
- Zhang, S.X., Zhu, X., Chen, L., Hou, J.B., Yin, X.C.: Arbitrary shape text detection via segmentation with probability maps. *IEEE Trans. Pattern Anal. Mach. Intell.* 2736–2750 (2023)
- Zhao, M., Feng, W., Yin, F., Zhang, X.Y., Liu, C.-L.: Mixed-supervised scene text detection with expectation-maximization algorithm. *IEEE Trans. Image Process.* 5513–5528 (2022)
- Zhu, J., et al.: TransText: Improving scene text detection via transformer. *Digit. Signal Process.* **130**, 103698 (2022)



FastTextSpotter: A High-Efficiency Transformer for Multilingual Scene Text Spotting

Alloy Das¹, Sanket Biswas²(✉), Umapada Pal¹, Josep Lladós²,
and Saumik Bhattacharya³

¹ CVPR Unit, Indian Statistical Institute, Kolkata, Kolkata, India
umapada@isical.ac.in

² Computer Vision Center, Universitat Autònoma de Barcelona, Bellaterra, Spain
{sbiswas, josep}@cvc.uab.cat

³ ECE, Indian Institute of Technology, Kharagpur, Kharagpur, India
saumik@ece.iitkgp.ac.in

Abstract. The proliferation of scene text in both structured and unstructured environments presents significant challenges in optical character recognition (OCR), necessitating more efficient and robust text spotting solutions. This paper presents FastTextSpotter, a framework that integrates a Swin Transformer visual backbone with a Transformer Encoder-Decoder architecture, enhanced by a novel, faster self-attention unit, SAC2, to improve processing speeds while maintaining accuracy. FastTextSpotter has been validated across multiple datasets, including ICDAR2015 for regular texts and CTW1500 and TotalText for arbitrary-shaped texts, benchmarking against current state-of-the-art models. Our results indicate that FastTextSpotter not only achieves superior accuracy in detecting and recognizing multilingual scene text (English and Vietnamese) but also improves model efficiency, thereby setting new benchmarks in the field. This study underscores the potential of advanced transformer architectures in improving the adaptability and speed of text spotting applications in diverse real-world settings. The dataset, code, and pre-trained models have been released in our [Github](#).

Keywords: Text Spotting · Vision Transformers · Multilingual · Attention

1 Introduction

In the rapidly evolving field of pattern recognition, text spotting- the task of localizing and recognizing text within natural scenes - poses unique challenges. These challenges have been addressed through powerful optical character recognition (OCR) systems designed to handle text in both structured [11] and

A. Das and S. Biswas—Equal contribution.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15320, pp. 135–150, 2025.
https://doi.org/10.1007/978-3-031-78498-9_10

unstructured [1, 8] environments. These environments commonly feature text in multiple ranges of orientations (from arbitrary-shaped [3, 22, 36] to regular-shaped [15]), annotation styles (from rotated quadrilaterals [15], polygonal word-level [3], polygonal sentence-level [22] to hierarchical layout-level [25]) and diverse language domains (multilingual [28, 29] to low-resource languages [30] to different scripts [46]). The overall computational load of processing such high-resolution images to detect and recognize text accurately across different text orientations, languages and styles is substantial.

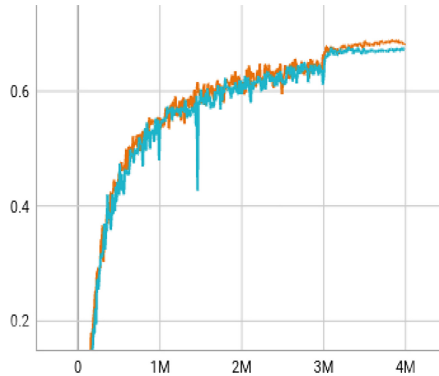


Fig. 1. Trade-off between text spotting performance h-mean vs number of training iterations: The blue curve indicates the model without the SAC2 attention module while the orange curve depicts the model performance with our proposed SAC2 module. (Color figure online)

Current state-of-the-art models have made significant contributions towards improving text detection and recognition capabilities, which employs both Convolutional Neural Networks (CNNs) [21, 23, 32, 33] and Transformers [4, 8, 14, 45, 47]. Despite significant advancements, these models typically struggle with balancing high accuracy and computational efficiency, especially under constrained resources or in real-time applications. This is particularly critical in scenarios where quick text interpretation is essential, such as in navigation aids for visually impaired individuals or instant translation services. Fast and reliable text interpretation can indeed help bridge the digital divide by making information more accessible to people who speak less common languages or dialects. Moreover, it can automate and improve the process of annotating vast amounts of data, which is essential for training more robust OCR systems that provide higher-quality, context-aware annotations.

The emergence of text spotting transformers (TESTR) [47] has prompted the adoption of detection transformer (DETR) architectures [2] as foundational backbones within text spotting frameworks as in [4, 45]. Recent methods [44, 45] have focused on enabling more efficient training and faster convergence using

deformable attention [49] on the dynamic control point queries of text coordinates. Using point coordinates to obtain positional queries rather than anchor boxes, as described in [47], allows the transformer decoder to dynamically update points for scene text detection. In this work, we explore the possibility of extending this dynamic attention mechanism towards the task of scene text spotting. Moreover, recent works [4, 5, 14] have recently shown the effectiveness of Swin Transformers [24] by generating hierarchical feature maps which are critical for fine-grained predictions necessary in text segmentation. This work introduces **FastTextSpotter**, a novel text spotting framework that combines a Swin-tiny backbone for visual feature extraction with a dual-decoder transformer encoder architecture [47] tailored for text spotting. To optimize training, we introduce a novel attention module, **SAC2** (Self-Attention with Circular Convolutions), inspired by [31, 44]. This novel component, integrated within our text spotting framework, not only competes well with existing state-of-the-art (SOTA) text spotters but also enhances operational efficiency, particularly in frames per second (FPS), setting a new benchmark in text spotting performance. Figure 1 illustrates the accuracy vs efficiency trade-off and the impact of the SAC2 attention module. In this context, we also explore the following key research questions to gain a more comprehensive understanding of the trade-offs between accuracy and efficiency in SOTA text spotting models. 1) How can the computational efficiency of attention mechanisms in Transformer models be improved without compromising on text detection and recognition accuracy? 2) What architectural modifications are necessary to adapt the Swin Transformer for optimal performance in diverse text spotting scenarios and orientations? 3) Can the model effectively handle multilingual text spotting, particularly in low-resource languages like Vietnamese?

The paper proposes a three-fold contribution: 1) We develop FastTextSpotter, integrating a Swin-Tiny backbone with an efficient text spotting setup, significantly enhancing scene text detection and recognition efficiency. 2) We introduce SAC2, a dynamic attention mechanism that accelerates training and convergence while maintaining high detection accuracy. 3) FastTextSpotter excels in detecting multilingual and variably oriented texts, including in resource-limited languages like Vietnamese, broadening its utility in diverse applications.

2 Related Work

Our FastTextSpotter framework is designed to refer to the previous works introduced below, aiming to handle scene text spotting in an efficient way that combines text spotting transformers with a dynamic and faster attention module.

End-to-End Object Detection. The DETR approach [2] proposed the first end-to-end transformer-based object detection as a set prediction task without complex hand-crafted anchor generation and post-processing. The Deformable-DETR [49] addressed the task by attending to sparse features using a deformable cross-attention operation which reduces the quadratic complexity of DETR to

linear complexity and leveraging multi-scaled features. DE-DETR [41] identifies that the main element which impacts the model efficiency is related to the sparse feature sampling, whereas DAB-DETR [19] handled the above issue using dynamic anchor boxes as position queries. In our study, we recast this query in point formulation to modify the Transformer Decoder backbone in TESTR [47] for both detection and recognition tasks to handle arbitrarily shaped scene texts and also fasten the training process. Recent methods like [48] focus attention on more information-rich tokens for improving trade-off between efficiency and model performance.

Scene Text Spotting. The rise of deep learning has significantly advanced the field of scene text spotting. Early methods treated text spotting as a two-stage process, training separate detection and recognition modules that were combined at inference time [18, 39]. More recent approaches have adopted end-to-end strategies [16, 20], simultaneously tackling detection and recognition through RoI operations to address arbitrary-shaped texts with some using quadrangle text region proposals [10, 37]. Other approaches employ the MaskTextSpotter [17, 27] series which employed binary maps for text and character-level segmentation tasks based on Mask-RCNN [12] to reduce segmentation errors. PAN++ [42] have further refined these methods by enhancing segmentation efficiency and reducing background interference, similar to those adopted in [34, 43]. While these approaches produced acceptable performance, the mask representation needed some further post-processing steps. MANGO [32] proposed a mask attention module to utilize global features across several text instances, however, it required center-line segmentation for the predictions. Other attempts to create customized representations for curved texts include Parametric Bezier curves [21, 23], Shape Transform module [33] etc.

Impact of Transformers. The introduction of Transformers [38] has marked a pivotal shift towards transformer-based architectures in text spotting, eliminating the need for RoI operations by leveraging global feature modelling, as seen in applications like ABINet and SwinTextSpotter [9, 14]. The introduction of vision transformers [7] also opened the floodgates to its application in STR application [1]. ABINet [8, 9] integrate advanced techniques such as bidirectional language modelling and Feature Pyramid Networks (FPNs) to improve text detection and recognition, particularly for texts entangled with complex backgrounds or small sizes. Our proposed framework, leveraging a Swin-Transformer [24] with a tiny variant for computational efficiency, utilizes a multi-scale deformable attention mechanism [49] used by TESTR approach [47] to optimize feature extraction across various text sizes *without necessitating post-processing for polygon vertices or Bezier control points*, thereby streamlining the text spotting process and enhancing overall system performance.

3 Methodology

The core objective of the FastTextSpotter framework is to enhance the efficiency and accuracy of scene text detection and recognition. This section details the

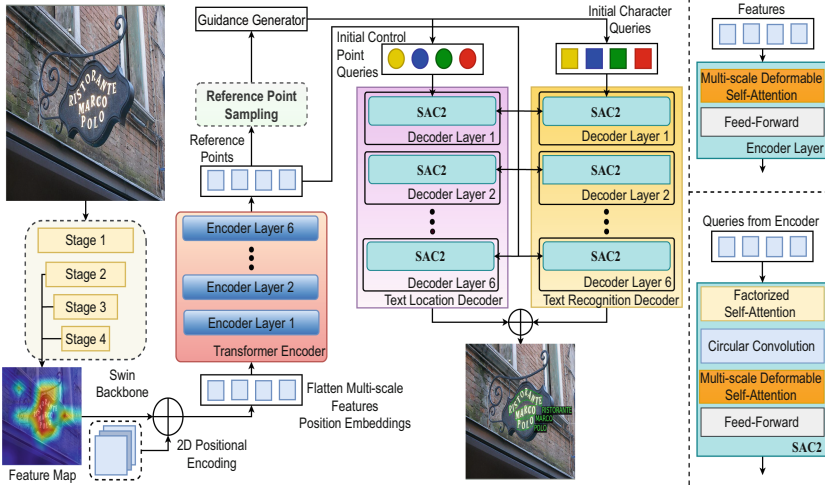


Fig. 2. Overview of FastTextSpotter illustrating a Swin Transformer visual backbone with a Transformer Encoder-Decoder framework. Key features include the SAC2 attention module, dual decoders for accurate text localization and recognition, and the Reference Point Sampling system for effective text detection across various shapes and languages.

architectural components and the novel self-attention unit, SAC2, which together form the backbone of our proposed system. Additionally, we outline the training objectives and processes that drive the performance of the entire framework.

3.1 Model Architecture

The overall architectural framework, as depicted in Fig. 2, is composed of three primary components: (1) a visual feature extraction unit that utilizes a Swin-Transformer [24] backbone for the extraction of multi-scale features; (2) a text spotting module that includes a Transformer encoder, which encodes the image features into positional object queries, followed by two separate Transformer decoder units that are responsible for predicting the locations of text instances and recognizing the corresponding characters.

Visual Feature Extraction Unit. vanilla convolutions, which operate locally at fixed sizes (e.g., 3×3), struggle to connect distant features effectively. Text spotting, however, demands the ability to capture relationships between various text regions within the same image, while also accounting for similarities in background, style, and texture. To address this, we selected a compact yet efficient Swin-Transformer [24] unit, referred to as Swin-tiny, to extract more detailed and fine-grained image features in Fig. 4.

Text Spotting Unit. The text-spotting module primarily comprises a transformer encoder and two transformer decoders dedicated to text detection and

recognition, following a schema similar to the TESTR framework [47]. We formulate this task as a set prediction problem inspired by DETR [2], aiming to predict a set of point-character pairs for each image. Specifically, we define it as $A = (E^{(j)}, F^{(j)})_{j=1}^K$, where j indicates the index of each instance. Here, $E^{(j)} = e_1^{(j)}, \dots, s_M^{(j)}$ represents the coordinates of M control points, and $F^{(j)} = f_1^{(j)}, \dots, f_M^{(j)}$ corresponds to the sequence of M text characters. In this unified framework, the text location decoder (TLD) predicts $E^{(j)}$, while the text recognition decoder (TRD) predicts $F^{(j)}$.

Text Location Decoder. In the location decoder, queries are transformed into composite queries that predict multiple control points for each text instance. We define these as Q queries, with each one corresponding to a text instance denoted as $E^{(j)}$. Each query comprises several sub-queries e_n , such that $e^{(j)} = e_1^{(j)}, \dots, e_M^{(j)}$. These initial control points are then processed through the location decoder, which consists of multiple layers. This is followed by a classification head that predicts confidence levels for the final control points, alongside a two-channel regression head that generates the normalized coordinates for each point. In this context, the control points are defined as the polygon vertices, starting from the top-left corner and proceeding in a clockwise direction.

Text Recognition Decoder. The character decoder operates similarly to the location decoder, with the key difference being that control point queries are replaced with character queries, denoted as $F^{(j)}$. Both $E^{(j)}$ and $F^{(j)}$ queries, sharing the same index, correspond to the same text instance. Consequently, during the prediction phase, each decoder simultaneously predicts the control points and the characters for the corresponding instance. Finally, a classification head is employed to predict multiple character classes based on the final character queries.

3.2 Query Point Formulation and SAC2 Attention Module

The training efficiency of the FastTextSpotter is primarily driven by a dynamic point update strategy, which updates prediction points during sampling from the transformer encoder unit. This is followed by the application of the SAC2 attention module in the subsequent text location and recognition decoders.

Reference Point Sampling. We adopt the box-to-polygon conversion method from TESTR [47], which effectively transforms axis-aligned box predictions into polygon representations of scene text. This approach, inspired by [44] simplifies and improves the scene text detection. Positional queries are generated from anchor boxes using a 2D positional encoding, enhanced with a multi-layered perceptron as implemented in [19], with the objective of making these queries learnable. Specifically, these dynamic anchor boxes-post the final Transformer encoder layer-are concatenated with M content queries for control points and A content queries for text characters, refining the text spotting process. The following Eq. 1 explains the used strategy of creating the compositional queries

$Q^{(j)}(j = 1, \dots, K) :$

$$Q^{(j)} = E^{(j)} + F = \theta((s, r, c, d)^{(j)}) + (e_1, e_2, \dots, e_M) \quad (1)$$

where S and R stand in for the relevant positional and content components of each composite query. The sine positional encoding function is followed by a normalising and linear layer. The center coordinate and scale details of each anchor box are represented by (s, r, c, d) . The M learnable control point content queries shared over K composite inquiries are (e_1, \dots, e_M) . Keep in mind that we used the detector with the Eq. 1 query formulation in our model. We sample $\frac{M}{2}$ point coordinates $point_m(m = 1, \dots, M)$ evenly on the top and bottom side of each anchor box, respectively, motivated by the positional label form and the shape prior that the top and bottom side of a scene text are often close to the corresponding side on bounding box as in Eq. 2:

$$point_n = \begin{cases} (s - \frac{c}{2} + \frac{(m-1) \times c}{\frac{M}{2}-1}, r - \frac{d}{2}), & m \leq \frac{M}{2} \\ (s - \frac{c}{2} + \frac{(M-m) \times c}{\frac{M}{2}-1}, r + \frac{d}{2}), & m > \frac{M}{2} \end{cases} \quad (2)$$

With $(point_1, \dots, point_N)$, we can generate composite queries using the following complete point formulation as in Eq. 3 (Fig. 3):

$$Q^{(i)} = \varphi((point_1, \dots, point_M)^{(i)}) + (e_1, \dots, e_M) \quad (3)$$



Fig. 3. Visualization of attention maps for Resnet-50 feature backbone. (L) to (R) shows attention maps starting from the first layer.

The φ function in Eq. 3 shows the dynamic point query update and is differentiable. This results in the best training convergence since each of the N control point content queries has its own explicit position prior.

The SAC2 Attention Block. We use the Factorized Self- Attention (FSA) [6] in our model in accordance with [44, 47]. FSA takes advantage of an intra-group self-attention (SA_{intra}) across M subqueries that correspond to each of the $Q^{(j)}$ to capture the relationship between various points within each text instance. *FastTextSpotter captures the relationship between various objects by introducing an inter-group self-attention (SA_{inter}) across K composite inquiries is used after SA_{intra} .* We hypothesize that the non-local self-attention mechanism, SA_{intra} , does not adequately capture the spherical shape of polygon

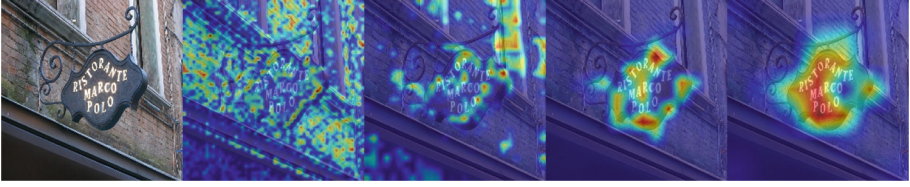


Fig. 4. Visualization of attention maps for Swin-Tiny feature backbone. (L) to (R) shows attention maps from the first layer.

control points. To address this, we incorporate local circular convolution [31] to bolster factorized self-attention (FSA). Initially, SA_{intra} processes to produce internal queries $Q_{intra} = SA_{intra}(Q)$, using identical keys as Q and values that exclude positional elements. Concurrently, locally enhanced queries are formed: $Q_{local} = ReLU(BN(CirConv(Q)))$. These are then integrated to create fused queries $Q_{fuse} = LN(FC(C + LN(Q_{intra} + Q_{local})))$, where C represents content queries acting as a shortcut, and FC , BN , and LN denote fully connected layer, BatchNorm, and LayerNorm, respectively. Subsequently, Q_{inter} , which explores inter-positional relations, is derived from Q_{fuse} using SA_{inter} and passed to the deformable cross-attention module [49]. Optimal performance and inference speed are achieved using the aforementioned training setup.

3.3 Loss Functions

The overall losses used for FastTextSpotter can be summarised under the encoder \mathcal{L}_{enc} and decoder \mathcal{L}_{dec} blocks shown in Eq. 4 and Eq. 5 respectively.

$$\mathcal{L}_{enc} = \sum_j \left(\lambda_{cls} \mathcal{L}_{cls}^{(j)} + \lambda_{coord} \mathcal{L}_{coord}^{(j)} + \lambda_{gIoU} \mathcal{L}_{gIoU}^{(j)} \right) \quad (4)$$

$$\mathcal{L}_{dec} = \sum_i \left(\lambda_{cls} \mathcal{L}_{cls}^{(i)} + \lambda_{coord} \mathcal{L}_{coord}^{(i)} + \lambda_{char} \mathcal{L}_{char}^{(i)} \right) \quad (5)$$

Here, $\mathcal{L}_{cls}^{(i)}$ represents the focal loss for text instance classification, while $\mathcal{L}_{coord}^{(i)}$ denotes the L-1 loss used for control point coordinate regression. $\mathcal{L}_{char}^{(i)}$ corresponds to the cross-entropy loss for character classification, and \mathcal{L}_{gIoU} is the generalized IoU loss for bounding box regression, as defined in [35]. The weighting factors for these losses are represented by λ_{cls} , λ_{char} , λ_{coord} , λ_{cls} , and λ_{gIoU} .

Instance Classification Loss. We use the focal loss as the classification loss of text instances. For the i -th query, the loss is denoted as:

$$\begin{aligned} \mathcal{L}_{cls} = & -\mathbb{1}_{\{i \in Pic(\sigma)\}} \alpha (1 - \hat{b}^{(i)})^\gamma \log(\hat{b}^{(i)}) \\ & -\mathbb{1}_{\{j \notin Pic(\sigma)\}} (1 - \alpha) (\hat{b}^{(j)})^\gamma \log(1 - \hat{b}^{(i)}) \end{aligned} \quad (6)$$

Control Point Loss. We used L-1 loss for control point regression the loss is defined for the i -th query:

$$\mathcal{L}_{\text{coord}}^{(i)} = \mathbf{1}_{\{i \in Pic(\sigma)\}} \sum_{j=1}^M \left\| e_j^{(\sigma^{-1}(i))} - \hat{e}_j^{(i)} \right\| \quad (7)$$

Character Classification Loss. Character recognition seems like a classification problem, where each class is a specific character. We use cross-entropy loss, it defined as:

$$\mathcal{L}_{\text{char}}^{(i)} = \mathbf{1}_{\{i \in Pic(\sigma)\}} \sum_{j=1}^A (-f_j^{(\sigma^{-1}(i))} \log \hat{f}_j^{(i)}) \quad (8)$$

4 Experimentations

4.1 Datasets

Table 1. Results of scene text spotting on Total-Text and CTW1500. “None ” denotes recognition without a lexicon. The “Full” lexicon contains all the words in the test set. Results style: **best**, second best.

Methods	Total-Text					CTW1500					FPS
	Detection			End-to-end		Detection			End-to-end		
	P	R	F	None	Full	P	R	F	None	Full	
Text Perceptron [33]	88.8	81.8	85.2	69.7	78.3	87.5	81.9	84.6	57.0	—	—
ABCNet [21]	—	—	—	64.2	75.7	—	—	81.4	45.2	74.1	6.9
MANGO [32]	—	—	—	72.9	83.6	—	—	—	58.9	78.7	4.3
ABCNet v2 [23]	90.2	84.1	87.0	70.4	78.1	85.6	<u>83.8</u>	84.7	57.5	77.2	10
Swintextspotter [14]	—	—	88.0	74.3	84.1	—	—	<u>88.0</u>	51.8	77.0	—
Abinet++ [8]	—	—	—	<u>79.4</u>	85.4	—	—	—	61.5	<u>81.2</u>	10.6
TESTR [47]	93.4	81.4	86.90	73.25	83.3	<u>89.7</u>	83.1	86.3	53.3	79.9	<u>5.5</u>
DeepSolo [45]	<u>93.1</u>	<u>82.1</u>	<u>87.3</u>	79.7	87.0	—	—	—	<u>60.01</u>	78.4	10
FastTextSpotter	90.58	85.46	<u>87.95</u>	75.14	<u>86.0</u>	91.45	85.16	88.19	56.02	82.91	5.38

For comparison with state-of-the-art methods, we selected the following benchmarks for experimental validation: **ICDAR 2015** [15], the official dataset of the ICDAR 2015 robust reading competition, is used for evaluating regular text spotting, employing the same test-train split as in the competition. **Total-Text** [3] is a widely recognized benchmark for arbitrary-shaped text spotting, providing word-level text instances for evaluation. **CTW1500** [22] serves as another benchmark for arbitrary-shaped text spotting, featuring sentence-level text instances. **Vin-Text** [30] is a Vietnamese text dataset utilized for assessing the performance of multilingual text spotting systems.

4.2 Implementation Details

The hyper-parameters for the deformable transformer [49] were configured similarly to the original implementation, with 8 attention heads and 4 sampling points utilized for the deformable attention mechanism. The number of layers for both the encoder and decoder was set to 6.

Data Augmentation. During pre-training, we apply data augmentation by randomly resizing the images, with the shorter edge ranging from 480 to 896 pixels, while constraining the longest edge to a maximum of 1600 pixels. Additionally, an instance-aware random cropping technique is employed.

Pre-training. We pre-trained FastTextSpotter using the SynthText150k [21], MLT 2017 [29], and TotalText [3] datasets over 4000K iterations, starting with a learning rate of 2×10^{-5} , which was reduced by a factor of 0.1 after 3000K iterations. The AdamW [26] optimizer was employed, with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 10^{-5} . We utilized $Q = 100$ composite queries, with 20 control points, and set the maximum text length to 25. The entire pre-training process was conducted on an RTX 3080 Ti GPU with a batch size of 1, spanning a total of 16 days.

Fine-Tuning. Following pre-training, we fine-tuned on the Total-Text and ICDAR2015 datasets for 200K iterations. For the CTW1500 dataset, FastTextSpotter was fine-tuned over 2000K iterations, with the maximum text length set to 100, given its sentence-level annotations that require additional training iterations. For the VinText dataset, the model was trained for 1000K iterations.

4.3 Comparison with State-of-the-Art

Table 1 summarizes the results of FastTextSpotter compared to other text spotting methods. We surpass the state-of-the-art Abinet++ [8] in end-to-end recognition on both the word-level Total-Text [3] and sentence-level CTW1500 [22] benchmarks. For scene text detection, our model outperforms SwinTextSpotter [14] on CTW1500 and achieves the second-best F-score on Total-Text. Notably, our model excels in recall metrics across both benchmarks. Qualitative examples are provided in Fig. 5.

We evaluated our method on the ICDAR15 benchmark [15] and compared it with state-of-the-art approaches (Table 2). For text detection, we achieved a 5% precision gain over TESTR [47] and slightly exceeded the best F-measure. In text spotting, our method delivered the *top performance in the challenging “Strong” category*, where each image contains a lexicon of only 100 words. We outperformed TESTR, SwinTextSpotter, and Abinet++ by roughly 1.5%, 2.5%, and 0.5%. Figure 5 shows our model’s performance on this dataset in the third column.

Text Spotting in Low-Resource. We also evaluated Vintext [30], a low-resource benchmark for Vietnamese scene text detection, to demonstrate our

Table 2. Results of scene text spotting ICDAR-15 datasets. “S”, “W”, and “G” are “Strong”, “Weak”, “Generic”, lexicas to recognise, respectively. Results style: **best**, second best.

Methods	Detection			End-to-end		
	P	R	F	S	W	G
Text Perceptron [33]	89.4	82.5	87.1	80.5	76.6	65.1
Unconstrained [34]	89.4	87.5	87.5	83.4	79.9	68.0
MANGO [32]	–	–	–	81.8	78.9	67.3
ABCNet v2 [23]	90.4	86.0	88.1	82.7	78.5	73.0
Swintextspotter [14]	–	–	–	83.9	77.3	70.5
TESTR [47]	90.3	89.7	90.0	85.2	79.4	73.6
Abinet++ [8]	–	–	–	86.1	81.9	77.8
PGNet [40]	91.8	84.8	88.2	83.3	78.3	63.5
DeepSolo [45]	<u>92.8</u>	<u>87.4</u>	<u>90.10</u>	86.8	81.9	<u>76.9</u>
FastTextSpotter	95.03	85.70	90.13	<u>86.63</u>	<u>81.67</u>	75.44

model’s generalizability. As shown in Table 3, our method outperforms the state-of-the-art by nearly 2%. Figure 5 illustrates our model’s performance on Vintext in the last column.

Why ABINet++ has Better Performance in Text Recognition? ABINet++ [8] and MANGO [32] leverage linguistic information for text spotting, enhancing their performance in this metric. Despite relying solely on a visual approach, our method outperforms both.

Performance vs Efficiency Trade-Off. Compared to previous approaches, our method shows optimum performance in terms of FPS which is **5.38** reported in Table 1. Our FPS is almost halved in comparison to ABINet++ and DeepSolo [45] approach. As depicted in Fig. 1, we observe the effectiveness of the SAC2 attention module introduced in the model (as shown in orange curve) when compared to the one with normal deformable attention [49]. The key explanation behind this phenomenon is the usage of cyclic convolutions which was previously proposed for real-time instance segmentation [31]. Using this layer on top of the self-attention module along with the reference point resampling strategy used for both position and character queries helps in faster convergence during training and a better end-to-end spotting h-mean.

Efficiency Comparison with MANGO. The MANGO text spotter [32] adopts a position-aware mask attention module to generate attention weights on each text instance and its characters to recognize character sequences without RoI operation. They achieve a slightly better FPS of 4.3 compared to ours owing to the fact that it’s a single stage model with no self-attention. However, Table 1 and Table 2 highlight the fact that utilizing self-attention and trans-

former frameworks helps in significant improvement in metrics for detection and recognition tasks.

Table 3. Text recognition performance of the proposed and the state-of-the-art systems on the Vintext datasets. Results style: **best**, second best.

Methods	H-mean
ABCNet + D [30]	57.4
ABCNet [21]	54.2
Mask Textspotter v3 + D [30]	68.5
Swintextspotter [14]	<u>71.1</u>
Mask Textspotter v3 [30]	53.4
FastTextSpotter(w/o fine-tune)	21.54
FastTextSpotter	72.95



Fig. 5. Some illustration of our method on different datasets. Zoom in for better visualization. First two images from Total-Text, third and fourth images from CTW1500, fifth and sixth images from ICDAR15, and the last two images from Vintext.

4.4 Ablation Studies

We conducted ablation studies to assess the significance of the various components within the FastTextSpotter framework, leading to the following insights.

Swin Transformer Serves as a Robust Visual Backbone for Text Spotting. We show a comparison of attention maps visualized in different layers for ResNet-50 backbone [13] when compared to the Swin-Tiny [24] variant. The Swin attention better captures the global interactions between the different objects to localize the text region which helps them to get a substantial gain over ResNet-50 which primarily captures more local spatial relationships with the attention. The hierarchical shifted-window mechanism is highly useful to control the attention on top of the text region boundaries to have a more complete understanding using better contextual cues. More empirical results on the utility of the Swin backbone for the text spotting has been illustrated in previous works [4, 5].

Effect of Reference Point Resampling and SAC2 Module. The effect of adding the SAC2 and reference point resampling strategies is shown in Table 4

where we clearly observe substantial gain in end-to-end recognition performance along with an optimal gain in detection too. Also using these modules helps us to gain faster convergence during the pre-training of FastTextSpotter.

Table 4. Effectiveness of SAC2 and Reference Point Resampling Modules. Performance obtained by pre-training model under this setting. “LD” and “CD” stand for Text Location Decoder and Text Recognition Decoder, respectively. ‘None’ indicates that no lexicon was used.

Modules			Detection			End-to-End
Reference Points Reampling	SAC2 in LD	SAC2 in CD	P	R	F	None
✗	✗	✗	89.63	70.34	78.82	60.56
✓	✗	✗	91.36	72.52	80.85	62.38
✓	✗	✓	91.75	74.35	82.13	65.46
✓	✓	✗	93.59	75.20	83.40	67.06
✓	✓	✓	91.3	77.33	83.68	68.87

5 Conclusion and Future Work

We proposed a novel efficient transformer model for text spotting, FastTextSpotter, which not only establishes itself as a robust and efficient solution in the field of text spotting but also excels in operational efficiency. It outperforms previous state-of-the-art models in both end-to-end text recognition and scene text detection tasks, notably achieving top recall metrics for the Total-Text and CTW1500 benchmarks. The model’s efficiency is highlighted by its enhanced processing speed and reduced computational demands compared to the existing SOTA models, making it well-suited for real-time applications. Moreover, we show the effectiveness of the model for spotting in multiple languages, namely English and Vietnamese. Expanding its capabilities to include a broader array of languages, especially those with complex scripts could significantly increase its applicability and utility. We plan to extend our evaluation to include diverse languages and scripts such as Arabic, Chinese, and Hindi, to further validate and enhance the model’s versatility and effectiveness in various linguistic contexts.

Acknowledgements. This work acknowledges the Spanish projects GRAIL PID2021-126808OB-I00, DocAI 2021-SGR-01559, the CERCA Program/Generalitat de Catalunya, and PhD Scholarship from AGAUR 2023 FI-3- 00223.

References

1. Atienza, R.: Vision transformer for fast and efficient scene text recognition. In: ICDAR 2021, Part I, pp. 319–334. Springer (2021)

2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV 2020, Part I, pp. 213–229. Springer (2020)
3. Ch'ng, C.K., Chan, C.S., Liu, C.L.: Total-text: toward orientation robustness in scene text detection. *Int. J. Doc. Anal. Recogn. (IJ DAR)* **23**(1), 31–52 (2020)
4. Das, A., Biswas, S., Banerjee, A., Lladós, J., Pal, U., Bhattacharya, S.: Harnessing the power of multi-lingual datasets for pre-training: Towards enhancing text spotting performance. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 718–728 (2024)
5. Das, A., Biswas, S., Pal, U., Lladós, J.: Diving into the depths of spotting text in multi-domain noisy scenes. In: 2024 IEEE International Conference on Robotics and Automation (ICRA), pp. 410–417. IEEE (2024)
6. Dong, Q., Tu, Z., Liao, H., Zhang, Y., Mahadevan, V., Soatto, S.: Visual relationship detection using part-and-sum transformers with composite queries. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3550–3559 (2021)
7. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
8. Fang, S., Mao, Z., Xie, H., Wang, Y., Yan, C., Zhang, Y.: ABINet++: autonomous, bidirectional and iterative language modeling for scene text spotting. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022)
9. Fang, S., Xie, H., Wang, Y., Mao, Z., Zhang, Y.: Read like humans: autonomous, bidirectional and iterative language modeling for scene text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7098–7107 (2021)
10. Feng, W., He, W., Yin, F., Zhang, X.Y., Liu, C.L.: TextDragon: an end-to-end framework for arbitrary shaped text spotting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9076–9085 (2019)
11. Garcia-Bordils, S., Karatzas, D., Rusiñol, M.: Step-towards structured scene-text spotting. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 883–892 (2024)
12. 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)
13. 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)
14. Huang, M., et al.: SwinTextSpotter: scene text spotting via better synergy between text detection and text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4593–4603 (2022)
15. Karatzas, D., et al.: ICDAR 2015 competition on robust reading. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1156–1160. IEEE (2015)
16. Li, H., Wang, P., Shen, C.: Towards end-to-end text spotting with convolutional recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5238–5246 (2017)
17. Liao, M., Pang, G., Huang, J., Hassner, T., Bai, X.: Mask TextSpotter v3: segmentation proposal network for robust scene text spotting. In: European Conference on Computer Vision, pp. 706–722. Springer (2020)
18. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: TextBoxes: a fast text detector with a single deep neural network. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)

19. Liu, S., et al.: DAB-DETR: dynamic anchor boxes are better queries for DETR. In: International Conference on Learning Representations (2022)
20. Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., Yan, J.: FOTS: fast oriented text spotting with a unified network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5676–5685 (2018)
21. Liu, Y., Chen, H., Shen, C., He, T., Jin, L., Wang, L.: ABCNet: real-time scene text spotting with adaptive Bezier-curve network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9809–9818 (2020)
22. Liu, Y., Jin, L., Zhang, S., Luo, C., Zhang, S.: Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recogn.* **90**, 337–345 (2019)
23. Liu, Y., et al.: ABCNet v2: adaptive Bezier-curve network for real-time end-to-end text spotting. arXiv preprint [arXiv:2105.03620](https://arxiv.org/abs/2105.03620) (2021)
24. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)
25. Long, S., Qin, S., Pantelev, D., Bissacco, A., Fujii, Y., Raptis, M.: Towards end-to-end unified scene text detection and layout analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1049–1059 (2022)
26. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101) (2017)
27. Lyu, P., Liao, M., Yao, C., Wu, W., Bai, X.: Mask TextSpotter: an end-to-end trainable neural network for spotting text with arbitrary shapes. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 67–83 (2018)
28. Nayef, N., et al.: ICDAR2019 robust reading challenge on multi-lingual scene text detection and recognition-RRC-MLT-2019. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1582–1587. IEEE (2019)
29. Nayef, N., et al.: ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification-RRC-MLT. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 1454–1459. IEEE (2017)
30. Nguyen, N., et al.: Dictionary-guided scene text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7383–7392 (2021)
31. Peng, S., Jiang, W., Pi, H., Li, X., Bao, H., Zhou, X.: Deep snake for real-time instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8533–8542 (2020)
32. Qiao, L., et al.: MANGO: a mask attention guided one-stage scene text spotter. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 2467–2476 (2021)
33. Qiao, L., et al.: Text perceptron: towards end-to-end arbitrary-shaped text spotting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11899–11907 (2020)
34. Qin, S., Bissacco, A., Raptis, M., Fujii, Y., Xiao, Y.: Towards unconstrained end-to-end text spotting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4704–4714 (2019)
35. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: a metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 658–666 (2019)

36. Singh, A., Pang, G., Toh, M., Huang, J., Galuba, W., Hassner, T.: TextOCR: towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8802–8812 (2021)
37. Sun, Y., Zhang, C., Huang, Z., Liu, J., Han, J., Ding, E.: TextNet: irregular text reading from images with an end-to-end trainable network. In: ACCV 2018, Part III, pp. 83–99. Springer (2019)
38. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
39. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 International Conference on Computer Vision, pp. 1457–1464. IEEE (2011)
40. Wang, P., et al.: PGNet: real-time arbitrarily-shaped text spotting with point gathering network. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 2782–2790 (2021)
41. Wang, W., Zhang, J., Cao, Y., Shen, Y., Tao, D.: Towards data-efficient detection transformers. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022)
42. Wang, W., et al.: PAN++: towards efficient and accurate end-to-end spotting of arbitrarily-shaped text. *IEEE Trans. Pattern Anal. Mach. Intell.* (2021)
43. Wang, X., Jiang, Y., Luo, Z., Liu, C.L., Choi, H., Kim, S.: Arbitrary shape scene text detection with adaptive text region representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6449–6458 (2019)
44. Ye, M., Zhang, J., Zhao, S., Liu, J., Du, B., Tao, D.: DPText-DETR: towards better scene text detection with dynamic points in transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 3241–3249 (2023)
45. Ye, M., et al.: DeepSolo: let transformer decoder with explicit points solo for text spotting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19348–19357 (2023)
46. Zhang, R., et al.: ICDAR 2019 robust reading challenge on reading Chinese text on signboard. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1577–1581. IEEE (2019)
47. Zhang, X., Su, Y., Tripathi, S., Tu, Z.: Text spotting transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9519–9528 (2022)
48. Zheng, D., Dong, W., Hu, H., Chen, X., Wang, Y.: Less is more: focus attention for efficient DETR. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6674–6683 (2023)
49. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: deformable transformers for end-to-end object detection. *arXiv preprint [arXiv:2010.04159](https://arxiv.org/abs/2010.04159)* (2020)



Vietnamese Scene Text Detection via Edge Information and Text Region Feature Enhancement

Liyu Jiang¹, Shaoliang Shi², Wenhui Huang¹, Zhengli Xu¹, Vinh Loc Cu³,
and Yimin Wen¹(✉)

¹ Guilin University of Electronic Technology, Guilin, China
xu_zhengli@gsgsg.uum.edu.my, ymw@guet.edu.cn

² China Academy of Science & Technology Development GuangXi Branch,
Nanning, China

³ Can Tho University, Can Tho, Vietnam
cvloc@ctu.edu.vn

Abstract. Scene text detection plays a crucial role in the development of computer vision. However, current scene text detection algorithms mainly focus on Chinese and English, while Vietnamese scene text detection still remains a challenging task. The current algorithms to detect Vietnamese scene text frequently result in the incapacity of detecting Vietnamese diacritics and wrongly detecting background as text. To address these challenges, in this paper, a Vietnamese scene text detection algorithm is proposed to concentrate on diacritics and effectively reduce background interference. Specifically, an Edge Information Enhancement Module (EIEM) is first proposed to enhance the edge features of Vietnamese characters by combining a gradient filter with an attention mechanism. Secondly, a Text Region Enhancement Module (TREM) is proposed to enhance the feature representation of text regions by capturing global contextual information and dependencies among Vietnamese characters, thereby enhancing the distinction between background and text. Experiments on the Vintext dataset illustrate that the proposed method performs better in Vietnamese scene text detection tasks compared with several contemporary scene text detection algorithms. The code of the proposed algorithm is available at <https://github.com/mlmmwym/VSTD-EITRFE>.

Keywords: Scene text detection · Vietnamese characters · Diacritics · Feature pyramids

To whom correspondence should be addressed. This work was supported by the Key R&D Program of Guangxi under Grant (AB21220023), the National Natural Science Foundation of China (62366011), and Guangxi Key Laboratory of Image and Graphic Intelligent Processing (GIIP2306), Innovation Project of GUET Graduate Education(2023YCXB11,2024YCXS047).

1 Introduction

Scene text detection, a critical task in computer vision, holds vast potential applications across various domains such as education, logistics, and tourism. In recent years, significant advancements have been achieved in this field, showcasing a substantial body of outstanding work [1–5]. However, most of these studies predominantly focus on detecting Chinese or English texts in natural scenes, while there is limited research dedicated to Vietnamese scene text. However, according to statistics [6], about 85.3 million people speak Vietnamese globally, and it is the 20th language in the world in terms of the number of speakers. The number of international tourists visiting Vietnam has increased from 10 million in 2016 to 18 million in 2019. After 2022, Vietnam’s tourism industry recovered rapidly, and in 2023, Vietnam ranked the 8th among the top 10 countries globally with the highest growth rates. Therefore, addressing the detection and recognition challenges of Vietnamese texts in natural scenes will bring significant convenience to tourists.

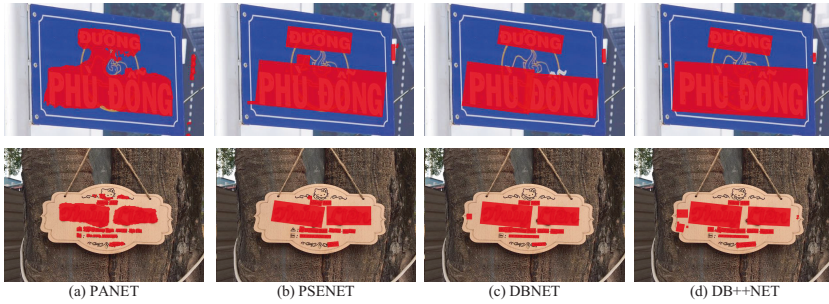


Fig. 1. The detection results of several contemporary scene text detection algorithms on the dataset of Vintext

Vietnamese is a tonal language, and its characters are composed of Latin letters and diacritics. There are primarily nine diacritics, with four symbols combined with vowels, and the remaining five indicating tones in Vietnamese. The different combinations of Latin letters and diacritics will result in distinct meanings. For example, the meanings of “thiên” and “thiện” represent the sky and virtue, respectively. “dưa”, “dừa”, and “dứa” are three distinct fruits: watermelon, coconut, and pineapple, respectively. Thus, subtle changes in a Vietnamese character can result in significantly different meanings.

Even though there have been many excellent works in scene text detection tasks for Chinese and English characters, applying these methods to handle Vietnamese scene text detection is still challenging. In Fig. 1, we show some results of several contemporary scene text detection methods, while in Fig. 6, we show the qualitative results of these methods on the Vintext dataset [7]. It can be observed that diacritics are often omitted or insufficiently detected, and

backgrounds are mistakenly recognized as text. We selected 100 images from the Vintext dataset and performed a statistical analysis of diacritics detection, as shown in Table 4. It can be observed that these contemporary scene text detection methods cannot detect diacritics very well.

There are several reasons for these wrong detection results. Firstly, the diacritics of Vietnamese are smaller compared to Latin letters, which makes the diacritics to be easily mixed with its surrounding background. Existing methods do not pay special attention to this situation during feature extraction, which leads to ineffective extraction of diacritic’s features and makes them more susceptible to interfere with background information during detection. Secondly, the unique structure formed by the combination of Latin letters and diacritics can make certain text-like objects more susceptible to being mistakenly identified as text targets.

In this paper, a new Vietnamese scene text detection method is proposed to effectively address the challenges mentioned above. Specifically, we use DBNet [2] as the baseline. Our choice of DBNet is supported by its outstanding performance in scene text detection, coupled with its exceptional ability to process texts of varying shapes, sizes, and orientations. The main contributions of this paper are as follows:

- We proposed an edge information enhancement module (EIEM) to utilize gradient filter which consists of a Sobel filter and improved channel attention. It enhances the diacritic’s features by extracting relevant Vietnamese text edge information which is then integrated into the low layers of the backbone network.
- We proposed a text region enhancement module (TREM) to first capture global contextual information and dependencies between Vietnamese characters, and then enhance the feature representation of text region by utilizing this information to adjust all shallow features to intensify the differentiation between background and text. Thus, TREM reduces the interference of background information.
- We conducted extensive experiments on the Vintext dataset and compared it with five well-established scene text detection methods to demonstrate the superiority and reasonableness of our proposed method.

2 Related Work

In this section, we briefly introduce the research related to scene text detection, edge detection methods, and feature pyramids.

2.1 Scene Text Detection

Currently, there has already been a substantial amount of excellent work on scene text detection tasks. Here, we only select several well-known segmentation-based algorithms for a brief introduction. And then, we especially introduce some work for Vietnamese scene text detection.

Segmentation-based methods classify pixels to obtain a probability map of text regions, and bounding boxes are obtained through a post-processing algorithm. This type of approach primarily relies on Mask R-CNN [1] and Fully Convolutional Networks (FCN) [8] for text detection. Wang et al. proposed PSENet [5], which learns the segmentation regions of text through a progressive scale expansion algorithm. Subsequently, PANNet [9] was introduced to address the complex post-processing in PSENet, proposing a cost-effective segmentation module and a learnable post-processing approach. Liao et al. proposed DBNet [2], which makes the threshold binarization process differentiable to optimize segmentation prediction results. DBNet++ [3] adds the Adaptive Scale Fusion (ASF) module on top of DBNet, enhancing the robustness of the fused features.

There are limited studies on Vietnamese scene text detection, Nguyen et al. [7] proposed the Vintext dataset, which is the first publicly available Vietnamese scene text dataset. In addition, they proposed a new way of incorporating language models into text recognition to solve the problem of visual ambiguity in scene text recognition. Pham et al. [10] conducted an empirical study using the Vintext dataset in DBNet [2], PMTD [11], PANNet [9] and FCEN [12] to test the effectiveness of these four detection models. Huang et al. [13] proposed a new recognition transformation mechanism that co-optimizes the detection and recognition parts in the same architecture, which outperforms ABCNet+D [7] on the Vintext dataset. Huang et al. [14] first proposed an algorithm for Vietnamese scene text detection, which improves detection accuracy by enhancing diacritic's features, introducing IoU-v matching boundaries, and fusing diacritic's features.

2.2 Edge Detection and Feature Pyramid

Edge detection and feature pyramid are pivotal techniques in image processing and pattern recognition.

Edge in an image is delineated by a string of neighboring pixels that display pronounced variations in their brightness, marking the transition between distinct visual elements [15]. The edge detection algorithm proposed by Sobel in 1970 has evolved into a fundamental technique in image processing [16]. Chetia et al. [17] designed an improved Sobel detection method, which employs non-maximum suppression and dual thresholding strategies to effectively overcome the problem of incomplete extraction of information from the edges of the image by the traditional Sobel algorithm. Tian et al. [18] improved Sobel operator and proposed a new image denoising algorithm, which improves the performance of Sobel operator in processing noisy images.

The feature pyramid makes the development of computer vision faster. Liu et al. proposed SSD [19] to utilize feature maps of various sizes to detect targets at different scales, but due to its unique structure, it often performs poorly in detecting small objects. To solve the shortcomings of SSD, Lin et al. proposed FPN [20], which fuses the features of different feature maps through a top-down path setup so that the bottom-level feature maps also have better semantic information. Liu et al. proposed PANet [4], which is based on FPN, and adds a

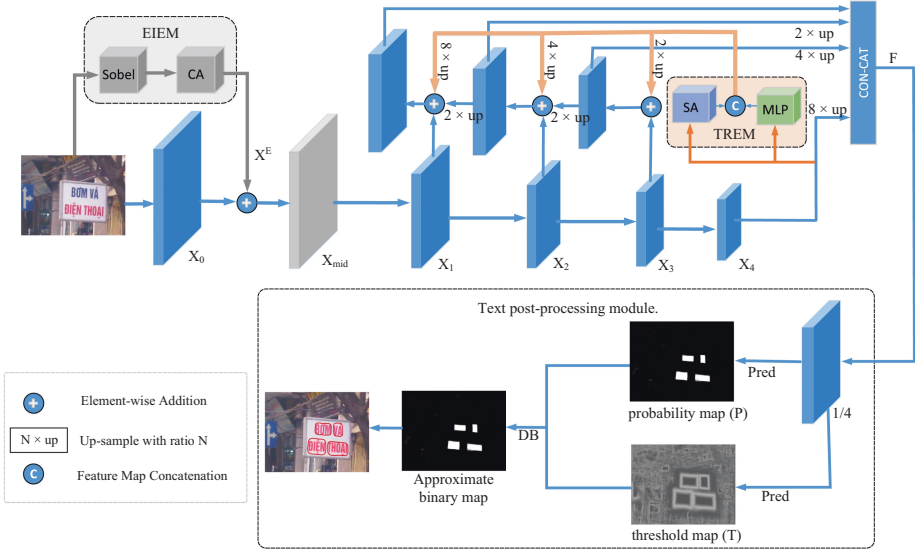


Fig. 2. The architecture of the proposed Vietnamese scene text detection algorithm

bottom-up path so that the high-level features can also obtain enough information about the bottom-level features. Overall, the feature pyramid, by extracting features at multiple scales, provides more comprehensive information for computer vision tasks, enhancing the adaptability and performance of models across different scenarios.

Inspired by the above work, we combine the sobel operator with an attention mechanism to enhance the edge information of Vietnamese scene texts in the network. Then, we propose a text region enhancement method that allows efficient interactions between features from deep and shallow layers, thus enabling the network to extract richer and more effective features.

3 Methodology

In this section, we present the implementation details of the proposed Vietnam Scene Text Detection Algorithm. Firstly, the overall framework of the algorithms is introduced in Sect. 3.1, and then the two proposed modules, EIEM and TREM, are introduced in Sects. 3.2 and 3.3, respectively.

3.1 Overall Framework

The overall framework of the proposed method is illustrated in Fig. 2. It includes four main components: (1) A CNN backbone network for extracting visual feature pyramids; (2) Edge information enhancement module (EIEM); (3) Text region enhancement module (TREM); (4) Text post-processing module.

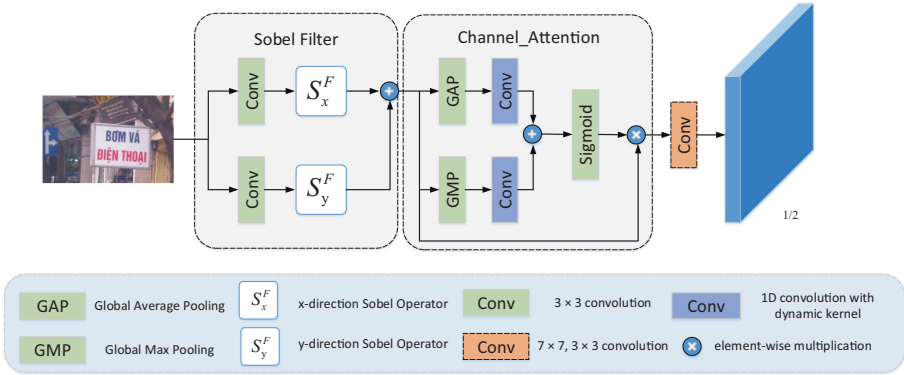


Fig. 3. The framework of the edge information enhancement module

Specifically, an image is first input into the backbone network (i.e., Resnet-50) to extract the features of X_i ($i = 0, 1, 2, 3, 4$), and their sizes are in $1/2, 1/4, 1/8, 1/16$, and $1/32$ of the input image, respectively. EIEM is designed to extract text edge detail information X^E from the input image. After EIEM, X^E is added with X_0 to obtain the intermediate feature X_{mid} which includes rich edge information.

$$X_{mid} = Conv_{3 \times 3, 1 \times 1} (X^E \oplus X_0) \tag{1}$$

TREM is implemented at the top level of the feature pyramid (i.e., X_4) to capture the global contextual information in X_4 and the dependencies between Vietnamese characters.

Then, the output of TREM is up-sampled to modulate X_1, X_2 , and X_3 , respectively, cascading to generate new feature maps. These new feature maps, along with X_4 , are all then upsampled to the same size and concatenated to obtain the feature map F . Then F is input into the text post-processing module to obtain a probability map P and an adaptive threshold map T that matches the dimensions of the original input image. Finally, a differentiable binarization operation is performed on P and T to obtain an approximate binary map, thereby determining the boundaries of text boxes.

3.2 Edge Information Enhancement Module

In Vietnamese scene text detection, the detailed information on the edge of Vietnamese characters cannot be well extracted using only convolutional neural networks (CNN), because the extracted feature of small-sized diacritics easily interfered with background noise in natural scenes. Moreover, with increasing network depth, the down-sampling process of CNN may lose the detailed information of diacritics.

Therefore, inspired by Park et al. [21] and as shown in Fig. 3, EIEM is proposed to enhance the edge information of Vietnamese scene text to achieve higher

detection accuracy. The horizontal gradient X^x is first obtained through a 3×3 convolution and a horizontal Sobel operation, while a 3×3 convolution with a vertical Sobel operation is deployed in the vertical direction to compute the vertical gradient X^y . Subsequently, both the horizontal and vertical gradient information are added to output X^G . X^G is then fed into the subsequent channel attention to compute attention weights, which are then multiplied by the corresponding elements of the feature maps X^G , and then passed through a convolutional layer to obtain the final feature maps X^E . The above process can be expressed as follows:

$$X^x = Sobel_x (Conv_{3 \times 3} (X^{RGB})) \quad (2)$$

$$X^y = Sobel_y (Conv_{3 \times 3} (X^{RGB})) \quad (3)$$

$$X^G = X^x + X^y \quad (4)$$

where $Sobel_x(\cdot)$ and $Sobel_y(\cdot)$ represent sobel filters. $Conv_{3 \times 3}(\cdot)$ denotes the convolution of 3×3 , X^x and X^y represent the horizontal and vertical gradient information, respectively.

$$X^E = Conv (Channel_Attention (X^G)) \quad (5)$$

where $Channel_Attention$ represents our proposed channel attention, $Conv(\cdot)$ denotes the convolution of 7×7 , 3×3 .

Traditional efficient channel attention first utilizes global average pooling to obtain aggregated features, and then generates attention weights by 1D convolution with dynamic kernel and sigmoid activation function. In EIEM, we improved the traditional channel attention by adding an additional global max pooling branch, and then fusing the outputs of global average pooling and global max pooling after 1D convolution to obtain the weights for each channel via a sigmoid function. The global max pooling operation can help to retain the most significant edge features in the input features, and combining it with the global average pooling can make the network focus more on the edge part. Finally, these weights are multiplied by the corresponding elements of X^G and the results are passed through two convolution operations to obtain X^E which includes detailed information on character's edge.

3.3 Text Region Enhancement Module

As shown in Fig. 4, the proposed TREM is composed of two parallel parts, where the MLP part is used to capture global contextual information from the top-level feature X_4 while the Strip Attention (SA) part is used to capture the dependencies between Vietnamese characters. The results of these two parts are fused as the final output of TREM. The above process can be expressed as:

$$X_{out} = Conv_{1 \times 1} (Concat (MLP (X_{in}), SA (X_{in}))) \quad (6)$$

where X_{out} denotes the output of TREM, X_{in} is the output of performing convolution, batch normalization, and ReLU on X_4 . $Concat(\cdot)$ denotes the connection

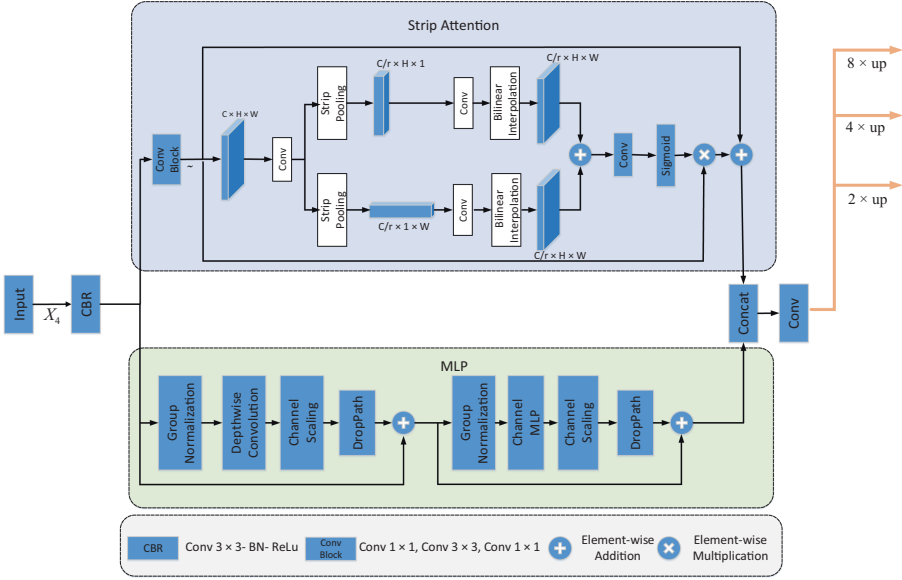


Fig. 4. The framework of the text region enhancement module

along channel dimension. $SA(X_{in})$ and $MLP(X_{in})$ denote the outputs of the SA and MLP part, respectively. $Conv_{1 \times 1}(\cdot)$ represents the 1×1 convolution. The computing of X_{in} is presented in (7).

$$X_{in} = ReLU(BN(Conv_{7 \times 7}(X_4))) \tag{7}$$

where $Conv_{7 \times 7}(\cdot)$ denotes a 7×7 convolution with step size 1, $BN(\cdot)$ denotes the batch normalisation, and $ReLU(\cdot)$ denotes the ReLU activation function.

Inspired by PoolForme [22], in this section, the MLP part is designed to capture the global contextual information while effectively preserving the spatial information in the 2D feature representation of a scene image, which helps the network to recognize text regions. The MLP part is mainly composed of two serial residual blocks. Specifically, X_{in} is first processed by the group normalization and then sent to the depth-wise convolution layer. Because traditional pooling reduces the spatial dimension of the data by down-sampling, resulting in loss of spatial information, which may have a great negative impact on Vietnamese scene detection. Thus, compared with pooling, depth-wise convolution is chosen to perform feature extraction while maintaining spatial resolution, which can better preserve important spatial information. Its output is subsequently processed through channel scaling and dropout. Finally, the residual connection with X_{in} is element-wisely added. The above process is specifically represented as follows:

$$\hat{X}_{in} = (Drop(Scale(DConv_{1 \times 1}(GN(X_{in})))) \oplus X_{in} \tag{8}$$

where \hat{X}_{in} represents the output of the first residual block, $GN(\cdot)$ represents the group normalisation, $DConv_{1 \times 1}(\cdot)$ represents the depth-wise convolution with kernel size 1×1 , $Scale(\cdot)$ represents channel scaling, $Drop(\cdot)$ represents dropout, and \oplus represents element-wise addition.

In the latter residual block, group normalization is first performed on \hat{X}_{in} . Subsequently, channel MLP [23] is applied to the outputs of the previous operation. Finally, after implementing channel scaling, dropout, and residual connection on \hat{X}_{in} , the final output is presented as follows:

$$MLP(X_{in}) = \left(Drop \left(Scale \left(CMLP \left(GN \left(\hat{X}_{in} \right) \right) \right) \right) \right) \oplus \hat{X}_{in} \quad (9)$$

where $CMLP(\cdot)$ represents the channel MLP [23], and the rest of the symbols have the same meaning as above.

In the SA part, we proposed to employ strip pooling [24] to implement an attention mechanism, for the reason that strip pooling only captures both horizontal and vertical contexts while our attention mechanism not only captures contextual information but also reinforces dependencies between characters.

X_{in} is first passed through multiple serial convolution layers to yield \tilde{X}_{in} .

$$\tilde{X}_{in} = Conv(X_{in}) \quad (10)$$

where $Conv(\cdot)$ denotes the serial convolutions of 1×1 , 3×3 , and 1×1 . Then, the number of channels of \tilde{X}_{in} is compressed to $\frac{C}{r}$ through 1×1 convolution operation. In this paper, the hyper-parameter r is set to 4.

Subsequently, two strip pooling layers are employed in parallel to produce horizontal and vertical stripe features. Subsequently, a 1D convolution with a kernel size of 3 is used to adjust the dependency relationships between features at each position, and bi-linear interpolation is then employed for up-sampling to obtain $y^h \in \mathbb{R}^{\frac{C}{r} \times H \times W}$ and $y^v \in \mathbb{R}^{\frac{C}{r} \times H \times W}$, which are then fused together by element-wise addition to obtaining $z \in \mathbb{R}^{\frac{C}{r} \times H \times W}$.

$$z = y^h \oplus y^v \quad (11)$$

Then, the attention matrices are obtained using a 1×1 convolution and a sigmoid activation function. The weighted sum of the relevant attention matrices with \tilde{X}_{in} is element-wisely added with \tilde{X}_{in} to obtain the final output. The final output can then be expressed as:

$$SA(X_{in}) = \left(\tilde{X}_{in} \otimes \sigma(Conv_{1 \times 1}(z)) \right) \oplus \tilde{X}_{in} \quad (12)$$

where $Conv_{1 \times 1}(\cdot)$ represents the 1×1 convolution, $\sigma(\cdot)$ denotes the sigmoid function, and \otimes denotes element-wise multiplication, and \oplus represents the element-wise addition.

3.4 Text Post-processing Module and Loss Function

Both the text post-processing module and loss function are consistent with those in DBNet [2]. Differentiable binarization algorithm is used in the post-processing

part, and its formula is:

$$\widehat{B}_{i,j} = \frac{1}{1 + e^{-K(P_{i,j} - T_{i,j})}} \quad (13)$$

where $\widehat{B}_{i,j}$, $\widehat{P}_{i,j}$, and $\widehat{T}_{i,j}$ represents the value of pixel (i, j) in the binary map, the probability map, and the adaptive threshold map, respectively. K is a parameter set to 50.

In this paper, we use DBNet’s overall loss function, where the specific parameter settings remain consistent with it. The expression is as follows:

$$L = L_s + \alpha \times L_b + \beta \times L_t \quad (14)$$

where L_s , L_b , and L_t denote the loss of the probability map, the binary map, and the threshold map, respectively. α , β are hyperparameters for balancing loss.

4 Experiments

Our experiments were performed on the Vintext dataset [7]. The entire Vintext dataset contains 2000 images, including 1200 training images, 300 validation images, and 500 test images, with 56,084 text instances.

We use ResNet-50 pre-trained on the ImageNet dataset [25] as the backbone. During the training process, we employed the adaptive moment estimation (Adam) optimizer with an initial learning rate of 0.001, while enabling the AMSGrad optimization method. At the same time, we employ a “poly” learning rate policy, which adjusts the learning rate for each iteration based on the rule: the learning rate of the current iteration is equal to the initial learning rate multiplied by $\left(1 - \frac{iter}{max_iter}\right)^{0.9}$. The *max_iter* denotes the maximum number of iterations, depending on the maximum epochs and the number of batches. The network is trained on the Vintext dataset for 250 epochs with the batchsize set to 16. In addition, our enhancement methods for the training data include: (1) random cropping; (2) random flipping; (3) random rotation, with the rotation range set to $(-10^\circ, 10^\circ)$. Finally, resize the image to 640×640 . Our method employs precision, recall, and F-measure as performance evaluation metrics.

Table 1. Ablation Results

EIEM	TREM	Precision	Recall	F-measure
✘	✘	88.1	78.9	83.2
✓	✘	92.6	79.6	85.6
✘	✓	93.3	79.4	85.8
✓	✓	92.7	81.0	86.5

4.1 Ablation Experiment

To verify the validity of EIEM and TREM, a series of experiments have been performed on the Vintext dataset. Table 1 illustrates the results of the ablation experiments, where “✗” means that the corresponding module is not used while “✓” means that the corresponding module is used. And thus, the first row of Table 1 presents the results of DBNet [2], which means the baseline.

The second row of Table 1 shows that the employment of the EIEM results in improvements of 4.5%, 0.7%, and 2.4% on precision, recall, and F-measure, respectively, compared to the baseline. The reason is that the EIEM provides more detailed textual edge information, which enhances the diacritic’s features for subsequent processing. To further verify the effectiveness of the EIEM, we visualized the feature maps after EIEM employment and the feature maps of baseline, as shown in Fig. 5. It can be observed that EIEM employment can obtain richer text edge information, and the features of diacritics are more prominent.

The third row of Table 1 shows that the employment of TREM results in improvements of 5.2%, 0.5%, and 2.6% on precision, recall, and F-measure, respectively, compared to the baseline. The observed outcome can be attributed to TREM’s capability to augment text region feature representation by capturing global contextual information and dependencies among Vietnamese characters. This information conditions the shallow network, enhancing its ability to recognize text regions while reducing the interference of the background.

Finally, the fourth row of Table 1 demonstrates that significant improvements have been achieved on all three evaluation metrics. Compared with the baseline, they are improved by 4.6%, 2.1%, and 3.3% respectively.

4.2 Comparison with Other Methods

Firstly, our method is compared with five segmentation-based methods, namely Mask R-CNN [1], PANNET [9], PSNet [5], DBNet [2], and DBNet++ [3] on the Vintext dataset. The results are presented in Table 2.

According to Table 2, it can be observed that our method achieves on precision, recall, and F-measure by 92.7%, 81.0%, and 86.5%, respectively. When compared with Mask R-CNN [1], PANNET [9], PSENET [5], DBNET [2], and

Table 2. The results of different methods on the Vintext dataset

Method	Precision	Recall	F-measure	Params (10^6)	BS	Optimizer	Lr	Epochs
Mask R-CNN	83.2	86.8	85.0	42.3	4	SGD	$2e-3$	26
PANNET	84.1	60.0	70.0	24.8	8	Adam	$1e-3$	250
PSENET	85.6	78.2	81.3	29.2	8	Adam	$1e-3$	250
DBNET	88.1	78.9	83.2	28.7	16	Adam	$1e-3$	250
DBNET++	89.4	79.8	84.3	29.3	16	Adam	$1e-3$	250
Ours	92.7	81.0	86.5	29.1	16	Adam	$1e-3$	250



Fig. 5. Visualization of feature maps after the first stage of Resnet, where the top row includes original images, the middle row presents the feature maps when EIEM was not used, and the bottom row presents the feature maps when EIEM was used. In addition, to make the images clear, we have standardized the scale changes

DBNET++ [3], our method achieves improvements on F-measure by 1.5%, 16.5%, 5.2%, 3.3%, and 2.2%, respectively.

Secondly, our method is compared with two special scene text recognition methods, ABCNET+D [7] and SwinTextSpotter [13], which have been evaluated on the Vintext dataset, and the results are shown in Table 3. It can be observed that our method still outperforms both on the F-measure.

Table 3. Comparison between two special methods and ours on the Vintext dataset

Method	Precision	Recall	F-measure
ABCNET+D	90.1	80.2	84.8
SwinTextSpotter	95.6	75.1	84.1
Ours	92.7	81.0	86.5

Thirdly, to further illustrate the effectiveness of our method in diacritics detection. We selected the first 100 images from Vintext’s test set and performed a statistical analysis on the instances whose diacritics were detected incorrectly. As shown in Table 4, the error rate of our method is only 11.7%, significantly lower than the other methods.

Fourthly, to evaluate whether the background is wrongly detected as text, experiments are conducted by applying the aforementioned detection algorithms

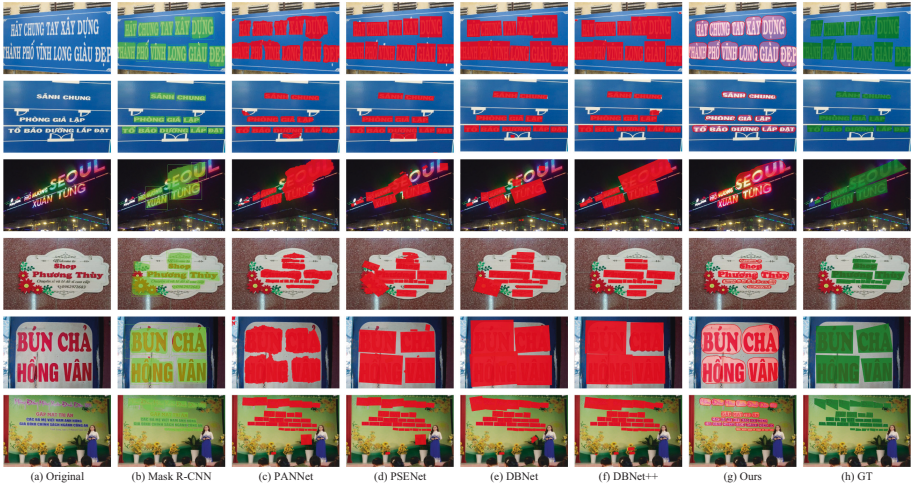


Fig. 6. Qualitative results of different algorithms on the dataset of Vintext

and ours on the ICDAR2015 and Vintext datasets. From Table 5, it is very interesting to observe that all comparison methods achieved higher rates of wrongly detecting background as text on Vintext than on ICDAR2015. These results suggest that the complex construction of Vietnamese characters makes the detection of Vietnamese scene text more challenging compared to English scene text detection. Furthermore, it can be observed that our method gets a significant decrease on the rates of wrongly detecting background as text on Vintext. These results illustrate our method is more suitable for Vietnamese scene text detection tasks.

Table 4. Detection results of different algorithms in diacritics detection

Method	Total Instances	Errors Instances	Error Rate
Mask R-CNN	1116	204	18.3%
PANNET	1116	520	46.6%
PSENET	1116	231	20.7%
DBNET	1116	212	19.0%
DBNET++	1116	190	17.0%
Ours	1116	131	11.7%

It should be noted that the experimental results in Table 4 and Table 5 were obtained by taking the average of the statistical results obtained by two researchers in our laboratory, so there may be certain errors. However, the differences in the experimental results are significant, and it can be believed that our experimental results still have some reliability.

Table 5. The rates of wrongly detecting background as text on ICDAR2015 and Vintext

Method	ICDAR2015	Vintext
Mask R-CNN	22%	29.6%
PANNET	7.8%	24.6%
PSENET	17.2%	21.4%
DBNET	13%	30%
DBNET++	10%	41.6%
Ours	–	5.6%

To visually explain the above results more intuitively, we selected several text images from Vintext to perform text detection. The results are shown in Fig. 6. The detection results indicate that the above-mentioned detection methods struggle to fully detect diacritics, however, there are still serious wrongly detecting background as text and incorrect diacritics detection. In contrast, our method demonstrates better detection of diacritics and distinguishes backgrounds more effectively.

Table 6. Dection results of DBNet and ours on the ICDAR 2015 dataset

Method	Precision	Recall	F-measure
DBNet	88.06	77.14	82.24
Ours	91.74	73.12	81.37

Finally, we verified the generalization performance of our proposed method on other tasks. We employ the ICDAR 2015 [26] dataset for evaluation. From Table 6, It can be observed that our proposed method only reduces 0.87% on F-measure, while ours does better than DBNet on precision and worse on recall. Therefore, it can be believed that even though our method is specially designed for Vietnamese scene text detection task, it also generalizes comparably to English scene text detection tasks.

5 Conclusion

In this paper, we proposed a new method for Vietnamese scene text detection. Firstly, we proposed an Edge Information Enhancement Module to enhance the edge detail information of Vietnamese text to augment the features of diacritics. Then, we proposed a Text Region Enhancement Module to improve the distinction between background and text, effectively reducing the background interference with Vietnamese scene text. Comprehensive experiments illustrated that the shortcomings of several existing algorithms in Vietnamese scene text

detection have been overcome to a certain extent and our method achieves better performance in Vietnamese scene text detection.

In addition, we believe that our approach is not only applicable to Vietnamese scene text detection but also to other similar tonal languages. However, the absence of public datasets of other similar tonal languages makes it difficult to prove our idea. For future work, we aim to develop scene text detection datasets for more tonal languages and evaluate our model's performance on these datasets.

References

1. 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)
2. Liao, M., Wan, Z., Yao, C., Chen, K., Bai, X.: Real-time scene text detection with differentiable binarization. In: Proceedings of the AAAI conference on artificial intelligence, vol. 34, pp. 11474–11481 (2020)
3. Liao, M., Zou, Z., Wan, Z., Yao, C., Bai, X.: Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(1), 919–931 (2022)
4. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018)
5. Wang, W., et al.: Shape robust text detection with progressive scale expansion network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9336–9345 (2019)
6. Thinh, V.X.: Exploring the Vietnamese language: history, dialects, and essential tourist phrases (2024). Accessed 11 Aug 2024
7. Nguyen, N., et al.: Dictionary-guided scene text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7383–7392 (2021)
8. 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)
9. Wang, W., et al.: Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8440–8449 (2019)
10. Pham, N.T., Nguyen-Van, Q., Nguyen, B.H., Dang, D.N.M., Nguyen, S.D., et al.: Vietnamese scene text detection and recognition using deep learning: an empirical study. In: Proceedings of the IEEE Conference on Green Technology and Sustainable Development (GTSD), pp. 213–218 (2022)
11. Liu, J., Liu, X., Sheng, J., Liang, D., Li, X., Liu, Q.: Pyramid mask text detector. arXiv preprint [arXiv:1903.11800](https://arxiv.org/abs/1903.11800) (2019)
12. Zhu, Y., Chen, J., Liang, L., Kuang, Z., Jin, L., Zhang, W.: Fourier contour embedding for arbitrary-shaped text detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3123–3131 (2021)
13. Huang, M., et al.: SwinTextSpotter: scene text spotting via better synergy between text detection and text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4593–4603 (2022)

14. Huang, W., Shi, S., Wen, Y.: Focusing on diacritics to improve Vietnamese scene text detection (to be in press). In: Proceedings of the International Joint Conference on Neural Networks (2024)
15. Zheng, Z., Zha, B., Yuan, H., Xuchen, Y., Gao, Y., Zhang, H.: Adaptive edge detection algorithm based on improved grey prediction model. *IEEE Access* **8**, 102165–102176 (2020)
16. Jain, R., Kasturi, R., Schunck, B.G., et al.: *Machine Vision*, vol. 5. McGraw-Hill, New York (1995)
17. Chetia, R., Boruah, S.M.B., Sahu, P.P.: Quantum image edge detection using improved sobel mask based on NEQR. *Quantum Inf. Process.* **20**, 1–25 (2021)
18. Tian, R., Sun, G., Liu, X., Zheng, B.: Sobel edge detection based on weighted nuclear norm minimization image denoising. *Electronics* **10**(6), 655 (2021)
19. Liu, W., et al.: SSD: single shot multibox detector. In: Proceedings of the ECCV Conference on Computer Vision, pp. 21–37 (2016)
20. 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)
21. Park, J.M., Murphey, Y.L.: Edge detection in grayscale, color, and range images. In: *Wiley Encyclopedia of Computer Science and Engineering*, pp. 1–16 (2007)
22. Yu, W., et al.: Metaformer is actually what you need for vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10819–10829 (2022)
23. Tolstikhin, I., et al.: MLP-mixer: an all-MLP architecture for vision. In: Proceedings of the 35th International Conference on Neural Information Processing Systems, pp. 24261–24272 (2021)
24. Hou, Q., Zhang, L., Cheng, M.M., Feng, J.: Strip pooling: rethinking spatial pooling for scene parsing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4003–4012 (2020)
25. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
26. Karatzas, D., et al.: ICDAR 2015 competition on robust reading. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 1156–1160 (2015)



Scene Uyghur Text Detection Based on Adaptive Feature Fusion

Dong Wang¹, Elham Eli¹, Alimjan Aysa^{1,2}, Xuebin Xu^{1,2}, Hornisa Mamat¹,
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. In recent years, with the development of deep learning, text detection research has achieved good research results. However, there is still relatively little research on the detection of Uyghur text in natural scenes. Therefore, this paper proposes a scene Uyghur text detection model based on adaptive feature fusion for scene Uyghur texts with special writing styles, complex and variable backgrounds, and different text scales. First, a normalization-based attention module is introduced into the feature extraction network to enhance text features while suppressing background noise. Second, in order to better extract the features of multi-scale text, this paper adds the proposed adaptive feature fusion module in the feature fusion stage. The efficient fusion of text features at different scales is realized by adaptively adjusting the weights between features at different levels. Finally, experiments on the Scene Uyghur text dataset and the ICDAR2015 dataset show the effectiveness and robustness of the proposed method in this paper.

Keywords: Uyghur Text Detection · Natural scenes · Adaptive feature fusion

1 Introduction

Text detection is a subtask of optical character recognition, whose main task is to accurately locate the text area in a picture. With the rapid development of artificial intelligence, it has been applied in many fields, such as traffic cue recognition [1], document key information extraction [2], wearable smart devices [3], etc. In recent years, good progress has been made for the research of text detection in natural scenes, but most of these researches are for Chinese and English, while the research of text detection for Uyghur is still in its infancy, so the research of scene text detection for Uyghur is very meaningful.

Uyghur letters have their own specific writing forms, which consist of strokes as well as dots and symbols. In addition, Uyghur words are adhesive, that is, the letters in a word tend to be closely linked together to form a whole. At the same time, the form and rules of writing letters vary according to their position at the beginning, middle and end of the word. Therefore, the detection algorithms for Uyghur text, which has special writing rules and adhesion characteristics, are faced with multiple challenges.

First of all, Uyghur words are prone to inaccurate word localization in the text detection algorithms due to their unique writing style, as well as the complex backgrounds, noises, and even text-like textures that are extremely similar to text textual textures that may exist in text images of natural scenes. Second, the scale of text in natural scenes tends to vary greatly due to a variety of influences, including the text itself and the angle at which it is shot. All these factors bring considerable challenges to the accurate detection of text. Therefore, in this paper, we improve the DBNet [4] network and design a proposed Scene Uyghur text detection model based on adaptive feature fusion. The main contributions of this paper are shown below:

- (1) In order to alleviate the problem of inaccurate word localization due to the special writing style of Uyghur and the complex background interference in text images, this paper introduces a normalization-based attention module in the feature extraction network.
- (2) In order to better and more accurately detect multi-scale text in text images, this paper adds an adaptive feature fusion module in the feature fusion stage.

2 Related Work

2.1 Scene Text Detection

In recent years, many scholars have made significant progress in the field of natural scene text detection using deep learning algorithms. Deep learning-based natural scene text detection methods can be roughly categorized into regression-based methods and segmentation-based methods.

Regression-based scene text detection methods usually utilize a convolutional neural network to directly predict the text's enclosing frame through the regression layer, which is direct, efficient, and suitable for many scenarios. The EAST method proposed by Zhou et al. [5] is an anchor-free region suggestion network, which directly predicts words or lines of text in the full image in any direction and quadrilateral shape, eliminating the intermediate steps and the complex reference frame design, thus significantly improving the detection efficiency. However, due to the sensory field limitation, there may be a more difficult recognition problem for longer text. TextBoxes proposed by Liao et al. [6] is an improvement on SSD [7] by modifying the anchors and scales of the convolutional kernel for text detection. Despite adapting to various image sizes in the detection process, using multiple anchors of different sizes increases the computational burden and

is not ideal for detecting multi-directional text instances. To solve the problems of TextBoxes, Liao et al. [8] proposed TextBoxes++. The method further adjusts the horizontal bounding box to a quadrilateral with a specific angle, which enhances the detection of non-horizontal linear text instances. Tang et al. [9] proposed SegLink++, a text detection method based on instance perception and component combination, which solves the problem of detecting dense and irregular text in natural scenes through a bottom-up approach and significantly improves the effectiveness of dense text detection.

However, regression-based scene text detection methods may suffer from some limitations when dealing with complex text scenarios such as curved or multi-directional text. Segmentation-based scene text detection methods, on the other hand, segment the image by pixel level to accurately determine the boundaries of the text. The key of the method is to classify each pixel as text or non-text, generating a segmentation map of the same size as the input image that visualizes the shape and contour of the text. This method performs well in dealing with a variety of text morphologies including curved, multi-directional and irregularly shaped text, but may have higher computational complexity compared to regression-based methods. He et al. [10] viewed text detection as an instance segmentation problem using multi-scale image input. Firstly, FCN is used to predict text blocks, followed by text line prediction through two CNN branches and instance-aware segmentation from the estimated text blocks. Deng et al. [11] proposed a segmentation network, PixelLink, to achieve high-precision text localization and recognition by predicting text regions and inter-pixel connectivity on a pixel-by-pixel basis. Wang et al. [12] proposed PSENet to generate a series of kernels of different scales for each text instance, and ultimately achieve accurate detection of arbitrarily shaped text instances by progressively expanding the kernel scale from a larger one to a smaller one. DBNet proposed by Liao et al. [4] uses a differentiable binarization module to combine the binarization process in the text detection task with network training, and improves the accuracy of text detection by adaptively learning the appropriate binarization threshold, which is especially suitable for dealing with complex backgrounds and arbitrarily shaped text instances. To further improve the robustness of text detection Liao et al. [13] proposed an effective adaptive scale fusion module based on DBNet, which improves the scale robustness by adaptively fusing features of different scales. Qu et al. [14] proposed an adaptive inflationary network focusing on the reconstruction process from shrinking polygons aiming to provide a rigorous and complete textual representation. Xu et al. et al. [15] proposed a model called Text Location Aware Pixel Aggregation Network that aims to improve the performance of segmentation-based scene text detection models when dealing with crowded or overlapping text. Yang et al. [16] proposed a Bidirectional Perspective strategy based Network (BiP-Net) to simultaneously achieve high detection accuracy and fast detection of arbitrarily shaped text instances. Zhu et al. [17] proposed a new method called Expand Kernel Network (EK-Net) as a way to achieve precise localization of arbitrarily shaped text by detection networks.

2.2 Uyghur Text Detection

For the detection of Uyghur text Yan et al. [18] proposed a fast and stroke-specific FASTroke keypoint extractor, in addition to the proposed clustering method based on component similarity, which reduces the computational cost while improving the accuracy of the detection. Peng et al. [19], based on the unique characteristics of Uyghur characters, made the detection network more effective in adapting to the requirements of Uyghur language detection by adjusting the aspect ratio of the anchor frame. Abdulweli Ruzhe [20] et al. improved Yolo (You Only Look Once) by generating three types of fixed-width anchor frames, and then performed multi-directional Uyghur text detection by regression. Li et al. [21] proposed a Uyghur text detection method based on Sobel edge detection algorithm. The method can effectively detect the Uyghur text regions. In addition, accurate localisation of Uyghur text is achieved through the merging operation of text regions. Wang et al. [22] improved DBNet by firstly using Res2Net in the feature extraction network so as to adapt to the Uyghur language with adhesion. Secondly, in the process of feature fusion, an adaptive multi-level feature map fusion strategy is used to overcome the inconsistency of information in the fusion process. Thus, the detection accuracy can be effectively improved.

3 Method

The proposed detection model in this paper is shown in Fig. 1. The model is mainly divided into three modules, which are feature extraction module, feature fusion module and text box inference module.

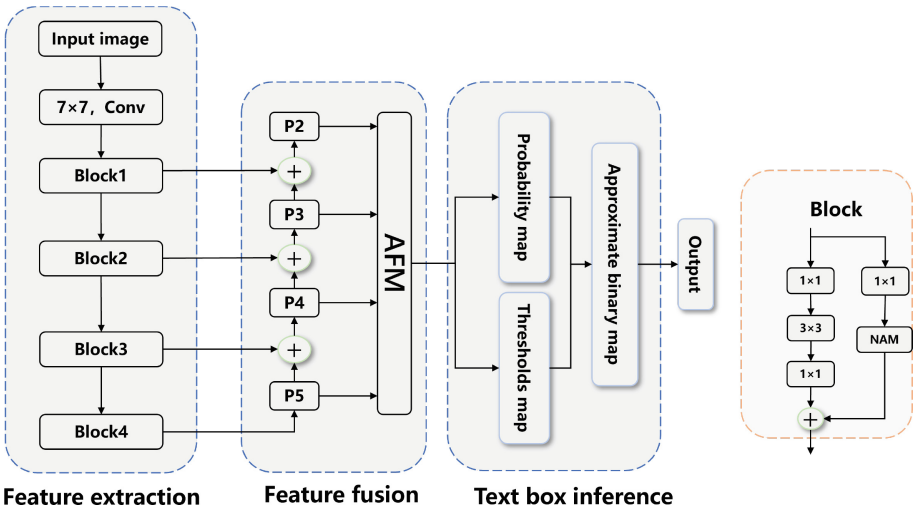


Fig. 1. Architecture of Scene Uyghur text detection model based on adaptive feature fusion.

3.1 Feature Extraction Network

In this paper, ResNet-50 is used as a feature extraction network for extracting text features from images. However, due to the unique writing form of Uyghur text and the presence of complex background noise in natural scenes, ResNet-50 may have difficulty in adequately capturing the key features of text regions when processing Uyghur text. Therefore, in this paper, normalized attention based (NAM) [23] is introduced into the residual network to enhance the network’s attention to text regions. NAM is introduced to enhance the network’s focus on the text region. NAM helps the model capture the key information of Uyghur text more accurately during feature extraction by focusing on and enhancing the representation of Uyghur text features while suppressing the non-text features in the complex background. In this way, not only the problem of inaccurate word localization can be alleviated, but also the interference of background texture and noise on text recognition can be significantly reduced. In addition, unlike other attentions NAM adopts a normalization-based approach, which avoids the addition of fully connected and convolutional layers, thus reducing the complexity and computation of the model. The residual block with the addition of NAM module is shown in Fig. 2.

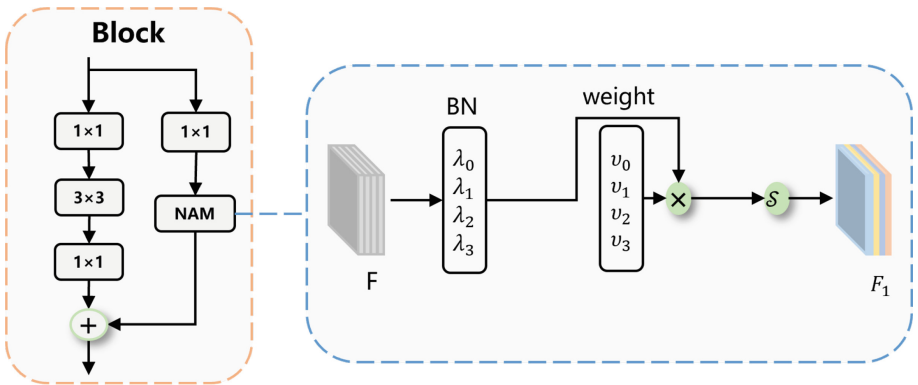


Fig. 2. Improved residual block.

In NAM module, the feature map is first input into the BN layer, and for each channel, the BN layer learns its corresponding scaling factor λ . The BN layer is calculated as shown in equation (1):

$$B_{out} = BN(B_{in}) = \lambda \frac{B_{in} - \mu}{\sigma^2 + \epsilon} + \gamma \tag{1}$$

where B_{in} is the input, μ is the mean, σ is the standard deviation, and ϵ is a very small number; γ is the offset parameter and λ is the scaling factor of the channel, where both γ and λ are trainable parameters.

Next, the scaling factor λ_i corresponding to each channel is used to find its corresponding weight w_i . A larger weight indicates that the channel will contain richer information, and conversely, a smaller weight indicates that the channel will contain less information. The calculation formula is shown in Eq. (2):

$$w_i = \frac{\lambda_i}{\sum_{j=0}^n \lambda_j} \quad (2)$$

where n represents the number of channels. Finally, the obtained weights are dot-multiplied with the feature map that has gone through the BN layer and input to the Sigmoid function to obtain the feature map F_1 , which is calculated as shown in Eq. (3):

$$F_1 = \text{sigmoid}(w_i(\text{BN}(F))) \quad (3)$$

3.2 Adaptive Feature Fusion Module

Feature pyramid networks (FPN) [24] are often used in image segmentation networks to effectively utilize feature maps at different scales. FPN enables the network to capture both low level detailed features and high level semantic features by constructing a multi-scale feature hierarchy. In order to achieve effective fusion of these feature maps at different scales, most of the methods will use concatenation or summation. However, this approach may make it difficult for the fused feature maps to capture the features of both small-scale and large-scale text, which in turn affects the accuracy of text detection. In addition, high-level feature maps usually contain rich semantic information, while low-level feature maps are more detail-oriented. Direct splicing fusion may lead to redundancy of high-level semantic information, especially when detecting small-scale text, too much high-level semantic information may drown out the detailed features of small text, which further affects the detection accuracy.

To address the above problems, this paper proposes an Adaptive Feature Fusion Module (AFM), which can adaptively learn the importance of each feature map according to the semantic information of the feature maps at different scales and adjust its weight in the fusion process accordingly to be more adaptive in feature fusion. This adaptive fusion makes the model more flexible in capturing the information of different scales of text, which enables the model to better detect different scales of text. The AFM module is shown in Fig. 3.

The main idea of the adaptive feature fusion module is to adaptively generate the corresponding weights for the feature maps at different scales in the FPN. The AFM module first upsamples the feature maps P_5 , P_4 and P_3 by $8x$, $4x$, and $2x$ and resizes them to match the size of P_2 . Second, the resized feature maps are concatenated with P_2 . Next, the feature map β of size $4 \times H \times W$ is obtained by convolution operation. Then softmax operation is performed on the feature map β in the channel dimension to generate the corresponding weight map $\alpha_l, l \in 2, 3, 4, 5$ for each feature map P_i . The calculation formula is shown in Eq. (4).

$$\alpha_l = \frac{e^{\beta_{i,j}^l}}{e^{\beta_{i,j}^1} + e^{\beta_{i,j}^2} + e^{\beta_{i,j}^3} + e^{\beta_{i,j}^4}} \quad (4)$$

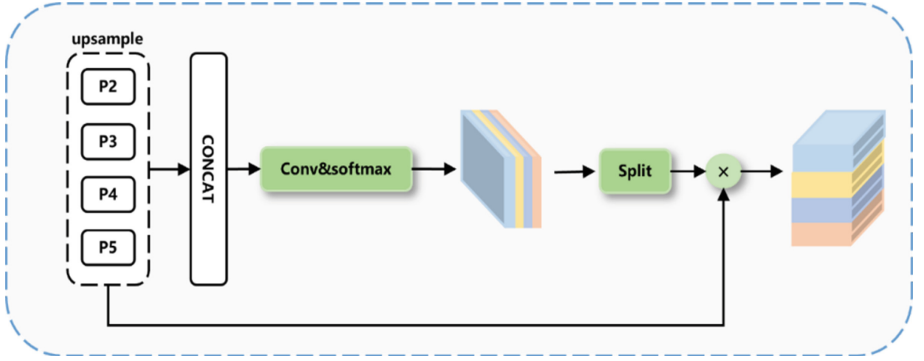


Fig. 3. Adaptive feature fusion modules.

where $\beta_{i,j}^c$ denotes the value of the feature map β at position (i, j) , the c th channel. Finally, the feature maps P_i of different scales are adjusted by multiplying them with their corresponding weight maps α_l so that the model is able to weight each feature map individually, highlighting the key features, thus capturing the multi-level semantic information more effectively and improving the performance of the model in processing text of different scales.

3.3 Text Bounding Boxes Inferencing

In the inference of text box, the feature map output from the feature fusion module is first predicted to obtain the probability map and threshold map respectively. Then, the approximate binary map is generated using differentiable binarization method, and finally the text box is obtained by post-processing. The formula for the differentiable binarization is given in Eq. (5):

$$\hat{B}_{i,j} = \frac{1}{1 + e^{-K(P_{i,j} - T_{i,j})}} \quad (5)$$

where $P_{i,j}$ is the pixel point in the probability map, $T_{i,j}$ represents the pixel point in the threshold map, and K represents the magnification factor, which is set to 50 in this paper.

3.4 Label Generation

When training the detection network, labels need to be generated for supervised learning. These labels include the probability map, the binary map and the threshold map. The probability map and the approximate binary map use the same supervised signal, so two labels need to be generated. Probability map labels are obtained by shrinking the labels of the text box by a distance D , while threshold map labels are obtained by expanding the labels of the text box outwards by D . The offset distance D is calculated as shown below:

$$D = \frac{A \times (1 - r^2)}{C} \quad (6)$$

where A represents the area of the text box, C represents the perimeter of the text box, and r represents the shrinkage factor, which is set to 0.4 in this paper.

3.5 Loss Function

The loss function L of the text detection model proposed in this chapter consists of a probabilistic graph loss function L_p , a binary graph loss L_t and a threshold graph loss L_d .

$$L = L_p + \omega \times L_t + \varphi L_d \tag{7}$$

where ω and φ represent the weight coefficients set to 1 and 10, respectively.

For the probabilistic graph loss L_p and the approximate binarised graph loss L_t a binary cross-entropy loss function is used, as shown in Eq. (8):

$$L_p = L_t = \sum_{i \in S_t} y_i \log x_i + (1 - y_i) \log(1 - x_i) \tag{8}$$

where S_t is the set of samples with a positive to negative ratio of 1:3, while y_i represents the labelled values in the probability map and x_i represents the predicted values of the network. The threshold graph loss function L_d used in this paper is the L_1 loss function, Eq. (9):

$$L_d = \sum_{i \in R_t} |y_i^* - x_i^*| \tag{9}$$

where y_i^* and x_i^* represent the labelling of the threshold map and the prediction of the threshold map, respectively.

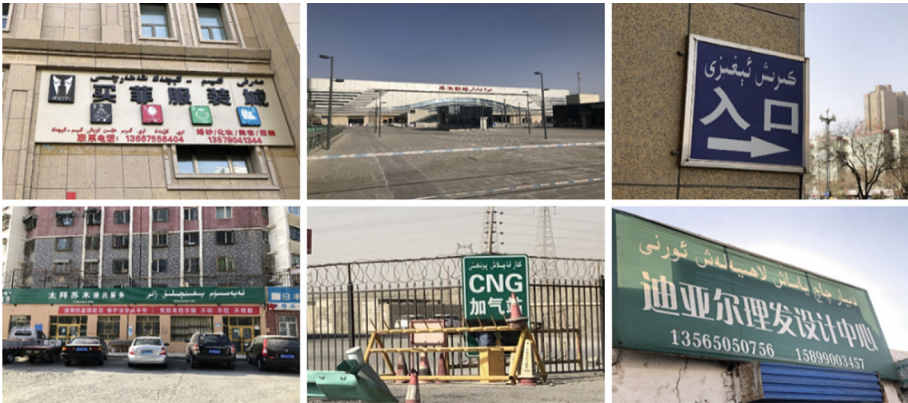


Fig. 4. Sample scene Uyghur text dataset.

4 Experiments

4.1 Dataset

Scene Uyghur Text Dataset. The scene Uyghur text dataset was collected and produced by Wang et al. [22] in Xinjiang, China and other places. The dataset contains 2400 sheets of training set, 800 sheets of validation set, 800 sheets of test set, and the total number of samples is 4000 sheets in total. These text images are text images of natural scenes collected by shooting, and the images contain street scenes such as traffic signs, shop signs, and promotional banners. Due to the influence of lighting, equipment, and shooting techniques, the text images are characterised by multiple scales and directions. A sample data set is shown in Fig. 4:

ICDAR2015 Dataset. The ICDAR2015 dataset [25] is the official dataset used in the Scene Text Detection Competition organised by the International Conference on Document Analysis and Recognition (ICDAR) in 2015, which contains 1,000 images in the training set and 500 images in the test set, with a total of 1,500 images in the total number of samples. The text images in this dataset were taken with Google Glass, and the shooting method is relatively casual. There is a large amount of fuzzy text in the images, and the text scale varies a lot, so the detection is more difficult. The sample of the ICDAR2015 dataset is shown in Fig. 5:



Fig. 5. Sample ICDAR2015 dataset.

4.2 Experimental Details And Configuration Environment

In this paper, we use the pre-trained ResNet-50 network weights on the ImageNet dataset to initialise the feature extraction network, and the overall training process is carried out for a total of 1200 epochs. In this paper, the initial learning

rate and batch size are set to 0.0001 and 10, respectively, and the learning rate is adjusted using the WarmupPolyLR strategy, where the warm-up phase lasts for 3 cycles. For the optimiser, AdamW is chosen in this paper. The details of the required environment parameter settings for the experiment are shown in Table 1. In this paper, Accuracy (P), Recall (R), F-Measure (F) and Frames per Second (FPS) are used as evaluation metrics.

Table 1. Experimental environment parameters configuration.

Type	Configuration
CPU	Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50 GHz
GPU	NVIDIA RTX 3090
Operating System	ubuntu20.04
Deep Learning Framework	PyTorch 1.11.0

4.3 Ablation Study

In order to verify the effectiveness of the added individual modules, this paper evaluates and analyses the scene text dataset and the ICDAR2015 dataset respectively. The experimental results are shown in Table 2, where the first row represents the experimental results of the baseline model.

Table 2. Ablation experiments with different modules.

NAM	AFM	Scene Uyghur text dataset				ICDAR2015 dataset			
		R	P	F	FPS	R	P	F	FPS
		91.1	96.1	93.6	39.3	79.5	88.5	83.8	35.2
✓		91.2	96.3	93.7	36.5	80.3	89.0	84.4	31.9
	✓	91.3	96.6	93.9	38.7	79.5	89.3	84.1	34.6
✓	✓	92.1	95.9	94.0	35.8	82.9	88.8	85.8	30.5

The first set of experiments, shown in the second row of Table 2, adds NAM module to the feature extraction network of the baseline model. As the NAM module can adjust the weights of the feature map to highlight the key information in the text while mitigating the interference of the complex background in the text image. As can be seen in Table 2, on the Uyghur text dataset, the module resulted in a 0.2% improvement in the accuracy of the model, while the F composite metrics also gained 0.1%. On the ICDAR2015 dataset, the effect of the NAM module is even more significant, resulting in a 0.8% increase in the model’s recall and a corresponding 0.6% increase in the F composite metric.

These improvements fully demonstrate the effectiveness of the NAM module in improving the performance of text detection models.

The second set of experiments, shown in the third row of Table 2, adds the adaptive feature fusion module to the feature fusion module of the baseline model. The experimental data show that the introduction of the AFM module improves the F-integrated metrics of both datasets by 0.3% compared to the baseline model. This is because the AFM module flexibly adjusts the weights of feature fusion thus enabling the model to better capture and adapt to text features at different scales.

The third set of experiments, shown in the fourth row of Table 2, incorporates both the NAM and AFM modules into the baseline model. Compared to the baseline model, the Baseline+NAM+AFM network model improves the recall and F-integrated metrics by 1% and 0.4% on the scene Uyghur text detection dataset, while the recall and F-integrated metrics on the ICDAR2015 dataset improve by 3.4% and 2%, respectively. In addition, its F-composite metrics improved in both datasets with the addition of both modules, relative to the comparison using only a single module. The experimental results demonstrate that the simultaneous use of these two improved modules is more beneficial for enhancing the performance of the text detection model, and further validate the effectiveness of these two modules.

4.4 Comparative Experiments On Uyghur Text Dataset

In order to further demonstrate the effectiveness of this paper’s method in Uyghur text detection. In this paper, DBNet and EAST [5] algorithms are reproduced and compared with the method proposed in this paper for experiments on the scene Uyghur text dataset, in addition to comparing the scene Uyghur text Detection method proposed by wang [22]. The experimental comparison results are shown in Table 3. It should be noted that the data marked with citation corners in the table come from the experimental results in the original literature, while the part without citation is the experimental data reproduced by this paper.

Table 3. Experimental results of different methods on the Scene Uyghur dataset.

Method	R	P	F	FPS
EAST	81.1	90.8	85.7	77.8
Wang [22] (736)	91.5	96.6	93.9	26.4
DBNet+Resnet18 (736)	89.8	96.0	92.8	62.8
DBNet+Resnet50 (736)	91.1	96.1	93.6	39.3
Ours (736)	92.1	95.9	94.0	35.8
Ours (1024)	93.1	96.3	94.7	21.9

From the fifth row of the table, we can see that the proposed method in this paper achieves 94% of the F composite index, and the F composite index even reaches 94.7% when the image size is adjusted to 1024. However, since the FPS decreases by more than 1.6 times, the image size of 736 is used in this paper. The model used in this paper is ResNet50 network selected for feature extraction. The reason for choosing ResNet50 as the feature extraction network can be seen through the experimental results in the third and fourth rows of Table 3. Although the FPS of using RenNet18 as the feature extraction network is much faster, the experimental results show that the network using RenNet50 improves the recall, accuracy and F composite metrics by 1.3%, 0.1% and 0.8% respectively than the one using RenNet18, so RenNet50 is chosen as the feature extraction network. Compared with the regression-based EAST algorithm, the method proposed in this paper achieves higher performance in all the three metrics of R, P, and F, and the improvement in speed is also significant. Compared with the improved DBNet-based scene Uyghur text detection method proposed by Wang et al. [22], the proposed method in this paper achieves 0.6% and 0.1% improvement in recall and F composite metrics, respectively, which further indicates that the proposed method in this paper is competitive on scene Uyghur text datasets.

Table 4. Experimental results of different methods on the ICDAR2015 dataset.

Method	R	P	F	FPS
EAST [5]	72.8	80.5	76.4	6.5
TextBoxes++ [8]	76.8	87.2	81.7	11.6
SegLink++ [9]	73.7	86.3	79.5	9.5
PixeLink [11]	80.0	85.5	83.7	3.0
DBNet [4]	82.7	88.2	85.4	26.0
Wang [22]	81.8	88.3	84.9	18.9
BiP-Net [16]	82.1	86.9	83.9	24.8
EK-Net [17]	80.2	92.0	85.7	35.4
Ours	82.9	88.8	85.8	30.5

4.5 Comparative Experiments With The ICDAR2015 Dataset

In order to further verify the effectiveness and generalization of the model, the experimental results of this paper on the public dataset ICDAR2015 are compared with a variety of text detection algorithms, and the experimental comparison results are shown in Table 4. From the table, it can be seen that the proposed method in this paper achieves the best performance in terms of Recall and F-Measure, which further illustrates the effectiveness of the improved method.

Since the segmentation approach labels the text and background at the pixel level, it can be more flexible to adapt to irregularly shaped text. Therefore, compared with several regression methods, such as EAST [5], TextBoxes++ [8], and SegLink++ [9], the method proposed in this paper shows better performance in all metrics. The reason why the improved method in this paper is better than other segmentation-based methods may be that the NAM module introduced in this paper can reduce the interference of the background in the text image while enhancing the network’s attention to the text region, and the adaptive feature fusion module can make the feature extraction network better adapt to the text features at different scales so as to better detect the text at multiple scales. Therefore, the accuracy of the proposed method in this paper is higher than other models.

4.6 Comparison Of Some Test Results

Some test results of this paper’s method with the baseline model DBNet are shown in Fig. 6, where (a) and (b) are the detection results on the scene Uyghur text dataset and the ICDAR2015 dataset, respectively.

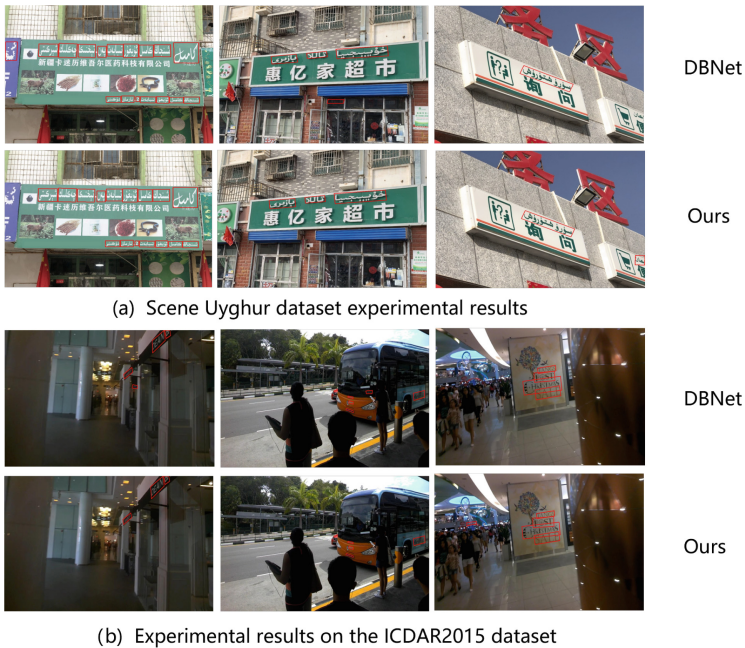


Fig. 6. Visualization of experimental results for comparison.

As can be seen from Fig. 6, in the first and second columns, the baseline algorithm has misdetection because of the interference of text-like texture and

background noise in the text image. In the third column, the baseline algorithm has leakage detection when detecting multi-scale text images, and the detection effect is not good. Compared with the baseline algorithm, the improved method in this paper can correctly localize the text region. Through these sets of comparisons, it can be seen that the improved model has good detection effect when facing complex background interference and multi-scale text. This further proves the robustness of the method proposed in this paper.

5 Conclusion

In order to improve the accuracy of Uyghur text detection, this paper proposes a Uyghur text detection model based on adaptive feature fusion. First, in order to accurately locate Uyghur text in complex backgrounds, this paper introduces a normalization-based attention module in the feature extraction network. Second, in order to improve the model's ability to accurately recognize multi-scale Uyghur text in natural scenes, an adaptive feature fusion module is designed in this paper. Finally, a series of experiments are conducted to verify the effectiveness and robustness of the proposed model in this paper on the task of Uyghur text detection in natural scenes.

Acknowledgments. This work was supported by the Xinjiang Uygur Autonomous Region “Tianshan Talents” Science and Technology Innovation Leading Talents Program (2023TSYCLJ0025) and the National Natural Science Foundation of China (No. 62266044, 62061045, 61862061).

References

1. Yan, C., Xie, H., Liu, S., Yin, J., Zhang, Y., Dai, Q.: Effective Uyghur language text detection in complex background images for traffic prompt identification. *IEEE Trans. Intell. Transp. Syst.* **19**(1), 220–229 (2018)
2. Guo, P., Song, Y., Deng, Y., Xie, K., Xu, M., Liu, J., Ren, H.: DCMAI: a dynamical cross-modal alignment interaction framework for document key information extraction. *IEEE Trans. Circuits Syst. Video Technol.* (2023)
3. Tanveer, M., Ahmad, M., Nguyen, T.N., Abd El-Latif, A.A., et al.: Resource-efficient authenticated data sharing mechanism for smart wearable systems. *IEEE Trans. Netw. Sci. Eng.* (2022)
4. Liao, M., Wan, Z., Yao, C., Chen, K., Bai, X.: Real-time scene text detection with differentiable binarization. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 11474–11481 (2020)
5. Zhou, X., et al.: East: an efficient and accurate scene text detector. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 5551–5560 (2017)
6. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: Textboxes: a fast text detector with a single deep neural network. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31 (2017)
7. Liu, W., et al.: SSD: single shot multibox detector. In: *ECCV 2016, Part I*, pp. 21–37. Springer (2016)

8. Liao, M., Shi, B., Bai, X.: TextBoxes++: a single-shot oriented scene text detector. *IEEE Trans. Image Process.* **27**(8), 3676–3690 (2018)
9. Tang, J., Yang, Z., Wang, Y., Zheng, Q., Xu, Y., Bai, X.: SegLink++: detecting dense and arbitrary-shaped scene text by instance-aware component grouping. *Pattern Recogn.* **96**, 106954 (2019)
10. He, D., et al.: Multi-scale FCN with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3519–3528 (2017)
11. Deng, D., Liu, H., Li, X., Cai, D.: PixelLink: detecting scene text via instance segmentation. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32 (2018)
12. Wang, W., et al.: Shape robust text detection with progressive scale expansion network. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9336–9345 (2019)
13. Liao, M., Zou, Z., Wan, Z., Yao, C., Bai, X.: Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(1), 919–931 (2022)
14. Qu, Y., Xie, H., Fang, S., Wang, Y., Zhang, Y.: ADNet: rethinking the shrunk polygon-based approach in scene text detection. *IEEE Trans. Multimedia* **25**, 6983–6996 (2022)
15. Xu, J., Lin, A., Li, J., Lu, G.: Text position-aware pixel aggregation network with adaptive gaussian threshold: Detecting text in the wild. *IEEE Trans. Circuits Syst. Video Technol.* (2023)
16. Yang, C., Chen, M., Yuan, Y., Wang, Q.: BIP-net: bidirectional perspective strategy based arbitrary-shaped text detection network. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2255–2259 (2022)
17. Zhu, B., Liu, F., Chen, X., Tang, Q.: EK-Net: Real-time scene text detection with expand kernel distance. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6380–6384 (2024)
18. Yan, C., et al.: A fast Uyghur text detector for complex background images. *IEEE Trans. Multimedia* **20**(12), 3389–3398 (2018)
19. Yong, P.: Uyghur detection in natural scenes based on deep learning. Master's thesis, Xinjiang University (2018)
20. Ruze, A.: Multi-directional Uyghur region detection algorithm based on deep learning. Master's thesis, Xinjiang University (2020)
21. Li, X., Li, J., Gao, Q., Yu, X.: Uyghur text detection in natural scene images. In: *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1542–1547. IEEE (2019)
22. Wang, Y., Mamat, H., Xu, X., Aysa, A., Ubul, K.: Scene Uyghur text detection based on fine-grained feature representation. *Sensors* **22**(12), 4372 (2022)
23. Liu, Y., Shao, Z., Teng, Y., Hoffmann, N.: NAM: normalization-based attention module. *arXiv preprint arXiv:2111.12419* (2021)
24. 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)
25. Karatzas, D., et al.: ICDAR 2015 competition on robust reading. In: *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1156–1160. IEEE (2015)



Handwriting Trajectory Recovery Via Trajectory Transformer With Global Radical Context-Aware Module

Junxiang Lin, Zhounan Chen, Lingyu Liang, Wenjie Peng,
and Shuangping Huang^(✉)

South China University of Technology, Guangzhou, China
eehsp@scut.edu.cn

Abstract. Handwriting trajectory recovery aims to reconstruct the writing trajectories from images of handwritten characters, holding applications in various areas such as text recognition, signature authentication, and forensic handwriting analysis. Current methods commonly used CNNs for feature extraction from character images, excelling in local feature identification but lacking in capturing global handwriting structures. Subsequent trajectory generation is typically handled by RNNs, which model the temporal sequence of writing but are hindered by gradient vanishing issues, leading to accuracy reduction in longer sequences. In this paper, we propose a Trajectory Transformer with a Global Radical Context-Aware (**GRCA**) module to realize precise trajectory recovery by analyzing intricate structural relationships in handwriting characters and modeling contextual correlations within trajectory sequences. Concretely, the GRCA module utilizes dilated convolutions to extract character radical features across various scales and perceives the structural associations embedded within the multi-scale features. Additionally, we introduce a Transformer to capture the contextual correlations among trajectory sequences, thus alleviating the issue of trajectory drift. Experiment results show that our proposed Trajectory Transformer achieves state-of-the-art performance on four benchmark datasets.

Keywords: Handwriting Trajectory Recovery · Global Radical Context-Aware Module · Transformer

1 Introduction

Handwritten pattern analysis encompasses two primary categories: offline image-based patterns captured via cameras [7, 8, 13], and online patterns delineating pen-tip trajectories from touch-sensitive or pen-enabled devices [20, 21]. Handwriting trajectory recovery is a sophisticated cross-modal text generation technology that aims to recover online handwriting trajectories from offline images, thus promoting the development of text recognition and related fields [9–11, 15, 29].

Early traditional methods in trajectory recovery [3, 4, 14, 16, 22–25] relied on heuristic rules which were effective for single-stroke images and simple alphanumeric characters. However, these methods exhibited limited generalization capabilities for complex multi-stroke character images (Fig. 1).

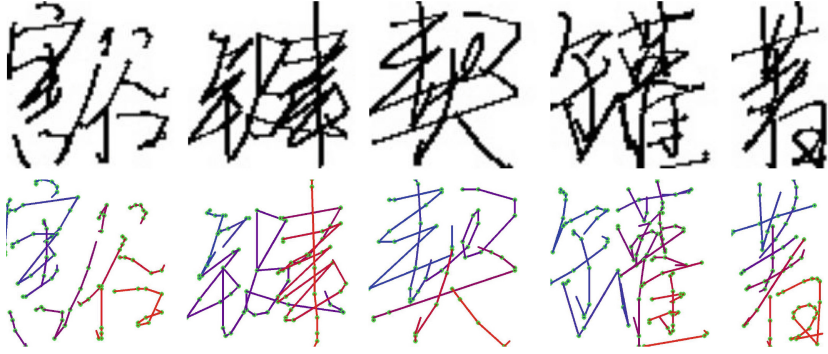


Fig. 1. Visualization of handwriting text images and trajectory, where green dots represent the trajectory coordinate points, and the strokes of the writing trajectory are depicted in a gradient from blue to red to indicate the sequence of writing. (Color figure online)

To address this, some approaches [1, 2, 26–28, 31, 32] leveraged the CRNN architecture to extract features from character images and recover trajectory points based on these extracted features. Specifically, Zhao et al. [31, 32] used CNNs to extract image features and generate a pen movement heat map. Bhunia et al. [2] extended this paradigm by introducing LSTM to capture context within trajectory sequences. Archibald et al. [1] further extended the idea of [2] and introduced the Dynamic Time Warping (DTW) loss function to optimize model learning. Sumi et al. [27] employed a Cross Variational Autoencoder to establish a shared latent space for offline-to-online character conversion. However, these methods suffer from the forgetfulness of RNNs, which may lead to position drift when predicting lengthy trajectory sequences.

Recently, some works considered introducing the attention mechanisms to fuse 2D features of character images [5, 19]. Nguyen et al. [19] introduced 2D attention and incorporated a Gaussian Mixture Model to enhance model robustness. Chen et al. [5] proposed a two-stream parsing encoder to compress feature maps from vertical and horizontal directions to improve the accuracy of 2D coordinates recovery. They also proposed a global tracking mechanism that incorporates global features to predict each trajectory point, which alleviates the position drift phenomenon to some extent.

Despite the promising progress of existing trajectory recovery models, we find that they still suffer from two major challenges: **(1) Failing to adequately analyze structural relationships in complex characters.** In complex character sets like Chinese and Japanese, radicals are frequently distributed across

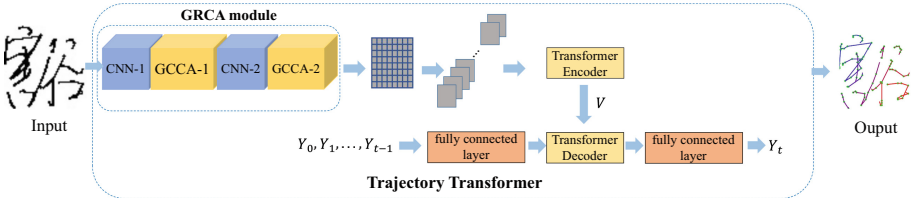


Fig. 2. The proposed Trajectory Transformer consists of a GRCA module and a Transformer architecture. The GRCA module extracts global character image features and analyzes the structural relationships, while the Transformer models the contextual correlations of trajectory sequences and further predicts trajectory points based on the extracted features.

distinct positions and exhibit various sizes, thereby resulting in multi-level structural relationships. **(2) Failing to sufficiently model contextual correlations among trajectory sequences.** Intuitively, each trajectory point is linked to previous trajectory points and global context information. Understanding these contextual relationships among trajectory sequences is critical for trajectory recovery.

In this paper, we propose a Trajectory Transformer with a Global Radical Context-Aware (**GRCA**) module to tackle the above two challenges. For the first challenge, we introduce the GRCA module, which uses dilated convolution to extract radical features of various scales and analyze their structural relationships. For the second, we propose a Transformer architecture to capture context correlations among trajectory sequences based on the features extracted by the GRCA module, thus enhancing the precision of lengthy trajectory recovery. Our contributions are summarized as follows:

- We propose a Trajectory Transformer for handwriting trajectory recovery, which can effectively capture the contextual correlations of trajectory sequences.
- We design a novel GRCA module for complex character feature extraction that can comprehensively analyze its internal structural relationships.
- Our method achieves state-of-the-art performance on four datasets across various languages, which outperforms previous methods by a large margin.

2 Method

2.1 Trajectory Transformer

Figure 2 illustrates the architecture of the proposed Trajectory Transformer, which integrates a Global Radical Context-Aware (GRCA) module with a Transformer framework. This framework processes a handwritten image, denoted as \mathbf{I} , and outputs predicted trajectory sequences represented by $P = (p_1, \dots, p_l)$, where l denotes the trajectory's length. Each trajectory point,

$p_i = (x_i, y_i, s_i^1, s_i^2, s_i^3)$, includes the 2D coordinates (x_i, y_i) along with pen tip states (s_i^1, s_i^2, s_i^3) , corresponding to “pen-down”, “pen-up”, and “end-of-sequence” actions, respectively. The GRCA module is crucial in processing the input image \mathbf{I} , enhancing the perception across multiple scales and extracting a detailed 2D feature map \mathbf{O} . This map is subsequently flattened and fed into the Transformer encoder, a key step for obtaining the global context feature \mathbf{V} that is intricately linked with the input image \mathbf{I} . By employing a self-attention mechanism, the Transformer encoder embeds positional information into the sequence features and analyzes relationships within the sequence. It calculates and compares the similarity of each position to every other, using these calculations as weights to integrate trajectory features into the encoded feature \mathbf{V} , which encapsulates comprehensive global context information.

During each sequential step t in the Transformer decoder, the feature \mathbf{V} is combined with the historical trajectory sequences p_1, p_2, \dots, p_{t-1} to predict the next trajectory point p_t . This iterative process, based on an auto-regressive model, continues until the complete trajectory sequence p_1, p_2, \dots, p_l is reconstructed.

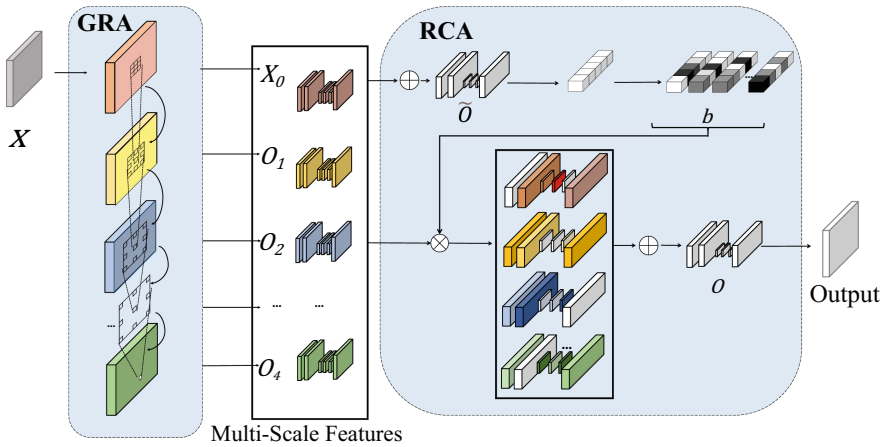


Fig. 3. The GRCA module consists of a GRA module and a RCA module, in which the GRA module captures multi-scale features, and the RCA module conducts attention perception of multi-scale features.

2.2 Global Radical Context-Aware Module

As previously noted, prevailing methodologies predominantly utilize standard convolution for feature extraction from character images. However, this approach may limit the model’s comprehensive understanding of character radicals, primarily due to the constrained scope of continuous receptive fields. In response

to this limitation, our approach is inspired by the multi-scale contextual analysis method detailed in Li et al. [17]. We have developed a novel module, termed the Global Radical Context-Aware (GRCA) module, which integrates dilated convolutions within the backbone network. This integration facilitates the extraction of multi-scale radical features from handwritten images, thereby allowing for a more subtle perception of the structural interrelations inherent within these features. The GRCA module’s design substantially augments the sequence model’s capability to discern intricate details in character images, consequently improving the accuracy of writing trajectory predictions.

Figure 3 outlines the structure of the GRCA module, which is divided into two distinct yet interconnected submodules: the Global Radical-Aware (GRA) module and the Radical Context-Aware (RCA) module. The GRA module focuses on recognizing and perceiving the global aspects of radicals, going beyond mere local feature extraction. It leverages dilated convolutions to encompass a wider field of view, thereby capturing the essence of radicals more comprehensively. Meanwhile, the RCA module specifically targets the contextual relationship of these radicals. It employs techniques such as dynamic attention mechanisms and contextual embedding to analyze the spatial and semantic relationships among different radical features, providing a deeper layer of contextual understanding. Together, these modules work to enhance the model’s cognition and understanding of complex character structures in handwritten text images.

The GRA Module: The GRA module is an innovative aspect of our architecture that integrates dilated convolutions to enhance the capture of multi-scale features within character images. This approach is pivotal in augmenting the global representation of radical features. The rationale behind employing dilated convolutions lies in their capability to expansively augment the receptive field without altering the convolution kernel’s dimensions. This approach is essential for facilitating the neurons’ perception of the intricate radical relationships present in characters.

A critical step in our methodology involves dynamically modulating the receptive field of neurons to optimize feature perception. We commence this process by computing the maximal receptive field for each pixel within the feature map, dimensions of which are denoted by $H \times W$. It is important to note that pixels located at the boundary of the image have a reduced receptive field due to the absence of surrounding pixels, leading to distinct calculations for boundary and non-boundary pixels.

The dilation rates in horizontal and vertical directions for the i -th scale feature map are represented as D_i^W and D_i^H , respectively. To determine the maximal receptive field, we apply the following equations for non-boundary pixels (RP_i) and boundary pixels (RE_i) within the feature map \mathbf{O}_i : $RP_i = 1 + 2 \times \sum_{j=1}^i D_j^W$ and $RE_i = 1 + \sum_{j=1}^i D_j^H$

To maintain continuity in feature extraction, we first employ standard convolution, setting the initial dilation rate D_1 to 1, to procure the initial output feature map \mathbf{X}_0 . The calculation of the maximum receptive field guides our selection of dilation rates for various convolution layers within the GRA module. By

judiciously choosing dilation rates within the bounds of the calculated maximum receptive field, we progressively expand the receptive field. This strategic expansion is crucial for perceiving multi-scale features while preserving the contextual continuity of the receptive field. Ultimately, this methodical approach culminates in the extraction of multi-dimensional features, thereby enhancing the overall functionality of our module.

The RCA Module: In our model, the Radical Context-Aware (RCA) module is specifically designed to focus on salient features within characters. This module enhances the model’s capabilities by attending to the multi-scale feature maps produced by the Global Radical-Aware (GRA) module. The primary objective of the RCA module is to judiciously assign varying attention weights to these feature maps, thereby generating global-aware feature maps that encapsulate a comprehensive understanding of the character structures.

As depicted in Fig. 3, the initial step in the RCA module involves the amalgamation of multi-scale features. This is accomplished by summing the features across different scales, resulting in an aggregated output denoted as $\tilde{\mathbf{O}}$. Subsequently, to condense these aggregated features spatially, we apply average pooling, which effectively transforms the feature maps into a vector $\mathbf{z} \in \mathbb{R}^C$.

The critical phase of the RCA module involves assessing the importance of various scale contexts associated with \mathbf{z} . To achieve this, we utilize a series of five fully connected layers applied to the feature vector \mathbf{z} . This architecture facilitates the learning of weight coefficients $b \in \mathbb{R}^{5 \times C}$, corresponding to each scale. The weight coefficients are determined through softmax normalization executed independently for each channel, as represented by the following equation:

$$b_i = \frac{\exp(fc_i(\mathbf{z}))}{\sum_{j=1}^5 \exp(fc_j(\mathbf{z}))}. \quad (1)$$

The culmination of this process involves the application of the learned weights b to integrate the feature maps across various scales, resulting in a global-aware feature representation \mathbf{O} : $\mathbf{O} = b_0 \mathbf{X}_0 + \sum_{i=1}^4 b_i \mathbf{O}_i$.

This global-aware feature \mathbf{O} is then employed by the Trajectory Transformer in predicting trajectory sequences. The RCA module’s ability to discern and prioritize different scale features plays a crucial role in the Trajectory Transformer’s overall accuracy and efficiency. By focusing on both local and global character features, the RCA module ensures that the Trajectory Transformer has a comprehensive understanding of the intricacies involved in character trajectories.

2.3 Loss Functions

We employ L1 regression loss L_{reg} and cross-entropy loss L_{CE} to optimize the coordinates and pen states of the trajectory points respectively. The cross-entropy loss is formulated as follows:

$$L_{\text{CE}} = -\frac{1}{3N_p} \sum_{i=1}^{N_p} \sum_{k=1}^3 w_k s_i^k \log(p(\hat{s}_i^k)), \quad (2)$$

where \hat{s}_i^k denote the predicted pen tip state, while s_i^k represent the ground truth, and w_1 , w_2 , and w_3 are the weight coefficients for “pen-down”, “pen-up”, and “end-of-sequence” states, which are set to 1, 5, and 5. In summary, the overall optimization objective is as follows:

$$L = \lambda_1 L_{\text{reg}} + L_{\text{CE}}, \quad (3)$$

where λ_1 is a balance factor, which is set to 0.5.

3 Experiments

3.1 Datasets and Evaluation Metrics

We use the following datasets as the benchmark for validation.

Chinese Dataset: CASIA-OLHWDB [18] is a million-level online handwritten character dataset, covering Chinese characters, numbers, English characters, and some special characters. We extract all Chinese characters from it for experiments, including the most frequently used Chinese characters.

English Dataset: We collect all the English characters in CASIA-OLHWDB as the English dataset for training and testing.

Japanese Dataset: Referring to [19], we train on the Nakayosi_t-98-09 dataset and test on the Kuchibue_d-96-02 dataset.

Indian Dataset: We use the Tmail online text dataset [2], which was used as the dataset for the IWFHR2006 online Tamil Handwriting character recognition competition.

We employ four metrics to evaluate the writing order and font fidelity according to [5]. For writing order evaluation, we compare the recovered writing trajectory with the ground truth trajectory using Length-independent Dynamic Time Warping (LDTW) and Dynamic Time Warping (DTW) [12]. In terms of font fidelity evaluation, we render the recovered writing trajectory into text images and compare these with ground truth text images, utilizing Adaptive Intersection over Union (AIoU) [6] and Learned Perceptual Image Patch Similarity (LPIPS) [30].

3.2 Implementation Details

In the experiment process, we normalize the coordinates of the writing trajectory to the size range [0,64] and keep the aspect ratio constant. In addition, for the Japanese and Indian datasets, the high density of writing trajectory points may result in overlapping trajectory points after normalization. Therefore, we remove the overlapping points after scaling and downsample the remaining trajectory points. The model performs 500,000 iterations on the Chinese and Japanese datasets and 200,000 iterations on the English and Indian datasets. The batch size is set to 128, and the Adam optimizer is adopted for training with a learning rate of 0.001.

Table 1. Comparison with four current SOTA methods on four benchmarks. The best results are highlighted in bold.

Dataset	Method	AIoU \uparrow	LPIPS \downarrow	LDTW \downarrow	DTW \downarrow
Chinese Dataset	Cross-VAE [27]	0.146	0.402	13.64	1038
	Kanji-Net [19]	0.326	0.186	5.51	443
	DED-Net [2]	0.397	0.136	4.08	303
	PEN-Net [5]	0.450	0.113	3.11	234
	Ours	0.641	0.033	2.33	160
English Dataset	Cross-VAE [27]	0.238	0.206	7.43	177
	Kanji-Net [19]	0.356	0.121	5.98	150
	DED-Net [2]	0.421	0.089	4.70	110
	PEN-Net [5]	0.461	0.074	3.21	77
	Ours	0.608	0.035	3.06	74
Indian Dataset	Cross-VAE [27]	0.235	0.228	4.89	347
	Kanji-Net [19]	0.340	0.163	3.04	234
	DED-Net [2]	0.519	0.084	2.00	130
	PEN-Net [5]	0.546	0.074	1.62	105
	Ours	0.637	0.048	1.50	84
Japanese Dataset	Cross-VAE [27]	0.164	0.346	22.7	1652
	Kanji-Net [19]	0.290	0.236	6.92	395
	DED-Net [2]	0.413	0.150	4.70	214
	PEN-Net [5]	0.476	0.125	3.39	145
	Ours	0.564	0.074	3.22	119

3.3 Evaluation Results

Our research entailed comprehensive experiments to benchmark our proposed model against current state-of-the-art (SOTA) methods. These methods include DED-NET [2], Cross-VAE [27], Kanji-Net [19], and PEN-Net [5]. The comparative results, as tabulated in Table 1, unequivocally demonstrate the superior performance of our method across various evaluation metrics and datasets. Significantly, our approach exhibits remarkable improvements in datasets featuring complex characters, particularly Chinese scripts.

The observed enhancement in performance can be ascribed to two pivotal aspects of our model:

Global Radical Context-Aware (GRCA) Module: Our uniquely designed GRCA module plays a crucial role in comprehensively understanding the structural nuances within complex character images. This module’s ability to discern and integrate multi-scale radical features significantly contributes to the model’s efficacy in handling intricate character representations.

Integration of Transformer Architecture: Another key factor is the incorporation of Transformers within our model. The Transformer architecture effectively mitigates the issue of trajectory drift, a common challenge in character trajectory prediction. By leveraging self-attention mechanisms, the Transformer is adept at maintaining consistency in trajectory sequences, especially for lengthy and complex characters.

Furthermore, the visualizations presented reinforce our findings. While previous methods demonstrated proficiency in recovering simple character trajectories, they often faltered with more complex characters, leading to issues such as stroke repetition and trajectory drift. In stark contrast, our method consistently and accurately predicts complex character trajectories, showcasing remarkable reliability.

In summary, compared to existing SOTA methods, our approach not only excels in handling complex characters but also demonstrates unparalleled accuracy in predicting extensive trajectory sequences. This advancement marks a significant step forward in the field of handwriting recognition and trajectory prediction, particularly for scripts that pose intricate structural challenges.

3.4 Ablation Studies

In order to rigorously evaluate the efficacy of the individual components of our model, we conducted a series of ablation experiments. For these experiments, we utilized the PEN-Net [5], based on the Convolutional Recurrent Neural Network (CRNN) architecture, as our baseline model. The results, delineated in Table 2, provide a clear quantitative assessment of the enhancements contributed by each module in our proposed system.

Impact of Transformer Integration: The introduction of the Transformer into our model architecture has resulted in substantial improvements across all performance metrics. The key to this enhancement lies in the Trajectory Transformer’s ability to capture the contextual correlations within trajectory sequences. Unlike traditional approaches, the Transformer’s self-attention mechanism allows for a more nuanced understanding of the sequential dependencies and structural nuances in handwriting, resulting in more accurate trajectory predictions.

Effectiveness of the GRCA Module: The addition of the GRCA module has also shown a significant positive impact on the model’s performance. This improvement underscores the GRCA module’s robust capabilities in feature extraction and analysis, particularly for characters with complex structures. The module’s design, focusing on multi-scale radical features, equips the model with an advanced understanding of the intricate details and contextual relationships present in complex characters. This is especially pertinent in scripts where radicals play a crucial role in character formation, such as in Chinese handwriting (Fig. 4).

	Label	Cross-VAE	Kanji-Net	DED-Net	PEN-Net	Ours
Chinese	檄					
	競					
	檣					
	翼					
Japanese	子					
	錢					
	藤					
Indian						
English	K					
	A					

Fig. 4. Visualization of the trajectories recovered by different methods for qualitative analysis, with each color representing a specific stroke. Compared with our proposed methods, it shows that the precision of the previous methods is poor when recovering complex character images and a noticeable drift phenomenon arises in the recovery of lengthy sequences.

In conclusion, these ablation studies validate the substantial contributions of the Transformer and GRCA modules to our model’s overall performance. By systematically analyzing the performance improvements with each added module, we demonstrate that both the nuanced context handling of the Trajectory Transformer and the sophisticated feature analysis of the GRCA module are integral to the model’s success in accurately predicting handwriting trajectories (Fig. 5).

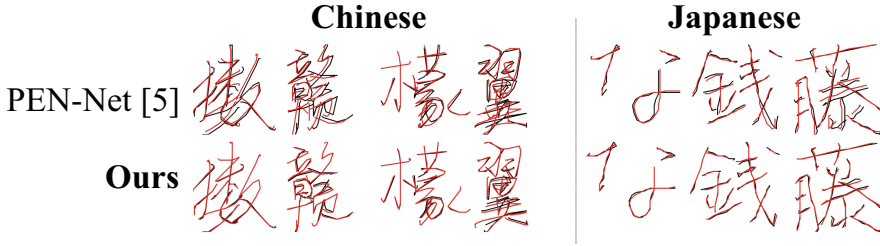


Fig. 5. Our method is compared with the visualized results of PEN-Net, where the black text is Groundtruth and the red text is the model prediction result. Although PEN-Net can recover the glyphs of text images as a whole, there is a certain degree of offset in the stroke details. (Color figure online)

Table 2. Ablation studies on our designed modules.

Transformer	GRCA	AIoU \uparrow	LPIPS \downarrow	LDTW \downarrow	DTW \downarrow
\times	\times	0.450	0.113	3.11	233.8
\checkmark	\times	0.627	0.042	2.55	182.9
\checkmark	\checkmark	0.641	0.033	2.33	159.2

4 Conclusions

In this research paper, we introduce the Trajectory Transformer, a novel approach devised for the precise prediction of trajectory points from handwritten text images. Our model is specifically tailored to address the intricate, multi-level structural relationships inherent in handwritten images, a challenge that has been a focal point in the field of handwriting analysis and recognition.

A key innovation in our approach is the development of the Global Radical Context-Aware (GRCA) module. This module is meticulously engineered to extract and analyze the nuanced relationships between various scale features of radicals in handwriting. Radicals, especially in certain scripts like Chinese, play a critical role in character structure and meaning. The GRCA module’s ability to dissect these features and their interrelationships is instrumental in enhancing the overall accuracy of character recognition.

To further refine our model, we have integrated Transformer architecture, renowned for its efficiency in modeling contextual relationships. In the realm of handwriting trajectory recovery, one of the persistent challenges is the drift that occurs in lengthy trajectory sequences. The Transformer’s self-attention mechanism adeptly tackles this issue, ensuring consistency and precision in trajectory prediction over extended sequences.

Evaluation Through Extensive Experiments and Ablation Studies: The efficacy of our Trajectory Transformer has been thoroughly evaluated through extensive experiments on benchmark datasets. Additionally, we have conducted detailed ablation studies to confirm the effectiveness of each individual compo-

ment within our model. These studies are crucial in isolating and understanding the contributions of the GRCA module and the Transformer architecture to the model's overall performance.

In conclusion, the combination of the GRCA module and Transformer architecture in our Trajectory Transformer presents a significant advancement in the field of handwriting recognition. Our model not only addresses key challenges such as the handling of complex structural relationships and trajectory drift but also sets a new benchmark in trajectory prediction accuracy.

Acknowledgements. The research is partially supported by the National Key R&D Program of China (No. 2023YFC3502900), National Natural Science Foundation of China (No. 62176093, 61673182), Key Realm R&D Program of Guangzhou (No. 202206030001), Guangdong Provincial Science and Technology Plan (No. 2023A0505030016), and Guangdong Natural Science Foundation (No. 2024A1515012217).

References

1. Archibald, T., Poggemann, M., Chan, A., Martinez, T.: Trace: a differentiable approach to line-level stroke recovery for offline handwritten text (2021)
2. Bhunia, A.K., et al.: Handwriting trajectory recovery using end-to-end deep encoder-decoder network. In: 2018 24th ICPR, pp. 3639–3644. IEEE (2018)
3. Boccignone, G., Chianese, A., Cordella, L.P., Marcelli, A.: Recovering dynamic information from static handwriting. *Pattern Recogn.* **26**(3), 409–418 (1993)
4. Cao, Z.S., Su, Z.W., Wang, Y.Z.: A model for recovering writing sequence from offline handwritten Chinese character image. In: 2008 Congress on Image and Signal Processing, vol. 1, pp. 298–302. IEEE (2008)
5. Chen, Z., Yang, D., Liang, J., Liu, X., Wang, Y., Peng, Z., Huang, S.: Complex handwriting trajectory recovery: evaluation metrics and algorithm (2022)
6. Cheng, B., Girshick, R., Dollár, P., Berg, A.C., Kirillov, A.: Boundary iou: improving object-centric image segmentation evaluation. In: CVPR, pp. 15334–15342 (2021)
7. Dai, G., Zhang, Y., Ke, Q., Guo, Q., Huang, S.: One-shot diffusion mimicker for handwritten text generation. In: European Conference on Computer Vision (2024)
8. Dai, G., et al.: Disentangling writer and character styles for handwriting generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5977–5986 (2023)
9. Diaz, M., Crispo, G., Parziale, A., Marcelli, A., Ferrer, M.A.: Impact of writing order recovery in automatic signature verification. In: International Graphonomics Conference, pp. 11–25. Springer (2022)
10. Diaz, M., Ferrer, M.A., Impedovo, D., Malik, M.I., Pirlo, G., Plamondon, R.: A perspective analysis of handwritten signature technology. *ACM Comput. Surv. (CSUR)* **51**(6), 1–39 (2019)
11. Faundez-Zanuy, M., Fierrez, J., Ferrer, M.A., Diaz, M., Tolosana, R., Plamondon, R.: Handwriting biometrics: applications and future trends in e-security and e-health. *Cogn. Comput.* **12**, 940–953 (2020)
12. Hassaine, A., Al Maadeed, S., Bouridane, A.: ICDAR 2013 competition on handwriting stroke recovery from offline data. In: 2013 12th ICDAR, pp. 1412–1416. IEEE (2013)

13. Huang, H., et al.: AGTGAN: unpaired image translation for photographic ancient character generation. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 5456–5467 (2022)
14. Jager, S.: Recovering writing traces in off-line handwriting recognition: using a global optimization technique. In: ICPR 1996, vol. 3, pp. 150–154. IEEE (1996)
15. Ji, N., et al.: Content-independent online handwriting verification based on multi-modal fusion. In: 2021 ICME, pp. 1–6 (2021). <https://doi.org/10.1109/ICME51207.2021.9428239>
16. Kato, Y., Yasuhara, M.: Recovery of drawing order from single-stroke handwriting images. TPAMI **22**(9), 938–949 (2000)
17. Li, H., Yang, D., Huang, S., Lam, K.M., Jin, L., Zhuang, Z.: Two-dimensional multi-scale perceptive context for scene text recognition. Neurocomputing **413**, 410–421 (2020)
18. Liu, C.L., Yin, F., Wang, D.H., Wang, Q.F.: Casia online and offline Chinese handwriting databases. In: 2011 ICDAR, pp. 37–41. IEEE (2011)
19. Nguyen, H.T., Nakamura, T., Nguyen, C.T., Nakagawa, M.: Online trajectory recovery from offline handwritten Japanese kanji characters. arXiv preprint [arXiv:2009.04284](https://arxiv.org/abs/2009.04284) (2020)
20. Nguyen, V., Blumenstein, M.: Techniques for static handwriting trajectory recovery: a survey. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, pp. 463–470 (2010)
21. Noubigh, Z., Kherallah, M.: A survey on handwriting recognition based on the trajectory recovery technique. In: 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), pp. 69–73. IEEE (2017)
22. Plamondon, R., Privitera, C.M.: The segmentation of cursive handwriting: an approach based on off-line recovery of the motor-temporal information. IEEE Trans. Image Process. **8**(1), 80–91 (1999)
23. Plamondon, R., Srihari, S.N.: Online and off-line handwriting recognition: a comprehensive survey. TPAMI **22**(1), 63–84 (2000)
24. Qiao, Y., Nishiara, M., Yasuhara, M.: A framework toward restoration of writing order from single-stroked handwriting image. TPAMI **28**(11), 1724–1737 (2006)
25. Qiao, Y., Yasuhara, M.: Recover writing trajectory from multiple stroked image using bidirectional dynamic search. In: 18th ICPR'06, vol. 2, pp. 970–973. IEEE (2006)
26. Rabhi, B., Elbaati, A., Hamdi, Y., Alimi, A.M.: Handwriting recognition based on temporal order restored by the end-to-end system. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1231–1236. IEEE (2019)
27. Sumi, T., Iwana, B.K., Hayashi, H., Uchida, S.: Modality conversion of handwritten patterns by cross variational autoencoders. In: 2019 ICDAR, pp. 407–412. IEEE (2019)
28. Wang, T.Q., Liu, C.L.: Handwriting trajectory recovery from off-line multi-stroke characters by deep ordering prediction and heuristic search. In: 2021 ICME, pp. 1–6 (2021). <https://doi.org/10.1109/ICME51207.2021.9428463>
29. Wu, X., Kimura, A., Iwana, B.K., Uchida, S., Kashino, K.: Deep dynamic time warping: End-to-end local representation learning for online signature verification. In: 2019 ICDAR, pp. 1103–1110. IEEE (2019)
30. 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)

31. Zhao, B., Yang, M., Tao, J.: Pen tip motion prediction for handwriting drawing order recovery using deep neural network. In: 2018 24th ICPR, pp. 704–709. IEEE (2018)
32. Zhao, B., Yang, M., Tao, J.: Drawing order recovery for handwriting Chinese characters. In: 2019 ICASSP, pp. 3227–3231. IEEE (2019)



Handwriting Intra-Variability Across Surface Transitions: Implications for Writer Identification

Kumari Priya¹, Chandranath Adak¹, Bidyut B. Chaudhuri²,
and Michael Blumenstein³

¹ Department of CSE, Indian Institute of Technology Patna, Patna 801106, India
{priya_2321cs02, chandranath}@iitp.ac.in

² Department of CSE, Techno India University, Salt Lake City, Kolkata 700091, India

³ FEIT, University of Technology Sydney, Ultimo 2007, Australia

Abstract. Handwriting exhibits intra-variability within the same writer due to friction between different writing surfaces, such as transitioning from paper to a computer tablet. This study investigates such intra-variability in handwriting characteristics across different writing surfaces and its implications for writer identification. An empirical study is conducted to assess the performance of state-of-the-art deep architectures in identifying writers amidst such intra-variation. Additionally, a transformer-based model is proposed to capture writer identification under these intra-variable circumstances. A dataset comprising 1560 handwritten English text-line images from 130 writers is created and utilized for experimentation. The results reveal insightful outcomes regarding the utilization of deep architectures and the proposed model in handling intra-variability for writer identification. This study contributes to advancing the understanding of intra-variability in handwriting and offers practical implications for forensic analysis and document authentication in the digital age.

Keywords: Biometric · Computer Forensics · Handwriting
Intra-Variability · Transformer Networks · Writer Identification

1 Introduction

In the realm of forensic investigation and beyond, the analysis of handwriting has long been a fundamental tool for identifying individuals and authenticating documents. However, with the advent of digital technology, traditional pen-and-paper writing is progressively giving way to computer tablets for various writing tasks [14]. This shift raises pertinent questions about the consistency and reliability of handwriting characteristics across different writing surfaces. Forensic contexts, such as writer identification in legal documents or criminal investigations, necessitate a deep understanding of handwriting intra-variability (variations within an individual's writing) [2] across surface transitions to ensure

accurate analysis and interpretation [10,25]. Moreover, in the education sector, studying handwriting analysis amidst surface transitions can provide valuable insights into students' adaptability [15].

The friction between the writing instrument and the writing surface plays a pivotal role in shaping handwriting [15]. For instance, on a computer tablet, factors like pressure sensitivity, screen responsiveness, and the angle of the writing instrument contribute to variations in writing output. As individuals adapt their writing behaviors to accommodate digital mediums, it becomes imperative to examine the degree of intra-variability that may emerge within their handwriting patterns [3]. Writings among different writers are distinguished by unique styles, known as inter-class variance or *inter-variability*. Moreover, within the writings of a single person, considerable variations can occur due to various mechanical, physical, and psychological factors [22], termed intra-class variance or *intra-variability* [4]. Despite significant intra-variations in ink-strokes among handwritten samples of a writer, individuals familiar with certain writing over an extended period may still identify it. This ability may stem from implicit stroke characteristics present in the writing [1].

This paper explores the intra-variability inherent in individual handwriting across surface transitions, particularly focusing on the transition from paper to computer tablets, and its implications for writer identification. While the field of pattern recognition lacks studies similar to ours, research in the domains of education and psychology has addressed related topics [14,15]. Gerth et al. [15] studied whether age-related effects exist in graphomotor execution due to variations in writing surfaces. Their another research [14] also indicated that proficient writers can adjust their handwriting movements to suit the writing surface. The study by Alamargot et al. [6] examined the impact of writing on tablet screens on students' graphomotor skills across different grade levels. Some past studies also explored the impact of varying writing instruments (e.g., pen, and pencil) on individual handwriting, revealing insights into these tools' influence on intra-variability [19,23].

In this paper, we recognize the potential of transformer networks in addressing the challenges posed by handwriting intra-variability across different writing surfaces. Through our empirical study, we aim to assess the effectiveness of deep architectures in accurately identifying writers amidst intra-variation and discerning subtle nuances in handwriting patterns. The past researches on writer identification can be found in [8,26,36]. In recent days, contemporary deep convolutional architectures have also been employed for writer identification, including models like CaffeNet [13], AlexNet [27], SqueezeNet, GoogLeNet, Xception Net, VGG, ResNet, etc. [2]. Integration of global and local features in architectures can also be seen, e.g., FragNet [17], GR-RNN [18]. Very recently, papers employing spatial attention [30] and multi-head self-attention [5,21] have emerged. By bridging the gap between traditional handwriting analysis and emerging digital technologies, our paper aims to provide valuable insights for practitioners across diverse fields.

Our **contributions** to this paper are outlined as follows:

(i) We conducted a thorough study on writer identification using the intra-variable characteristics found within an individual’s handwriting. This involves a comprehensive and detailed investigation into the unique attributes and variations present in individual writing style.

(ii) We investigated both traditional pen-and-paper writing and computer tablet writing, demonstrating intra-variation influenced by a range of writing instruments. We systematically investigate the variables within each category to mimic real-world writing conditions, encompassing writing on paper, on-screen display tablets, and off-screen graphics tablets. This comprehensive approach facilitates research in handwriting analysis by capturing the nuanced variations observed across different writing mediums and environments.

(iii) We have proposed a transformer-based network and conducted rigorous experiments aimed at thoroughly investigating the effects of intra-variability in handwriting. Through systematic testing and data collection, we aimed to gain a comprehensive understanding of how surface transitions influence handwriting characteristics within individuals. Our experiments were designed to provide insights into the nuances of intra-variability and its implications for handwriting analysis on surface changes, with the goal of enhancing the accuracy and reliability of automated systems for writer identification.

The rest of the paper is organized as follows. In Sect. 2, we mention the employed dataset details and associated challenges. Section 3 presents the proposed method. The experimental analysis and results are discussed in Sect. 4. Finally, Sect. 5 concludes this paper.

2 Dataset Details and Challenges

This study aims to examine intra-variability in handwriting across surface transitions and its significance for writer identification. Given the absence of publicly available datasets meeting our specific requirements, we undertook the task of creating our own dataset.

Writer Details: Our dataset includes contributions from 130 distinct writers from various regions of India, all of whom have at least a professional working proficiency in English. None of the writers are known to be native English speakers, and all have completed at least a higher-secondary level of education with English as part of the curriculum. The ages of the writers range from 16 to 33 years, with an average age of 19.14 and a standard deviation of 2.22; among them, 101 are male, and 29 are female. The writers have different levels of experience with writing on computer tablets, ranging from extensive to minimal.

Text-Dependent Writing: In our dataset, all writers were tasked with writing a standard English pangram, “*The quick brown fox jumps over a lazy dog*”. This enabled us to closely examine the characteristics of each English character in a text-dependent manner.

Writing Surface: For each of the 130 writers, we engaged three writing surfaces as mentioned below:

(i) *Paper:* Each writer was provided with a standard form printed on 75 GSM white A4 paper, featuring blank $24.5 \text{ cm} \times 2 \text{ cm}$ sized rectangular boxes. Participants were instructed to scribble the above English pangram within this designated box. We provided all the writers with the same 5 writing tools, i.e., pencil, gel pen, fountain pen, 0.5 mm and 1 mm ball pens. Here, we have incorporated 2 distinct *paper* surfaces to capture a comprehensive array of handwriting variations. Firstly, participants were provided with a stack of paper containing fifty A4 sheets as platform, on which they kept the printed paper form to write, allowing for a *regular* writing experience akin to standard note-taking conditions. Secondly, we offered a wooden exam clipboard as an alternative to place only the printed paper form, providing a *hard* platform for writing tasks. Here, each writing tool offers unique tactile feedback and line thickness, contributing to the diverse range of handwriting styles observed in the dataset. By offering participants a selection of tools commonly encountered in everyday writing contexts, we aimed to capture the full spectrum of handwriting variability across different mediums and tools. Thus, using 5 tools on paper placed on both *regular* and *hard* platforms, each writer wrote $10 (= 5 \times 2)$ copies of the above pangram. The pages were scanned on an autofed flatbed scanner (EPSON DS-1630) to convert into digital images.

(ii) *On-screen display tablet:* Each writer wrote 1 copy of the above pangram on Wacom One Pen DTC133W0C Display Tablet (size: medium). We captured this writing as an image.

(iii) *Off-screen display tablet:* Each writer scribbled 1 copy of the abovementioned pangram on Wacom Intuos CTL-4100/K0-CX Digital Graphics Tablet (size: small). This device features a decoupled writing surface without a built-in display, requiring the writer to view the connected computer screen while writing. Here also, the writing was captured as an image.

In this way, we have collected $12 (= 10 + 1 + 1)$ handwriting samples from each of the 130 writers. Therefore, our dataset contains a total of $1560 (= 12 \times 130)$ handwritten text lines.

Major Challenges Observed in Dataset: This database offers valuable insights into intra-variant handwriting, which encompasses the natural variations in a person’s handwriting due to diverse mechanical factors. We have observed various challenging cases in our database, some of which are mentioned below:

(i) *Micro-structure intra-variation:* This type of variation occurs when the writer is less accustomed to touch tablets and must concentrate on the screen to which the tablet is connected. It reflects the deviations in the micro-level details of handwriting strokes due to the writer’s adjustment to the digital writing interface. These deviations are often observable in the subtle details of pen movement, line thickness, and overall flow of the handwriting, highlighting the impact of the digital medium on the intricacies of handwritten expression. Figure 1 shows the impact of micro-structure variations in Fig. 1(a3) where the writer has lost the smoothness of the writing that was maintained in Figs. 1(a1), (a2).

Writer-a	(a1)	The quick brown fox jumps over a lazy dog.
	(a2)	The quick brown fox jumps over a lazy dog
	(a3)	The quick brown fox jumps over a lazy dog
	(a4)	The quick brown fox jumps over a lazy dog
Writer-b	(b1)	The quick brown fox jumps over a lazy dog
	(b2)	The quick brown fox jumps over a lazy dog
	(b3)	The quick brown fox jumps over a lazy dog
	(b4)	The quick brown fox jumps over a lazy dog
Writer-c	(c1)	The quick brown fox jumps over a lazy dog.
	(c2)	The quick brown fox jumps over a lazy dog.
	(c3)	The quick brown fox jumps over a lazy dog.
	(c4)	The quick brown fox jumps over a lazy dog
Writer-d	(d1)	The quick brown fox jumps over a lazy dog
	(d2)	The quick brown fox jumps over a lazy dog
	(d3)	The quick brown fox jumps over a lazy dog
	(d4)	The quick brown fox jumps over a lazy dog

Fig. 1. Some samples from our database. Here, Writer-a has written four samples (a1)-(a4). Similarly, Writers-b, c, d have written (b1)-(b4), (c1)-(c4), (d1)-(d4), respectively. (a1), (b1), (c1), (d1): Writing on *paper* while the paper is placed on a hard surface. (a2), (b2), (c2), (d2): Scribbling on *paper* placed on a medium/ regular surface. (a3), (b3), (c3), (d3): *On-screen* display tablet writing. (a4), (b4), (c4), (d4): *Off-screen* graphics tablet scribbling. *Writing tools*: (a2), (d1): pencil;(b1), (c2): gel pen; (d2): fountain pen; (b2): 0.5 mm ball pen; (a1), (c1): 1 mm ball pen.

(ii) *Stroke reduction intra-variation*: It occurs due to the continuous surface of the tablet screen, which requires the writer to maintain contact without lifting the pen as frequently as with traditional paper writing. Consequently, the writing process on a tablet involves longer strokes and fewer interruptions in pen movement, leading to alterations in the typical patterns of handwriting strokes. This change in writing behavior can result in differences in stroke length, spacing between letters, and overall fluidity of the handwriting, reflecting the influence of the digital medium on the intricacies of writing style. Figures 1(b1), (b2), (b3), (b4) reflect the stroke reduction in the handwriting samples.

(iii) *Ink type intra-variation*: It refers to the variations in handwriting characteristics resulting from the distinct visco-elastic properties of the inks used in these two types of pens. The gel pen's smooth, consistent lines from gel-based ink contrast with the intermittent flow of the ballpoint pen's oil-based ink, leading to varying stroke widths and intensities in handwriting. Figures 1(c1), (c2), (c3), (c4) illustrate the variation in ink flow within the handwriting samples.

(iv) *Idiosyncratic intra-variation*: Arises from the writer's tendency to employ different forms of letters while writing. This phenomenon reflects the unique stylistic choices and habits of the writer, resulting in variations in the shapes, sizes, and embellishments of individual letters within the handwriting samples. These idiosyncrasies contribute to the distinctive and personalized appearance of the handwriting, highlighting the individuality and nuances in the writer's writing style. In the writing structure of the first character 'T', we can see a clear difference in Fig. 1(d4) from Figs. 1(d1), (d2), (d3).

3 Proposed Methodology

The aim of this study is to examine handwriting intra-variability across transitions between various writing surfaces and explore its relevance to writer identification. Therefore, we first formally define the problem in the context of writer identification, and subsequently discuss the methodology to address it.

3.1 Problem Formulation

A handwriting image (\mathcal{I}) has been given as input. The task is to identify the writer (w_i) from a set of writers (\mathcal{W}), who has scribbled the text; for $i = 1, 2, \dots, |\mathcal{W}|$. Therefore, we formulate the undertaken task as a multiclass classification problem to classify the correct writer-class $w_i \in \mathcal{W}$ of handwriting image \mathcal{I} . The database employed in this paper contains samples from 130 writers; i.e., $|\mathcal{W}| = 130$.

3.2 Solution Architecture

Before moving on to the main processing module, we conducted some preprocessing steps.

Pre-processing: A handwritten text-line image (I) is input to our model, which is first resized into $d_h \times d_w$ sized I_T without distorting the aspect ratio. Such distortion is undesirable in the context of writer identification tasks due to preserving the ink-stroke individuality or writer inter-variability. We introduce zeros into certain rows or columns to maintain the aspect ratio. We here empirically fix $d_h = 192$, and $d_w = 1920$.

Transformer Network: For the task at hand, our approach involves utilizing a transformer network due to its minimal inductive bias and resilience to noise [12]. However, unlike directly employing the Vision Transformer (ViT) [12] that feeds raw image patches directly to the transformer encoder, our model initially extracts deep features from the image patches before embedding in the transformer encoder. This workflow is visually depicted in Fig. 2.

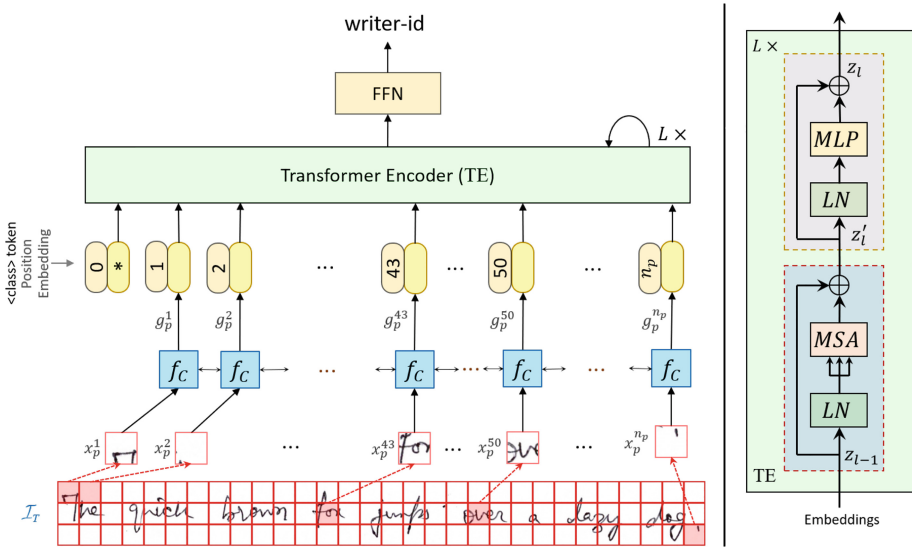


Fig. 2. Workflow of the proposed writer identification model

I_T is partitioned into a sequence of non-overlapping n_p patches denoted as x_p^i (for $i = 1, 2, \dots, n_p$), each having dimensions $d_p \times d_p \times c_p$, where c_p represents the channel count of I_T . Each patch x_p^i undergoes processing through a convolutional architecture f_c to extract the corresponding deep feature g_p^i with a dimension of d_g . In f_c , we utilize the layers before the global_average_pool of ResNeXt-50 [35], as it exhibited superior performance compared to contemporary models like VGG19 [29], ResNet [16], MobileNetV2 [28], etc. The weights of f_c 's are shared across all patches. We refrain from utilizing a distinct objectness network [34] to delineate between background and foreground ink-stroke (object) patches, since transformer network inherently leverage the attention mechanism [33] to prioritize important patches. Here, $c_p = 3$, since I_T is an RGB image. Empirically, we

fix $d_p = 64$; therefore, $n_p = \lfloor (d_h \times d_w) / (d_p \times d_p) \rfloor = \lfloor (192 \times 1920) / (64 \times 64) \rfloor = 90$. We also choose $d_g = 2048$.

Each g_p^i undergoes further flattening and is then transformed into a D -dimensional vector, i.e., embedding z_0 through transformer layers [12], utilizing the following linear projection:

$$z_0 = [g_{class} ; g_p^1 \mathbb{E} ; g_p^2 \mathbb{E} ; \dots ; g_p^{n_p} \mathbb{E}] + \mathbb{E}_{pos} \quad (1)$$

Here, $\mathbb{E} \in \mathbb{R}^{d_g \times D}$ is the embedding projection matrix; $\mathbb{E}_{pos} \in \mathbb{R}^{(n_p+1) \times D}$ denotes the position embedding added to the deep feature embeddings extracted from patches, serving to retain positional information; $g_{class} = z_0^0$ refers to a learnable embedding [11]. Following the embedding space, a sequence of transformer encoder is incorporated [12, 33]. The right-hand side of Fig. 2 depicts the internal structure of a transformer encoder, which consists of alternating layers of *MSA* (Multi-head Self-Attention) [12] and *MLP* (Multi-Layer Perceptron) [37] modules. *LN* (Layer Normalization) [7], and residual connections [37] are applied before and after each of these modules, respectively. This composition is formally presented in Eq. 2 with general semantics. The *MLP* module utilized here comprises two layers with $4D$ and D neurons, respectively, incorporating the GELU (Gaussian Error Linear Unit) non-linear activation function [12].

$$z'_l = MSA(LN(z_{l-1})) + z_{l-1}; z_l = MLP(LN(z'_l)) + z'_l; l = 1, 2, \dots, L \quad (2)$$

where, L denotes the total count of transformer blocks. The core element of the transformer encoder is *MSA*, which incorporates h (> 1) number of *heads*. Each *head* ^{i} , $\forall i \in \{1, 2, \dots, h\}$ uses *SA* (Scaled dot-product Attention) [12, 33], wherein the input consists of query (Q), key (K), and value (V) matrices. The *SA* module calculates the attention assigned to the input patches. The results of *SA* computations across all heads are concatenated within the *MSA* module, as illustrated below.

$$\begin{aligned} MSA(Q, K, V) &= [head^1, head^2, \dots, head^h]; \\ head^i &= SA(Q \cdot W_q^i, K \cdot W_k^i, V \cdot W_v^i); \\ SA(Q, K, V) &= softmax\left(QK^T / \sqrt{D_h}\right)V \end{aligned} \quad (3)$$

where, W_q, W_k, W_v are the weight matrices for the linear transformation; $D_h = D/h$. Following L transformer encoder blocks, the <class> token [11] is enriched with contextual information. The learnable embedding state resulting from the transformer encoder (z_L^0) serves as the image representation y [12]; $y = LN(z_L^0)$.

Feed Forward Network (FFN): The final stage of our model integrates an FFN consisting of two hidden layers, sequentially added with 1024 and D nodes, respectively, which engages GELU activation function. The output layer consists of n_w neurons with *softmax*, resulting in the distribution $s^{<j>}$, from which the writer-id w is identified as below.

$$w = \arg \max_j s^{<j>} ; \text{ for } j = 1, 2, \dots, n_w ; \quad (4)$$

where, n_w is the number of writers in the database, i.e., $n_w = |\mathcal{W}|$. Here, we utilize cross-entropy loss, as it has been found effective for multi-class classification tasks [37]. Furthermore, we employ the Adam optimizer [20]. The details regarding hyper-parameter tuning and training are elaborated in Sect. 4.

4 Experiments and Discussions

This section starts with an outline of the dataset and experimental setups, followed by experimental results to evaluate the effectiveness of our model and some contemporary deep architectures for the undertaken task.

4.1 Dataset Employed and Experimental Setups

As mentioned in Sect. 2, we have collected a total of 1560 handwritten English text-line images from 130 individuals. Each writer scribbled 10 samples on “*paper*”, 1 sample on “*on-screen*” display tablet, and 1 sample on “*off-screen*” graphics tablet.

We have created 4 experimental setups (*ES*), as below:

- *ES-1*: All samples written on *paper* (i.e., 10×130 samples) were used as training set, and *on-screen* 1×130 samples were engaged for testing.
- *ES-2*: The training set was kept similar to *ES-1*, and *off-screen* 1×130 samples were used for testing.
- *ES-3*: The training set was the same as *ES-1*. The test set combined the *on-screen* and *off-screen* samples used for testing in *ES-1* and *ES-2* (i.e., 2×130 samples).
- *ES-4*: We randomly split all 1300 ($= 10 \times 130$) samples written on *paper* only into training and test sets with a ratio of 8 : 2. We ensured that the training set included samples from all 130 writers.

In each of the above experimental setups, 10% of the training data was allocated for validation purposes. During model training, we augmented the training set samples by introducing random changes in image saturation, brightness, and contrast to mitigate overfitting.

4.2 Results

We performed the experiments on an Intel(R) Xeon(R) CPU @ 2.00 GHz with 52 GB RAM and Tesla T4 16 GB GPU. The hyperparameters of the employed models were tuned and fixed during the model training, considering the performance of the validation set [37]. For training, the mini-batch size was equal to 16. In this study, 100 epochs were used for model training. The Adam optimizer [20] parameters were selected as follows: the initial learning rate was set to 10^{-4} , the exponential decay rates for the 1st and 2nd moment estimates, β_1 and β_2 , were set to 0.9 and 0.999, respectively, and the zero-denominator removal

parameter (ϵ) was set to 10^{-8} . For transformer network, we empirically chose $L = 6$, $D = 192$, and $h = 12$. All results presented in this paper were obtained from the testing set. We utilize average *top-1 accuracy* % over all writers as the evaluation metric for assessing model performance [2].

Table 1. Performance analysis on various experimental setups and comparative study

Methods		Top-1 Accuracy %			
		<i>ES-1</i>	<i>ES-2</i>	<i>ES-3</i>	<i>ES-4</i>
Baseline	VGG19 [29]	56.4391	56.3369	57.3468	67.4889
	ResNet50-V2 [16]	64.3467	64.2342	65.4734	79.3504
	Inception-V3 [31]	69.3458	69.3456	69.3456	80.0581
	Xception [9]	69.3563	69.3873	69.5647	80.1904
	MobileNet-V2 [28]	69.4737	69.3884	70.4098	80.4259
	EfficientNet-B3 [32]	72.3422	<u>72.8463</u>	<u>72.6833</u>	81.2626
	RAM [24]	<u>72.4523</u>	72.2346	72.5842	<u>81.4558</u>
SOTA	Fiel et al. [13]	54.5692	54.1956	54.8073	65.4442
	GR-RNN [18]	71.7432	71.6591	72.1226	81.0562
	Koepf et al. [21]	73.2389	73.2420	73.3460	81.6678
	Srivastava et al. [30]	75.1104	75.0188	75.3365	81.7277
	WiT [5]	75.3602	<u>75.3600</u>	75.5308	83.1328
	FragNet [17]	<u>75.3613</u>	75.0404	<u>76.2904</u>	<u>83.4105</u>
Ours	75.6061	75.5975	76.8509	83.6481	

Table 1 presents the performance of our model, and provides a comparison with some major baseline deep architectures [9, 16, 24, 28, 29, 31, 32] and state-of-the-art (SOTA) writer identification methods [5, 13, 17, 18, 21, 30] across above-mentioned four experimental setups (i.e., *ES-1*, *ES-2*, *ES-3*, and *ES-4*).

From the results presented in Table 1, we have the following major **observations**:

(i) Our method outperformed major baseline deep architectures and SOTA methods by attaining 75.6061%, 75.5975%, 76.8509%, and 83.6481% top-1 accuracies for *ES-1*, *ES-2*, *ES-3*, and *ES-4*, respectively.

(ii) Among the compared baseline and SOTA methods, EfficientNet-B3 [24] and FragNet [17] obtained superior performances, respectively, in *ES-3*. In all individual setups, the best performances of baseline and SOTA methods are underlined in Tabel 1.

(iii) Overall, the comparable methods, including ours, achieved better performances in *ES-4* than other setups. One possible reason is that the training and test samples in *ES-4* encompass handwriting produced by various tools (e.g., pencil, gel pen, fountain pen, 0.5 mm and 1 mm ball pens) on *paper* while placed on regular and hard surfaces.

(iv) Overall, our method and some major baseline/ SOTA methods encountered challenges stemming from the surface transition from paper to computer tablets. This is evident from performances on *ES-1*, *ES-2*, and *ES-3* setups, where training samples were written on paper while test samples were scribbled on computer tablets.

(v) We also noted that overall, the comparable methods, including ours, demonstrated better performance in *ES-1* compared to *ES-2*. One plausible reason is that individuals encountered more challenges in *ES-2* when writing on an *off-screen* graphics tablet, which involves a decoupled writing surface while viewing the computer screen. However, for some writers, the act of scribbling on a *on-screen* relatively smoother surface of a display tablet posed challenges in *ES-1*.

These observations highlight the presence of intra-variation in handwriting resulting from surface transitions, as evidenced by the performance of writer identification methods. As a matter of fact, while baseline and state-of-the-art (SOTA) methods achieve high accuracy in benchmark datasets [36], their performance is notably poorer in the dataset examined in this paper.

Ablation Study: We also performed an ablation study by removing the f_C component from our model (refer to Fig. 2 and Sect. 3.2). After ablating f_C , we obtained top-1 accuracies of 73.2389%, 73.2420%, 73.3460%, and 81.6678% in *ES-1*, *ES-2*, *ES-3*, and *ES-4* setups, respectively. This ablation led to a decrease in accuracy ranging from 1% to 3.5%.

5 Conclusion

This paper studies the intricate challenge of understanding intra-variability in handwriting, which encompasses the variations observed across different writing surfaces, ranging from traditional paper sheets to modern computer tablets. The study explores these diverse writing contexts, including writing on both traditional paper and digital tablets, with and without visual displays, to shed light on how they influence the characteristics and patterns of handwriting. By examining the variations arising from these different conditions, the paper aims to provide insights into the underlying factors contributing to intra-variability in handwriting. We utilized a transformer network with deep features to assess performance in this study. We curated an intra-variable handwriting dataset across various surfaces, incorporating English handwriting samples from 130 distinct writers, totaling 1560 samples. Our model demonstrated an overall accuracy of 76.8509% on this dataset, showcasing promising outcomes. In future research endeavors, we aim to further investigate intra-variation stemming from various writing tools and explore multiple scripts to deepen our understanding of handwriting characteristics across diverse contexts.

Acknowledgment. The authors heartily thank all the writers/ volunteers, and interns/ trainees/ researchers, who helped in database generation. C. Adak acknowledges the partial support from the SERB, DST, Govt. of India, under grant no. SRG/2022/002151.

References

1. Adak, C.: A study on automated handwriting understanding. Ph.D. thesis, University of Technology Sydney, Australia (2019). <https://opus.lib.uts.edu.au/handle/10453/134139>
2. Adak, C., Chaudhuri, B.B., Blumenstein, M.: An empirical study on writer identification and verification from intra-variable individual handwriting. *IEEE Access* **7**, 24738–24758 (2019). <https://doi.org/10.1109/ACCESS.2019.2899908>
3. Adak, C., Chaudhuri, B.B., Blumenstein, M.: A deep reinforcement learning-based study on handwriting difficulty analysis. In: *Advances in Pattern Recognition and Artificial Intelligence*, pp. 97–117. World Scientific (2022). <https://doi.org/10.1142/9789811239014.0006>
4. Adak, C., Chaudhuri, B.B., Lin, C.T., Blumenstein, M.: Intra-variable handwriting inspection reinforced with idiosyncrasy analysis. *IEEE TIFS* **15**, 3567–3579 (2020). <https://doi.org/10.1109/TIFS.2020.2991833>
5. Adak, C., Jaswanth, B., Akhtar, Z., Kåsen, A., Chanda, S.: Writer identification from nordic historical manuscripts using transformer networks. In: *IJCB*, pp. 1–9 (2023). <https://doi.org/10.1109/IJCB57857.2023.10448665>
6. Alamargot, D., Morin, M.F.: Does handwriting on a tablet screen affect students' graphomotor execution? A comparison between grades two and nine. *Hum. Mov. Sci.* **44**, 32–41 (2015). <https://doi.org/10.1016/j.humov.2015.08.011>
7. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. In: *Deep Learning Symposium, NIPS* (2016)
8. Bulacu, M., Schomaker, L.: Text-independent writer identification and verification using textural and allographic features. *IEEE TPAMI* **29**(4), 701–717 (2007). <https://doi.org/10.1109/TPAMI.2007.1009>
9. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: *CVPR*, pp. 1251–1258 (2017). <https://doi.org/10.1109/CVPR.2017.195>
10. Costain, J.: Questioned documents and the law: handwriting evidence in the federal court system. *J. Forensic Sci.* **22**(4), 799–806 (1977). <https://doi.org/10.1520/JFS10422J>
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: *NAACL-HLT*, pp. 4171–4186. ACL (2019). <https://doi.org/10.18653/v1/N19-1423>
12. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: *ICLR* (2021). <https://openreview.net/forum?id=YicbFdNTTy>
13. Fiel, S., Sablatnig, R.: Writer identification and retrieval using a convolutional neural network. In: *CAIP*, pp. 26–37. Springer (2015). https://doi.org/10.1007/978-3-319-23117-4_3
14. Gerth, S., et al.: Adapting to the surface: a comparison of handwriting measures when writing on a tablet computer and on paper. *Hum. Mov. Sci.* **48**, 62–73 (2016). <https://doi.org/10.1016/j.humov.2016.04.006>
15. Gerth, S., et al.: Is handwriting performance affected by the writing surface? comparing preschoolers', second graders', and adults' writing performance on a tablet vs. paper. *Front. Psychol.* **7**, 211618 (2016). <https://doi.org/10.3389/fpsyg.2016.01308>
16. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: *ECCV*, pp. 630–645 (2016). https://doi.org/10.1007/978-3-319-46493-0_38

17. He, S., Schomaker, L.: Fragnet: writer identification using deep fragment networks. *IEEE TIFS* **15**, 3013–3022 (2020). <https://doi.org/10.1109/TIFS.2020.2981236>
18. He, S., Schomaker, L.: GR-RNN: global-context residual recurrent neural networks for writer identification. *Pattern Recogn.* **117**, 107975 (2021). <https://doi.org/10.1016/j.patcog.2021.107975>
19. Hilton, O.: Effects of writing instruments on handwriting details. *J. Forensic Sci.* **29**(1), 80–86 (1984). <https://doi.org/10.1520/JFS11637J>
20. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *ICLR (Poster)* (2015)
21. Koepf, M., Kleber, F., Sablatnig, R.: Writer identification and writer retrieval using vision transformer for forensic documents. In: *DAS*, pp. 352–366 (2022). https://doi.org/10.1007/978-3-031-06555-2_24
22. Koppenhaver, K.M.: *Forensic document examination: principles and practice* (2007)
23. Mathyer, J.: The influence of writing instruments on handwriting and signatures. *J. Crim. L. Criminol. Police Sci.* **60**, 102 (1969). <https://doi.org/10.2307/1141743>
24. Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K.: Recurrent models of visual attention. *NeurIPS* **27** (2014). <https://dl.acm.org/doi/10.5555/2969033.2969073>
25. Morris, R.N.: *Forensic handwriting identification: fundamental concepts and principles* (2020)
26. Plamondon, R., Lorette, G.: Automatic signature verification and writer identification—the state of the art. *Pattern Recogn.* **22**(2), 107–131 (1989). [https://doi.org/10.1016/0031-3203\(89\)90059-9](https://doi.org/10.1016/0031-3203(89)90059-9)
27. Rehman, A., Naz, S., Razzak, M.I., Hameed, I.A.: Automatic visual features for writer identification: a deep learning approach. *IEEE Access* **7**, 17149–17157 (2019). <https://doi.org/10.1109/ACCESS.2018.2890810>
28. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: inverted residuals and linear bottlenecks. In: *CVPR*, pp. 4510–4520 (2018). <https://doi.org/10.1109/CVPR.2018.00474>
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *ICLR* (2015)
30. Srivastava, A., Chanda, S., Pal, U.: Exploiting multi-scale fusion, spatial attention and patch interaction techniques for text-independent writer identification. In: *ACPR*, pp. 203–217 (2021). https://doi.org/10.1007/978-3-031-02444-3_15
31. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: *CVPR*, pp. 2818–2826 (2016). <https://doi.org/10.1109/cvpr.2016.308>
32. Tan, M., Le, Q.: Efficientnet: rethinking model scaling for convolutional neural networks. In: *ICML*, pp. 6105–6114. *PMLR* (2019). <https://proceedings.mlr.press/v97/tan19a.html>
33. Vaswani, A., et al.: Attention is all you need. *NeurIPS* **30** (2017)
34. Wang, J., Tao, X., Xu, M., Lu, J.: Boundary objectness network for object detection and localization. In: *ICASSP*, pp. 2336–2340 (2018). <https://doi.org/10.1109/ICASSP.2018.8462434>
35. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: *CVPR*, pp. 1492–1500 (2017). <https://doi.org/10.1109/cvpr.2017.634>
36. Xiong, Y.J., Lu, Y., Wang, P.S.: Off-line text-independent writer recognition: a survey. *IJPRAI* **31**(05), 1756008 (2017). <https://doi.org/10.1142/S0218001417560080>
37. Zhang, A., Lipton, Z.C., Li, M., Smola, A.J.: *Dive into Deep Learning*. Cambridge University Press, Cambridge (2023)



Enhancing Table Structure Recognition via Bounding Box Guidance

Lei Hu and Shuangping Huang^(✉)

South China University of Technology, Guangzhou, China
eehulei@mail.scut.edu.cn, eehsp@scut.edu.cn

Abstract. Table Structure Recognition (TSR) aims to extract the bounding boxes of cells and table structure (e.g., HTML) from table images. Although current approaches have made significant progress, the latest image-to-sequence methods overlook the explicit utilization of the bounding box information when predicting HTML sequences, leading to error predictions in complex scenes. In this paper, we introduce a novel framework **BGTR** (**B**ounding **B**ox-**G**uided **T**able **R**ecognizer). To more effectively utilize bounding box information, we first predict the bounding boxes of cells and then use this information to guide the generation of HTML sequences. While utilizing bounding box information can enhance the accuracy of HTML sequences, for natural scene tables, the data volume is too small to allow for sufficient training of bbox-guided HTML generation. In response, we adopt a progressive training method for natural scene tables and introduce **SNSTab**, a synthetically generated natural scene table dataset. Our experiments on five benchmark datasets demonstrate SOTA performance.

Keywords: Table structure recognition · Image-to-sequence · Bounding box guidance · Dataset

1 Introduction

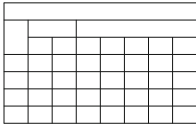
Tables are a crucial medium for structured information dissemination. Table detection (TD) aims to extract the position of tables from document images, and many methods [1, 8, 20] have shown excellent results. Table Structure Recognition (TSR) aims to transform images containing tables into structured data, which is both crucial and challenging. Leveraging the advancements of transformer [23], which have proven highly effective in various fields [7], image-to-sequence methods [3, 11, 18, 27] have demonstrated promising results in TSR. These methods employ an encoder-decoder architecture to simultaneously predict the HTML (Hyper Text Markup Language) sequence and the bounding box (bbox) of table cells. In predicting HTML sequences, they rely solely on image information and

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78498-9_15.

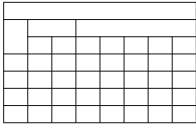
overlook the explicit utilization of bbox information. However, table structure and formatting can be highly complex, and bbox information is essential for parsing the structure of tables. Therefore, exclusive reliance on image information may lead to error predictions in complex scenes, like spanning cells (Fig. 1(a)). In this paper, we introduce a novel framework **BGTR** (**B**ounding **B**ox-**G**uided **T**able **R**ecognizer). Unlike previous image-to-sequence methods [3, 11, 18, 27], we explicitly utilize bbox information to obtain accurate HTML sequences. We first use a Bbox Predictor to predict bboxes. Then, during HTML sequence decoding, we enable the Bbox-Guided Structure Decoder to perceive both the image and bbox information of the table, resulting in accurate HTML sequences.

Maximum time to record video and photo size							
Memory card(Gb)	VIDEO(minutes)		PHOTO				
	1920*1080P(Full HD)	1280*720P(HD)	4000*3000(12M)	3648*2736(10M)	3264*2448(8M)	2560*1920(5M)	1920*1080(2M)
32G	368	640	9232	11056	13584	21616	47552
16G	184	320	4616	5528	6792	10808	23776
8G	92	160	2308	2764	3396	5404	11888
4G	46	80	1154	1382	1698	2702	5944

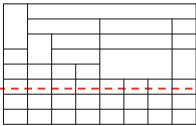
w/ bbox-guided ✓



ground truth



w/o bbox-guided ✗



(a)

Digital Document Tables			
Dataset	Samples	Datasets	Samples
PubTabNet [31]	500K	FinTabNet [30]	100K
SynthTabNet [18]	600K	PubTables-1M [21]	758K
Natural Scene Tables			
Dataset	Samples	Datasets	Samples
TabRecSet [26]	38K	WTW [15]	14K
iFLYTAB [28]	17K	TAL [5]	15K

(b)

Fig. 1. The motivation behind the proposed method. (a) Comparison of HTML visualization with and without **bbox-guided** generation on TabRecSet [26], the **red** dotted boxes indicate error results, the **blue** dotted boxes indicate spanning cells. The results indicate that using bboxes for guidance achieves better performance in spanning cells. (b) A comparison between two types of datasets reveals that the sample size of the **digital document** table dataset is significantly larger than that of the **natural scene** table dataset. (Color figure online)

Although utilizing bbox information can improve the accuracy of the HTML sequence, for natural scene tables, as shown in Fig. 1(b), the data volume is

small, and the structure and style of tables in natural scenes are complex, making it insufficient for adequate training of bbox-guided HTML generation. In response, we adopt a progressive training method for natural scene tables and introduce **SNSTab**. Progressive training method includes a foundation training stage and an advancement training stage. In the foundation training stage, we aim to train the model with a large number of tables from natural scenes, thereby enabling it to learn how to more effectively utilize bbox information for guiding the generation of HTML sequences, this approach leads to improve the model’s foundational understanding of tables. In the advancement training stage, training is conducted on a specific natural scene table dataset (e.g., TabRecSet [26] and iFLYTAB [28]). **SNSTab** is a synthetically generated natural scene table dataset containing 500k table images for the foundation training stage. It includes wired tables, wireless tables, inclined tables, and curved tables, featuring diverse table structures and backgrounds. This variety enables the model to comprehensively learn various aspects of table knowledge during the foundation training stage, thereby achieving better results in complex scenes like spanning cells and deformed tables, as shown in Fig. 7.

Extensive experiments demonstrate the effectiveness of our proposed BGTR and the progressive training method, achieving state-of-the-art performance on five public benchmarks.

To sum up, our contributions are as follows:

- We propose **BGTR**, a novel framework that explicitly utilizes bbox information for guiding HTML sequence generation, which aims at enhancing structural recognition accuracy in challenging table scenes.
- To ensure that bbox-guided HTML generation is adequately trained in natural scenes, we adopt a progressive training method and introduce **SNSTab**, a synthetically generated natural scene table dataset for the foundation training stage.
- Our experiments on five benchmark datasets demonstrate state-of-the-art performance.

2 Related Work

2.1 Table Structure Recognition

With the rapid development of deep learning, a variety of table structure recognition methods have emerged, which can be divided into three categories: graph-based methods, split-and-merge methods, and image-to-sequence methods.

Graph-Based Methods. These methods utilize cells or text boxes as the basic elements of the table, employing a graph network to determine the row and column relationships between them. GraphTSR [4] utilized graph attention networks to the TSR task, determining the row and column relationships of adjacent cells through graph edge classification. TabStruct-Net [19] implemented a unified end-to-end framework for cell detection and cell relationship analysis.

GFTE [12] employed a graph-based convolutional network that integrates image features, position features, and textual features to predict relationships between cells. NCGM [14] enabled cooperation among geometry, appearance, and content modalities, leveraging their interaction to enhance multi-modal representation in intricate situations. However, these methods are limited by their reliance on additional bbox data or OCR accuracy, leading to potential errors in table structure recognition, and additionally, they need complex post-processing methods.

Split-and-Merge Methods. Typically, these methods comprise two models: the split model and the merge model. The split model initially detects the row and column regions of the table and then intersects them to obtain the grid cells of the table. Subsequently, the merge model is employed to determine which adjacent grid cells need to be merged. SPLERGE [22] became the first to use the split-and-merge framework for the TSR task, addressing an issue where previous methods struggled with resolving spanning cells. By utilizing textual information, SEM [29] achieved enhanced results on complex tables with spanning cells. To address geometric distortion in table images, TSRFormer [13] approached the detection of row and column regions as a linear regression problem. However, two-stage training can be complex and resource-intensive, potentially leading to longer training times and difficulties in optimization compared to more streamlined, end-to-end methods.

Image-to-Sequence Methods. These methods treat the table as a structured sequence (e.g., HTML or LATEX), using an encoder-decoder framework to convert the table image into a structured sequence that fully describes the table structure. EDD [31] employed a CNN-based encoder to extract the visual features from table images and utilized two LSTM-based decoders to simultaneously recognize the table structure and cell content. TableMaster [27] introduced a transformer-based [23] architecture, achieving significant progress in the TSR task by recognizing the table structure and cell bboxes simultaneously. Based on TableMaster [27], VAST [11] treated bbox prediction as a coordinate sequence generation task and introduced a visual-alignment loss that significantly improved bbox accuracy. However, bbox information is essential for parsing the structure of table, unlike previous methods that produce inaccurate HTML sequence predictions in complex table scenes due to the lack of bbox information, this paper utilizes bbox information to guide the generation of HTML sequences, resulting in more accurate HTML sequences.

2.2 Existing Datasets

While the size of table datasets has significantly increased, existing datasets primarily focus on digital documents [18, 21, 30, 31], such as PDF files. Building a digital document table dataset is relatively straightforward because annotated information can be directly extracted from PDF files. However, tables captured in natural scenes through cameras cannot be automatically annotated, and manual annotation is a time-consuming process. Additionally, natural scene tables are more complex, often inclined, rotated, and curved, further increasing the

annotation difficulty. Due to these challenges, there is a substantial disparity in the number of table datasets between natural scenes and digital documents, as illustrated in Fig. 1(b). To address this issue, we propose a large-scale synthetically generated natural scene table dataset.

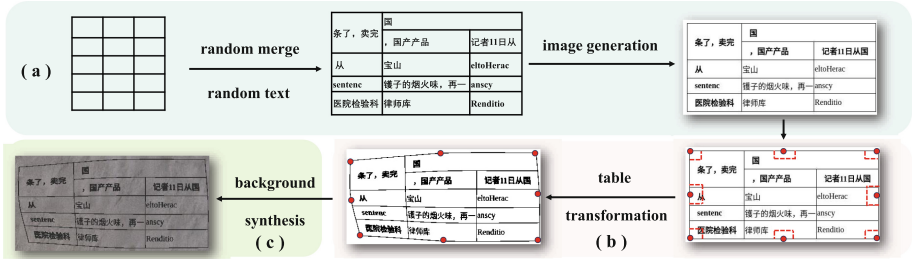


Fig. 2. The production process of SNSTab: (a) table generation. (b) table transformation. (c) background synthesis.

3 SNSTab

SNSTab contains 500k synthetic images of natural scene tables, including wired tables, wireless tables, provincial line tables, inclined tables, curved tables and large tables. Although image generation has achieved significant success in other fields [6], its application in table recognition remains quite limited. To our knowledge, SNSTab is the **first** large-scale natural scene table synthesis dataset. SNSTab’s annotations contain the coordinates of the table cells, the text inside the cells, and the HTML sequence that describes the table structure. The creation of the SNSTab dataset involves three phases: table generation, table transformation, and background synthesis, as shown in Fig. 2.

Table Generation. This step is to generate digital document table images. We randomly generate table images based on the open source tool Table Generation¹. First, we will generate a grid with a random number of rows and columns; Then, we will randomly merge the adjacent grids to get spanning cells, and generate random text for each grid; Finally, we convert the above table into HTML sequences, and get the final table image through the browser rendering.

Table Transformation. Since tables in the nature scene tend to be inclined or rotated. Therefore, after automatically generating tables, we apply thin plate spline (TPS) [2] to randomly transform them. This simulation captures the complexities observed in natural scenes. As shown in Fig. 2(b), we take the four vertices of the table image and the midpoints of the four sides as the source points, the target points are then obtained by randomly moving the source points within

¹ <https://github.com/WenmuZhou/TableGeneration>.

the range of the red dotted line. After the TPS transformation, the coordinates of the cells are also transformed.

Background Synthesis. In addition to their complex structures, tables in natural scenes often exhibit a variety of backgrounds. We captured 400 background images of natural scenes, including paper, walls, daily-life items, and more. For each table image, a random background image is first selected, and then a random area of the same size as the table image is extracted from the background image. Finally, the table image is merged with the selected background to produce the final image.

For more details of the dataset and for additional dataset samples, please refer to the supplementary materials.

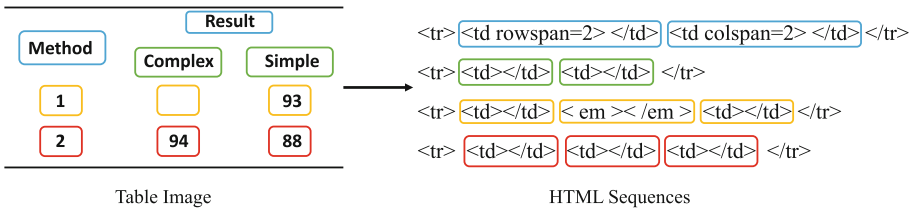


Fig. 3. A simple example of using HTML sequences to represent a table structure, different colors in the figure represent different rows of the table.

4 Method

4.1 Preliminary

In this paper, we utilize HTML sequences to represent the table structure, as shown in Fig. 3. Given a table image, our model outputs HTML sequences of the table and the corresponding cell bboxes. To better facilitate prediction, we tokenize HTML sequences into HTML tokens. For cells without spanning, non-empty cells and empty cells are denoted by $\langle td \rangle \langle /td \rangle$ and $\langle em \rangle \langle /em \rangle$, respectively. In the case of spanning cells, the tokens are divided into three parts: $\langle td, colspan = N \text{ or } rowspan = N, \text{ and } \rangle \langle /td \rangle$. Here, $\langle td$ indicates the beginning of the spanning cells, N specifies the count of cells that are spanning, and $\rangle \langle /td \rangle$ marks the end of the spanning cells. $\langle tr \rangle$ and $\langle /tr \rangle$ respectively represent the beginning and the end of each row in a table. We use $H_N = \{h_i\}_{i=1}^N \in \mathbb{R}^{N \times 1}$ to denote HTML sequences, where N is the sequence length and h_i denotes the i -th HTML token. We use $B_N = \{b_j\}_{j=1}^N \in \mathbb{R}^{N \times 4}$ to denote the bbox of table cells. For each cell, its bbox is represented as $[x_1, y_1, x_2, y_2]$, where $[x_1, y_1]$ denotes the coordinates of the top-left corner, and $[x_2, y_2]$ represents the coordinates of the bottom-right corner. Moreover, the HTML tokens have a one-to-one correspondence with the bboxes, and the bbox value is non-zero only if the HTML token is $\langle td \rangle \langle /td \rangle$ and $\langle td$.

4.2 Overall Architecture

The overall framework of BGTR is illustrated in Fig. 4. Given a table image, denoted as $\mathbf{P} \in \mathbb{R}^{H \times W \times 3}$, where H and W represent the height and width of the image, respectively. We employ an Image Encoder to extract image features, resulting in the feature map $\mathbf{F}_{encoder} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times d}$, where d denotes the dimension of the features. After applying 2D positional encoding, the flattened image features are obtained as $\mathbf{F}_{image} \in \mathbb{R}^{\frac{HW}{64} \times d}$. The image features \mathbf{F}_{image} are further fed into the Shared Decoder for decoding, resulting in decoded features $\mathbf{F}_{share}^1 \in \mathbb{R}^{N \times d}$. The Shared Decoder is used to reduce the gap between the image and the sequence, making it more aligned with the sequential features. \mathbf{F}_{share}^1 is first fed into the Bbox Decoder to obtain bboxes B_N . Then, \mathbf{F}_{share}^1 is sent into the Bbox-Guided Structure Decoder (Sect. 4.3), using the predicted bbox information to guide the generation of HTML sequences H_N . For additional details regarding the Image Encoder, Shared Decoder, and Bbox Decoder, please refer to Sect. 5.2.

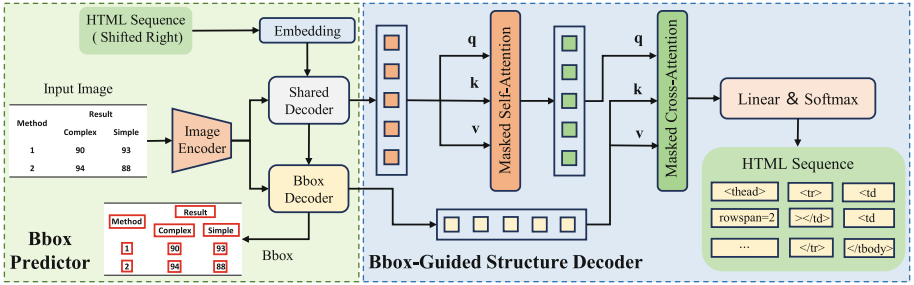


Fig. 4. Architecture of BGTR. The Bbox Predictor aims to acquire the bboxes of table cells and consists of three parts: an image encoder, a shared decoder, and a bbox decoder. The Bbox-Guided Structure Decoder generates the HTML sequence.

4.3 Bbox-Guided Structure Decoder

To more effectively utilize bbox information, we first predict the bboxes of cells and then utilize the bbox information to enhance the accuracy of HTML sequence prediction. Since we employ an autoregressive decoding approach, we utilize parallel training methods during training to accelerate the training speed. Specifically, the Bbox-Guided Structure Decoder receives $\mathbf{F}_{share}^1 \in \mathbb{R}^{N \times d}$ from the Shared Decoder and $\mathbf{F}_{bbox} \in \mathbb{R}^{N \times d}$ from the Bbox Decoder as input. In the Bbox Decoder, after \mathbf{F}_{bbox} passes through a linear layer and a sigmoid layer, the bboxes $B_N \in \mathbb{R}^{N \times 4}$ are obtained. \mathbf{F}_{share}^1 initially passes through a masked self-attention layer, resulting in $\mathbf{F}_{share}^2 \in \mathbb{R}^{N \times d}$. Here, the mask refers to the prediction of the current time step HTML token being based on the output of previous time steps. Subsequently, \mathbf{F}_{share}^2 and \mathbf{F}_{bbox} are fed into a masked

cross-attention layer. Here, the mask indicates that the prediction of the current time step HTML token is based on the bbox outputs of both the current and previous time steps. \mathbf{F}_{share}^2 serves as the query vector, while \mathbf{F}_{bbox} serve as the key/value vectors. By utilizing the cross-attention mechanism, bbox information \mathbf{F}_{bbox} becomes effectively integrated into \mathbf{F}_{share}^1 . The use of \mathbf{F}_{bbox} for decoding allows the model to comprehensively understand the position and relative relationships of each cell while predicting HTML sequences. This process allows the model to generate HTML sequences guided by bbox information. After passing through a linear layer and a softmax layer, the decoder’s output yields the final HTML sequences $H_N = \{h_i\}_{i=1}^N \in \mathbb{R}^{N \times 1}$.

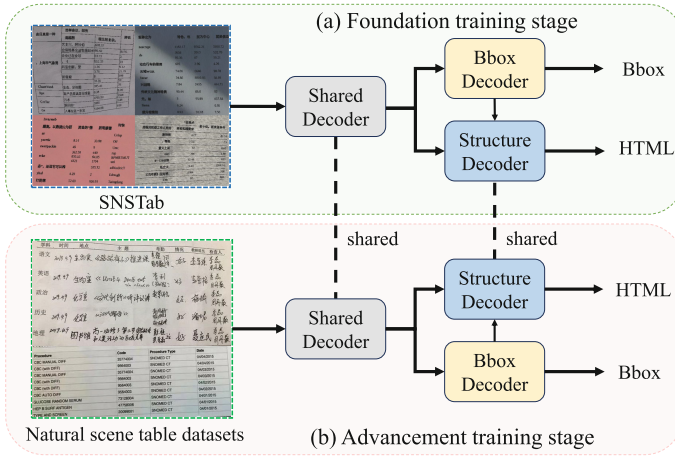


Fig. 5. An overview of the progressive training method. (a) Foundation training stage training on SNSTab. (b) Advancement training stage training on natural scene table datasets (e.g., TabRecSet [26] and iFLYTAB [28]).

4.4 Progressive Training Method

As illustrated in Fig. 5, this section will discuss the implementation of the progressive training method.

Foundation Training Stage. In the foundation training stage, the purpose is for the model to acquire common knowledge about tables. Due to the diverse types and varied structures of tables in natural scenes, the foundation training stage requires a large number of data samples. Based on this, we introduce SNSTab, a large synthetic table dataset in natural scenes. For further details about SNSTab, please refer to Sect. 3 and the supplementary material. After completing the foundation training stage on SNSTab, the model develops a foundational capability for recognizing table structures in natural scenes. Additionally, it can learn how to effectively utilize bbox information to guide the

generation of HTML sequences, particularly in complex scenes such as spanning cells and deformed tables (Fig. 7).

Advancement Training Stage. Building on the foundation training stage, the advancement training stage is conducted on a specific natural scene table dataset. With the common knowledge acquired in the foundation training stage, the model demonstrates improved convergence speed and enhanced overall training effectiveness in the advancement training stage. And in this stage, the Shared Decoder and the Bbox-Guided Structure Decoder are initialized using the training from the foundation training stage. Due to certain differences in the data between the two stages, the Bbox decoder is trained from scratch.

Through the progressive training process, the issue of insufficient data leading to inadequate training of bbox-guided HTML generation in natural scenes has been significantly alleviated.

4.5 Loss Functions

Our model adopts an end-to-end training approach and includes two loss functions. For the Bbox Decoder, L_1 loss is employed to supervise the prediction of bboxes, which is denoted as \mathcal{L}_{bbox} . For Bbox-Guided Structure Decoder, cross-entropy loss is utilized to supervise the prediction of HTML tokens, which is denoted as \mathcal{L}_{html} . The final loss function is formulated as follows:

$$\mathcal{L} = \lambda \mathcal{L}_{html} + \mathcal{L}_{bbox}, \quad (1)$$

where λ is the hyperparameter.

5 Experiments

5.1 Datasets and Evaluation Metric

Datasets. Our method is evaluated on five popular public benchmarks, including TabRecSet [26], iFLYTAB [28], PubTabNet [31], FinTabNet [30] and SynthTabNet [18].

TabRecSet [26] is a natural scene table dataset featuring tables from diverse scenes with various forms. It has 32.07K images and 38.17K tables, the number of images is not equal to the number of tables because some images contain multiple tables. As TabRecSet did not provide a predefined split, we randomly divided the dataset into train and test splits (80%, 20%), resulting in 30.6k training table images and 7.5k testing table images.

iFLYTAB [28] has 12,104 training samples and 5,187 testing samples. It contains both wired and wireless tables from natural scenes and digital documents.

PubTabNet [31] contains 500,777 training images and 9,115 validating images, each accompanied by annotation information detailing the table structure and text content along with their positions. All the tables are extracted from the

scientific articles, and annotations are automatically obtained from the PDF source files.

FinTabNet [30] is a large-scale dataset containing 91596 training tables, 10,635 validating tables and 10,656 testing tables. All the tables are sourced from the annual reports of the S&P 500 companies. Following [11, 17, 18, 30], we use validating sets for testing.

SynthTabNet [18] is a synthetically generated dataset with diverse table styles, complex structures, and an increased number of rows and columns. It contains 480k training images, 60k validating images, and 60k testing images. In addition to the bounding boxes of the non-empty cell, it also has the bounding boxes of the empty cell.

Evaluation Metric. The Tree-Edit-Distance-based Similarity (TEDS) [31] is employed as the evaluation metric, treating tables as tree structures. To mitigate the impact of OCR errors on the final score, we also utilize TEDS-S to assess the accuracy of the table structure without the table content.

Table 1. Comparison with state-of-the-art methods. PT indicates that progressive training method is used. S indicates simple tables. C indicates complex tables. **Bold** indicates the best performance, while underline indicates the second-best performance. \star indicates the image-to-sequence method. \dagger means pre-training on PubTabNet [31].

Method	PubTab		FinTab	SynthTab	TabRecSet			iFLYTAB
	TEDS-S	TEDS	TEDS-S	TEDS-S	TEDS-S			TEDS-S
					S	C	All	
EDD \star [31]	89.90	88.30	90.06	-	95.01	77.71	91.03 \dagger	-
GTE [30]	93.01	-	87.10	-	-	-	-	-
TableMaster \star [27]	96.04	96.16	-	-	97.20	84.11	94.14	84.63
SEM [29]	-	93.70	-	-	-	-	-	75.90
NCGM [14]	-	95.40	-	-	-	-	-	-
TableFormer \star [18]	96.75	93.60	96.80	<u>96.70</u>	-	-	-	-
VAST \star [11]	97.23	<u>96.31</u>	<u>98.63</u>	-	-	-	-	-
GridFormer [17]	97.00	95.84	<u>98.63</u>	-	-	-	-	-
SEMv2 [28]	<u>97.50</u>	-	-	-	-	-	-	92.00
TSRFormer [13]	<u>97.50</u>	-	-	-	-	-	-	-
BGTR \star	97.63	96.57	98.89	99.11	<u>98.35</u>	<u>89.27</u>	<u>96.23</u>	<u>91.02</u>
BGTR (PT) \star	-	-	-	-	98.65	92.47	97.21	92.00

Table 2. Comparison of cell bbox detection results on PubTabNet. PP indicates the post-processing.

Method	mAP	mAP (PP)
EDD + BBox [18]	79.2	82.7
TableFormer [18]	82.1	86.8
BGTR	91.9	-

5.2 Implementation Details

In this paper, the experimental settings are as follows: the table images are resized to 480×480 , and the flattened image sequence length is 3600. The dimension of the features d is 512. The multi-head number is 8. The maximum HTML sequence length is 500. We used Ranger [25] as the optimizer, the mini-batch size is set to 8. For TabRecSet, PubTabNet, FinTabNet and SynthTabNet, we trained 25 epochs, the initial learning rate is established at $1e-3$, and divided by 10 at 17 and 22 epochs. For iFLYTAB, we trained 120 epochs, the initial learning rate is established at $1e-3$, and divided by 10 at 75 and 105 epochs. For the foundation training stage on SNSTab, we trained 3 epochs, the initial learning rate is established at $1e-3$. Experiments are conducted using 2 NVIDIA GeForce RTX 3090 GPUs with 24 GB of RAM memory.

We use the ResNet-50 [10] combined with the Multi-Aspect GCA [16] module and 2D positional encoding to form the Image Encoder. To enhance the model’s understanding of the 2D topology of table images, we employ 2D positional encoding to encode image features. The Shared Decoder comprises two identical stacked transformer [23] decoding layers. The Bbox Decoder comprises a single transformer [23] decoding layer. The Bbox-Guided Structure Decoder comprises two identical stacked transformer [23] decoding layers.

5.3 Comparison with Previous State-of-the-Arts

As shown in Table 1, our method not only outperforms non-image-to-sequence methods, but also outperforms the best image-to-sequence method.

Results on Natural Scene Tables. We evaluate the performance of our model on two natural scene table datasets: TabRecSet [26] and iFLYTAB [28]. Given the absence of a baseline method in TabRecSet, TableMaster [27] is adopted as the baseline. We divide the dataset into two categories: simple (S) and complex (C). A table is considered complex if it contains spanning cells, otherwise, it is classified as a simple table. On TabRecSet, a TEDS-S score of 98.65% for simple tables and 92.47% for complex tables is achieved. Compared with baseline TableMaster, our method demonstrates improvements of 1.45% on simple tables, 8.36% on complex tables, and 3.07% overall. On iFLYTAB, a TEDS-S accuracy of 92.00% is achieved by our method, comparable to SEMv2 [28] and outperforms other methods.

Results on Digital Document Tables. The performance of our model is also evaluated on three digital document table datasets: PubTabNet [31], FinTabNet [30] and SynthTabNet [18]. For PubTabNet, similar to previous methods [9, 11, 17], the OCR results are from the text detection method PSENet [24] and text recognition method MASTER [16], and we match the text bboxes to the cell bboxes as described in [27]. A TEDS-S score of 97.63% and a TEDS score of 96.57% are achieved on PubTabNet which outperforms other methods. For FinTabNet and SynthTabNet, TEDS-S scores of 98.89% and 99.11% are achieved, respectively. Compared with TableFormer [18], our method exhibits improvements of 2.09% and 2.41% on FinTabNet and SynthTabNet, respectively.

In addition, we evaluate the performance of cell bbox detection on PubTabNet [31] using the PASCAL VOC mAP metric. As shown in Table 2, our method outperforms TableFormer [18] by 5.1% even without using post-processing.

The results on five datasets validate the effectiveness of using bbox to guide the generation of HTML sequences.

5.4 Visualization

We illustrate some visualization of BGTR in PubTabNet [31], FinTabNet [30], SynthTabNet [18], TabRecSet [26] and iFLYTAB [28]. As shown in Fig. 6, BGTR is adept at handling a wide range of scenarios and complex table structures. This includes tables with row and column spans, those containing multi-line text, as well as instances with empty cells. Moreover, it demonstrates strong robustness in both digital documents and natural scene environments.

Table 3. Ablation studies of module design. **BG** signifies bbox-guided HTML generation. **PT** signifies the progressive training method.

Methods		TEDS-S		
BG	PT	Simple	Complex	All
		98.30	86.50	95.54
✓		98.35	89.27	96.23
✓	✓	98.65	92.47	97.21

5.5 Ablation Studies

For simplicity, we conduct ablation experiments on TabRecSet [26]. Several experiments were conducted to validate the effectiveness of our methods.

Effectiveness of Module Design. As indicated in Table 3, **BG** signifies bbox-guided HTML generation. **PT** signifies the progressive training method, we constructed the baseline experiment following the previous methods [3, 11, 18, 27] which overlook the explicit utilization of bbox information when predicting

HTML sequences. Utilizing **BG** significantly improves the TEDS-S score by 2.77% on complex tables, indicating the effectiveness of guiding HTML sequence generation with bbox information in complex table scenes. Meanwhile, **PT** enhances the model’s generalization capabilities, particularly in handling complex tables, proving the effectiveness of the progressive training method. As shown in Fig. 7, using the progressive training method can yield better results on spanning cells and deformed tables.

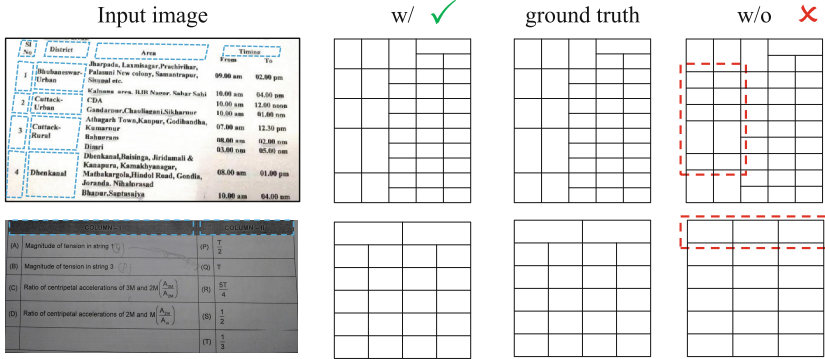


Fig. 7. Comparison of HTML visualization w/ and w/o progressive training on TabRec-Set [26], the red dotted boxes indicate error results, the blue dotted boxes indicate spanning cells. (Color figure online)

Table 5. Ablation studies of λ in loss function.

λ	TEDS-S		
	Simple	Complex	All
0.5	98.49	91.82	96.94
1	98.65	92.47	97.21
2	98.57	91.92	97.02

Effectiveness of Advancement Training Stage Training Method. As indicated in Table 4, placing a check mark (✓) signifies that the module continues to the advancement training stage of training, building upon the foundation training stage, while its absence indicates starting the training anew. From the results, we can see that **SD** (Share Decoder) and **BGD** (Bbox-Guided Structure Decoder) are very helpful for the training in the advancement training stage. This indicates that training in the foundation training stage with a large amount of data enables the model to learn a wide variety of table structures. However,

due to the data differences between two stages, **BD** (Bbox Decoder) is not much of a help for the training in the advancement training stage.

Effectiveness of λ in Loss Function. As indicated in Table 5, the table indicates that deep supervision positively impacts performance. However, the numerical results demonstrate a notable consistency across various trade-off parameter settings. For simplicity in model training, we recommend using $\lambda = 1$ in practical applications.

6 Conclusion

In this paper, we introduced BGTR, a novel framework that explicitly use bbox information to guide the generation of HTML sequences. Besides, to alleviate the problem of insufficient data leading to inadequate training of bbox-guided HTML generation in natural scenes, we adopted a progressive training method for natural scene tables and introduced SNSTab, a large synthetic table dataset in natural scenes. Experimental results on five benchmark datasets demonstrate that the proposed method achieves state-of-the-art performance.

Acknowledgements. The research is partially supported by National Key R&D Program of China (2023YFC3502900), National Natural Science Foundation of China (No. 62176093, 61673182), Key Realm R&D Program of Guangzhou (No. 202206030001), Guangdong Provincial Science and Technology Plan (No. 2023A0505030016).

References

1. Agarwal, M., Mondal, A., Jawahar, C.: Cdec-net: Composite deformable cascade network for table detection in document images. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 9491–9498. IEEE (2021)
2. Bookstein, F.L.: Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(6), 567–585 (1989)
3. Chen, B., Peng, D., Zhang, J., Ren, Y., Jin, L.: Complex table structure recognition in the wild using transformer and identity matrix-based augmentation. In: International Conference on Frontiers in Handwriting Recognition, pp. 545–561. Springer (2022)
4. Chi, Z., et al.: Complicated table structure recognition. arXiv preprint [arXiv:1908.04729](https://arxiv.org/abs/1908.04729) (2019)
5. Contributors, T.: Tal_ocr_table: a scene table structure recognition benchmark (2021). <https://ai.100tal.com/dataset>
6. Dai, G., Zhang, Y., Ke, Q., Guo, Q., Huang, S.: One-shot diffusion mimicker for handwritten text generation. In: European Conference on Computer Vision (2024)
7. Dai, G., et al.: Disentangling writer and character styles for handwriting generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5977–5986 (2023)
8. Gemelli, A., Vivoli, E., Marinai, S.: Graph neural networks and representation embedding for table extraction in pdf documents. In: 2022 26th International Conference on Pattern Recognition (ICPR), pp. 1719–1726. IEEE (2022)

9. Guo, Z., et al.: Trust: an accurate and end-to-end table structure recognizer using splitting-based transformers. arXiv preprint [arXiv:2208.14687](https://arxiv.org/abs/2208.14687) (2022)
10. 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)
11. Huang, Y., et al.: Improving table structure recognition with visual-alignment sequential coordinate modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11134–11143 (2023)
12. Li, Y., et al.: Gfte: graph-based financial table extraction. In: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part II, pp. 644–658. Springer (2021)
13. Lin, W., et al.: Tsrformer: table structure recognition with transformers. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 6473–6482 (2022)
14. Liu, H., Li, X., Liu, B., Jiang, D., Liu, Y., Ren, B.: Neural collaborative graph machines for table structure recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4533–4542 (2022)
15. Long, R., et al.: Parsing table structures in the wild. In: ICCV, pp. 944–952 (2021)
16. Lu, N., et al.: Master: multi-aspect non-local network for scene text recognition. *Pattern Recogn.* **117**, 107980 (2021)
17. Lyu, P., et al.: Gridformer: towards accurate table structure recognition via grid prediction. In: Proceedings of the 31st ACM International Conference on Multimedia, pp. 7747–7757 (2023)
18. Nassar, A., Livathinos, N., Lysak, M., Staar, P.: Tableformer: table structure understanding with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4614–4623 (2022)
19. Raja, S., Mondal, A., Jawahar, C.: Table structure recognition using top-down and bottom-up cues. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16, pp. 70–86. Springer (2020)
20. Shehzadi, T., et al.: Towards end-to-end semi-supervised table detection with deformable transformer. In: International Conference on Document Analysis and Recognition, pp. 51–76. Springer (2023)
21. Smock, B., Pesala, R., Abraham, R.: Pubtables-1m: towards comprehensive table extraction from unstructured documents. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4634–4642 (2022)
22. Tensmeyer, C., Morariu, V.I., Price, B., Cohen, S., Martinez, T.: Deep splitting and merging for table structure decomposition. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 114–121. IEEE (2019)
23. Vaswani, A.: Attention is all you need. arXiv preprint [arXiv:1706.03762](https://arxiv.org/abs/1706.03762) (2017)
24. Wang, W., et al.: Shape robust text detection with progressive scale expansion network. In: CVPR (2019)
25. Wright, L., Demeure, N.: Ranger21: a synergistic deep learning optimizer. arXiv preprint [arXiv:2106.13731](https://arxiv.org/abs/2106.13731) (2021)
26. Yang, F., Hu, L., Liu, X., Huang, S., Gu, Z.: A large-scale dataset for end-to-end table recognition in the wild. *Sci. Data* **10**(1), 110 (2023)
27. Ye, J., et al.: Pingan-vcgroup’s solution for icdar 2021 competition on scientific literature parsing task b: table recognition to html. arXiv preprint [arXiv:2105.01848](https://arxiv.org/abs/2105.01848) (2021)
28. Zhang, Z., et al.: Semv2: table separation line detection based on instance segmentation. *Pattern Recogn.* **149**, 110279 (2024)

29. Zhang, Z., Zhang, J., Du, J., Wang, F.: Split, embed and merge: an accurate table structure recognizer. *Pattern Recogn.* **126**, 108565 (2022)
30. Zheng, X., Burdick, D., Popa, L., Zhong, X., Wang, N.X.R.: Global table extractor (GTE): a framework for joint table identification and cell structure recognition using visual context. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 697–706 (2021)
31. Zhong, X., ShafieiBavani, E., Jimeno Yepes, A.: Image-based table recognition: data, model, and evaluation. In: *European Conference on Computer Vision*, pp. 564–580. Springer (2020)



A Deep-Learning Based Real-Time License Plate Recognition System for Resource-Constrained Scenarios

Karthik Mohan¹(✉) and Suraj Kumar Pandey²

¹ School of Computing, SASTRA University, Trichy-Tanjore Road,
Thirumalaisamudram, Thanjavur 613401, Tamil Nadu, India
125156054@sastra.ac.in

² Department of Computer Science, IIT Guwahati, Amingaon, North Guwahati,
Guwahati 781039, Assam, India
suraj18a@iitg.ac.in

Abstract. License Plate Recognition plays a pivotal role in modern traffic law enforcement, ensuring public safety and order. However, conventional surveillance systems such as CCTVs and static cameras lack real-time response capabilities and have limited mobility. With the growing number of vehicles, the requirement for automated and mobile methods for license plate recognition has soared. The noisy and dynamic environment of license plates further exacerbates this issue. While Deep Learning (DL) can help automate such tasks, the computational demands of DL pose a significant hurdle for real-time usage and mobility. In this regard, the declining costs and enhanced computational capabilities of microcontrollers offer promising potential for enabling the implementation of DL-based techniques in license plate recognition in resource-constrained scenarios. This paper introduces an approach for automated license plate recognition designed to guarantee mobility and real-time responsiveness. The proposed framework integrates various elements, encompassing microcontrollers, Internet of Things (IoT), Deep Neural Networks, and computer vision technologies. Furthermore, to alleviate the computational overhead on the microcontroller, the system leverages Transfer Learning and Cloud Computing for enhanced efficiency. The system was tested for real-time performance using a camera onboard a microcontroller, which was used to detect the license plates. The system delivered good accuracy for license plate recognition, both across existing datasets and for real-time images on multiple metrics. This system can also be integrated with wearable devices such as helmets or goggles and used by traffic law officials to facilitate easy monitoring and surveillance of traffic laws.

Keywords: IoT · Deep Learning · Microcontrollers

1 Introduction

Road safety continues to be a significant developmental issue, a public health concern, and a leading cause of death and injury worldwide. In 2021, road accidents resulted in a total of 1.19 million fatalities globally, emerging as the primary cause of death for individuals aged 5 to 29 [1]. The Global Status Report on Road Safety 2023 from the WHO further underscored the importance of enhancing the effectiveness of enforcing traffic laws for various violations as a key objective. Manual methods of license plate recognition are often faced with several problems compromising their efficacy [2]. They encounter complications in scalability, especially with handling high traffic [3]. Moreover, efficiently recalling the license plates of moving vehicles is a task of considerable difficulty for humans [2]. In light of these shortcomings, there has been a growing shift towards increased adoption of Automatic License Plate Recognition (ALPR) systems, aiming to establish a more sophisticated and intelligent transportation system [2]. With its capability of reading and detecting data from heavy volumes of fast-moving vehicles, ALPR has been integrated into various domains such as parking management, tolling, traffic management, policing, and many others [4,5]. Furthermore, the additional information provided by ALPR makes it well-suited to aid tasks such as search and surveillance.

However, ALPR often faces issues with noisy data that is rife with complex variations, which hampers the performance of the system. License plate deflection, which refers to tilted and turned images of license plates, makes accurate recognition challenging for models trained on simple data sets with well-centered images [6]. Another prominent issue is noise in license plate images caused by varying weather conditions such as rain or snow, resulting in blurred and unevenly lit license plates, further impeding the accuracy [7].

While Deep Learning (DL) based methods can tackle noisy data, the computational demands of DL models are high. This requires the models to be kept in immobile computationally rich servers, which impedes the mobility of the solution required for real-time on-site usage.

To overcome challenges related to noisy data and enable mobile usage, the proposed solution integrates a microcontroller for capturing license plate images. These captured images are then transmitted to a remote server utilizing IoT-based methods, and the computational requirements of the DL model are offloaded to the remote server. Through the integration of these components, the system guarantees portability while efficiently handling noisy data to achieve precise license plate recognition.

2 Related Works

2.1 Overview of ALPR Works

There has been considerable research focusing on the development of accurate and efficient systems for license plate recognition in recent years. This research can be mainly divided into two areas: License Plate (LP) Detection and LP

Character Recognition. We begin with a brief overview of previous works in both these areas and then discuss various end-to-end approaches in detail, along with their drawbacks.

LP Detection. LP detection involves identifying the presence and location of the license plate and its characters within an image. Traditional LP detection methods often rely on edge information [18–22] which make use of the distinctive edges and aspect ratio of license plates to detect them, or background analysis [23, 25, 26] which examine the differences in color between the license plate and the vehicular body. These techniques are generally lightweight and fast, however, at times they are often sensitive to irrelevant edges and varying illumination, leading to errors. Works like [27–29] used texture-based methods that leverage the unique texture and color transitions of license plates to detect and localize them. Despite these techniques being generally considered robust, they also suffer from the issue of high computational complexity and inability to deal with changes in lighting conditions. With the rise of deep neural networks, many researchers have shifted towards these advanced techniques. Several studies such as [30–33] have utilized You Only Look Once (YOLO) [24] based networks and their variants for the detection and localization of license plates within images.

LP Character Recognition. LP Character recognition involves accurately extracting the sequence of characters present in the license plate. Works like [20, 34] use template matching techniques, where known fonts and character sizes are leveraged to classify license plate characters. These methods face challenges with generalization, variations in typography as well as a high computation time due to additional templates for rotated characters. There have also been multiple approaches which made use of feature extraction techniques like eigenvector transformation, Gabor Filters, etc. [35] in conjunction with traditional Machine Learning models like Support Vector Machines (SVM) [36] or Hidden Markov Models (HMM) [37] to recognize characters. Similar to LP detection, in recent years, there has been a major shift towards the usage of neural networks, especially Convolutional Neural Networks (CNNs) [38–40] which act as both feature extractors and classifiers directly from raw pixel data.

End-To-End Approaches. Zhang et al. [8] used a modified CycleGAN model for generating license plate images and an image-to-sequence network for license plate recognition. However, due to being trained only with license plates from China and Taiwan, this work presents limited practical applicability in other countries. The work done by Xu et al. [17], presents the widely used Chinese City Parking Dataset (CCPD) dataset and a baseline for recognition of Chinese license plates. They use a model RpNet which is capable of taking a single RGB image as input and predicting the bounding box of the plate and the corresponding characters. However, this work also suffers from the same issue of

limited suitability towards non-Chinese license plates as well as greater computational complexity. Pustokhina et al. [9] utilized K-Means clustering for image segmentation followed by a Convolutional Neural Network (CNN) for license plate recognition. The experimental evaluation of this work focused on data sets such as the Stanford Cars dataset and FZU Cars dataset, which do not accurately depict real-world scenarios by overlooking various challenges such as partial occlusion, diverse lighting conditions, and variations in plate sizes. Björklund et al. [10] proposed the usage of a tailored CNN architecture trained on a synthetic image data set, generated by modeling the critical variables of license plates. While synthetic generation can be useful for extending data sets, the realism of the generated images does not always represent the wide variability found in real-time images of license plates. Another issue in implementing license plate recognition is the usage of static cameras for license plate image capture, potentially reducing the quantity and quality of the captured data, as their installation locations may not necessarily align with the optimal locations [11].

2.2 ALPR for Resource-Constrained Scenarios

Despite the extensive research done for ALPR, work done in the context of resource-constrained scenarios remains limited. Most of the existing solutions are catered towards scenarios with sufficient computational resources. In recent times, there have been explorations towards edge computing and IoT-based approaches [41, 42]. However, the demand for ALPR systems that are capable of effectively operating in diverse, and challenging environments still persists.

Keeping in view the limitations of the existing approaches, the following requirements were identified for the proposed approach.

1. The solution should be in the form of a portable component, enhancing the application's flexibility and enabling it to perform more diverse tasks such as search and surveillance.
2. A deep learning framework should be utilized to ensure that the solution is robust and performs well under noisy and variable environments as well as handle variations in license plates.
3. In order to accommodate deep-learning-based inference, the proposed method should facilitate effective communication between the application user and a remote server capable of performing computationally intensive tasks.

3 Methodology

Based on the requirements identified in the previous section, as depicted in Fig. 1, the proposed license plate recognition system involves a portable device for capturing the image of the license plate, a deep-learning framework hosted on a remote server that is able to process the image and predict the license plate characters, and a proper channel to ensure effective communication between the two. The proposed ALPR system comprises the following components:

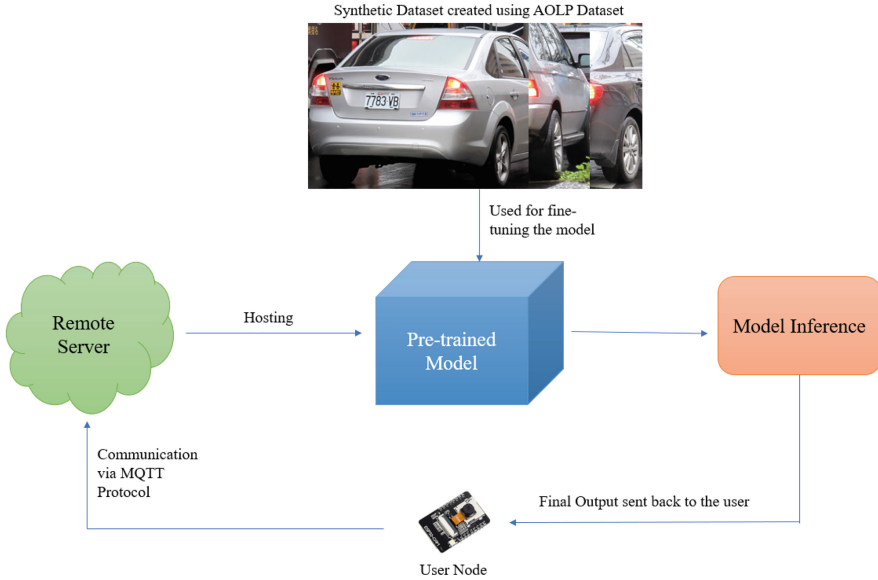


Fig. 1. The input image is captured by the microcontroller device, which is sent to the remote server hosting the pre-trained model, which has been fine-tuned using the synthetic dataset. Once the model has completed the prediction, it is sent communicated back to the user.

1. *User Node (U)* : It is a microcontroller device U equipped with a camera module for capturing real-time images of license plates. For its power source, either a power bank or a 5 V battery can be used. The microcontroller captures an image I of the license plate using the onboard camera. In order to perform training, processing, or inference, the image I is transmitted to a remote server R , using an IoT-based technique. In order to outsource the computational load, the microcontroller leverages the internet to connect to a GPU-enabled remote server. The internet connection can be realized by using a local WiFi hotspot or by using a microcontroller with built-in GSM support. The compactness of the microcontroller device enables seamless integration with wearable devices or handheld equipment to be used by traffic law officials
2. *Remote Server:* The remote server R comprises a Graphics Processing Unit (GPU) hosted on the cloud and is used for training, processing, and drawing inferences on the transmitted image I . The remote server is connected to the User Node U under an IoT setup using the Message Queueing Telemetry Transport (MQTT) protocol. The user node U publishes the data to the remote server subscriber. By leveraging the capabilities of the remote server, resource-intensive tasks are offloaded from the microcontroller. Within the server, the image I is processed to make it suitable for character recognition. The processed image is fed to a Deep Neural Network (DNN) deployed on



Fig. 2. t-stochastic Neighbour Embedding Visualization of the Synthetic Dataset used for fine-tuning the pre-trained model

the server, which yields the final inference for image I . This inference is then relayed back to the user node.

3. *Pretrained Model:* A pre-trained CNN model is used as the base of the DNN model by the remote server R . To prepare the model, we leverage transfer learning for license plate recognition by creating a synthetic dataset consisting of cropped images of license plate characters ranging from 0–9 and A–Z. Keeping the rest of the layers frozen, the output layer of the DNN model is trained using the synthetic data. Before the image I is passed onto the model for prediction, it is processed and the characters of the license plate are segmented. The model treats each segmented character as a separate image and makes the predictions. These predictions are concatenated to obtain the final prediction for the complete license plate. The remote server communicates the final prediction back to the user node. The model’s deep and complex network structure allows it to learn a wide range of visual patterns, making it a suitable candidate for license plate identification. The diversity of the training dataset in terms of character textures, font styles, and orientations ensures the model’s robustness in the prediction of license plates captured in real-life scenarios.
4. *Synthetic Dataset:* Due to the limited size as well as the low diversity of existing datasets for license plate recognition, a synthetic dataset was created by isolating characters from images of the AOLP Dataset [16]. This newly generated dataset consisted of 36 distinct classes - alphabets from A to Z and numbers from 0 to 9. Each image of a character underwent identical processing procedures as their counterparts in the original images, ensuring uniformity and consistency. The processing steps included grayscale conversion, adaptive thresholding for contrast enhancement, contour detection, and segmentation to obtain each individual character of the license plate for prediction, as fur-

ther elaborated in the later sections. To provide an intuitive understanding of this dataset’s class distribution, t-distributed Stochastic Neighbor Embedding (t-SNE) visualization was used as shown in Fig 2, with the colors indicating different classes. By utilizing this synthetic dataset, which focuses on characters extracted from license plates, the model gains improved adaptability to variations encountered in real-world license plate images, thereby enhancing its performance across diverse datasets.

3.1 Preprocessing Steps for the Captured Image

The image captured by the microcontroller is rife with noise and is not suitable in its raw form for feeding it into the model for LP Character Recognition. Consequently, the captured image is subjected to a series of processing steps to prepare it for feeding into the model (Fig. 3).

The preprocessing commences by converting the raw image (Fig. 3a) captured by the microcontroller into a grayscale image (Fig. 3b). This is followed by applying top-hat and blackhat morphological transformations over the resulting image to increase its contrast. Adaptive thresholding is further performed over the image to convert it to a binary image (Fig. 3c). This helps in segmenting the characters from the background of the license plate. After thresholding, contour detection is used to identify continuous regions of white pixels in the binary image (Fig. 3d). These contours represent regions of interest in the image, i.e. characters of the license. Among the detected contours, potential license plate regions are filtered out using criteria such as height and width (Fig. 3e). Within these filtered regions, individual license plate characters are segmented by isolating each character. The segmented characters are then resized to the dimensions of the training data of the classifications model as represented in Fig. 3f. The model is used to predict the characters within these segmented regions. The individual character-wise predictions are aggregated to represent the prediction for the complete license plate (Fig. 3g).

4 Experiments and Results

In order to evaluate the proposed approach’s real-time performance, multiple experiments were performed on various datasets. The experiments help in analyzing the robustness of the system in handling various lighting and weather conditions and differences in the types of license plates.

For the proposed approach, an ESP32-CAM was used as the microcontroller device. It is a small, portable camera module based on the ESP32-S chip. It hosts an OV2640 camera, multiple GPIO pins, and a microSD card slot. It also has a low form factor with a 3.3 V or a 5 V voltage rating that allows it to run on a battery. The image captured on the ESP32-CAM was sent via the *MQTT* [12] protocol using the *Mosquitto* broker to a cloud-based *Nvidia* T4 GPU system acting as the remote server. The *Keras* library [13], the TensorFlow [14] framework, and the *Jupyter* Notebook [15] were used to realize the DNNs used in the experimentation.

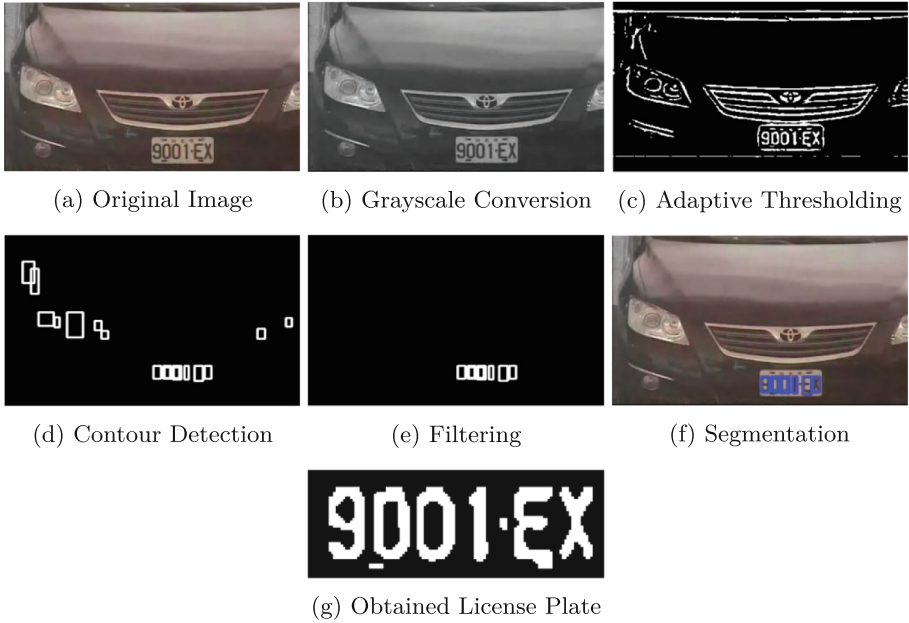


Fig. 3. Preprocessing steps for the image captured by the ESP32CAM

4.1 Performance Evaluation

In this section, we have evaluated the performance of the system using two different approaches, each utilizing a distinct model. In the first approach, a custom Convolutional Neural Network (CNN), as shown in Fig. 4, initially trained on the Street View House Numbers (SVHN) dataset, was used as the base model. The SVHN dataset was chosen due to its relevance in digit recognition tasks, which are crucial for LP Character Recognition; it provides a diverse set of real-world digit images, allowing the model to develop robust features that are transferable to our task. The second approach used a pre-trained model trained on a larger dataset, as the base model. For both these cases, the output layers of the models were fine-tuned using the synthetic dataset.

To choose an appropriate pre-trained model for LP character recognition for the second approach, we tested various standard pre-trained architectures by fine-tuning them on our synthetic dataset and analyzing their performance. All the models were trained for 30 epochs with a fixed learning rate of 0.01. The results are shown in Table 1.

Among the tested models, InceptionResNetV2 consistently achieved the highest accuracy, making it the most suitable candidate for further experiments. However, given that other pre-trained models also performed well, they also could be expected to achieve similar performances in future experiments.

In order to test our approaches outlined earlier, two different datasets were used. The first dataset comprised images from the AOLP Dataset, distinct from

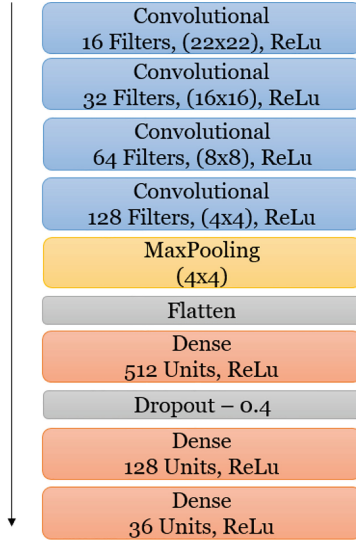


Fig. 4. Network Architecture

Table 1. Performance of various pre-trained models on the synthetic dataset

Model	Accuracy
InceptionResNetV2	99.71%
ResNet50	99.57%
InceptionV3	99.28%
VGG19	98.43%

the images used for creating the synthetic dataset while the second dataset comprised 60 images captured using an ESP32 CAM Module. Further, to address stochastic variations, each of these experiments was carried out 5 times. This yielded a total of 20 experiments for both the approaches.

AOLP Dataset. The AOLP dataset was used as the test data for analyzing the performance of the approach in each of the experiments.

The performance of the approach was evaluated by using two accuracy-based metrics detailed as follows:

1. *Metric 1:* This metric was obtained by dividing the summation of correctly classified characters across all the license plates by the total number of characters across all the license plates.

$$\text{Metric 1} = \frac{\sum_{i=1}^N C_i}{\sum_{i=1}^N T_i} \tag{1}$$

where C_i is the number of correctly classified characters in the i -th license plate, T_i is the total number of characters in the i -th license plate, and N is the total number of license plates.

- 2. *Metric 2*: This metric was obtained as the percentage of accurately classified license plates, where a license plate prediction is regarded to be accurate only if all the corresponding characters are correctly classified.

$$\text{Metric 2} = \left(\frac{\sum_{i=1}^N L_i}{N} \right) \times 100 \tag{2}$$

where L_i is 1 if the i -th license plate is accurately classified (i.e., all characters are correct), and 0 otherwise.

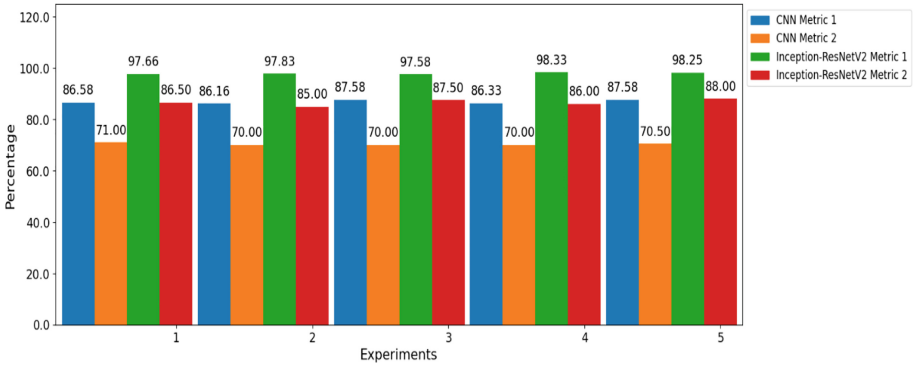


Fig. 5. Comparison of Inception-ResNet-v2 and Custom CNN on the AOLP Dataset

As observed through Fig. 5, the performance of the Inception-ResNet-v2 model was much better than the custom CNN with both metrics. This could be attributed to the greater diversity of images used for the initial training of the Inception ResNet-v2 model. The Inception-ResNet-v2 model was further tested on the real-time images captured with the ESP32 CAM.

ESP32 CAM Dataset. The ESP32 CAM dataset consisted of license plates that had a variable number of characters. To accommodate this variability, along with *Metric 1* and *Metric 2*, an additional metric was introduced as *Metric 3*. This new metric was calculated as the average of correctly classified characters per license plate and was used to assess how well the model performed character recognition on each license plate, as shown in Table 2.

$$\text{Metric 3} = \frac{1}{N} \sum_{i=1}^N \left(\frac{C_i}{T_i} \right) \quad (3)$$

where C_i is the number of correctly classified characters in the i -th license plate, T_i is the total number of characters in the i -th license plate, and N is the total number of license plates.

Table 2. Evaluation of the Inception ResNet model on the ESP32 CAM Dataset, across our three custom defined metrics

Expt.	Metric 1	Metric 2	Metric 3
1	94.72	85.00	92.27
2	94.44	85.00	92.00
3	95.27	86.67	93.33
4	95.00	85.00	93.00
5	94.44	85.00	93.33

It was observed that the performance was slightly worse on the images captured by the ESP32 CAM. This could be attributed to greater variations in the images including a variable number of characters across license plates, noisy images due to real-time capturing as well as lower resolution of the images. It was also seen that the values of *Metric 3* were lower than that of *Metric 1*. This is because the number of characters on each license plate is taken into consideration in the latter case. If a plate has a smaller number of characters, even a single misclassified character could have a substantial impact on the average.

The performance of the model on the real-time images and its ability to handle diverse weather and lighting conditions showcased its suitability for practical situations.

Comparison with Other Works. To further evaluate the strength of our approach, we compare it with their works by testing on benchmark license plate recognition datasets: CCPD Dataset and the PKUData.

1. CCPD (Chinese City Parking Dataset): This dataset was introduced Xu et al. [17] and it is one of the largest publicly available LP datasets with over 250,000 unique images. We test our approach on images from the Base (common case), Rotate (rotated images), Weather (images with rain, snow or fog) and challenge subsets.
2. PKUData: It was introduced in the work done by Yuan et al. [45]. It includes images of vehicles in diverse scenarios such as images taken from city roads, highways, nighttime and daytime. Similar to CCPD, we experiment on various subsets of this data to evaluate our approach's ability to generalize to varying conditions.

Table 3. Comparison of various methods on the different subsets of PKUData using LP Detection accuracy and inference time in milliseconds

Approach	AP	Base	Rotate	Weather	Inference Time (ms)
Ren et al. [43]	92.8	97.2	82.9	85.5	57.6
Wang et al. [48]	96.6	98.9	91.9	95.4	18.5
Luo et al. [49]	98.3	99.5	98.1	97.6	18.2
Zherzdev et al. [40]	93.0	97.8	79.4	92.0	7.5
Liu et al. [46]	95.2	98.3	88.4	87.3	25.6
Ke et al. [33]	99.8	99.9	99.9	99.6	10.2
Ours	97.5	98.3	97.7	97.3	5.4

As our approach is only capable of handling alphanumeric characters in the Latin Script, we do not consider the Chinese characters whilst evaluating on CCPD and PKUData. The results are presented in Table 3 and Table 4.

Table 4. Comparison of various methods on the different subsets of CCPD Dataset using LP Detection accuracy and inference time in milliseconds

Approach	Base	Rotate	Weather	Challenge	Inference Time (ms)
Ren et al. [43]	97.2	82.9	85.5	76.3	57.6
Li et al. [44]	97.8	87.9	86.8	81.2	31.0
Xu et al. [17]	98.5	94.7	84.1	92.8	11.7
Liu et al. [46]	99.1	95.6	83.4	93.1	24.6
Zhang et al. [47]	99.8	98.1	98.6	89.7	24.9
Wang et al. [39]	99.9	99.9	99.1	94.8	11.7
Ours	97.2	97.6	94.2	90.3	6.2

Despite our method not outperforming SOTA methods, it remains highly effective due to its lightweight nature and low inference time. These characteristics allow our approach to achieve comparable results to more complex approaches while being faster and more efficient, making it a strong contender against existing methods, especially for real-time usage.

4.2 Analysis of the Preprocessing Steps

The preprocessing steps outlined earlier are vital for the detection of the license plates, and subsequently the character recognition process. Hence, we analyze the performance of these methods in this section.

In the preprocessing experiments for LP detection on the AOLP data, key hyperparameters include the structuring element size (3×3) for morphological

operations, and specific contour filtering criteria such as a minimum area of 80, minimum width of 2 pixels, minimum height of 8 pixels, and an aspect ratio between 0.25 and 1.0. These parameters help in accurately isolating potential character regions, ensuring that only relevant contours are considered for further processing. These values were empirically chosen by qualitatively analyzing the results at each step.

The performance of these steps is calculated for both the AOLP Dataset and the ESP32 CAM Dataset and is outlined in Table 5 using the following measures:

1. Percentage of License Plates Correctly Identified in step Fig. 3e, calculated as the total percentage of license plates that were identified in their entirety with every character present.
2. Percentage of Characters correctly identified during segmentation in Fig. 3f, given as the percentage of segments that correspond to actual license plate characters.

Table 5. Evaluation of the image preprocessing for the AOLP and the ESP32 CAM Dataset

Experiment	Percentage of Correctly Identified License Plates	Percentage of Correctly Identified License Plate Characters
ESP32 CAM Dataset	95.0	96.4
AOLP	95.5	96.8

The reported figures in Table 5 were obtained with fixed threshold values for the height and width used to filter potential license plate regions for each of the datasets. By maintaining consistent threshold values across all images during the analysis, the preprocessing steps applied the same filtering criteria uniformly. The primary advantage of using this approach is its computational efficiency, particularly when compared to more resource-intensive techniques involving Deep Learning techniques. Unlike these methods which require significant computational resources and processing time, the approach utilized here is lightweight and efficient. However, this efficiency comes with a trade-off - the reliance on manually set threshold values for filtering out license plate regions. Practical images display diverse characteristics in license plate sizes, making it challenging to establish a universal threshold applicable to all images.

Conclusion

The work demonstrated in this paper introduces a real-time Automated License Plate Recognition (ALPR) system tailored specifically for resource-constrained scenarios. By combining microcontrollers, IoT, Deep Neural Networks, and

Cloud Computing, the proposed approach is able to show promising results in terms of accuracy and deployability. The experiments involving the Inception-ResNet-v2 model fine-tuned on a synthetic dataset, showcase robust performance across multiple diverse datasets, including real-time images captured by an ESP32CAM. The system's capability to detect license plates in real time, as well as its portability and maneuverability, highlight its practical viability. This is further enhanced by the ability to handle variable lighting, weather conditions, and license plate fonts and configurations. The integration of Transfer Learning and Cloud Computing addresses computational constraints, thereby enabling deployment in portable devices like wearables for effective traffic law enforcement. Future works could include the implementation of onboard learning mechanisms to enhance the system's adaptability and responsiveness. Onboard learning would allow the system to improve its performance continuously by adapting to new environments, license plate variations, and emerging patterns. The preprocessing steps could also be enhanced by adopting a more adaptive approach that allows for fine-tuning the threshold parameters on a per-image basis. By dynamically adjusting the thresholds to suit the specific characteristics of each image, the preprocessing algorithm could better accommodate the variability inherent in real-time images, providing improved performance in license plate recognition.

References

1. World Health Organization. Global Status Report on Road Safety (2023). <https://www.who.int/teams/social-determinants-of-health/safety-and-mobility/global-status-report-on-road-safety-2023>
2. Shashirangana, J., Padmasiri, H., Meedeniya, D., Perera, C.: Automated license plate recognition: a survey on methods and techniques. *IEEE Access*. **9**, 11203–11225 (2020). <https://doi.org/10.1109/ACCESS.2020.3047929>
3. Shafi, I., Hussain, I., Ahmad, J., et al.: License plate identification and recognition in a non-standard environment using neural pattern matching. *Complex Intell. Syst.* **8**, 3627–3639 (2022). <https://doi.org/10.1007/s40747-021-00419-5>
4. Lubna Mufti, N., Shah, S.A.A.: Automatic number plate recognition: a detailed survey of relevant algorithms. *Sensors*. **21**(9), 3028 (2021). <https://doi.org/10.3390/s21093028>
5. Kanteti, D., Srikar, D.V.S., Ramesh, T.K.: Intelligent smart parking algorithm. In: 2017 International Conference on Smart Technologies for Smart Nation (Smart-TechCon), pp. 1018–1022. *IEEE* (2017)
6. Wang, W., Tu, J.: Research on license plate recognition algorithms based on deep learning in complex environment. *IEEE Access*. 1–1 (2020). <https://doi.org/10.1109/ACCESS.2020.2994287>
7. Padmasiri, H., Shashirangana, J., Meedeniya, D., Rana, O., Perera, C.: Automated license plate recognition for resource-constrained environments. *Sensors* **22**, 1434 (2022). <https://doi.org/10.3390/s22041434>
8. Zhang, L., Wang, P., Li, H., Li, Z., Shen, C., Zhang, Y.: A robust attentional framework for license plate recognition in the wild. *IEEE Trans. Intell. Transp. Syst.* **22**(11), 6967–6976 (2021). <https://doi.org/10.1109/TITS.2020.3000072>

9. Pustokhina, I.V., et al.: Automatic vehicle license plate recognition using optimal k-means with convolutional neural network for intelligent transportation systems. *IEEE Access* **8**, 92907–92917 (2020). <https://doi.org/10.1109/ACCESS.2020.2993008>
10. Björklund, T., Fiandrotti, A., Annarumma, M., Francini, G., Magli, E.: Robust license plate recognition using neural networks trained on synthetic images. *Pattern Recogn.* **93**, 134–146 (2019). <https://doi.org/10.1016/j.patcog.2019.04.007>
11. Shobayo, O., Olajube, A., Ohere, N., Odusami, M., Okoyeigbo, O.: Development of smart plate number recognition system for fast cars with web application. *Appl. Comput. Intell. Soft Comput.* **2020**, 8535861 (2020). <https://doi.org/10.1155/2020/8535861>
12. Mishra, B., Kertesz, A.: The use of MQTT in M2M and IoT systems: a survey. *IEEE Access* **8**, 201071–201086 (2020). <https://doi.org/10.1109/ACCESS.2020.3035849>
13. Chollet, F., et al.: Keras. GitHub (2015). <https://github.com/fchollet/keras>
14. Mart'in, A., et al.: Tensorflow: a system for large-scale machine learning. In: 12th \$USENIX\$ Symposium on Operating Systems Design and Implementation (\$OSDI\$ 16), pp. 265–283 (2016)
15. Kluyver, T., et al.: Jupyter Notebooks - a publishing format for reproducible computational workflows. In: Loizides, F., Schmidt, B. (eds.) Positioning and Power in Academic Publishing: Players, Agents and Agendas, pp. 87–90 (2016)
16. Hsu, G.-S., Chen, J.-C., Chung, Y.-Z.: Application-oriented license plate recognition. *IEEE Trans. Veh. Technol.* **62**(2), 552–561 (2013). <https://doi.org/10.1109/TVT.2012.2226218>
17. Xu, Z., et al.: Towards end-to-end license plate detection and recognition: a large dataset and baseline. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11217, pp. 261–277. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01261-8_16
18. Hsieh, J.W., Yu, S.H., Chen, Y.S.: Morphology-based license plate detection from complex scenes. In: Proceedings of the 16th International Conference on Pattern Recognition, vol. 3, pp. 176–179. IEEE (2002)
19. Wu, H.H.P., Chen, H.H., Wu, R.J., Shen, D.F.: License plate extraction in low resolution video. In: 18th International Conference on Pattern Recognition, ICPR 2006, vol. 1, pp. 824–827. IEEE (2006)
20. Sarfraz, M., Ahmed, M.J., Ghazi, S.A.: Saudi Arabian license plate recognition system. In: Proceedings of International Conference on Geometric Modeling Graph, pp. 36–41 (2003)
21. Luo, L., Sun, H., Zhou, W., Luo, L.: An efficient method of license plate location. In: Proceedings of 1st International Conference on Information Science Engineering, pp. 770–773 (2009)
22. Heo, G., Kim, M., Jung, I., Lee, D.-R., Oh, I.-S.: Extraction of car license plate regions using line grouping and edge density methods. In: Proceedings of International Symposium on Information Technology Convergence (ISITC), pp. 37–42 (2007)
23. Yohimori, S., Mitsukura, Y., Fukumi, M., Akamatsu, N., Pedrycz, N.: License plate detection system by using threshold function and improved template matching method. In: Proceedings of IEEE Annual Meeting Fuzzy Information Processing (NAFIPS), vol. 1, pp. 357–362 (2004)
24. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)

25. Jia, W., Zhang, H., He, X., Piccardi, M.: Mean shift for accurate license plate localization. In: Proceedings of IEEE Intelligent Transport System, pp. 566–571 (2005)
26. Yao, Z., Yi, W.: License plate detection based on multistage information fusion. *Inf. Fusion* **18**, 78–85 (2014)
27. Xu, H.-K., Yu, F.-H., Jiao, J.-H., Song, H.-S.: A new approach of the vehicle license plate location. In: Proceedings of 6th International Conference on Parallel Distributed Computing Application Technology (PDCAT), pp. 1055–1057 (2005)
28. Deb, K., Chae, H.-U., Jo, K.-H.: Vehicle license plate detection method based on sliding concentric windows and histogram. *J. Comput.* **4**(8), 1–7 (2009)
29. Anagnostopoulos, C.N.E., Anagnostopoulos, I.E., Loumos, V., Kayafas, E.: A license plate-recognition algorithm for intelligent transportation system applications. *IEEE Trans. Intell. Transp. Syst.* **7**(3), 377–392 (2006)
30. Xie, L., Ahmad, T., Jin, L., Liu, Y., Zhang, S.: A new CNN-based method for multi-directional car license plate detection. *IEEE Trans. Intell. Transp. Syst.* **19**(2), 507–517 (2018)
31. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788 (2016)
32. Sung, J.-Y., Yu, S.-B., S.-h. P. Korea.: Real-time automatic license plate recognition system using YOLOv4. 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia) Seoul, Korea (South), pp. 1–3 (2020). <https://doi.org/10.1109/ICCE-Asia49877.2020.9277050>
33. Ke, X., Zeng, G., Guo, W.: An ultra-fast automatic license plate recognition approach for unconstrained scenarios. *IEEE Trans. Intell. Transp. Syst.* **24**(5), 5172–5185 (2023). <https://doi.org/10.1109/TITS.2023.3237581>
34. Rahman, C.A., Badawy, W., Radmanesh, A.: A real time vehicle’s license plate recognition system. In: Proceedings of IEEE Conference Advance Video Signal Based Surveillance, pp. 163–166 (2003)
35. Hu, P., Zhao, Y., Yang, Z., Wang, J.: Recognition of gray character using Gabor filters. In: Proceedings of 5th International Conference on Information Fusion (FUSION), vol. 1, pp. 419–424 (2002)
36. Kim, K.K., Kim, K.I., Kim, J.B., Kim, H.J.: Learning-based approach for license plate recognition. In: Proceedings of Neural Network Signal Processing X, IEEE Signal Processing Soc. Workshop, vol. 2, pp. 614–623 (2000)
37. Llorens, D., Marzal, A., Palazón, V., Vilar, J.M.: Car license plates extraction and recognition based on connected components analysis and HMM decoding. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) IbPRIA 2005. LNCS, vol. 3522, pp. 571–578. Springer, Heidelberg (2005). https://doi.org/10.1007/11492429_69
38. Li, H., Chunhua, S.: Reading car license plates using deep convolutional neural networks and LSTMs. *arXiv preprint* arXiv:1601.05610 (2016)
39. Wang, Y., Bian, Z.-P., Zhou, Y., Chau, L.-P.: Rethinking and designing a high-performing automatic license plate recognition approach. *IEEE Trans. Intell. Transp. Syst.* **23**(7), 8868–8880 (2022)
40. Zherzdev, S., Gruzdev, A.: LPRNet: license plate recognition via deep neural networks (2018). [arXiv:1806.10447](https://arxiv.org/abs/1806.10447)
41. Ammar, A., Koubaa, A., Boulila, W., Benjdira, B., Alhabashi, Y.: A multi-stage deep-learning-based vehicle and license plate recognition system with real-time edge inference. *Sensors* **23**(4), 2120 (2023)

42. Abdellatif, M.M., Elshabasy, N.H., Elashmawy, A.E., AbdelRaheem, M.: A low cost IoT-based Arabic license plate recognition model for smart parking systems. *Ain Shams Eng. J.* **14**(6), 102178 (2023)
43. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: *Proceedings of Advance Neural Information Processing System*, pp. 91–99 (2015)
44. Li, H., Wang, P., Shen, C.: Toward end-to-end car license plate detection and recognition with deep neural networks. *IEEE Trans. Intell. Transp. Syst.* **20**(3), 1126–1136 (2019)
45. Yuan, Y.L., Zou, W.B., Zhao, Y., Wang, X., Hu, X.F., Komodakis, N.: A robust and efficient approach to license plate detection. *IEEE Trans. Image Process.* **26**(3), 1102–1114 (2016)
46. Liu, W. et al.: SSD: single shot MultiBox detector. In: *Proceedings of European Conference on Computer Vision*, pp. 21–37 (2016)
47. Zhang, L., Wang, P., Li, H., Li, Z., Shen, C., Zhang, Y.: A robust attentional framework for license plate recognition in the wild. *IEEE Trans. Intell. Transp. Syst.* (2020)
48. Wang, T., et al.: Decoupled attention network for text recognition. *Proc. AAAI Conf. Artif. Intell.* **34**(7), 12216–12224 (2020)
49. Luo, C., Jin, L., Sun, Z.: A multi-object rectified attention network for scene text recognition (2019). [arXiv:1901.03003](https://arxiv.org/abs/1901.03003)



MTIQA360: An Easily Trainable Multitasking Network for Blind Omnidirectional Image Quality Assessment

Qinghai Wang and Shiguang Liu^(✉)

College of Intelligence and Computing, Tianjin University, Tianjin, China
{qwqhai, lsg}@tju.edu.cn

Abstract. Omnidirectional images are widely used in various fields such as virtual reality (VR), augmented reality (AR), and panoramic photography. However, most existing reference-free (NR) omnidirectional image quality assessment methods provide a single quality metric and are unable to identify the types of distortion that may exist in omnidirectional. This prevents subsequent image restoration tasks from automatically selecting an appropriate restoration method based on the distortion type. Furthermore, these methods often require extensive training resources. To address these two issues, we propose an ordinal prompt that can assess the quality level of omnidirectional images and classify distortion types. To better extract the distortion information of images, we further propose a proportional viewport sampling method that adapts to human browsing patterns. We conduct extensive experiments on two mainstream datasets (CVIQD and OIQA) and compare our method with state-of-the-art methods in terms of correlation coefficient, accuracy, and generalization ability. Various experimental results show that when trained with few parameters, our method still outperforms existing methods and has better generalization ability to different datasets and distortion types. The models and code are available on GitHub at <https://github.com/w-qhai/MTIQA360>.

Keywords: Omnidirectional Image · Image Quality Assessment · Multitask

1 Introduction

Omnidirectional imaging captures the entire surrounding scene from a single viewpoint, offering viewers an immersive experience akin to being physically present. With the rapid advancement and widespread adoption of virtual reality technology, omnidirectional images have become a crucial medium in diverse fields such as tourism, education, and entertainment. However, due to their inherently spherical nature, these images typically require conversion into a 2D image format via equirectangular projections (ERP) or cubemap projections (CMP) for storage and transmission. This conversion process renders traditional 2D

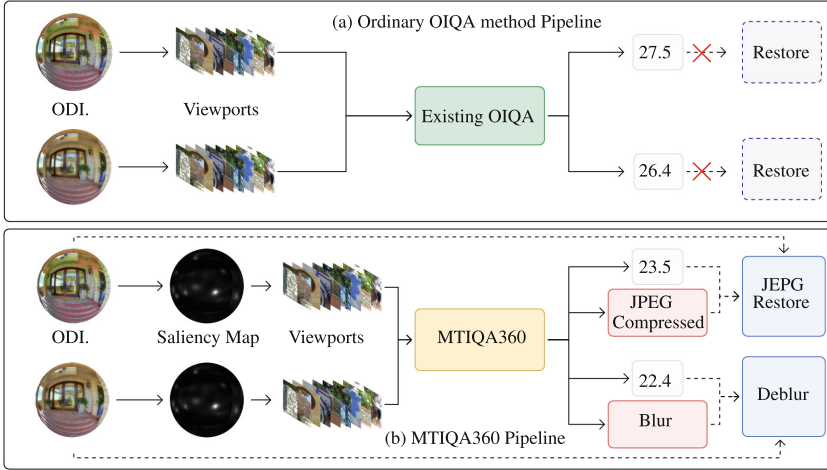


Fig. 1. (a) The previous method cannot provide sufficient differentiation information for two omnidirectional images (ODIs) with different distortion types and similar quality. This is inconducive to the continuation of subsequent tasks. (b) Our method, while giving the quality score, also suggests the distortion type, and can use different restoration strategies for specific tasks.

image quality assessment (IQA) metrics unsuitable for omnidirectional images [36]. Furthermore, omnidirectional images are prone to degradation or distortion during acquisition, compression, storage, and transmission, leading to diminished image quality. Low-quality omnidirectional images not only impair the viewing experience but may also induce viewer discomfort, such as dizziness and nausea [25, 37]. Consequently, there is an urgent need for an effective omnidirectional image quality assessment (OIQA) algorithm.

OIQA methods are categorized based on the availability of reference images into full-reference (FR), reduced-reference (RR), and no-reference (NR) methods. FR and RR methods necessitate the original or partial information of the reference images, which limits their practical application. NR methods, also known as blind omnidirectional image quality assessment (BOIQA) methods, do not require any reference information and rely solely on the distorted images for quality assessment. BOIQA methods are more suitable for real-world scenarios where reference images are difficult to obtain.

In the realm of BOIQA, existing methods [17, 19, 23, 24, 30] have primarily focused on predicting image quality, but they often overlook a critical aspect: the interplay between image content and distortion types. As illustrated in Fig. 1, existing methods miss out on the synergistic potential that arises from understanding both factors simultaneously. Consequently, their performance remains suboptimal, especially when dealing with critical low-level features like lines and textures. Additionally, the tendency to overscale images during viewport extraction exacerbates the loss of essential details.

When discussing the realm of traditional IQA, there are indeed success stories. These methods [20, 34] have demonstrated the positive impact of multitasking on quality evaluation. However, these methods often require multiple models to work together or significant training resources and time. Furthermore, their training processes do not fully exploit the powerful zero-shot capabilities inherent in visual language models.

Instead of fine-tuning the entire model, which would narrow its scope to a specialized image quality assessment tool, we propose an innovative approach. Our method involves an ordinal regression-based prompt specifically tailored for omnidirectional images. By doing so, we retain the generality of the original model while achieving excellent performance with minimal parameter training. Additionally, this approach allows parameters to be shared across tasks, ultimately reducing model deployment costs.

Overall, our contributions are as follows:

- We propose a multi-task network for blind omnidirectional image quality assessment (MTIQA360) for learning image quality features using image distortion pairs and outperforms current state-of-the-art methods
- We designed multitasking prompt based on ordinal regression, retaining the generalization ability of the original model while achieving better performance with minimal parameter training.
- We developed an equal-scale viewport extraction algorithm using saliency detection, proving that scaling images doesn’t improve their low-level details. It extracts appealing viewports from omnidirectional images at various resolutions, overcoming the excessive scaling and detail loss issues of past methods, and aligning viewport selection more closely with the human visual system.

2 Related Work

In this section, we will introduce previous BOIQA methods and other related works.

2.1 Quality Assessment for Omnidirectional Images

Traditional Methods. These methods are based on peak signal to noise ratio (PSNR) and structural similarity (SSIM), which are common metrics for 2D image quality assessment. They modify these metrics to account for the spherical projection and saliency effects of omnidirectional images. However, these methods do not consider the human perception and viewing behavior of omnidirectional images.

Deep Learning Based Methods. These methods use neural networks to learn features and predict quality of omnidirectional images. In the field of omnidirectional image quality assessment, the DeepVR-IQA [9] method evaluates the quality of each image block using positional and visual features, optimized through

adversarial learning guided by human perception. Methods like MC360IQ [17], VGCN [24], JN [30], and ST360IQ [19] select viewports from the image and score them using CNN [5] or ViT [3], with scores aggregated by a global model or RNN. The Assessor360 [23] method models the observers viewing process through recursive sampling of viewports, fusing distortion and semantic features with a distortion perception module and a multi-scale feature aggregation module to produce the image quality score.

Current approaches in omnidirectional image quality assessment predominantly utilize single-modal data, which inherently limits their capacity to encapsulate the multi-modal quality attributes of such images [35]. Furthermore, these methodologies frequently incur a loss of low-level detail due to the excessive scale of images during viewport extraction. As a result, these methods cannot relearn low-level information and features, such as distortion information. An enhanced BOIQA algorithm should, therefore, be capable of not only determining the quality score but also identifying the specific type of distortion present in omnidirectional images. This dual capability would significantly aid in the selection of the most suitable restoration algorithm tailored to the identified distortion type.

2.2 Image-Text Pretraining

Image-text pre-training exploits the natural correspondence between images and text. It first adopts contrastive learning to align image and text representations, and then applies self-supervised learning objectives such as masked region reconstruction, masked object prediction and word region alignment. Recently, several methods, such as CoCa [29], ALIGN [6] and CLIP [14], have introduced image-text backbone models trained on large-scale image-text datasets. These methods typically use billions of image-text pairs scraped from the web and achieve excellent performance on a variety of tasks, such as retrieval, classification, and subtitles. However, on the Internet, information about image quality and its text description is relatively scarce, which makes it difficult for the model to understand image quality through text. CLIP-IQA [20] and LIQE [34] use fine-tuning models to let the model learn image quality information., but this reduces the versatility of the original model. We believe that the model pre-trained on a large number of image and text pairs has learned enough image information, but lacks a suitable prompt to guide the model to establish the relationship between image and quality. Therefore, the development of a nuanced prompt that can effectively bridge the gap between image content and quality assessment is essential for leveraging the full potential of pre-trained image-text models in this field.

3 Proposed Method

This section introduces the entire process of the proposed method, including dataset preprocessing, prompts design, model training, and loss function.

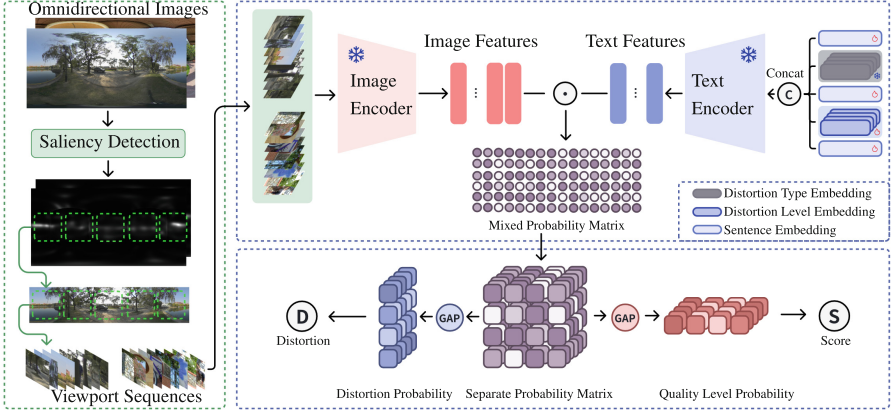


Fig. 2. Architecture of the proposed MTIQA360 for BOIQA. First, the image is input into the saliency detection model to obtain a saliency map, and the most attractive viewport is extracted based on the saliency map. The extracted viewport sequence and ordinary prompt are encoded separately, and then the cosine similarity is calculated to obtain the mixing probability matrix. Finally, the distortion type and quality score of each viewport sequence are calculated.

3.1 Viewport Sampling Strategy

Here we introduce the process of the viewport selection strategy, which aims to select viewports that can more effectively capture the viewers attention [27]. The purpose of the viewport selection strategy is to select a viewport that can more effectively capture the viewers attention based on the saliency information of the omnidirectional image. As shown in Fig. 2, we use the visual saliency detection algorithm [2] for omnidirectional images to obtain the saliency map. The specific process is as follows. We first input the omnidirectional image into the visual saliency detection model for omnidirectional images, obtain the saliency map as shown in Fig. 3,

$$I_s = SD(I), \quad (1)$$

where I represents the omnidirectional image, SD is the visual saliency detection algorithm for omnidirectional images, and I_s indicates the saliency map. Then extract the viewport according to the salient position in the I_s . In order to ensure the consistency of the content, we fixed the field of view and the aspect ratio of the viewport. Therefore, the viewport will be scaled by a certain proportion compared to the original image. The scaling factor λ is calculated as follows:

$$\lambda = \frac{fov \times W}{360 \times w}, \quad (2)$$

where fov represents the field of view (FOV), W is the original image width, and w stands for the viewport width. The new width and height of I_s are $\lambda \times W$, $\lambda \times H$, respectively. Based on the saliency map, we extract N viewports with



Fig. 3. The computed saliency map of an omnidirectional image.



Fig. 4. Some attractive viewpoints.

the highest saliency from high to low. The specific method is as follows: we first find the most salient point, convert the position of the point into longitude and latitude coordinates,

$$\begin{aligned} v &= -\frac{y}{\lambda \times H} \times 180 + 90, \\ u &= \frac{x}{\lambda \times W} \times 360 - 180, \end{aligned} \quad (3)$$

where (u, v) is the coordinate in I_s , (x, y) represents the position in I_s . We then extract the *fov* viewport centered on (u, v) and convert it to a 2D perspective to ensure that the viewport is consistent with the human eye. To highlight areas of high importance, areas of high attractiveness are sampled to multiple viewports. The above process is repeated N times to obtain a sequence containing N viewports, as shown in Fig. 4.

3.2 Ordinal Prompts

This section gives an overview of learnable ordinal prompts, to improve the performance of image quality assessment and distortion type recognition.

Human-designed prompts often have limitations [38,39]. Inspired by some works leveraging CLIP for number problems [10,11], we design ordinal prompts to improve the model performance. In the dataset we use, there are some distortion types $\mathcal{D} \in [\text{“jpeg compression”}, \text{“jpeg 2000 compression”}, \text{“hevc compression”}, \text{“avc compression”}, \text{“noise”}, \text{“blur”}, \text{“others”}]$. We first classify images into 5 quality levels $\mathcal{Q} \in [\text{“Unacceptable”}, \text{“Defective”}, \text{“Normal”}, \text{“Professional”}, \text{“Exceptional”}]$. We map the words of each quality level to a vector, and obtain L quality levels by interpolation. There are a total of $T_p = |\mathcal{D}| \times L$ prompts, where $|\cdot|$ denotes the number of elements in the set. Our methods multitasking capabilities come from the prompts we designed. We set two task goals in the prompt, which are image quality assessment and distortion type recognition.

3.3 Quality Regression and Loss Function

Through the viewport sampling and ordinal prompts, we obtain N viewports for each omnidirectional image and T_p prompts for all of omnidirectional image.

A viewport is a region that simulates the human FOV on the omnidirectional image. A prompt is a short text that describes the image quality level and distortion type of a viewport.

During training, we have a batch of viewports $X = \{x_i \mid 0 \leq i < B, x_i \in \mathbb{R}^{N \times w \times h \times C}\}$, where B is the batch size and C is the number of channels of the viewport. We use ViT to encode each viewport and a text GPT-2 [15] to encode each prompt. We then obtain a batch of N viewport features $F = \{f_i \mid 0 \leq i < B \times N, f_i \in \mathbb{R}^{dim}\}$ and T_p text features $P = \{p_i \mid 0 \leq i < T_p, p_i \in \mathbb{R}^{dim}\}$ for each prompt. We compute the similarity between each viewport feature and each text feature,

$$M = FP^\top, \quad (4)$$

where M is the mixed probability matrix, we convert $M_{(B \times N \times T_p)}$ to $M_{B \times N \times |\mathcal{D}| \times L}$. We take the average along the N dimension to obtain the separated probability matrix, and then take the average along the $|\mathcal{D}|$ and L dimensions to calculate the quality level and distortion type probabilities of the viewport sequence. We assign a weight $W = [1, 2, 3, \dots, L]$ to each prompts, indicating its contribution to the final score, and use the following formula to calculate the final score of the image:

$$S = \sum_{i=1}^L \mathcal{W}_i \mathcal{Q}_i, \quad (5)$$

where S is the final score, \mathcal{Q}_i is the probability of the i -th quality level. For the quality regression task, we use MAE loss as L_1 , and for the distortion type recognition task, we use cross entropy as L_2 . The overall loss $L = L_1 + L_2$.

4 Experiments and Results

In this section, we present our experimental settings and results. Then we compare MTIQA360 with the state-of-the-art BOIQA methods on two mainstream datasets: CVIQD and OIQA. We also analyze the performance of MTIQA360 on different distortion types. Finally, we conduct a series of ablation studies to validate the effectiveness of our approach.

4.1 Datasets and Evaluation Metrics

We use two publicly available datasets for omnidirectional image quality assessment: CVIQD [16] and OIQA [4]. CVIQD contains 16 reference images, each with 3 distortion types: JPEG compression, HEVC/H.265 compression, and AVC/H.264 compression. There are 10 distortion levels for each distortion type, for a total of 544 images. The resolution of each image is 4096×2048 . OIQA contains 16 reference images, each with 4 distortion types: JPEG compression,

JPEG 2000 compression, Gaussian noise, and Gaussian blur. There are 5 distortion levels for each distortion type, for a total of 336 images. The resolution of each image is 11332×5666 . Both datasets provide subjective ratings for each image, ranging from 0 to 100, where higher ratings indicate better quality.

We use three widely used evaluation metrics for BOIQA: Pearson’s correlation coefficient (PLCC), Spearman’s rank correlation coefficient (SRCC), and root mean square error (RMSE). PLCC measures the linear correlation between the prediction quality score and the subjective rating, while SRCC measures the monotonicity between the prediction quality score and the subjective rating, and RMSE represents the difference between the predicted value and the observed value. Higher values of the two correlation coefficient indicators indicate better performance. The lower the RMSE, the better.

4.2 Implementation Details

We implement MTIQA360 using PyTorch. We encode each viewport using the ViT-Base and each prompt using the GPT-2. We freeze the image encoder parameters and the text encoder parameters. We extract 20 viewports from each image, and the viewport size is 224×224 . We set the *fov* view to 45° , which is a common value for human vision. We use the Adam optimizer with a learning rate of $1e-2$ and a cosine annealing scheduler to train our model. We train for 300 epochs with a batch size of 16. We use a 80/20 split for training and testing, and report the average results of 10 random splits. We conduct our experiments on a Nvidia GeForce A100 GPU and less than 4 GB of memory is required for training.

4.3 Comparison with State-of-the-Art Methods

We compare MTIQA360 with 22 state-of-the-art quality assessment methods, which consist of 8 FR methods and 14 NR methods. The FR methods include PSNR, SSIM [21], MS-SSIM [22], WS-PSNR [18], WS-SSIM [40], VIF [31], DISTS [1], and LPIPS [32]. The BIQA methods include NIQE [13], BRISQUE [12], PaQ-2-PiQ [28], MUSIQ [8], MANIQA [26], CLIP-IQA [20], LIQE [34], SSP-BOIQA [33], MP-BOIQA [7], MC360IQ [17], VGCN [24], ST360IQ [19], and Assessor360 [23]. Note that some of the BIQA methods, such as NIQE, BRISQUE, PaQ-2-PiQ, MANIQA, MUSIQ, CLIP-IQA and LIQE are 2D image-based quality assessment metrics.

The quantitative results are shown in Table 1, which indicates that MTIQA360 has a significant improvement in performance and greatly outperforms the comparative BOIQA methods. On the CVIQD dataset, MTIQA360 achieves a PLCC of 0.9822 and a SRCC of 0.9781, which are about 1.1% and 1.9% higher than the second best method Assessor360, respectively. On the OIQA dataset, MTIQA360 achieves a PLCC of 0.9852 and a SRCC of 0.9868, which are 1.3% and 0.8% higher than the second best method Assessor360, respectively.

Table 1. Quantitative comparison of the state-of-the-art methods and proposed MTIQA360. The best are shown in **red**, and the second best (except ours) are **blue**. MTIQA360+ represents prompt and image encoder training at the same time. MTIQA360- indicates training using a viewport that retains more distortion information. Methods with * indicate that the data comes from Assessor360 [23] with the same experimental setup. “#Params” represents trainable parameters.

Methods	CVIQD			OIQA			
	PLCC↑	SRCC↑	RMSE↓	PLCC↑	SRCC↑	RMSE↓	#Params↓
PSNR*	0.8425	0.8015	-	0.3893	0.3929	-	-
SSIM*	0.7273	0.6737	-	0.2307	0.3402	-	-
MS-SSIM*	0.9272	0.9218	-	0.5084	0.575	-	-
WS-PSNR*	0.8410	0.8039	-	0.3678	0.3829	-	-
WS-SSIM*	0.7672	0.8632	-	0.3537	0.6020	-	-
VIF*	0.9370	0.9502	-	0.4158	0.4284	-	-
DISTS*	0.8613	0.8771	-	0.5809	0.574	-	-
LPIPS*	0.8242	0.8236	-	0.4292	0.5844	-	-
NIQE*	0.8392	0.9337	-	0.785	0.8539	-	-
BRISQUE*	0.8199	0.8269	-	0.8206	0.8213	-	-
PaQ-2-PiQ*	0.6500	0.7376	-	0.2102	0.1667	-	-
MANIQA*	0.6142	0.6013	-	0.4171	0.4555	-	-
MUSIQ*	0.3678	0.3483	-	0.3087	0.3216	-	-
CLIP-IQA*	0.4347	0.4884	-	0.2531	0.2330	-	-
LIQE*	0.8086	0.8594	-	0.7419	0.7634	-	-
SSP-BOIQA*	0.9077	0.8614	-	0.8600	0.8650	-	-
MP-BOIQA*	0.9390	0.9235	-	0.9206	0.9066	-	-
MC360IQ*	0.8240	0.8271	-	0.8925	0.9071	-	-
VGCN*	0.9651	0.9639	-	0.9584	0.9515	-	-
AHGCN*	0.9658	0.9617	-	0.9682	0.9647	-	-
ST360IQ	0.9698	0.9696	4.2719	0.9254	0.9235	5.5729	7.86×10^7
Assessor360	0.9713	0.9591	-	0.9724	0.9790	-	8.93×10^7
MTIQA360-	0.9817	0.9749	2.8727	0.9825	0.9824	2.8065	8.79×10^7
MTIQA360+	0.9824	0.9798	2.8310	0.9847	0.9840	2.6453	8.79×10^7
MTIQA360(Ours)	0.9822	0.9781	2.8651	0.9852	0.9868	2.5897	1.33×10^7

This demonstrates the effectiveness of our method for learning the quality representation of omnidirectional images by combining the complementary information between image levels and distortion types. Table 2 further demonstrate the performance of our method for different distortion types and degrees. We separate each dataset based on distortion type. PLCC and SRCC were calculated separately for each subset.

Table 2. Performance of our method on various distortion types in CVIQD and OIQA.

	CVIQD			OIQA			
	JPEG	HEVC	AVC	JPEG	J2K	Blur	Noise
PLCC \uparrow	0.9878	0.9809	0.9760	0.9860	0.9735	0.9906	0.9808
SRCC \uparrow	0.9811	0.9768	0.9726	0.9560	0.9560	0.9835	0.9708
RMSE \downarrow	3.0710	2.3892	2.8664	2.9262	3.0163	2.0391	2.4787
ACC \uparrow	1.0000	0.9722	0.9677	1.0000	1.0000	1.0000	1.0000



This is a SCHOOL image with AVC compression. Its quality is 36.84.



This is a PLAYGROUND image with AVC compression. Its quality is 45.45.



This is a SCHOOL image with AVC compression. Its quality is 63.13.



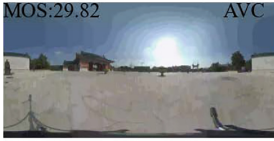
This is a PLAYGROUND image with HEVC compression. Its quality is 60.78.



This is a PLAYGROUND image with JPEG compression. Its quality is 71.28.



This is a PLAYGROUND image with HEVC compression. Its quality is 38.23.



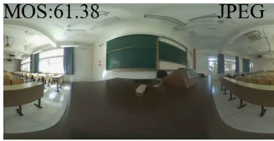
This is a LANDSCAPE image with AVC compression. Its quality is 31.67.



This is a SCHOOL image with JPEG compression. Its quality is 34.65.



This is a PLAYGROUND image with JPEG compression. Its quality is 62.96.



This is a CLASSROOM image with JPEG compression. Its quality is 60.23.



This is a PLAYGROUND image with AVC compression. Its quality is 63.24.



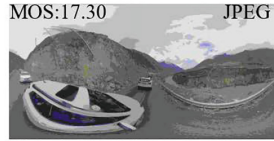
This is a SQUARE image with JPEG compression. Its quality is 60.03.



This is a HIGHWAY image with AVC compression. Its quality is 39.83.



This is a SMALL TOWN image with HEVC compression. Its quality is 61.63.



This is a MOUNTAIN ROAD image with JPEG compression. Its quality is 16.82.

Fig. 5. Our method targets quality score prediction for different scenarios, distortion types, and distortion levels. The upper left gives the true score of the image, and the upper right shows the distortion type of the image.

Figure 5 shows the quality prediction of our method in different scenarios, different distortion types, and different distortion levels. Affected by omnidirectional distortion, the scene prediction accuracy will deteriorate, but the prediction accuracy of distortion type and quality score is extremely high.

Table 3. Cross dataset validation.

		MC360IQ	VGCN	Assessor360	MTIQA360+	MTIQA360-	MTIQA360
Train on OIQA test on CVIQD	PLCC↑	0.5930	0.5929	0.8878	0.8763	0.8550	0.7825
	SRCC↑	0.5687	0.6932	0.8961	0.8948	0.8725	0.6523
	ACC↑	-	-	-	0.13	0.20	0.05
Train on CVIQD test on OIQA	PLCC↑	0.4295	0.2582	0.4030	0.8639	0.5955	0.6137
	SRCC↑	0.4189	0.2361	0.4613	0.8639	0.5513	0.6188
	ACC↑	-	-	-	0.24	0.24	0.24

4.4 Cross-Dataset Validation

In order to verify the generalization ability of the model, we conduct cross-validation on different data sets. And the results are compared with three state-of-the-art methods: MC360IQ, VGCN, Assessor360. The results are shown in Table 3. As can be seen from the data in Table 3, although the results of our method are slightly lower than other methods when only training a small number of parameters, they are still competitive. In addition, it can be seen from the data in the table that the generalization of the models trained on the CVIQD data set is generally poor. This is because OIQA mainly focuses on JPEG, JPEG2000, blur and noise distortion, each distortion type has only 5 distortion levels, while CVIQD mainly focuses on JPEG, AVC, HEVC distortion, each distortion type has 10 distortion levels. This gap allows models trained in CVIQD to learn relatively little distortion information and therefore have weak generalization capabilities. In addition, the performance of our method and ablation experiments exceeds other methods. This is because our method better utilizes the richer features of the CVIQD type distortion level. The accuracy of the distortion type can also support this. In CVIQD, the model trained on can accurately identify JPEG distortion from the 4 distortion types in the OIQA dataset, but the opposite effect is not good.

4.5 Ablation Study

Ordinal Prompts. To evaluate the effectiveness of prompts we proposed, we ablate our learnable prompts on the OIQA dataset. We first used CLIP to perform image reasoning without any modifications. We manually designed the prompts as “a photo with {distortion} artifacts, which is of {quality level} quality”, where quality level = [“abysma”, “terrible”, “poor”, “bad”, “acceptable”,

“good”, “great”, “excellent”, “outstanding”, “perfect”] correspond to the level 10. Compared with the results of 100 epoch of training in our original experiment, the results are shown in Table 4. This shows that our multi-task ordinal prompts are effective.

Table 4. Effectiveness of ordinal prompts on different dataset. **Bold** indicates the best results.

(a) OIQA					(b) CVIQD				
	PLCC↑	SRCC↑	RMSE↓	ACC↑		PLCC↑	SRCC↑	RMSE↓	ACC↑
handmade	0.4547	0.4453	46.7314	0.38	handmade	0.4185	0.4143	46.7509	0.09
ordinal	0.9762	0.9758	3.280	0.94	ordinal	0.9762	0.9708	3.3688	0.95

Train More Parameters. In cross-validation, we found that the model generalization ability was slightly lower than other methods. This is because there are relatively few image-text pairs on the Internet about image quality and distortion types, which may mean that the model has not been exposed to enough relevant data during the learning process to understand and improve image quality and distortion issues. To solve this problem, we tried to fine-tune the image encoder in the hope that the model would better understand the nuances of image quality.

During the fine-tuning process, we increase the number of parameters of the model so that the model can capture more features and patterns. As shown in Table 1, our method still outperforms other methods. Furthermore, as depicted in Table 3, the fine-tuned model demonstrates a notable leap in its generalization aptitude. This implies that the model not only excels in the training dataset but also upholds its robust performance on novel, unencountered data. Specifically, when the model is trained on the CVIQD dataset and evaluated against the OIQA dataset, it showcases superior performance, significantly outshining other methods.

Viewport Extraction Strategies. We trained MTIQA360+ on OIQA and tested it on CVIQD, achieving great generalization performance on scoring but bad performance on distortion type recognition, as shown in Table 3. We then ablated the viewport sampling strategy and extracted a 224×224 viewport from the original image, preserving the distortion information of the original image. Figure 6 shows the viewports sampled by the two methods, respectively. We can observe blocking artifacts due to JPEG compression in Fig. 6b, while in Fig. 6a most of the distortion information is lost since the scaling details are mosaiced. This explains why OIQA has poor generalization ability in distortion type recognition. This is unavoidable, as viewers see consistent FOV content when browsing omnidirectional image of any resolution. To ensure consistency between the

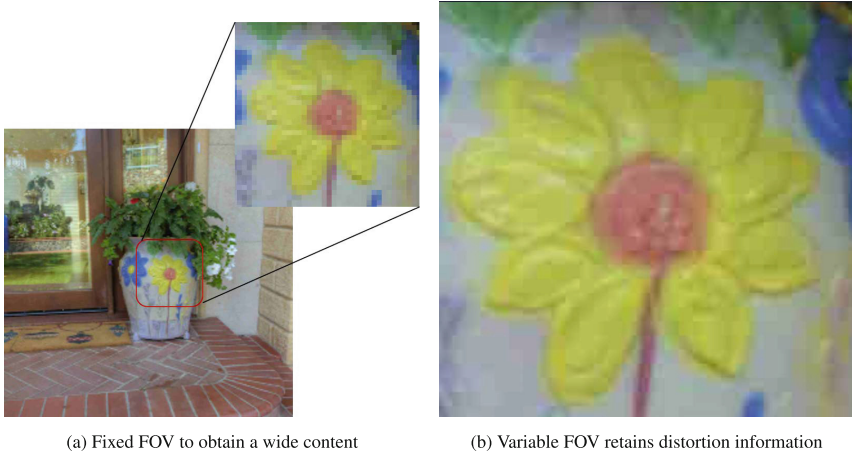


Fig. 6. Different viewport sampling strategies.

model and human eye input, we have to sacrifice some distortion information. We also trained on the original viewport and performed cross-validation. As shown in Table 3, MTIQA360- that the model’s accuracy in discriminating distortion types improved, but due to the inconsistent content of the FOV, PLCC and SRCC decreased significantly.

To enhance the validation of our viewport extraction method, we employed the viewport obtained through ST360IQ method as a training dataset. The outcomes, presented in Table 5, indicate a notable decline in several metrics, especially the accuracy of distortion type recognition. This reduction can be attributed to the methods process of resizing the image to 1024768 and subsequently scaling the extracted viewport, which significantly compromises the integrity of distortion information.

Table 5. Comparison of training results using different viewport extraction strategies on MTIQA360. Subscripts indicate viewport extraction strategies, **bold** indicates the best results.

Method	CVIQD				OIQA			
	PLCC \uparrow	SRCC \uparrow	RMSE \downarrow	ACC \uparrow	PLCC \uparrow	SRCC \uparrow	RMSE \downarrow	ACC \uparrow
MTIQA360 _{ST360IQ}	0.9780	0.9768	3.2261	0.8455	0.9661	0.9678	3.8454	0.7059
MTIQA360(Ours)	0.9822	0.9781	2.8651	0.9455	0.9852	0.9868	2.5897	0.9853

4.6 Discussion and Limitations

Although our method performs well in BOIQA, there are still some challenges that remain unsolved. First, regarding the dataset, we propose a new paradigm

for OIQA, viewing distortion type identification as an auxiliary task, which leads to higher requirements for the dataset, as the labels of the omnidirectional images not only need to include mean opinion score (MOS), but also distortion types. Second, omnidirectional images with slight distortions and high resolutions are still a difficulty for this problem. If we sample small viewports to preserve the distortion information, it is difficult to ensure the generalization ability of the model; if we sample large viewports to ensure the generalization ability, the distortion information will degrade into downsampling, which prevents the correct prediction of the image quality. Perhaps there will be more clever data preprocessing schemes in the future, which can guarantee both aspects of information. Finally, while the method demonstrates excellent performance in its current form, its effectiveness and adaptability are heavily dependent on the distortion types it has been trained to recognize. This reliance may limit its applicability in scenarios involving novel or uncharacterized distortions.

5 Conclusion and Future Work

In this paper, we have proposed a multi-task BOIQA method. Specifically, we designed a fixed viewport sampling scheme to extract regions that simulate human vision from omnidirectional images. Moreover, we designed learnable prompts for multi-task learning to fully exploit the prior knowledge of image-text pretraining model. Our method achieves great performance on two mainstream OIQA datasets and demonstrates the negative impact of scaling images on the distortion type identification of the model. Although the models complexity is increased, the training difficulty does not escalate because most parameters are locked. This approach reduces the resources required for training and avoids catastrophic forgetting caused by fine-tuning CLIP, thereby retaining its versatility.

Unfortunately, we have not yet found an optimal balance point between preserving large FOV and distortion information. However, we believe that our work can provide new insights for future BOIQA research. In the future, we will focus our work on how to extract sufficient distortion information while retaining large FOV, so that the model can adapt to images with different resolutions and correctly identify distortion types.

References

1. Ding, K., Ma, K., Wang, S., Simoncelli, E.P.: Image quality assessment: unifying structure and texture similarity. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **44**(5), 2567–2581 (2020)
2. Djilali, Y.A.D., Tliba, M., McGuinness, K., O’Connor, N.: ATSal: An attention based architecture for saliency prediction in 360 videos. *ArXiv, abs/2011.10600* (2020)
3. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: *International Conference on Learning Representations (ICLR)* (2021)

4. Duan, H., Zhai, G., Min, X., Zhu, Y., Fang, Y., Yang, X.: Perceptual quality assessment of omnidirectional images. In: IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5 (2018)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR), pp. 770–778 (2016)
6. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International Conference on Machine Learning, pp. 4904–4916. ICML (2021)
7. Jiang, H., Jiang, G., Yu, M., Luo, T., Xu, H.: Multi-angle projection based blind omnidirectional image quality assessment. IEEE Trans. Circ. Syst. Video Technol. (TCSVT) **32**(7), 4211–4223 (2021)
8. Ke, J., Wang, Q., Wang, Y., Milanfar, P., Yang, F.: MUSIQ: multi-scale image quality transformer. In: International Conference on Computer Vision (ICCV), pp. 5148–5157 (2021)
9. Kim, H.G., Lim, H.T., Ro, Y.M.: Deep virtual reality image quality assessment with human perception guider for omnidirectional image. IEEE Trans. Circ. Syst. Video Technol. (TCSVT) **30**(4), 917–928 (2019)
10. Li, W., Huang, X., Zhu, Z., Tang, Y., Li, X., Zhou, J., Lu, J.: OrdinalCLIP: learning rank prompts for language-guided ordinal regression. In: Conference on Neural Information Processing Systems (NeurIPS), vol. 35, pp. 35313–35325 (2022)
11. Liang, D., Xie, J., Zou, Z., Ye, X., Xu, W., Bai, X.: Crowdclip: unsupervised crowd counting via vision-language model. In: IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR), pp. 2893–2903 (2023)
12. Mittal, A., Moorthy, A.K., Bovik, A.C.: No-reference image quality assessment in the spatial domain. IEEE Trans. Image Process. (TIP) **21**(12), 4695–4708 (2012)
13. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a “completely blind” image quality analyzer. IEEE Signal Process. Lett. (SPL) **20**(3), 209–212 (2012)
14. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (ICML), pp. 8748–8763 (2021)
15. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
16. Sun, W., Gu, K., Zhai, G., Ma, S., Lin, W., Le Calle, P.: CVIQD: subjective quality evaluation of compressed virtual reality images. In: IEEE International Conference on Image Processing (ICIP), pp. 3450–3454 (2017)
17. Sun, W., et al.: MC360IQA: the multi-channel CNN for blind 360-degree image quality assessment. In: IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5 (2019)
18. Sun, Y., Lu, A., Yu, L.: Weighted-to-spherically-uniform quality evaluation for omnidirectional video. IEEE Signal Process. Lett. (SPL) **24**(9), 1408–1412 (2017)
19. Tofighi, N., Elfkir, M., Imamoglu, N., Ozcinar, C., Erdem, E., Erdem, A.: ST360IQ: no-reference omnidirectional image quality assessment with spherical vision transformers. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 4–9 (2023)
20. Wang, J., Chan, K.C., Loy, C.C.: Exploring CLIP for assessing the look and feel of images. In: AAAI Conference on Artificial Intelligence (AAAI), pp. 2555–2563 (2023)
21. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. (TIP) **13**(4), 600–612 (2004)

22. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: Asilomar Conference on Signals, Systems, and Computers (ACSSC), vol. 2, pp. 1398–1402. IEEE (2003)
23. Wu, T., et al.: Assessor360: multi-sequence network for blind omnidirectional image quality assessment. In: Conference on Neural Information Processing Systems (NeurIPS) (2023)
24. Xu, J., Zhou, W., Chen, Z.: Blind omnidirectional image quality assessment with viewport oriented graph convolutional networks. *IEEE Trans. Circ. Syst. Video Technol. (TCSVT)* **31**(5), 1724–1737 (2020)
25. Xu, M., Li, C., Chen, Z., Wang, Z., Guan, Z.: Assessing visual quality of omnidirectional videos. *IEEE Trans. Circ. Syst. Video Technol. (TCSVT)* **29**(12), 3516–3530 (2018)
26. Yang, S., et al.: MANIQA: multi-dimension attention network for no-reference image quality assessment. In: IEEE/CVF Computer Vision and Pattern Recognition Conference Workshop (CVPRW), pp. 1191–1200 (2022)
27. Yi, F., Chen, M., Sun, W., Min, X., Tian, Y., Zhai, G.: Attention based network for no-reference UGC video quality assessment. In: IEEE International Conference on Image Processing (ICIP), pp. 1414–1418 (2021)
28. Ying, Z., Niu, H., Gupta, P., Mahajan, D., Ghadiyaram, D., Bovik, A.: From patches to pictures (PaQ-2-PiQ): Mapping the perceptual space of picture quality. In: IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR), pp. 3575–3585 (2020)
29. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., Wu, Y.: Coca: contrastive captioners are image-text foundation models. [arXiv:2205.01917](https://arxiv.org/abs/2205.01917) (2022)
30. Zhang, C., Liu, S.: No-reference omnidirectional image quality assessment based on joint network. In: ACM International Conference on Multimedia (ACM MM), pp. 943–951 (2022)
31. Zhang, L., Shen, Y., Li, H.: VSI: a visual saliency-induced index for perceptual image quality assessment. *IEEE Trans. Image Process. (TIP)* **23**(10), 4270–4281 (2014)
32. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR), pp. 586–595 (2018)
33. Zhang, W., Ma, K., Yan, J., Deng, D., Wang, Z.: Blind image quality assessment using a deep bilinear convolutional neural network. *IEEE Trans. Circ. Syst. Video Technol. (TCSVT)* **30**(1), 36–47 (2020)
34. Zhang, W., Zhai, G., Wei, Y., Yang, X., Ma, K.: Blind image quality assessment via vision-language correspondence: a multitask learning perspective. In: IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR), pp. 14071–14081 (2023)
35. Zhang, Z., et al.: MM-PCQA: multi-modal learning for no-reference point cloud quality assessment. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, (IJCAI), pp. 1759–1767 (2023)
36. Zhang, Z., Sun, W., Min, X., Wang, T., Lu, W., Zhai, G.: No-reference quality assessment for 3D colored point cloud and mesh models. *IEEE Trans. Circ. Syst. Video Technol. (TCSVT)* **32**(11), 7618–7631 (2022)
37. Zheng, X., Jiang, G., Yu, M., Jiang, H.: Segmented spherical projection-based blind omnidirectional image quality assessment. *IEEE Access* **8**, 31647–31659 (2020)

38. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR), pp. 16816–16825 (2022)
39. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. *Int. J. Comput. Vis. (IJCV)* **130**(9), 2337–2348 (2022)
40. Zhou, Y., Yu, M., Ma, H., Shao, H., Jiang, G.: Weighted-to-spherically-uniform ssim objective quality evaluation for panoramic video. In: IEEE International Conference on Signal Processing (ICSP), pp. 54–57 (2018)



Full-Frequency Dynamic Convolution: A Physical Frequency-Dependent Convolution for Sound Event Detection

Haobo Yue, Zhicheng Zhang^(✉), Da Mu, Yonghao Dang, Jianqin Yin,
and Jin Tang

Beijing University of Posts and Telecommunications, Beijing, China
{hby, zczhang}@bupt.edu.cn

Abstract. Recently, 2D convolution has been found unqualified in sound event detection (SED). It enforces translation equivariance on sound events along frequency axis, which is not a shift-invariant dimension. To address this issue, dynamic convolution is used to model the frequency dependency of sound events. In this paper, we proposed the first full-dynamic method named *full-frequency dynamic convolution* (FFDConv). FFDConv generates frequency kernels for every frequency band, which is designed directly in the structure for frequency-dependent modeling. It physically furnished 2D convolution with the capability of frequency-dependent modeling. FFDConv outperforms not only the baseline by 6.6% in DESED real validation dataset in terms of PSDS1, but outperforms the other full-dynamic methods. In addition, by visualizing features of sound events, we observed that FFDConv could effectively extract coherent features in specific frequency bands, consistent with the vocal continuity of sound events. This proves that FFDConv has great frequency-dependent perception ability.

Keywords: Sound Event Detection · Full-Frequency Dynamic Convolution · Frequency-Dependent Modeling · Independent Representation spaces · Vocal Continuity

1 Introduction

Sound event detection (SED) is one of the subtasks of computational auditory scene analysis (CASA) [20], which helps machines understand the content of an audio scene. Similar to visual object detection [32] and segmentation [22], SED aims to detect sound events and corresponding timestamps (onset and offset), considered as a prior task of automatic speech recognition (ASR) and speaker verification. It has wide applications in information retrieval [9], smart homes [5], and smart cities [1].

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78498-9_18.

Initially, methods from other domains have greatly promoted the development of SED. Some methods from computer vision, such as SENet [7], SKNet [13], and CBAM [26] improved the capacity of feature representation of the network, adding the attention mechanism to the convolution network. The method from speech processing, for example, conformer [16] also improved the representation capacity, which added the local modeling to the transformer structure by inserting a convolution layer. These methods didn't study the characteristics of audio data, resulting in not great detection performance. Specifically, SENet, SKNet, and CBAM are designed on image data with a clear 2D spatial concept, while audio data is a time sequence. Conformer is designed on speech data containing only the speech sound event, meaning time-frequency patterns of speech data are distributed only in a certain fixed frequency band. However, audio data always contains multiple sound events, and so has diverse time-frequency patterns of sound events.

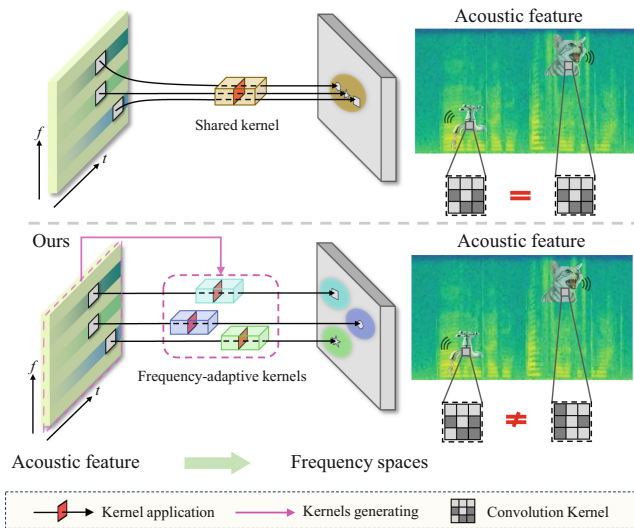


Fig. 1. Illustration of frequency-dependent modeling. Top models time-frequency patterns in the same space with a shared kernel. Bottom models them in several spaces with frequency-adaptive kernels, in which time-frequency patterns specific to sound events can be considered.

Recently, the characteristics of audio data started to be studied, and the dynamic convolution network has been tried in SED. Dynamic convolution network [8] was initially proposed for video prediction. It was designed to generate future frames based on the motion pattern within a particular video. The parameters of the dynamic convolution kernel are always adapted to the input. In SED, different sound events are distributed in different frequency regions, and this frequency dependence is invariant over time. This has motivated some

researchers to investigate whether the adaptivity of dynamic convolution can improve the capability of 2D convolution in modeling the frequency dependence of sound events. [17] proposed frequency dynamic convolution (FDConv), which found that the time-frequency spectrogram is not translation invariant on frequency dimension like image data. FDConv extracts frequency-adaptive attention weights from input for several pre-initialized convolution kernels. These kernels are then weightedly combined in the number dimension to obtain one convolution kernel. Then, the combined kernel is convoluted with the input in a standard manner. [27] proposed multi-dimensional frequency dynamic convolution (MFDCConv), which extends the frequency-adaptive dynamic properties of convolutional kernels to more dimensions of the kernel space, i.e. in-channels, out-channels, and kernel numbers.

Although FDConv and MFDCConv have achieved great performance, they are essentially the same as basic convolution, which is spatially shared. They belong to semi-dynamic convolution in the field of dynamic convolution. As shown in the upper part of Fig. 1, their perception abilities of different frequency bands are identical. They can only model time-frequency patterns in one representation space, where sound events are not easily recognized from each other. Compared with semi-dynamic convolution, full-dynamic convolution [8, 24, 25, 28, 31] attracts more attention recently, which uses a separate network branch to predict a specific filter for each pixel. [31] found this type of dynamic convolution is equivalent to applying attention on unfolded input features, which enables it more effective when modeling complex patterns. Sound events' time-frequency patterns are highly frequency-dependent, and full-dynamic convolution can model features of spatial pixels with different filters. Full-dynamic convolution may be optimal in dealing with recognizing sound events.

In this paper, we propose a novel method named *full-frequency dynamic convolution* (FFDCConv), which is the first full-dynamic convolution method for SED. As shown in the lower part of Fig. 1, FFDCConv generates frequency-specific kernels, resulting in distinct frequency representation spaces. This design is applied directly in the network structure for frequency-dependent modeling. In this way, the 2D convolution is physically furnished with the capability of frequency-dependent modeling, so that the specific time-frequency patterns can be acquired for different sound events. In the end, sound events can be easily recognized from each other in subsequent classification.

Contributions. (1) We proposed full-frequency dynamic convolution that can model time-frequency patterns in independent representation spaces. This method will extract more discriminative features of sound events, resulting in effective classification. (2) The Proposed method outperforms not only baseline but also pre-existing full dynamic filters method in other domain. (3) By visualizing features of sound events, we found the ability to model temporally coherent features is essential to the detection of sound events. And the FFDCConv has this ability.

2 Related Work

Recently, sound event detection has achieved great success with the help of deep learning. Existing methods include uninitialized learning and fine-tuning pretrained models.

Uninitialized Learning in SED. Most uninitialized Learning methods employ convolution networks. They either use initial version networks from the computer vision domain or design a new convolution network. As for methods from the computer vision domain, viewing the audio spectrogram as 2D image data, SENet [7], SKNet [13], and CBAM [26] directly extract features from the audio spectrograms, not considering the physical consistency between standard convolution and audio spectrogram. As for designing a new convolution network, finding the audio spectrogram is not translation invariant on frequency dimension like image data, FDConv [17] designed a frequency-dependent convolution, which equipped the basis convolution with the capacity to model frequency dependence of sound events. To further improve this capacity, MFDCConv [27] extended convolutional kernels' frequency-adaptive dynamic properties to more kernel space dimensions, i.e., in-channels, out-channels, and kernel numbers.

Fine-Tuning Pretrained Models in SED. In comparison, fine-tuning pretrained models always initialize the networks with weights from the upstream tasks. Chosen models either come from out-of-domain task (vision pre-training) or in-domain task (audio pre-training). As for fine-tuning models from the audio pre-training task, AST-SED [10] and PaSST-SED [11] fine-tuned the audio spectrogram transformer (AST) [6] with task-aware adapters in SED. ATST-SED [21] fine-tuned the ATST [12] and BEATs [4] with a two-stages training strategy. As for fine-tuning models from the vision pre-training task, HTS-AT [3] fine-tuned the swin-transformer [14] in the SED task.

3 Methodology

3.1 Full-Dynamic Convolution

A basic 2D convolution can be denoted as $y = \mathbf{W} * x + \mathbf{b}$, where $x \in \mathbb{R}^{T \times F \times C_{in}}$ and $y \in \mathbb{R}^{T \times F \times C_{out}}$ denote the input feature and output feature; $\mathbf{b} \in \mathbb{R}^{C_{out}}$ and $\mathbf{W} \in \mathbb{R}^{k \times k \times C_{in} \times C_{out}}$ denote the bias and weight of a basic convolution kernel. In contrast to basis convolution, full-dynamic convolution [8] leverages separate network branches to generate the filters for each pixel. Full-dynamic convolution operation can be written as:

$$\begin{aligned} y &= \mathbf{Concat}(\mathbf{W}_{t,f} * x(t, f)) \\ \mathbf{W}_{t,f} &= \mathbf{G}(x, t, f) \end{aligned} \quad (1)$$

where $\mathbf{W}_{t,f}$ denotes weights of filter for the current pixel; The \mathbf{G} is the filter generating function; \mathbf{Concat} here aims to convey that convolution operation of each pixel is independent and parallel. For simplicity, the bias term is omitted.

3.2 Overall of Proposed Method

As is commonly understood, different sound events have different frequency band distributions. For instance, catcall, which is sharp, shrill, and high-pitched, is often heard in the high-frequency range; running water, which is low, soft, and soothing, is often heard in the low-frequency range. Based on this, we explore designing a new convolution for SED, which can capture the distribution of frequency bands and model time-frequency patterns of sound events in different frequency representation spaces.

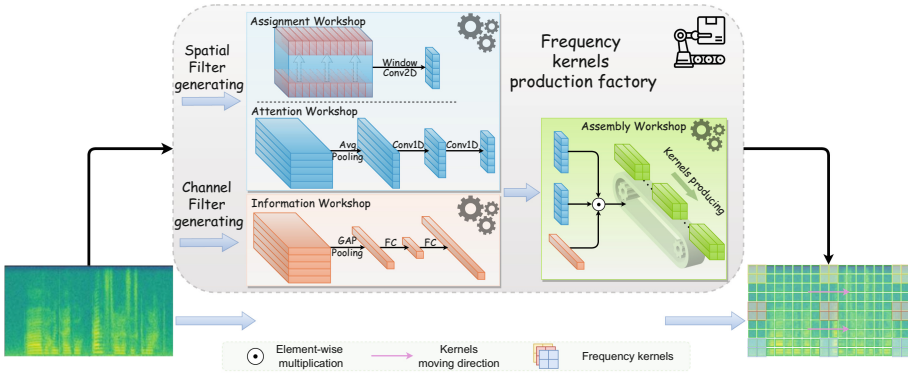


Fig. 2. Illustration of full-frequency dynamic convolution. In general, the factory produces frequency-dependent kernels from acoustic feature, and then kernels are convoluted with input along the time axis. In the factory, there are two workshops aiming to produce spatial filters and channel filters, respectively. And they are integrated in the assembly workshop.

Inspired by full dynamic convolution [31], we designed the full-frequency dynamic convolution (FFDConv) for SED. Overall, as shown in Fig. 2, FFDConv employs a separate branch to predict kernels for each frequency band, in which the content of kernels is based on input feature. In the kernel-generating branch, there are two sub-branches: the spatial filter-generating branch for the spatial space of kernels and the channel filter-generating branch for the channel space of kernels. After spatial and channel filters are obtained, they are combined and then convoluted with the input feature. Note that similarly, full-temporal dynamic convolution (FTDConv) predicts kernels for each temporal frame, and kernels are convoluted with input along the frequency axis.

3.3 Full-Frequency Dynamic Convolution

Unlike the previous semi-dynamic convolution, FFDConv is designed directly in the structure for frequency-dependent modeling. It models the feature along the

frequency axis in different representation spaces. Mathematically, FFDCConv can be written as:

$$y = \mathbf{Concat}(\mathbf{W}_f * x(f), \dim = f) \quad (2)$$

$$\mathbf{W}_f = G_s(x, f) \odot G_c(x, f)$$

where \mathbf{W}_f is the content-adaptive kernel for the f^{th} frequency band; $x(f) \in \mathbb{R}^T$ is the f^{th} frequency band of input feature; G_s and G_c are the spatial and channel filter-generating function; \odot denotes the elemental dot product operator. For clarity, **Concat** here aims to convey that \mathbf{W}_f is convolved with input along the time axis and the operation is parallel.

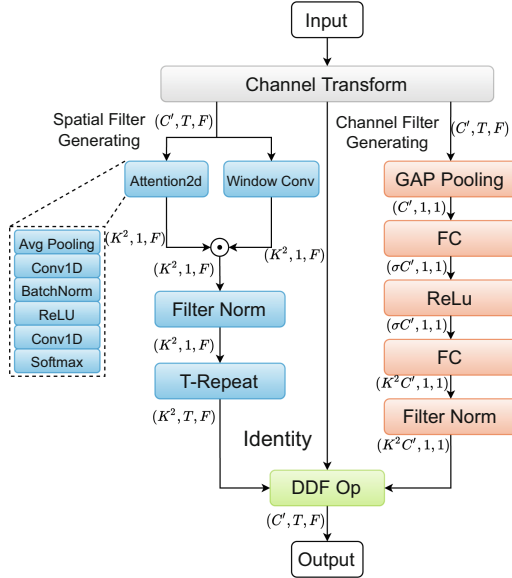


Fig. 3. Details of the FFDCConv

FFDCConv employs a separate branch to generate convolution kernels for each frequency band, in which there are two sub-branches: spatial filter-generating branch and channel filter-generating branch. The spatial filter-generating module is designed to predict the spatial content of dynamic kernels, and the channel-generating module is designed to predict the channel content of dynamic kernels. For efficiency, the dynamic filters are decoupled into spatial and channel ones, following [31].

Spatial Filter Generating. As illustrated in Fig. 3, we use a standard Conv2D to compress the time dimension of input and map channel dimension from C to K^2 , whose kernel weight $W \in \mathbf{R}^{C \times K^2 \times T \times W}$, where W is the window size of the kernel in the frequency dimension. It moves along the frequency axis when

convoluted with input. In this way, not only are the adjacent frequency components considered, but information along the time axis is aggregated. Then, the spatial filter of FFDCConv is obtained, which assigns $K \times K$ spatial weight to every frequency kernel and is highly related to the input. Consequently, FFDCConv can model features from different frequency bands of the input in independent representation spaces. In comparison, the convolution map in FTDCConv is $W \in \mathbf{R}^{C \times K^2 \times W \times F}$, and it moves along the time axis when convoluted with input, resulting in $K \times K$ spatial weight to every time kernel. The convolution map in full dynamic convolution [31] is $W \in \mathbf{R}^{C \times K^2 \times 1 \times 1}$, and assigns $K \times K$ spatial weight to every pixel finally.

Considering these representation spaces may be far apart from each other, we employ an attention2d module following [17] to limit individual differences between them so as not to be too large. Finally, the spatial filter is passed through a Filter-Norm module following [31], avoiding the gradient vanishing/exploding during training.

Channel Filter Generating. As illustrated in Fig. 3, the channel filter generating module is similar to the SE block [7]. It compresses the time and frequency feature of input by applying an average pooling and maps the channel dimension from C to CK^2 by two fully connected (FC) layers. Between two fully connected layers, the ReLU activation function is applied to introduce non-linearity. After input is passed through this module, the channel filter of FFDCConv is obtained, which assigns C channel weight to each spatial location of the frequency kernel. It should be noted that the channel filter for F frequency kernels is the same. In the end, the channel filter is also passed through the Filter-Norm [31]. The spatial and channel filters are mixed by dot product, and the full frequency kernels are obtained. We then use them to model time-frequency patterns of input features. Note that full dynamic convolution [31] and FTDCConv have the same channel filter generating branch.

3.4 FFDCConv Block

Considering that the frequency kernels of FFDCConv don't have the ability to change the channel dimension of input features, we design an FFDCConv block that contains the channel mapping. As illustrated in Fig. 3, firstly, the channel dimension of input is mapped from C_{in} to C_{out} after passing through the channel transformation module, where a standard 2D convolution is employed. Then, based on the input feature, the spatial and channel filters are obtained by passing through the spatial and channel filter generating module. Full-frequency dynamic kernels are obtained by mixing the spatial and channel filters. Finally, the kernels are convoluted with input along the time axis.

In the actual algorithm, following [31], spatial filters, channel filters, and input are sent to DDF operation to get the output, which is implemented in CUDA, alleviating any need to save intermediate multiplied filters during network training and inference. Note that the DDF op needs $H \times W$ spatial filters. We repeat the $1 \times F$ spatial filters to $T \times F$ so that the kernel's weights are the same along the time axis when convoluted with input in f^{th} frequency band.

4 Experiment

4.1 Dataset, Metrics and Implementation Details

Dataset. All experiments are conducted on the dataset of Task 4 in the DCASE 2022. The training set consists of three types of data: weakly labeled data (1578 clips), synthetic strongly labeled data (10000 clips), and unlabeled in-domain data (14412 clips). The real validation set (1168 clips) is used for evaluation. The input acoustic feature is the log Mel spectrogram extracted from 10-second-long audio data with a sampling rate of 16 kHz. The feature configuration is the same as [13], in which the input feature has 626 frames and 128 mel frequency bands.

Implementation Details. The baseline model is the CRNN architecture [33], which consists of 7 layers of conv blocks and 2 layers of Bi-GRU. Attention pooling module is added at the last FC layer for joint training of weakly labeled data, and mean teacher (MT) [23] is applied for consistency training with unlabeled data for semi-supervised learning. Data augmentations such as MixUp [29], time masking [19], frame-shift, and FilterAugment [18] are used. The data augmentation parameters are identical to [17]. The metrics hyper-parameters are identical to [17]. The model is trained using the Adam optimizer with a maximum learning rate of 0.001, and ramp-up is used for the first 80 epochs.

Metrics. Poly-phonic sound event detection scores (PSDS) [2], collar-based F1 score (EB-F1) [15], intersection-based F1 score (IB-F1) [2] are used to evaluate the model performance. Median filters with fixed time length are used for post-processing, and sound events have different thresholds from each other to obtain hard predictions for calculating EB-F1. These metrics have different focuses. PSDS1 and CB-F1 reflect more on the system’s capacity for detecting sound events. PSDS2 and IB-F1 reflect more on the system’s capacity for classifying and differing sound events.

Table 1. SED performance comparison between models using different convolutions on the real validation set. The best results are in **bold**, and the second best are in underlined. * denotes the results from our implementation using the codebase from [17].

Model	Params	PSDS1 \uparrow	PSDS2 \uparrow	CB-F1 \uparrow	IB-F1 \uparrow
Baseline* [33]	4M	0.370	0.579	0.469	0.714
DDFConv* [31]	7M	0.387	0.624	0.467	0.720
FTDConv*	7M	0.395	0.651	0.495	<u>0.740</u>
SKConv [30]	—	0.400	—	0.520	—
FDConv* [17]	11M	0.431	0.663	0.521	0.738
MFDCConv [27]	33M	0.461	<u>0.680</u>	0.542	—
FFDCConv*	11M	<u>0.436</u>	0.685	<u>0.526</u>	0.751

4.2 Full-Frequency Dynamic Convolution on SED

We compared the performances of baseline with full dynamic convolution methods, including decoupled dynamic convolution (DDFConv) [31], full-temporal dynamic convolution (FTDConv), and full-frequency dynamic convolution (FFDConv). For full dynamic convolution methods, dynamic convolution layers replaced all convolution layers except the first layer from the baseline model [33]. Besides, some typical convolution methods are also compared.

Compared with dynamic convolution methods, from Table 1, three types of full dynamic convolution can all outperform the baseline, which proves full dynamic convolution qualifies in SED. In addition, it can be seen that the effects of three types of convolution are in increasing order. First, FTDConv and FFDConv employ content-adaptive temporal or frequency kernels, which can be viewed as giving prior knowledge to SED compared with DDFConv. Second, FFDConv outperforms FTDConv, which can prove that time-frequency patterns of sound events are highly frequency-dependent, and this dependency is time-invariant. Moreover, FFDConv models acoustic features with different kernels along the frequency axis, which can be thought to be frequency components modeled in different representation spaces. As if components of the feature are split into different frequency spaces and then reassembled. This is consistent with the characteristics of sound events.

Compared with other typical methods, such as MFDCConv [27], SKConv [30], FDCConv [17], FFDConv still demonstrates competitive performances. Especially in terms of PSDS2 and IB-F1, FFDConv was the best among all convolution methods. It approves the effectiveness of FFDConv, which captures the information of different frequency band distributions for different sound events and then models more differentiated time-frequency patterns for them, favoring the classification of sound events. In terms of the PSDS1 and CB-F1, the FFDConv is suboptimal and slightly higher than FDCConv [17]. FFDConv’s kernels are time-invariant in some frequency band, the same as FDCConvs [17], leading to a close result. Compared to the MFDCConv [27], the latest state-of-the-art convolution method, FFDConv can also get competitive performances, with a 66.7% decrease in the number of parameters.

4.3 Fine-Grained Modeling Study

To explore FFDConv’s ability to understand acoustic spectral information at a fine-grained level. We visualized feature of the middle layer. More visualizations can be found in the supplementary material.

The visualization results are shown in Fig. 4. Comparing the features of FFDConv and CRNN, we can see that most of the time-frequency patterns modeled by CRNN are temporally isolated and disjoint. In contrast, FFDConv’s patterns and their neighbors are in a whole, thereby forming a distinct time-frequency representation. Moreover, this phenomenon can also be found in trends of frequency band features over time. The waveforms of FFDConv are smoother than CRNN. Specifically, the duration of peak and trough is longer in FFDConv’s waveform,

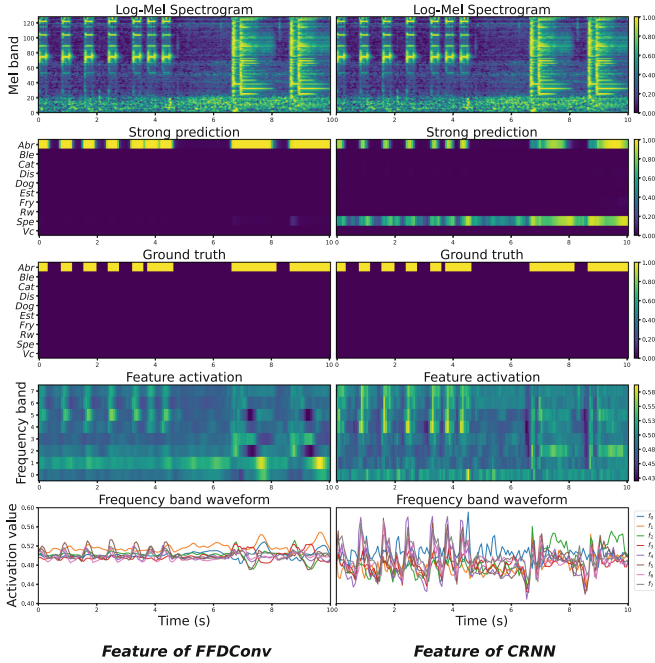


Fig. 4. Feature comparison of FFDCConv and CRNN. Features activation of the 5th Conv block are shown in the 4th row. The trends of frequency band features over time are shown in the 5th row. Note that y-axis labels of strong prediction are abbreviations of the sound event categories. For example, Abr stands for Alarm bell ringing.

which results from the feature being mostly coherent over time. There are more pulses in the resting state of CRNN’s waveforms, which are in a disorganized state. Besides, the distributions of frequency band features are consistent with alarm_bell_ring’s spectrogram in FFDCConv’s waveforms. The values of the low-frequency band features are smaller than those of the middle and high-frequency bands when the alarm bell rings. However, the differences between frequency bands in CRNN are ambiguous. As for the model’s prediction, the CRNN’s isolated features directly lead to the incoherent output compared with ground truth, which proves that the feature’s coherence over time is essential. Interestingly, the low-frequency white noise of the sound clip is filtered by FFDCConv, but CRNN tagged it as speech. This has to do with that dynamic convolution concentrates more on high-frequency texture information, and white noise in the spectrogram lacks clear contour information.

Most SED models are trained in a frame-based supervised way, which always leads to the feature and output being discrete over time. However, FFDCConv can alleviate this by frequency-dependent modeling, which models different patterns for frequency bands, leading to a distinct representation of a sound event. This modeling way is like an attention mechanism in which the distribution

of frequency band information of the spectrogram is maintained. Besides, the convolution kernel for a frequency band is shared in all frames, which produces temporally coherent representations. This is consistent with both the continuity of the sound waveform and the vocal continuity of sound events.

4.4 Ablation Study

We compared the performance of different window sizes of the build kernel when generating spatial filters. Note that the size of the spatial filter K is set to 3.

Table 2. Comparison of different window size, W .

Model	<i>Atten</i>	W	PSDS1	PSDS2
	✗	3	0.421	0.650
	✓	1	0.421	0.659
FFDConv	✓	3	0.436	0.685
	✓	5	0.423	0.656
	✓	7	0.432	0.666

The results are shown in Table 2. With constraints of the attention module, FFDConv can get better performance. This proves that before attention, spatial filters of different frequency spaces may have a large distance from each other. The performance of FFDConv is the best when window size is set to 3. This is because the adjacent frequency components are considered compared to size 1 when generating the spatial filter, and size 5 may suffer from overfitting. In addition, it’s interesting that the performance recovers when the window size is set to 7. This may have to do with the fact that dynamic convolutions are relatively unstable.

5 Conclusions

In this paper, we proposed full-frequency dynamic convolution, the first full-dynamic method for SED. Full-frequency dynamic convolution is designed to model time-frequency patterns in different frequency spaces. This design in structure physically furnished 2D convolution with the capability of frequency-dependent modeling. Experiments on the DESED show that full-frequency dynamic convolution is superior to not only baseline but also other full-dynamic convolutions, which proves FFDConv qualifies in SED. In addition, by visualizing features of sound events, we found that FFDConv can extract temporally coherent features in specific frequency bands, which is consistent with the vocal continuity of sound events. This proves that FFDConv has great frequency-dependent perception ability. In the future, we aim to explore new methods to model vocal continuity of sound events.

Acknowledgements. This work was supported partly by the National Natural Science Foundation of China (Grant No. 62173045, 62273054), partly by the Fundamental Research Funds for the Central Universities (Grant No. 2020XD-A04-3), and the Natural Science Foundation of Hainan Province (Grant No. 622RC675).

References

1. Bello, J., Mydlarz, C., Salamon, J.: Sound analysis in smart cities. In: Virtanen, T., Plumbley, M., Ellis, D. (eds.) *Computational Analysis of Sound Scenes and Events*, pp. 373–397. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63450-0_13
2. Bilen, Ç., Ferroni, G., Tuveri, F., Azcarreta, J., Krstulović, S.: A framework for the robust evaluation of sound event detection. In: *ICASSP*, pp. 61–65 (2020)
3. Chen, K., Du, X., Zhu, B., Ma, Z., Berg-Kirkpatrick, T., Dubnov, S.: HTS-AT: a hierarchical token-semantic audio transformer for sound classification and detection. In: *ICASSP*, pp. 646–650 (2022)
4. Chen, S., et al.: BEATs: audio pre-training with acoustic tokenizers. In: *ICML*, pp. 5178–5193 (2023)
5. Debes, C., Merentitis, A., Sukhanov, S., Niessen, M., Frangiadakis, N., Bauer, A.: Monitoring activities of daily living in smart homes: understanding human behavior. *IEEE SPM* **33**(2), 81–94 (2016)
6. Gong, Y., Chung, Y.A., Glass, J.: AST: audio spectrogram transformer. In: *INTERSPEECH*, pp. 571–575 (2021)
7. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: *CVPR*, pp. 7132–7141 (2018)
8. Jia, X., De Brabandere, B., Tuytelaars, T., Gool, L.V.: Dynamic filter networks. In: *NEURIPS* (2016)
9. Jin, Q., Schulam, P., Rawat, S., Burger, S., Ding, D., Metze, F.: Event-based video retrieval using audio. In: *INTERSPEECH*, pp. 2085–2088 (2012)
10. Li, K., Song, Y., Dai, L.R., McLoughlin, I., Fang, X., Liu, L.: AST-SED: an effective sound event detection method based on audio spectrogram transformer. In: *ICASSP*, pp. 1–5 (2023)
11. Li, K., Song, Y., McLoughlin, I., Liu, L., Li, J., Dai, L.R.: Fine-tuning audio spectrogram transformer with task-aware adapters for sound event detection. In: *INTERSPEECH*, pp. 291–295 (2023)
12. Li, X., Shao, N., Li, X.: Self-supervised audio teacher-student transformer for both clip-level and frame-level tasks. *IEEE/ACM TASLP* **32**, 1336–1351 (2024)
13. Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks. In: *CVPR*, pp. 510–519 (2019)
14. Liu, Z., et al.: Swin Transformer: hierarchical vision transformer using shifted windows, In: *ICCV*, pp. 10012–10022 (2021)
15. Mesaros, A., Heittola, T., Virtanen, T.: Metrics for polyphonic sound event detection. *AS* **6**(6) (2016)
16. Na, T., Zhang, Q.: Convolutional network with conformer for semi-supervised sound event detection. Technical report, DCASE2021 Challenge (2021)
17. Nam, H., Kim, S.H., Ko, B.Y., Park, Y.H.: Frequency dynamic convolution: frequency-adaptive pattern recognition for sound event detection. In: *INTERSPEECH*, pp. 2763–2767 (2022)
18. Nam, H., Kim, S.H., Park, Y.H.: FilterAugment: an acoustic environmental data augmentation method. In: *ICASSP*, pp. 4308–4312 (2022)

19. Park, D.S., : SpecAugment: a simple data augmentation method for automatic speech recognition. In: INTERSPEECH, pp. 2613–2617 (2019)
20. Rouat, J.: Computational auditory scene analysis: principles, algorithms, and applications. *IEEE TNN* **19**(1), 199–199 (2008)
21. Shao, N., Li, X., Li, X.: Fine-tune the pretrained ATST model for sound event detection. arxiv preprint (2023)
22. Sun, J., Li, Y., Lu, H., Kamiya, T., Serikawa, S.: Deep learning for visual segmentation: a review. In: COMPSAC, pp. 1256–1260 (2020)
23. Tarvainen, A., Valpola, H.: Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results. In: NEURIPS (2017)
24. Tian, Z., Shen, C., Chen, H.: Conditional convolutions for instance segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 282–298. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_17
25. Wang, X., Zhang, R., Kong, T., Li, L., Shen, C.: SOLOv2: dynamic and fast instance segmentation. In: NEURIPS, pp. 17721–17732 (2020)
26. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: CBAM: convolutional block attention module. In: ECCV, pp. 3–19 (2018)
27. Xiao, S., Zhang, X., Zhang, P.: Multi-dimensional frequency dynamic convolution with confident mean teacher for sound event detection. In: ICASSP, pp. 1–5 (2023)
28. Zamora Esquivel, J., Cruz Vargas, A., Lopez Meyer, P., Tickoo, O.: Adaptive convolutional kernels. In: ICCV Workshops (2019)
29. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: beyond empirical risk minimization. *ICLR* (2018)
30. Zheng, X., Song, Y., McLoughlin, I., Liu, L., Dai, L.R.: An improved mean teacher based method for large scale weakly labeled semi-supervised sound event detection. In: ICASSP, pp. 356–360 (2021)
31. Zhou, J., Jampani, V., Pi, Z., Liu, Q., Yang, M.H.: Decoupled dynamic filter networks. In: CVPR, pp. 6647–6656 (2021)
32. Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years: a survey. *Proc. IEEE* **111**(3), 257–276 (2023)
33. Çakır, E., Parascandolo, G., Heittola, T., Huttunen, H., Virtanen, T.: Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM TASLP* **25**(6), 1291–1303 (2017)



EDM10: A Polyphonic Stereo Dataset with Identical BGM for Musical Instrument Identification

Himadri Mukherjee¹(✉), Matteo Marciano², Ankita Dhar³, and Kaushik Roy¹

¹ TISA Lab, Department of Computer Science,
West Bengal State University, Barasat, India
himadrim027@gmail.com

² Gazelien Records Lab, Department of Arts and Humanities,
Music Program New York University Abu Dhabi, Abu Dhabi, UAE
matteo.marciano@nyu.edu

³ Department of Computer Science and Engineering,
Sister Nivedita University, New Town, India

Abstract. Music Signal Processing has significantly evolved in the past decades. One of the major areas of interest in this field has been automatic music transcription. It is a challenging task by itself that aggravates even more when the input audio is polyphonic (multiple instruments and timbres are played simultaneously). This requires the identification of musical instruments in the piece at the outset. The field of music signal processing that deals with this aspect is known as automatic music instrument identification. This field also has the potential of categorizing and recommending music based on instruments. Disparate datasets have been proposed to date for this task but none of them have interclass background similarity to the best of our knowledge. Further, the lead melody being played also varies from class to class in most cases. These aspects can introduce a possible bias for the machine learning models that can get manipulated unfairly by the additional variances in the classes other than the lead instrument itself. This sets the stage for a dataset where the classes are different only in terms of the tone of the lead instrument alone. In this paper, we introduce the first musical instrument dataset of 10 musical instruments with Electronic Dance Music melodies (EDM10) having identical background music (BGM) across instruments. This dataset is the first of its kind wherein synthetic tones have been used that have taken over the Globe. We introduce out-of-mood testing using exotic scales for musical instrument identification. The dataset is composed of 35800 polyphonic clips of 3 s each and a baseline result of 89.73% was obtained using a deep learning-based approach. The dataset is freely available for research purposes. <https://forms.glyV5e36TK1jHMKjpC6>.

H. Mukherjee and M. Marciano—The authors contributed equally.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15320, pp. 273–288, 2025.
https://doi.org/10.1007/978-3-031-78498-9_19

Keywords: Musical instrument identification · Polyphonic music · Electronic Dance Music · Identical Background · Synthetic Tones · Out-of-Mood Test

1 Introduction

The field of Computer Science that deals with automatic analysis, manipulation, and understanding of music signals is known as Automatic Music Signal Processing [15]. It is composed of multifarious aspects like music generation [4], genre identification [11], beat tracking [18], instrument identification [14], music transcription [1], etc. Leveraging concepts from digital signal processing, machine learning, and music theory, scientists in this field attempt to automate disparate aspects of music production and analysis that require human expertise. It involves the extraction of different meaningful information like rhythm, pitch, and timbre to name a few. One of the major areas of interest in this domain is Automatic Music Transcription which facilitates automatically notating a music piece. This is essential for music understanding, music generation, and other tasks.

A music piece can be broadly categorized into 2 groups - Monophonic and Polyphonic based on the number of notes that are played simultaneously. A Monophonic audio is one where a single musical note is played at a time which implicitly means that a single instrument is played at a time. In the case of polyphonic music multiple notes are played at a time which can be from the same or different instruments. The songs and instrumentals that we listen to are mostly polyphonic. Automatic Transcription for a monophonic piece is a challenging task that greatly increases when the piece is polyphonic. As multiple instruments can simultaneously play in this type of audio, it becomes necessary to identify the instrument being played and thereafter invoke the transcription mechanism for a specific instrument. This has led to the development of a field of research named Automatic Musical Instrument identification [14] from audio clips. Another application of this field is the categorization and recommendation of music pieces based on musical instruments. This field has the potential to contribute to the advancements in music technology and also enhance the accessibility and understanding of music for both professionals and enthusiasts.

Szeliga et al. [21] presented a convolutional neural network (CNN)-based approach for musical instrument recognition. They trained the system using 3 different monophonic datasets namely University of IOWA Musical Instrument Samples¹, Philharmonia Orchestra Sound Samples², and RWC Music Dataset [7]. They also experimented with polyphonic audio from the IRMAS dataset [3]. They used a staged training approach wherein, the system was initially trained with monophonic audio and thereafter using polyphonic audio. The experiments were performed on 7 instruments namely Cello, Clarinet, Flute, Guitar, Trumpet, Saxophone, and Violin. They reported the highest average training accuracy

¹ <https://theremin.music.uiowa.edu/MISPost2012Intro.html>.

² <https://philharmonia.co.uk/resources/sound-samples/>.

of 75.29% with the lowest average difference of 24% for the 2 stage approach. Han et al. [8] also experimented on the IRMAS dataset using a CNN-based approach coupled with Mel-spectrograms. They reported micro and macro F-Scores of 0.619 and 0.513 respectively which was 23.1% and 18.8% above the baseline results. Uruthiran and Ranathunga [23] attempted to distinguish Oriental musical instruments from the Philharmonia dataset. They used multifarious spectral and time domain features for categorizing the audios and reported the highest classification accuracy of 94.02% for 20 instruments using SVM. Toghiani-Rizi and Windmark [22] presented a neural network-based system for the distinction of 8 different musical instruments namely Trumpet, Oboe, Violin, Clarinet, Guitar, English Horn, Cello, and Banjo. They performed several experiments wherein the base experiment involved frequency-based analysis which yielded the highest accuracy of 93.5%. Nirozika et al. [17] attempted to distinguish 10 different Sri Lankan instruments with an SVM-based approach. The dataset consisted of solo instruments which were parameterized using Mel Frequency Cepstral Coefficient (MFCC)-based features and the highest accuracy of 86.8% was reported. Mukherjee et al. [13] attempted to distinguish 6 different types of pianos from a dataset where the backgrounds were the same for the different classes. Out of the 6 Pianos, 2 were electric while the rest were acoustic. The audio clips were parameterized using line spectral frequency-based features wherein the highest performance of 97.06% was obtained with neural network-based classification. Dutta et al. [5] presented a scalogram-based approach coupled with CNN for instrument identification. Experiments were performed on a monophonic dataset of 14 instruments wherein an accuracy of 85% was reported by using only 20% of the data for training and the rest for testing. Ghosh et al. [6] presented a Decision Tree-based approach for distinguishing 9 different instruments. The dataset was monophonic wherein the highest accuracy of 84.02% was reported by using spatial features that were extracted from the 2D representation of the audio. Blaszkę and Kostek [2] used a CNN-based approach for distinguishing musical instruments. They experimented with only 4 instruments from the Slakh dataset [12] namely Bass, Drum, Guitar, and Piano. They modeled the audio using MFCC features and reported class-level accuracies in the range of 86% to 99%. Nagawade and Ratnaparkhe [16] used MFCC-based features coupled with KNN classifier for distinguishing 5 instruments namely Piano, Cello, Violin, Flute, and Trumpet. The system was limited to working only for monophonic clips and an accuracy of 88.33% was reported. Solanki and Pandey [20] used Mel Spectrogram for the identification of musical instruments from the IRMAS dataset. The spectrograms were fed to an 8 layered CNN wherein the highest accuracy of 92.8% was reported.

It is observed from the literature that many works are based on monophonic datasets which is not ideal for real-World scenarios. There are works on polyphonic datasets as well, but most of them are not tested with data having identical backgrounds or melodies across instruments. Testing on such a scenario can help to identify a system's lead instrument tone-identifying capability.

2 Dataset

Data is an essential entity of any experiment. It should be robust enough to portray real-world characteristics that in turn facilitate the training of robust systems. The test-train split of data is a very important factor. It needs to be ensured that the test set is not biased in terms of the contributors in the training set. The dataset should be focused on the objective of a task and it needs to be ensured that all the factors that are variable in the dataset are directly related to the objective and do not provide additional variance within classes which in turn can unfairly enhance the performance of a system. Though such systems yield higher results, their application in the real world is limited at times.

In this paper, we introduce a dataset for Musical Instrument Recognition which has several distinctive features that are not available in the presently available datasets to the best of our knowledge. The dataset is based on Electronic Dance Music and encompasses 10 Musical Instruments, hence the name EDM10. The instruments include Cello, Clarinet, Guitar, Harmonium, Pan Flute, Piano, Santoor, Sitar, Trumpet, and Violin. These instruments were selected to accommodate both Eastern and Western instruments. The chosen instrument set covers multiple instrument families like Wind (Clarinet, Pan Flute, Harmonium, Trumpet), String (Guitar, Violin, Cello, Santoor, Piano), Reed (Clarinet, Harmonium), Key (Harmonium, Piano). Several instruments belong to multiple families like Piano, Harmonium, and Santoor to name a few. In the case of Piano, it is played by pressing keys, in the background strings are struck by the hammer of the respective keys on being pressed. Hence, it can be considered as a key, string, or even percussion instrument. Similarly for Santoor, it is a stringed instrument but is played by striking the strings with hammers. In the case of Harmonium, it produces sound when a key is pressed coupled with an inflow of air into the reed.

Electronic Dance Music was chosen to engender this dataset for several reasons. One of the primary reasons is its popularity over the years³. Another reason is that processing EDM can be very challenging due to multifarious aspects which are discussed as follows:

- Complicated Arrangements: EDM often contains multiple layers of synthetic sounds, samples, and effects which poses a challenge during analysis.
- High Dynamic Range: EDM tracks demonstrate a wide dynamic range in many cases. A single piece often has very quiet and very loud sections. This is challenging to process without the introduction of distortion or clipping.
- Fast Tempo and Rhythms: EDM can be characterized by a fast tempo with intricate rhythmic patterns in the background and foreground. Such intricacies of the background music (BGM) often hinder with the lead instruments during automated processing.
- Frequency Spectrum: EDM tracks mostly have a broad frequency envelope. This includes strong basslines, high-frequency, and complex synths.

³ <https://www.yourmusiccharts.com/50-best-edm-songs-2023/>.

- Dynamic Effects and Transitions: EDM often uses dynamic effects like sweeps, filters, drops, and transitions. These change the overall characteristics of the audio and are difficult to tackle for autonomous systems.
- Compression and Limiting: EDM tracks are often accompanied by heavy compression and limiting to achieve a loud and punchy sound. This poses a challenge in the automatic analysis and parameterization of a single aspect like the lead instrument.

Musicians were asked to play melodies in 6 different scales namely Major, Major Pentatonic, Minor Pentatonic, Blues, Hiraajoshi, and Arabic. They were provided with stereophonic background tracks consisting of a bassline, a synth arpeggio, and a percussive section. These layers were composed using multiple stems, for instance, the percussive section consisted of a kick stem, a hi-hat stem, and a snare stem in certain instances. In some cases, the BGM was formed using patterns extending across multiple bars. The MIDI session of the lead melody for the different scales was recorded. These sessions were thereafter assigned to the different instrument tones. This ensured that the exact same melody was played for the different instruments which were mixed with the BGM. Thus, the final pieces were only different from each other in terms of the tone of the lead instrument. Since every instrument class consisted of all the 6 scales, each of which had the exact same BGM and lead melody notation, it was ensured that the classes only differed in terms of the tone of the lead instrument thereby facilitating unbiased lead instrument identification. Different melodies in different instrument classes can at times introduce bias in the models by providing an unfair advantage which is avoided in this dataset. The finally rendered clips were split into lengths of 3 seconds for generating the train and test sets. Each of the instruments consisted of 2415 instances in the train set and 1165 instances in the test set. The train set was composed of melodies from the Major, Major Pentatonic, Minor Pentatonic, and Blues scale. The test set was composed of the remaining 2 exotic scales namely Arabian and Hiraajoshi. The mood of these 2 scales was completely different from the ones used in the training set thereby ensuring an unbiased out-of-mood test scenario. The details of the train and test set in terms of the encompassed scales, their root notes, the range of played notes, and the tempos are tabulated in Table 1.

Table 1. Scales along with their constituent notes, tempo, and used range of notes present in the Train and Test set

Partition	Scale	Notes	Range	Tempo
Train	Major	C, D, E, F, G, A, B, C	C5-C7	115
	Major Pentatonic	C, D, E, G, A, C	C4-C7	103
	Minor Pentatonic	C, D [#] , F, G, A [#] , C	F4-F7	110
	Blues	C, D [#] , F, F [#] , G, A [#] , C	C4-G7	108
Test	Arabic	C, C [#] , E, F, G, G [#] , B, C	C5-C7	120
	Hiraajoshi	C, D, D [#] , G, G [#] , C	C4-G [#] 6	106

The BGM and lead melody mixing methodology is illustrated in Fig. 1. The scheme has been demonstrated for a scenario consisting of 3 BGMs (B_1 , B_2 , and B_3) for 3 instruments (I_1 , I_2 , I_3) generating 3 types of track (A_1 , A_2 , and A_3). The mixing of a lead melody from an instrument with a BGM is demonstrated by a colour-coded arrow from B to I . Since the melody from every instrument was mixed with every BGM leading to 3 associations per background (demonstrated by 3 colour-coded arrows originating from each B). This led to 3 type tracks (based on BGM). This led to the production of 9 mixed tracks which were of 3 types as demonstrated in Fig. 1. The first A_1 indicates that it was formed by mixing the lead melody from instrument I_1 (hence the same coloured box) with the BGM B_1 (hence the red hue behind the box denoted as A_1). In a nutshell, every mixed track can be considered as a constrained walk from B to A via I . The constraint is that the incoming and outgoing edge for any I needs to be of the same colour wherein every coloured line can be considered as an edge and every box can be considered as a vertex.

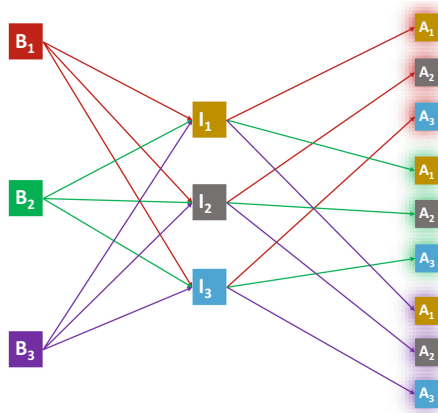


Fig. 1. Associativity of a BGM B with an instrument I leading to an audio A . It is a constrained walk from B to A , via I . The constraint is that the incoming and outgoing edge for any I needs to be of the same colour

The instrument tones were easy to distinguish when played without the BGM, but the overall mix made it difficult to distinguish them. The Mel Spectrogram of a Blues scale melody for the different instruments along with the BGM is presented in Fig. 2. The Mel Spectrograms for the same melody on mixing with the BGM are presented in Fig. 3. It is observed that similar components were introduced post-mixing. The spectrograms were generated by merging both the channels of the stereo tracks into a single track.

The overall tonal texture varied due to different BGMs which is illustrated in Fig. 4. In this Figure, a single clip of Harmonium for the different BGMs is shown. It is seen that for each of these BGMs, a variation is introduced in the audio.

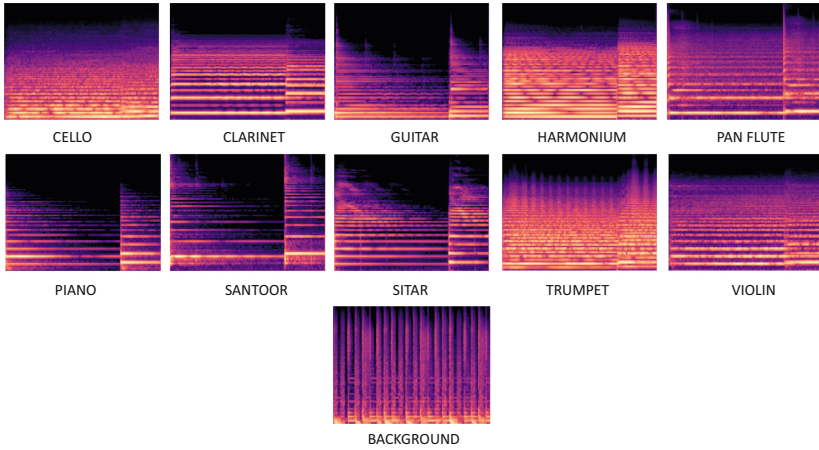


Fig. 2. Mel Spectrograms of the 10 instruments without the presence of BGM. The Mel Spectrogram of the BGM is shown at the bottom

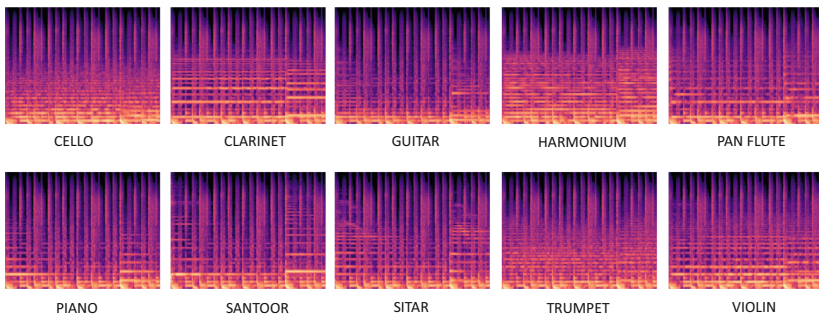


Fig. 3. Mel Spectrograms of the 10 instruments in the presence of BGM

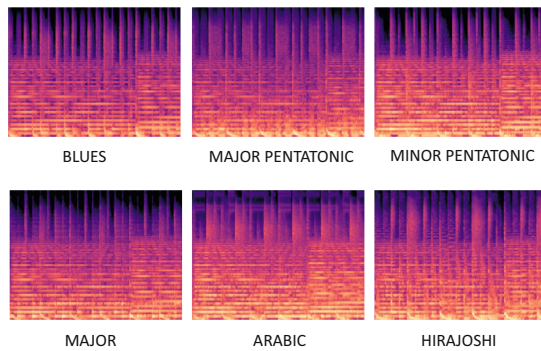


Fig. 4. Mel Spectrograms of Harmonium clip for different scales

The clips were initially recorded in the key of C which was then transposed to the key of B and A \sharp on the lower side (-1 and -2 semitones) as well as C \sharp and D on the upper side ($+1$ and $+2$) semitones. It was noted that differences in the overall audio were introduced due to this transposition which is shown in Fig. 5.

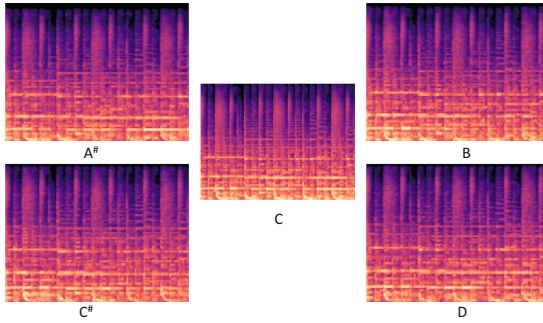


Fig. 5. Mel Spectrograms of a Piano clip in different Keys

The level of the Foreground melody in comparison to the BGM in terms of loudness is presented in Table 2. It presents the instrument-wise mean and standard deviation of the Foreground to BGM levels for the train and test. It is noted that for most of the instruments, the BGM was more dominant as compared to the foreground instrument. This is especially common for the lower octave lead melody sections wherein the bassline becomes more dominant.

Table 2. Foreground and Background power (dB) comparison in instrument level along with their designated symbols

Instrument	Symbol	Train $_{Mean}$	Train $_{Std}$	Test $_{Mean}$	Test $_{Std}$
Cello	CEL	-2.32	3.07	-0.37	1.92
Clarinet	CLA	-0.94	3.24	0.7	1.92
Guitar	GUI	-0.11	3.62	1.06	3.17
Harmonium	HAR	-2.68	4.37	-2.79	4.89
Pan Flute	PAN	-0.9	2.86	0.99	1.71
Piano	PIA	-2.64	4.09	-0.58	3.68
Santoor	SAN	-2.17	5.83	-0.12	4.8
Sitar	SIT	-5.02	4.87	-4.07	5.43
Trumpet	TRU	-8.3	2.86	-6.54	1.35
Violin	VIO	-5.77	3.38	-4.17	2.23

The main features of EDM10 can be summarized as follows:

- The dataset is composed of 10 Musical instruments from both the Eastern and Western World. It has exotic instruments like Santoor and Harmonium as well as closely linked instruments like Violin and Cello.
- The dataset consists of 35800 clips of 3 seconds duration wherein the mood of the clips in the train set is different from the test set thereby ensuring an out-of-mood test scenario. The test set is composed of exotic scales (Hirajoshi and Arabic) while the train set is composed of common scales.
- The BGM of the clips does not vary across instruments thus avoiding unwanted positive bias in instrument identification. Every BGM is present for every instrument in the dataset.
- The lead melody is the same across instruments thereby avoiding unfair positive bias in instrument identification.
- The dataset is composed of stereo audios with a sampling rate of 48000 Hz which is the studio standard.
- Synthetic tones are used in the dataset which is common in the present days. The BGMS are also multilayered and complex. The EDM genre makes analyzing the audio even more challenging due to the complex characteristics of the genre.

3 Baseline System

3.1 Audio Parameterization

The audio clips were parameterized using Mel Spectrograms [19]. A Mel Spectrogram is used to represent the frequency spectrum of an audio computed using the Mel scale. It involves the use of triangular bandpass filters each of which is centered around a particular Mel frequency and spans over a range of frequencies. The Mel Spectrogram represents the distribution of energy across the different frequency bands with temporal information. The regions with higher energy are more brightly coloured than the ones with lower energy distribution.

In this approach, an audio signal $x(t)$ is divided into short overlapping frames of length T where t represents time. Each of these frames is subjected to a windowing function $w(t)$ to avoid spectral leakage. The Fourier transform is computed for each of the frames $x_n(t)$ that produces its frequency domain representation $X_n(f)$ where f represents frequency. This is followed by the generation of the Mel Filterbank consisting of m filters. $H_m(f)$ represents the m^{th} filter whose center frequency is represented by f_m on the Mel scale. This is followed by computation of the power spectrum $P_n(f)$ wherein $H_m(f)$ is applied on $X_n(f)$. This procedure is represented as follows:

$$P_{nm} = \sum f |X_n(f)| \cdot H_m(f) \quad (1)$$

The summation symbol represents the summation of all frequencies covered by a Mel Filter. It aggregates the magnitude spectrum within the filter. This is followed by the conversion of P_{nm} to a logarithmic scale to mimic human perception. As this is done separately for every frame which is thereafter aggregated

to form the entire spectrogram, time resolution is also obtained. In this experiment, the left and right channels of the audio were merged into a single channel at the initial phase before parameterization. The Fourier transformation was performed with a length of 2048 and 512 samples were present in between 2 successive frames. The frames were subjected to the Hanning window which is represented as follows:

$$w(n) = 0.5 \left(1 - \cos \left(2\pi \frac{n}{N} \right) \right) \quad (2)$$

where, window length = $N + 1$ and $0 \leq n \leq N$.

3.2 Deep Learning-Based Classification

Deep learning [10] is a sub-field of Machine learning that leverages complex neural networks with multiple layers. These networks can learn complex patterns from data using hierarchical representations thereby introducing successive abstraction. They are different from traditional Machine Learning techniques that require handcrafted features.

Convolutional Neural Networks (CNNs) [9] are one such type of Deep Learning architecture designed to process grid data and are composed of 3 main components namely: Convolution layer, Pooling layer, and Dense/ Fully-Connected layer.

- Convolution layer: This layer is used for extracting spatial patterns/ features from data using a set of filters. These filters perform element-wise multiplication with the input data and produce feature maps.
- Pooling layer: This layer is used to reduce the spatial dimension of the extracted feature maps from the convolution layers. This achieved by down-sampling the maps which involves aggregating adjacent values which is governed by the filter size. This layer helps to reduce the computational complexity and controls overfitting.
- Dense/ Fully-Connected Layer: This layer is located at the end of a CNN which is responsible for performing the classification/ regression task. Every neuron of a dense layer is connected to every neuron of the subsequent dense layer.

In this work, the baseline architecture Instrument-Network (I-NET) was composed of 4 blocks. The first 3 were feature blocks while the last was a classification block. Each of the blocks was composed of 3 layers. Initially, the Mel spectrograms of 128×128 dimension were fed to the first block having the first convolution layer consisting of 64 filters of 5×5 dimension. This layer used a ReLU activation which is presented below.

$$f(x) = \max(0, x), \quad (3)$$

where x represents the input.

The feature maps extracted from this layer were passed on to a pooling layer where max-pooling was performed on 2×2 windows. This was followed by a 15% dropout and the resultant was passed on to the second block having the second convolution layer consisting of 32 filters of 3×3 dimension. This layer also used ReLU activation whose outcome was max-pooled with 2×2 windows. The resultant was subjected to 20% dropout and passed on to the third block having the third convolution layer consisting of 32 filters each of size 2×2 . The activation was similar to the previous convolutions whose result was again max-pooled with 2×2 filters and 25% of the parameters were discarded. This was passed on to the last block composed of 3 fully connected/ dense layers. The first 2 dense layers consisted of 256 and 128 neurons with ReLU activation. The final dense layer consisted of 10 neurons in accordance with the number of instruments and used softmax activation which is presented below.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}, \quad (4)$$

where z is an input vector of length K .

The network was trained for 100 epochs coupled with categorical cross-entropy and adam-based optimization. The network is illustrated in Fig. 6.

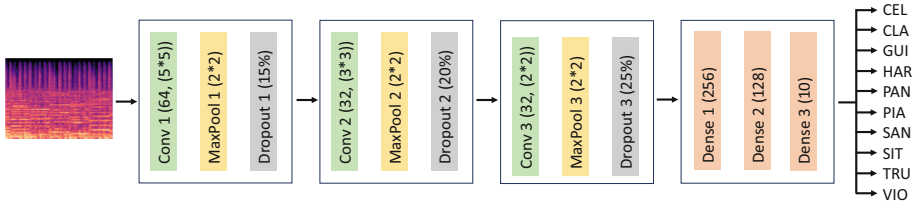


Fig. 6. Architecture of I-NET

4 Results and Analysis

The dataset was subjected to 2 networks wherein 10% of the training data was used for validation. The first was based on the architecture proposed by Solanki et al. [20] and the second (I-NET) was designed to enhance the recognition performance. The networks were trained for the same number of epochs along with the other parameters. However, a higher image size of 256×256 was used for [20] to avoid diminishing inputs. The experiments were repeated 42 times and the highest recognition rate of 88.32% was obtained for [20] whose confusion matrix is presented in Fig. 7. It is noted that the system confused several instances of “Cello” to be “Violin” and “Trumpet”. There were several clips where notes were played across different octaves. In the case of higher octave notes of “Cello” it might have been a reason of confusion with “Violin”. There were certain

instances, where the tone of the “Cello” appeared to be similar to that of the “Trumpet” in the lower octave which was further smoothed by the BGM. This also led to the confusion. There were lead arpeggios that often spanned across octaves. In the case of “Trumpet”, the higher octave seemed to be like that of the “Violin” in short clips which led to several clips of “Violin” being labeled as “Trumpet”.

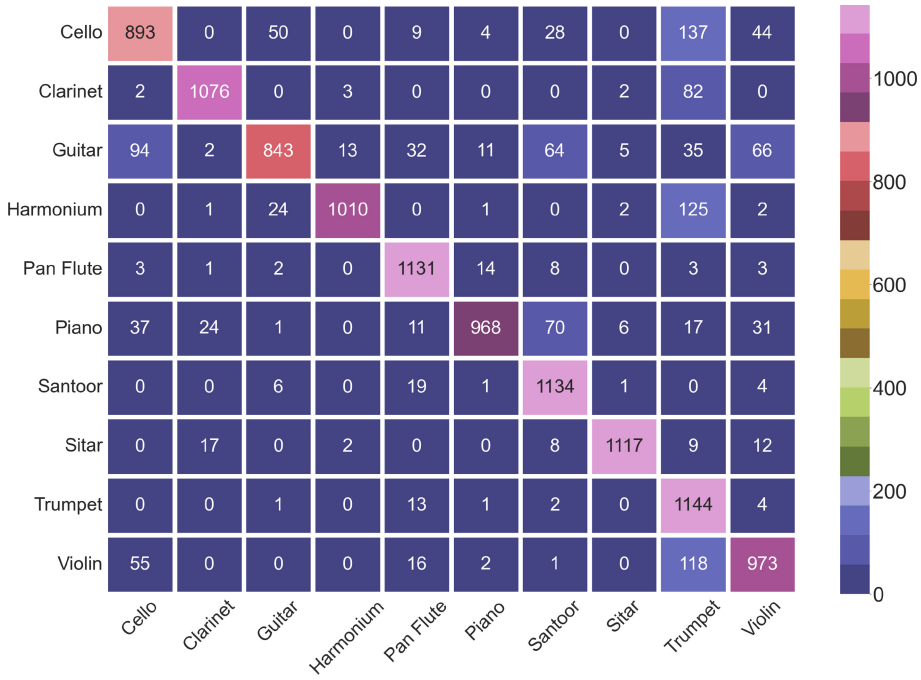


Fig. 7. Confusion matrix for the best result obtained using the architecture based on Solanki et al. [20]

In the case of I-NET, the highest accuracy of 89.73% was obtained whose confusion matrix is presented in Fig. 8. One of the highest false negatives was observed for “Cello” which was classified as “Trumpet”. In the case of the I-NET model, this confusion was reduced by 50.36% over the previous architecture. In this case, similar confusion as that for the model by [20] was observed. In certain instances, the lower octave of the “Harmonium” resembled a “Trumpet” in a clip of short duration. This led to misclassifications of several clips of “Harmonium” as “Trumpet”.

To get a clearer understanding of the results for both systems, the instrument level accuracies for the best performances are presented in Fig. 9. It is observed that out of the 10 instruments, the architecture of [20] performed better for 4 instruments namely “Cello”, “Clarinet”, “Santoor”, and “Trumpet”. The performance was better by 1.01%, 2.97%, 1.70%, and 1.69% for the aforementioned

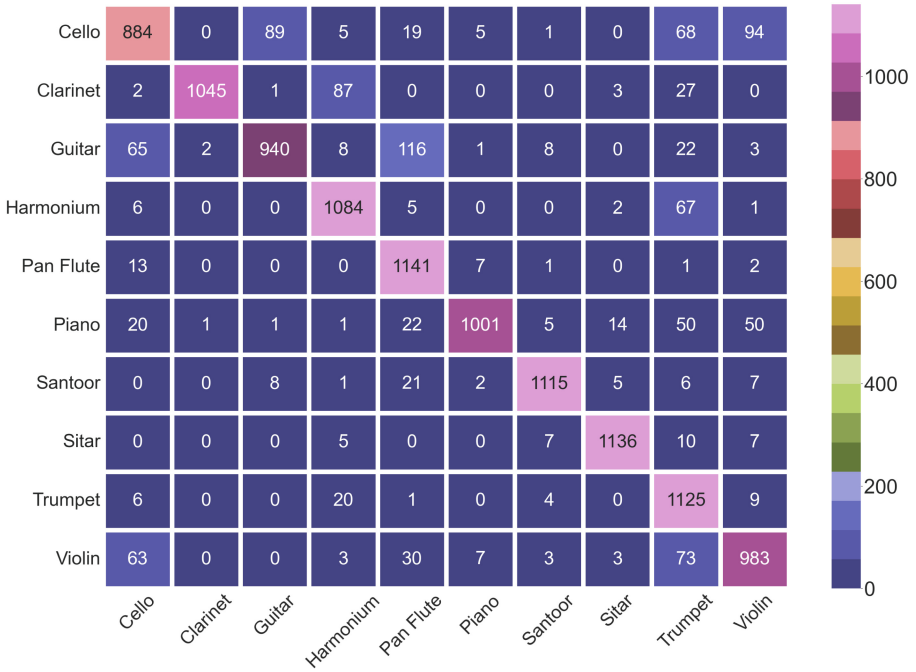


Fig. 8. Confusion matrix for the best result obtained using I-NET architecture

instruments respectively. In the case of the other 6 instruments, I-NET outperformed [20] by 11.51%, 7.32%, 0.89%, 3.41%, 1.70%, and 1.03% for “Guitar”, “Harmonium”, “Pan Flute”, “Piano”, “Sitar”, and “Violin” respectively.

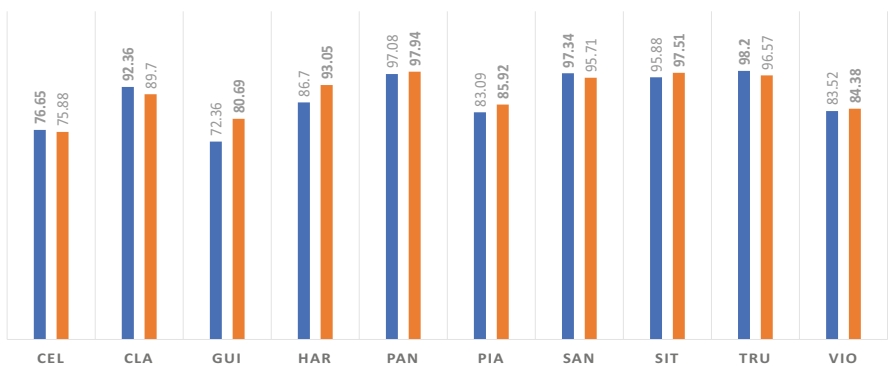


Fig. 9. Instrument-wise accuracies for the best results obtained using [20] (Blue) and I-NET (Red). Bold numbers on top of the bars signify higher values for each instrument. (Color figure online)

The mean accuracies in instrument level for the 2 systems were computed across the 42 runs whose results are presented in Fig. 10. A higher average accuracy of 85.91% was obtained for the I-NET architecture as compared to [20] which produced an average accuracy of 81.73%. On analyzing the instrument level results it was observed that [20] performed better for 4 instruments namely “Pan Flute”, “Santoor”, “Trumpet”, and “Violin”. The performance was better by 3.10%, 2.66%, 5.98%, and 0.71% for the aforementioned instruments respectively as compared to I-NET. In the case of the other 6 instruments, I-NET outperformed [20] by 13.90%, 6.62%, 10.71%, 9.77%, 39.44%, and 0.92% for “Cello”, “Clarinet”, “Guitar”, “Harmonium”, “Piano”, and “Sitar” respectively.

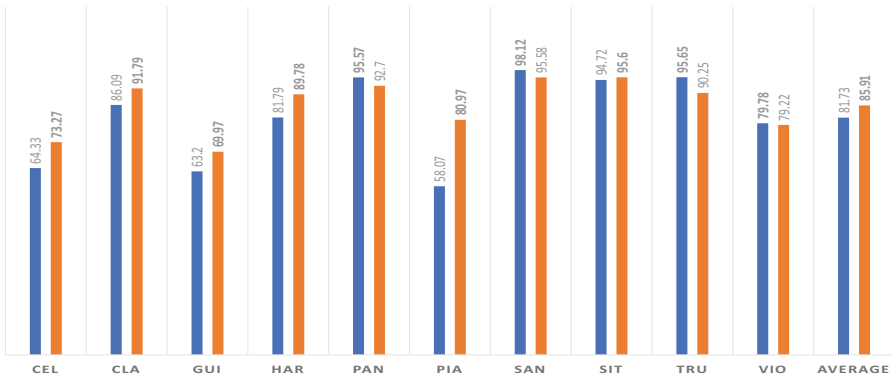


Fig. 10. Average Instrument-wise accuracies obtained using [20] (Blue) and I-NET (Red). Bold numbers on top of the bars signify higher values for each instrument. (Color figure online)

5 Conclusion

In this paper, a new stereo polyphonic dataset for Musical Instrument Identification (EDM10) is introduced. The dataset consists of Electronic Dance Music clips of 3 seconds duration. It also introduces an out-of-mood testing scenario and identical background and notation across the 10 instruments in the dataset. This ensures an unbiased Musical Instrument Identification scenario wherein models cannot get any unfair advantage from other varying characteristics within the different classes. The baseline results were obtained using a custom CNN architecture named I-NET that yielded the highest accuracy of 89.73%. Tests were also performed based on a CNN architecture proposed by [20] which yielded the highest accuracy of 88.32%.

Acknowledgement. This research was carried out on the High-Performance Computing resources at New York University Abu Dhabi. The authors thank Mr. Pradip Ghosh for his input during this work.

References

1. Benetos, E., Dixon, S., Duan, Z., Ewert, S.: Automatic music transcription: an overview. *IEEE Signal Process. Mag.* **36**(1), 20–30 (2018)
2. Blaszkze, M., Kostek, B.: Musical instrument identification using deep learning approach. *Sensors* **22**(8), 3033 (2022)
3. Bosch, J.J., Janer, J., Fuhrmann, F., Herrera, P.: A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In: *ISMIR*, pp. 559–564 (2012)
4. Carnovalini, F., Rodà, A.: A multilayered approach to automatic music generation and expressive performance. In: *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*, pp. 41–48. *IEEE* (2019)
5. Dutta, A., Sil, D., Chandra, A., Palit, S.: CNN based musical instrument identification using time-frequency localized features. *Internet Technol. Lett.* **5**(1), e191 (2022)
6. Ghosh, A., Pal, A., Sil, D., Palit, S.: Music instrument identification based on a 2-d representation. In: *2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, pp. 509–513. *IEEE* (2018)
7. Goto, M., Hashiguchi, H., Nishimura, T., Oka, R.: *Rwc music database: music genre database and musical instrument sound database* (2003)
8. Han, Y., Kim, J., Lee, K.: Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM Trans. Audio Speech Lang. Process.* **25**(1), 208–221 (2016)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **25** (2012)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
11. Lee, C.H., Shih, J.L., Yu, K.M., Lin, H.S.: Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features. *IEEE Trans. Multimed.* **11**(4), 670–682 (2009)
12. Manilow, E., Wichern, G., Seetharaman, P., Le Roux, J.: Cutting music source separation some Slakh: a dataset to study the impact of training data quality and quantity. In: *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. *IEEE* (2019)
13. Mukherjee, H., Marciano, M., Dhar, A., Obaidullah, S.M., Roy, K.: Distinguishing pianos: the difference in similarity. In: *2022 IEEE 7th International Conference for Convergence in Technology (I2CT)*, pp. 1–6. *IEEE* (2022)
14. Mukherjee, H., Obaidullah, S.M., Phadikar, S., Roy, K.: MISNA-A musical instrument segregation system from noisy audio with LPCC-S features and extreme learning. *Multimed. Tools Appl.* **77**(21), 27997–28022 (2018)
15. Muller, M., Ellis, D.P., Klapuri, A., Richard, G.: Signal processing for music analysis. *IEEE J. Sel. Top. Signal Process.* **5**(6), 1088–1110 (2011)
16. Nagawade, M.S., Ratnaparkhe, V.R.: Musical instrument identification using mfcc. In: *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 2198–2202. *IEEE* (2017)
17. Nirozika, K., Thulasiga, S., Krishanthi, T., Ramashini, M., Gamachchige, N.: Automatic Sri Lankan traditional musical instruments recognition in soundtracks. In: *2022 6th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI)*, pp. 1–6. *IEEE* (2022)

18. Oliveira, J.L., Gouyon, F., Martins, L.G., Reis, L.P.: IBT: a real-time tempo and beat tracking system. In: ISMIR, pp. 291–296 (2010)
19. Shen, J., et al.: Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4779–4783. IEEE (2018)
20. Solanki, A., Pandey, S.: Music instrument recognition using deep convolutional neural networks. *Int. J. Inf. Technol.* **14**(3), 1659–1668 (2022)
21. Szeliga, D., Tarasiuk, P., Stasiak, B., Szczepaniak, P.S.: Musical instrument recognition with a convolutional neural network and staged training. *Procedia Comput. Sci.* **207**, 2493–2502 (2022)
22. Toghiani-Rizi, B., Windmark, M.: Musical instrument recognition using their distinctive characteristics in artificial neural networks. arXiv preprint [arXiv:1705.04971](https://arxiv.org/abs/1705.04971) (2017)
23. Uruthiran, P., Ranathunga, L.: Optimization of feature selection and classification of oriental music instruments identification. In: 2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS), pp. 120–125. IEEE (2019)



SONNET: Enhancing Time Delay Estimation by Leveraging Simulated Audio

Erik Tegler^(✉), Magnus Oskarsson, and Kalle Åström

Lund University, Lund, Sweden

{erik.tegler,magnus.oskarsson,karl.astrom}@math.lth.se

Abstract. Time delay estimation or Time-Difference-Of-Arrival estimates is a critical component for multiple localization applications such as multilateration, direction of arrival, and self-calibration. The task is to estimate the time difference between a signal arriving at two different sensors. For the audio sensor modality, most current systems are based on classical methods such as the Generalized Cross-Correlation Phase Transform (GCC-PHAT) method. In this paper we demonstrate that learning based methods can— even based on synthetic data— significantly outperform GCC-PHAT on novel real world data. To overcome the lack of data with ground truth for the task, we train our model on a simulated dataset which is sufficiently large and varied, and that captures the relevant characteristics of the real world problem. We provide our trained model, SONNET (Simulation Optimized Neural Network Estimator of Timeshifts), which is runnable in real-time and works on novel data out of the box for many real data applications, i.e. without re-training. We further demonstrate greatly improved performance on the downstream task of self-calibration when using our model compared to classical methods.

Keywords: Time Delay Estimation · Time-Difference-of-Arrival · Generalized Cross-Correlation · Data Simulation · Audio

1 Introduction

Time Delay Estimation (TDE) is the problem of determining how much later (or earlier) a signal from a transmitter is received at two different receivers. The result is often denoted Time-Difference-Of-Arrival (TDOA), see Fig. 1. TDE is a pivotal problem, primarily due to its critical role in localization and positioning

This work was partially supported by the strategic research project ELLIIT and partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Computational resources were provided by the Swedish National Infrastructure for Computing at C3SE and NSC, partially funded by the Swedish Research Council, grant agreement no. 2018-05973.

systems. By enabling the determination of the TDOA of a signal at different receivers, TDE provides the foundational measurements for inferring the spatial location of senders and/or receivers using further methods. Examples of such methods are:

- Multilateration, where the positions of the receivers are known and the TDOA estimates are used to estimate the position of the sender, [12, 16].
- Direction of arrival, where prior knowledge of the geometry of a receiver array is used together with TDOA measurements from multiple pairs of receiver to compute from which direction the signal is received.
- Self-calibration, where the positions of both receivers and senders are estimated solely based on the measured TDOA, [17, 24, 28].

Accurate localization of sender and receiver nodes is crucial for various applications, including microphone array calibration, speaker diarization, beamforming, radio antenna array calibration, mapping, and positioning [20].

In these and many other applications, the initial signal processing step of obtaining reliable TDOA estimates, plays an important role. Currently, the state-of-the-art method is the Generalized Cross-Correlation Phase Transform (GCC-PHAT) and its variants [15]. However, recent research indicates that there is substantial room for improvement. In [30] it was shown that the average performance of existing methods fell below the desired threshold in nearly 40% of estimations based on a real dataset with ground truth.

The TDE problem is relevant across multiple different signal modalities such as audio and radio. However, in this paper the main focus is directed towards the analysis of audio, although studying radio signals is an interesting subject for future studies.

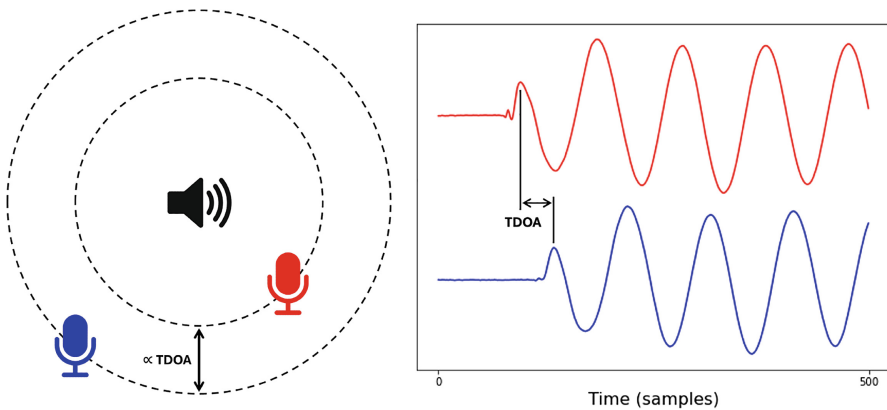


Fig. 1. Since the microphones are at different distances from the speaker, the signal arrives at different times for each of them. By estimating the timeshift in the signals (right figure), and combining it with the propagation speed of the signal, we get a measurement of distance difference (left figure)

1.1 Challenges

TDE in the audio domain presents a set of challenges that significantly complicates the estimation process. Unlike scenarios involving controlled signal transmission, for a variety of audio applications the sound source is not a controlled entity. Therefore our estimation techniques need to handle unknown signals with unpredictable characteristics. This difficulty is compounded by reverberations, a common phenomenon in acoustic spaces where sound waves reflect off surfaces, creating multiple delayed echoes that can obscure the true signal path. Another challenging aspect arises when dealing with moving sound sources, as this introduces a dynamic element to the TDE problem [7]. As the source moves, the relative distances to the receivers change continuously, altering the TDOA in real-time and demanding adaptive estimation techniques capable of handling these variations. Together, these factors-unknown signal characteristics, reverberations, and source mobility-make audio-based TDE a particularly demanding task, necessitating sophisticated algorithms and approaches to achieve reliable estimation.

1.2 Related Works

Previous methods for TDE in audio signal processing exhibit notable limitations in handling complex real-world scenarios. The Generalized Cross-Correlation method, and specifically its Phase Transform variant GCC-PHAT is often used as a starting point. It is robust to measurement noise and works well across diverse signal types. However, GCC-PHAT shows limitations when dealing with reverberations and moving sound sources.

In [7] methods were developed for estimation of TDOA with sound source or receiver motions. These methods were based on local optimization of initial estimates based on the GCC-PHAT.

Recent advancements have seen the adoption of machine learning techniques for TDE. For small baseline receiver arrays, where the receivers are typically placed equidistant, direction of arrival can be made using alternative techniques, for example steered-response power with phase transform [4], spectrograms [26] and raw waveforms [13]. For DOA estimation both traditional and data driven methods have been used successfully.

For large baseline arrays, where the receivers are placed in an ad-hoc fashion there has been some attempts at data-driven methods, see for example [6, 8–10, 25] and [11] for an extensive overview. Many of these are, however, trained and evaluated for specific subtasks. One of the problems has been the lack of real world data with accurate ground truth. Previous work circumvents this by using simulated data to train their models. While this approach is promising, these papers make no claim to have a model which works on novel real world data. Instead, only training and evaluating on similar simulated datasets.

1.3 Contribution

In this paper we propose more careful modeling of the problem, and thereby improving over previous learning methods by increasing the scope and quality of the simulations. Many learning based approaches focus primarily on reverberation effects, neglecting other critical factors. Thus, limiting the generalizability of these methods to real data. This paper aims to contribute to the body of knowledge on TDE by leveraging simulated data to explore and enhance estimation techniques for audio signals. To summarize, our contributions are:

- Demonstrating that data driven models trained on large scale simulated sound datasets, generalize to real data as well as to novel sounds for the TDE task.
- Providing a model, SONNET - Simulation Optimized Neural Network Estimator of Timeshifts, which outperforms state of the art methods for TDE and is evaluated on both simulated and real data.¹
- Demonstrating how the new estimators improve performance on downstream tasks.

2 Problem Setup

Consider a reverberant room containing two receivers positioned at $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{R}^3$ and a moving sender located at $\mathbf{s}(t) \in \mathbb{R}^3$. The sender is emitting an unknown signal $x(t)$ which is being recorded by the receiver at \mathbf{r}_i as the signal $x_i(t)$. The TDOA at time t for receivers i and j , is defined as

$$\Delta(t) = \frac{\|\mathbf{r}_i - \mathbf{s}(t)\| - \|\mathbf{r}_j - \mathbf{s}(t)\|}{v_x}, \quad (1)$$

where v_x is the propagation speed of the signal. We are in this paper primarily interested in this direct path TDOA, but an interesting extension would be to also consider TDOA measurements corresponding to multi-path components from reflective planes, that could potentially provide richer information, [5, 29].

For the modeling we assume that the received signal $x_i(t)$ can be modeled as

$$x_i(t) = \int h_i(t - \tau, \tau)x(\tau)d\tau + \epsilon_i(t), \quad (2)$$

where ϵ is the noise and $h_i(t, \tau)$ is the impulse response from the sender to receiver i at the position $\mathbf{s}(\tau)$. The impulse response captures the acoustic properties (position, orientation) of both the receiver and sender as well as the reverberant properties of the room. Typically there is a strong direct path component in the impulse response, corresponding to a time-delay of $\frac{\|\mathbf{r}_i - \mathbf{s}\|}{v_x}$, which allows for the TDOA estimation. While the TDOA is time dependent, we will for the rest of this paper refer to the TDOA of a pair recorded signals $x_i(t), x_j(t), 0 < t < T$ as the TDOA value at the middle of the signal, i.e. $\Delta(\frac{T}{2})$. The goal is to use two recorded signals x_i, x_j to estimate this TDOA value.

¹ Code available at: <https://vision.maths.lth.se/sonnet/>.

3 Data Simulation

Similar to earlier work by Berg et al. [2], we use Pyroomacoustics [22] to compute impulse responses using the image source method [1]. However, we augment the simulation in a number of important ways. Instead of simulating a single room we simulate a broader class of rooms in order to cover a larger set of possible impulse responses. Also, with the goal of making the simulation better reflect reality, we both simulate a moving sound source and also microphones and sound sources which are not omnidirectional. How and why will be explained in more detail in the following sections and motivated by our ablation study in Sect. 6.5.

3.1 Moving Sound Source

To simulate a moving sound source we first generate a path by constructing a quadratic Bézier curve $s(t)$, $0 < t < T$ with a length shorter than some maximum length. Because simulating the sound from a moving sound source is difficult within the Pyroomacoustics framework, we instead discretize the curve into k points

$$\{s(t_1), \dots, s(t_k)\}, \quad t_i = \frac{i-1}{k-1}T. \quad (3)$$

The sound from a moving sound source is then approximated by dividing the played sound $x(t)$ into k equally sized parts and simulating part $x(t)$, $\frac{i-1}{k}T < t < \frac{i}{k}T$ as a stationary speaker at point $s(t_i)$. This is essentially simulating that the sound source is jumping to a new location along a path after each time $\frac{T}{k}$. Following this methodology, the signal is computed as a sum of convolutions

$$x_i(t) = \sum_{j=1}^k h_i(t, \frac{j}{k}) * \bar{x}^{(j)}(t), \quad (4)$$

where $\bar{x}^{(j)}$ is the j th part of the signal, zero-padded to have the same shape as the original signal x .

3.2 Directionality

Previous work simulated both microphones and speakers as being omnidirectional, which means they emit/receive their signal equally well in all directions. However, the directional dependence in reality is rather complicated, since it depends on what hardware is used. This is further complicated by the directionality of a microphone not being constant for all frequencies, as demonstrated in [27]. We settled on using the subcardioid sensitivity pattern, since it is a common model for directional microphones and therefore already implemented in Pyroomacoustics [3].

4 Inference Model

One of the main contributions of this paper is to show that it is possible to close the sim2real gap, i.e. to demonstrate that simulated data generalizes to real data, given enough simulations of sufficiently rich character. Therefore we opted for a network architecture with the properties of simplicity and being easily trainable, see Fig. 2. The main part of our network is a ResNet, since it is a simple architecture which is easily trainable. However, an issue with audio data is that it has low information density, making it difficult to input the audio data directly into the ResNet, without significantly increasing the number of parameters of the model. We therefore use two common approaches to compress the audio data.

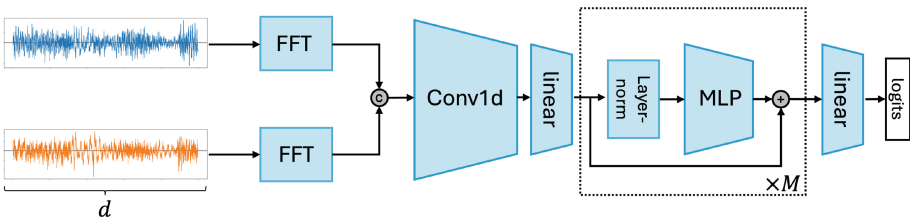


Fig. 2. System overview: Our model takes two audio recordings of length d as input data. The data is first converted to the frequency domain, using the fast Fourier transform, and stored with real and imaginary components as different channels. It is then sent through a series of 1d convolutional layers. The features are then processed using M stacked pairs of linear layers along with skip connections. Finally, the logits are acquired by adding a linear layer after the last residual block

First we use a fast Fourier transform and only store the values for frequencies which are below some threshold frequency f_{max} . Another advantage of using the Fourier transform is that it mitigates the problem of the spectral bias in neural networks [21]. Secondly, we use a backbone of 1d convolutional layers to extract more dense features from the data.

In the same manner as [2] our model performs regression-via-classification (RvC). This means that our network does not output TDOA values, but rather outputs logits for a fixed number of classes. Each of the classes then corresponds to a range of TDOA values.

To make the data have the right size we use linear layers as projections between the backbone and the ResNet, as well as a linear classifier between the ResNet and the final logits.

5 Implementation Details

In our code we have provided a pretrained model SONNET. In this section we have outlined implementation details on both how SONNET was trained, as well as how the dataset it was trained on was generated.

The sound used when training the model are from the *Musan* dataset [23], which is an openly available corpus containing about 18 GB of music, speech and noise. For our simulations we used a signal length of $d=10,000$ samples with a sampling frequency of 16 kHz. However, since we want reverberations to be present at the beginning of the recording we simulated 2000 extra samples at the beginning of each simulation. For each recording, we simulated a new room in the shape of a rectangular cuboid with each of the three dimensions having a length uniformly sampled in the interval $[1, 10]$ m. The reverberation level in the room was varied by sampling the reflection coefficient of the walls from the interval $[0.05, 0.99]$. The path of the sender was simulated in one of two different ways with equal probability, either as a stationary point source or as a randomly sampled quadratic Bézier curve with maximum velocity of 5 m/s. In each room we simulated 50 microphones recording the signal.

The dataset consist of 10,000 rooms, which means that it in total contains $10,000 \binom{50}{2} = 12$ million training examples of pairs of recordings. The memory footprint of the dataset is 19 GB.

For the model we set our threshold frequency $f_{max} = 4800$ Hz. The backbone consisted of three 1d convolutional layers. The ResNet consisted of $M = 4$ blocks. Throughout the network we used GELU [14] as activation function. We choose to have 1000 output classes as possible predictions for our model. Each class corresponded to bins of TDOA values with a width of 1 sample. This means our model makes predictions with the same resolution as GCC-PHAT.

The model was trained in PyTorch [19] with the AdamW optimizer [18], a batch size of 4096, a learning rate of 0.0003, during 20 epochs. We used the cross-entropy loss with label-smoothing of 0.1 as our loss function. The training was done on Tesla V100-PCIE-16GB GPU and took 3 h.

6 Experiments

In this section we first analyze the inference speed and memory footprint of the proposed system. We then study the performance of the system on both simulated and real data, by making several comparative studies against GCC-PHAT. Finally we show how the proposed system can be used to improve a previous state-of-the-art system for automatic self-calibration of an ad-hoc configuration of microphones. We quantitatively and qualitatively (see Figs. 3, 4 and 5) show that our model outperforms previous methods for performing TDE in novel real world settings.

6.1 Inference Speed and Memory Footprint

The ideas in this paper can be used train TDE models of different sizes, and can therefore be tailored to the available memory and computation requirements for a specific use case. However, we suggest as a starting point to use SONNET, which we have provided along with this paper. SONNET has 20 million parameters and a memory footprint of 75 MB.

Inference speed was evaluated on both CPU, Intel(R) Xeon(R) W-2125 CPU @ 4.00 GHz, and on a GPU, Tesla V100-PCIE-16GB. The results are shown in Table 1. To summarize, SONNET takes four times longer to run than GCC-PHAT, however, the inference time is still fast enough to run SONNET on a CPU in real-time without any issues.

Table 1. Computation time per pair using a batch size of 100 recording pairs

	SONNET (ms)	GCC-PHAT (ms)
CPU	0.94	0.32
GPU	0.022	0.005

6.2 Noise and Reverberation Sensitivity (simulated Data)

We have also evaluated our model’s robustness to noise and reverberation using simulated data, see Fig. 3. We use accuracy at 10 cm as our main evaluation metric. To motivate this, we would like to highlight the distribution of the residuals. The residual distribution for all three detectors shown here seems to be well explained by a combination of a normal distribution (inliers) and a uniform distribution (outliers), a common combination of distributions in the area of robust estimation. Because of this, using mean squared error as an evaluation metric is both noisy and highly dependent on the space of values the model can estimate making comparisons between models more difficult. When using these detections for downstream tasks, because they contain outliers one probably need to use methods from robust estimation. Because of this, we think reporting inlier ratio, at a given inlier threshold, is a better evaluation metric.

The evaluation examples are created in the same manner as the training data, with the change that instead of using audio from *Musan* we used the audio from *tdoa_20201016* (described in Sect. 6.3) in the simulation. Our model outperforms GCC-PHAT in a wide range of reverberant and noise environments, as shown in Fig. 3.

6.3 Real Data

Arguably, the most important evaluation of our model is the results on real world data. We have evaluated our model on the *tdoa_20201016* dataset provided by [30]. The advantage of using this dataset is that it contains ground truth values for the TDOA for any pair of two microphones. The dataset also contains recordings without accompanying ground truth but these were not used in our evaluation. The dataset is recorded in 96 kHz, which means that we have to down-sample, since our model is trained on 16 kHz. The total playing time of the speaker over all the experiments is around 600 s with 12 microphones recording.

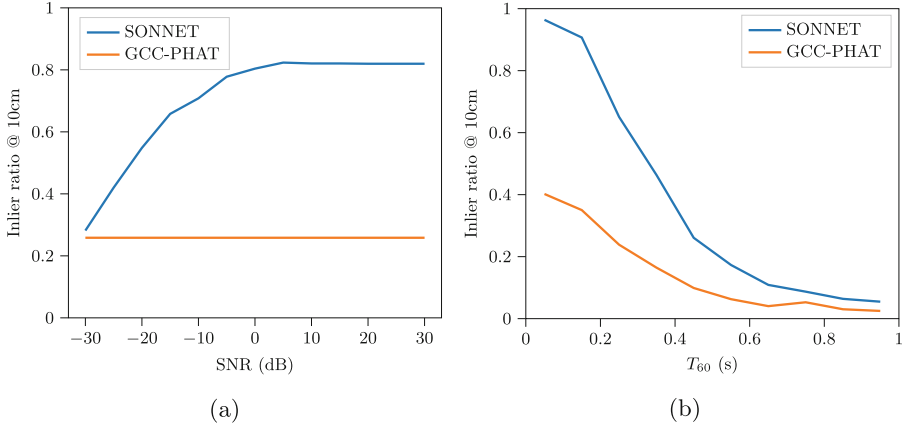


Fig. 3. Results on the simulated data. **(a)** Noise sensitivity evaluated at $T_{60} = 0.2$ s. Note that GCC-PHAT is very robust against white noise **(b)** Reverberation sensitivity evaluated at SNR = 10 dB

Using a window overlap of $5/6$, we get 384648 pairs of windows to estimate TDOA on. As shown in Fig. 4, the learned models significantly outperforms GCC-PHAT. We also show qualitative results over some of the recordings in the dataset in Fig. 5.

6.4 Downstream Application

Since the main reason for studying TDE is its use in downstream applications, we have evaluated our models on the task of self-calibration using the [30] dataset. In self-calibration the goal is to estimate the 3D geometry of both the receivers and senders using only the TDOA values as input, i.e. no prior position information.

To do this, we used the TDOA values acquired as input to a published self-calibration system [17, 30]. We then compare our estimated 3D positions with the ground truth positions provided in the dataset. In order to be able to do this comparison, we need to fix the gauge freedom in the solution, in this case the solution and ground truth might differ by a Euclidean transformation. We estimate this Euclidean transformation using the receiver positions of our solution compared to the ground truth receiver positions. This is similar to how to evaluate maps found using Structure-from-Motion (SfM) or Simultaneous Localization And Mapping (SLAM). After applying the Euclidean transformation to the solution, the residuals are then computed as the distances between corresponding receivers in the solution and ground truth.

As can be seen in Table 2, using the TDOA values from our learned models makes the self-calibration system converge to good solutions on all of the experiments. This is a significant improvement compared to using GCC-PHAT for which the system only manages to converge on some of the experiments and even when it converges it has larger errors. An example of a 3D reconstruction

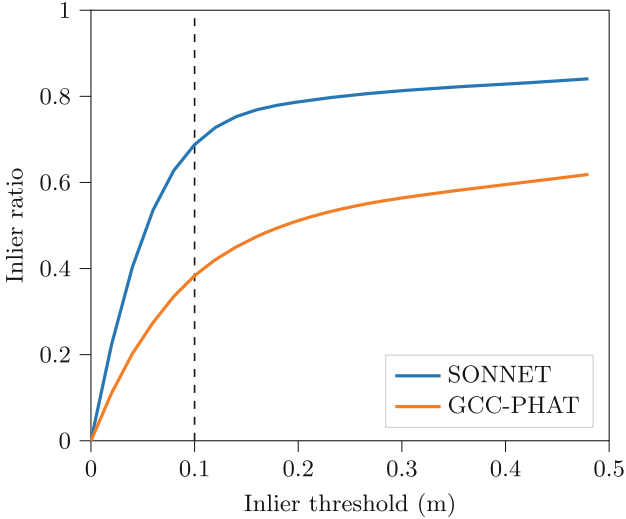


Fig. 4. Quantitative results on the dataset *tdoa_20201016* showing the probability of correct detection at different inlier thresholds. We have marked the 10 cm threshold which we use as our main evaluation metric

Table 2. RMS error of the receivers for the estimated 3D geometry after registration to the ground truth. Estimation is done using TDOA values from SONNET or GCC-PHAT. Experiments missing a value have an error larger than 1 m

Experiment	SONNET (m)	GCC-PHAT (m)
chirp_0001	0.05	0.80
chirp_0002	0.05	0.17
chirp_0004	0.05	0.40
iregchirp_0006	0.06	0.54
iregchirp_0007	0.04	0.59
music_0008	0.07	–
music_0009	0.06	–
music_0010	0.04	0.31
music_0011	0.05	–
music_0012	0.04	–
music_0013	0.03	0.35
music_0014	0.04	0.28
music_0015	0.04	0.10
metronom_0021	0.16	–
metronom_0022	0.10	–
median	0.05	0.59

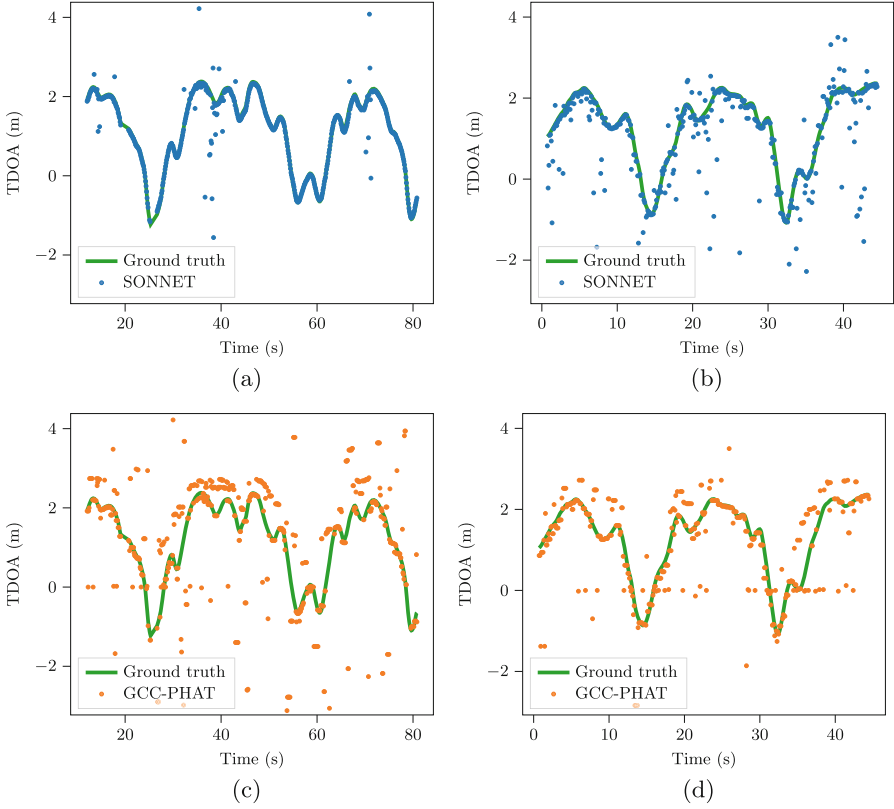


Fig. 5. Qualitative results of the estimated TDOA values on the dataset *tdoa_20201016*. (a) and (c) correspond to the recording *music_0014* while (b) and (d) correspond to *chirp_0001*. The microphone paired used for all four plots are microphone 1 and microphone 6. SONNET significantly outperforms GCC-PHAT when music is played while also achieving a performance gain when chirp sounds are played

resulting from using the learned model together with the self-calibration system can be seen in Fig. 6.

6.5 Ablation Studies

Earlier works which use simulated data to train models to solve TDE have not been demonstrated to generalize to real world data. We claim that we can achieve a good generalization to novel real world data by: scaling the dataset, simulating sound source with movement and directionality. To show that these three changes are helpful we have performed two ablation studies.

In the first ablation study we changed how the dataset was generated by including or excluding the simulation augmentations: moving the sound source or directionality of the sound source. For each of the four configuration we trained

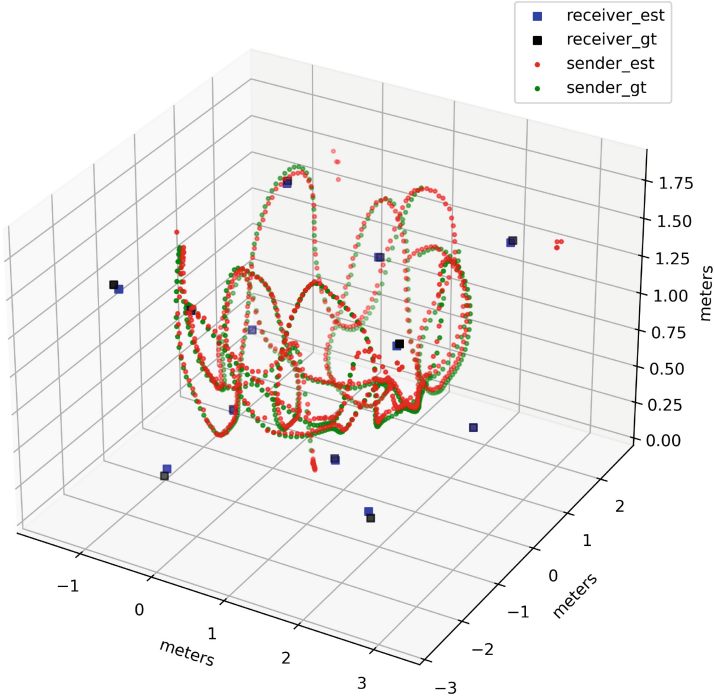


Fig. 6. Example of 3D reconstruction, on the experiment *music_0014*

a separate model in the same way as the full SONNET model. Each of the models were then evaluated on *tdoa_20201016* in the same way as in Sect. 6.3, and the results are shown in Fig. 7a. As we can see, using learning based methods on stationary omnidirectional data outperforms GCC-PHAT. However, we can further improve the method by augmenting the simulation.

For the second ablation study, we trained models on different sizes of the training dataset. The models were then evaluated on *tdoa_20201016* in the same way as in Sect. 6.3, the results are shown in Fig. 7b. As we can see, having a large enough dataset is important for generalization and scaling up the dataset might be a way to improve the model further.

7 Conclusions

As we have demonstrated in this paper, combining the ability to simulate data with learning based methods is a promising direction for further studies. In this paper we have shown that it is possible to improve on TDE, a key task when performing audio based localization. However, using simulation together with learning based methods is also a ripe area for further studies, since it enables an approach to harder versions of the TDE problem. Such examples include using multiple sound sources, multipath components of the sound, or utilizing

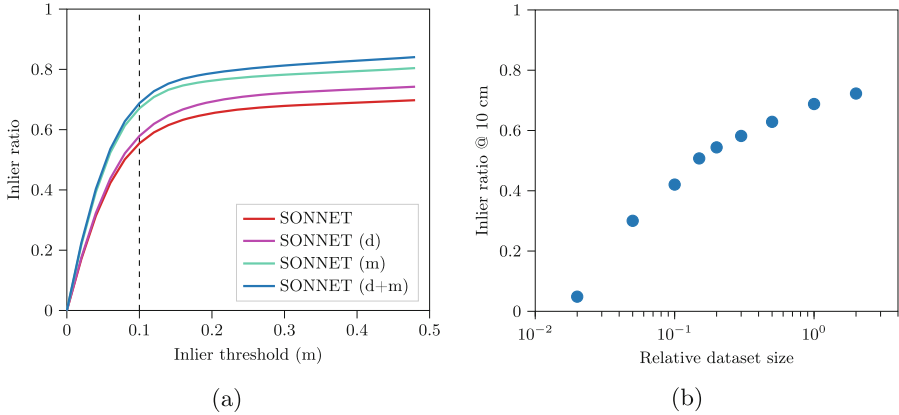


Fig. 7. Results from the ablation studies. **(a)** Ablation study on the effect of introducing the simulation augmentations: sound source movement (m) and directionality (d). Introducing sound source movement gives a larger performance gain. **(b)** Ablation study on the effect of the size of the simulated training dataset, when model is evaluated on the real data from Sect. 6.3. The size of the dataset is given in relative sizes to the dataset SONNET was trained on

the information from more than two microphones at the same time. Since we can simulate such data with ground truth, it might be possible to create detectors for such problems. It is our belief that this paper is a key stepping stone for further studies in the area.

References

1. Allen, J.B., Berkley, D.A.: Image method for efficiently simulating small-room acoustics. *J. Acoust. Soc. Am.* **65**(4), 943–950 (1979)
2. Berg, A., O’Connor, M., Åström, K., Oskarsson, M.: Extending GCC-PHAT using shift equivariant neural networks. In: *Proceedings of the Interspeech 2022*, pp. 1791–1795 (2022). <https://doi.org/10.21437/Interspeech.2022-524>
3. De Sena, E., Hacıhabiboğlu, H., Cvetković, Z.: A generalized design method for directivity patterns of spherical microphone arrays. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 125–128. IEEE (2011)
4. Diaz-Guerra, D., Miguel, A., Beltran, J.R.: Robust sound source tracking using srp-phat and 3D convolutional neural networks. *IEEE/ACM Trans. Audio Speech Lang. Process.* **29**, 300–311 (2020)
5. Dokmanić, I., Daudet, L., Vetterli, M.: How to localize ten microphones in one finger snap. In: *2014 22nd European Signal Processing Conference (EUSIPCO)*, pp. 2275–2279. IEEE (2014)
6. Feng, L., Gong, Y., Zhang, X.L.: Soft label coding for end-to-end sound source localization with ad-hoc microphone arrays. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE (2023)

7. Flood, G., Heyden, A., Åström, K.: Stochastic analysis of time-difference and doppler estimates for audio signals. In: De Marsico, M., di Baja, G.S., Fred, A. (eds.) ICPRAM 2018. LNCS, vol. 11351, pp. 116–138. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05499-1_7
8. Gong, Y., Liu, S., Zhang, X.L.: End-to-end two-dimensional sound source localization with ad-hoc microphone arrays. In: 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1944–1949. IEEE (2022)
9. Grinstein, E., Brookes, M., Naylor, P.A.: Graph neural networks for sound source localization on distributed microphone networks. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE (2023)
10. Grinstein, E., Neo, V.W., Naylor, P.A.: Dual input neural networks for positional sound source localization. *EURASIP J. Audio Speech Music Process.* **2023**(1), 32 (2023)
11. Grumiaux, P.A., Kitić, S., Girin, L., Guérin, A.: A survey of sound source localization with deep learning methods. *J. Acoust. Soc. Am.* **152**(1), 107–151 (2022)
12. Gustafsson, T., Rao, B.D., Trivedi, M.: Source localization in reverberant environments: modeling and statistical analysis. *IEEE Trans. Speech Audio Process.* **11**(6), 791–803 (2003). <https://doi.org/10.1109/TSA.2003.818027>
13. He, Y., Markham, A.: Sounddoa: learn sound source direction of arrival and semantics from sound raw waveforms. In: Interspeech, pp. 2408–2412 (2022)
14. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint [arXiv:1606.08415](https://arxiv.org/abs/1606.08415) (2016)
15. Knapp, C., Carter, G.: The generalized correlation method for estimation of time delay. *Acoust. Speech Signal Process. IEEE Trans.* **24**(4), 320–327 (1976). <https://doi.org/10.1109/TASSP.1976.1162830>, <https://www.ee.iitb.ac.in/course/~sachinnayak/finalpaper2.pdf>
16. Larsson, M., Larsson, V., Åström, K., Oskarsson, M.: Optimal trilateration is an eigenvalue problem. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5586–5590 (May 2019). <https://doi.org/10.1109/ICASSP.2019.8683355>, iSSN: 2379-190X
17. Larsson, M., Flood, G., Oskarsson, M., Åström, K.: Fast and robust stratified self-calibration using time-difference-of-arrival measurements. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2021)
18. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101) (2017)
19. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 32, pp. 8024–8035. Curran Associates, Inc. (2019). <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
20. Plinge, A., Jacob, F., Haeb-Umbach, R., Fink, G.A.: Acoustic microphone geometry calibration: an overview and experimental evaluation of state-of-the-art algorithms. *IEEE Signal Process. Mag.* **33**(4), 14–29 (2016)
21. Rahaman, N., et al.: On the spectral bias of neural networks. In: *International Conference on Machine Learning*, pp. 5301–5310. PMLR (2019)
22. Scheibler, R., Bezzam, E., Dokmanić, I.: Pyroomacoustics: a python package for audio room simulation and array processing algorithms. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 351–355. IEEE (2018)

23. Snyder, D., Chen, G., Povey, D.: MUSAN: a music, speech, and noise corpus (2015). [arXiv:1510.08484v1](https://arxiv.org/abs/1510.08484v1)
24. Tegler, E., Larsson, M., Oskarsson, M., Åström, K.: Sensor node calibration in presence of a dominant reflective plane. In: 30th European Signal Processing Conference, EUSIPCO 2022 - Proceedings, pp. 1941–1945. European Signal Processing Conference, European Signal Processing Conference, EUSIPCO (2022). 30th European Signal Processing Conference, EUSIPCO 2022 ; Conference date: 29-08-2022 Through 02-09-2022
25. Vera-Diaz, J.M., Pizarro, D., Macias-Guarasa, J.: Towards end-to-end acoustic localization using deep learning: from audio signals to source position coordinates. *Sensors* **18**(10), 3418 (2018)
26. Wang, Y., Yang, B., Li, X.: FN-SSL: full-band and narrow-band fusion for sound source localization. arXiv preprint [arXiv:2305.19610](https://arxiv.org/abs/2305.19610) (2023)
27. Zetterqvist, G., Gustafsson, F., Hendeby, G.: Using received power in microphone arrays to estimate direction of arrival. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE (2023)
28. Zhayida, S., Andersson, F., Kuang, Y., Åström, K.: An automatic system for microphone self-localization using ambient sound. In: 2014 22nd European Signal Processing Conference (EUSIPCO). IEEE (2014)
29. Zhayida, S., Rex, S.S., Kuang, Y., Andersson, F., Åström, K.: An automatic system for acoustic microphone geometry calibration based on minimal solvers. arXiv preprint [arXiv:1610.02392](https://arxiv.org/abs/1610.02392) (2016)
30. Åström, K., Larsson, M., Flood, G., Oskarsson, M.: Extension of time-difference-of-arrival self calibration solutions using robust multilateration. In: 29th European Signal Processing Conference (EUSIPCO) (2021). <https://doi.org/10.23919/EUSIPCO54536.2021.9616051>



The Effect of Lung Volume on Glottal Parameters: An Empirical Study

Gauri Deshpande^{1,2(✉)} and Björn W. Schuller^{2,3,4}

¹ TCS Research, Pune, India

² Chair EIHW, University of Augsburg, Augsburg, Germany
gauri1.d@tcs.com, schuller@tum.de

³ CHI, MRI, Technical University of Munich, Munich, Germany

⁴ GLAM, Imperial College London, London, UK

Abstract. In natural continuous speech, speakers articulate at both Inspiratory Reserve Volume (IRV) and Expiratory Reserve Volume (ERV), commonly referred to as high and low lung volumes respectively. This paper aims to investigate whether glottal parameters of a speaker portray the variations in their corresponding lung volumes during vowel articulation. It also seeks to ascertain the extent to which each parameter conveys this information. To capture these parameters, we employ two Glottal Closure Instant (GCI) detection algorithms: Speech Event Detection using the Residual Excitation And a Mean-based Signal (SEDREAMS) and Single Frequency Filtering (SFF). These observations are made while participants are engaged in continuous reading of a phonetically balanced paragraph over a span of 2–3 min. Our research delves into examining how lung volume influences glottal parameters among 94 speakers as they articulate vowels with the first formant exceeding 700 Hz and the second formant below 2500 Hz. Among a vector of 24 glottal parameters, three consistently show higher values during vowel articulation at high lung volume. Leveraging these three parameters, we can accurately differentiate between high and low lung volume in vowel articulation among 77% of the speakers.

Keywords: Glottal Patterns · Speech-Breathing Patterns · Acoustics Patterns

1 Introduction

The process of speech production, particularly in vowel articulation, is closely intertwined with respiration. Respiration entails a continuous cycle of inhalations and exhalations. Inhalation entails the expansion of the lungs, while exhalation involves their contraction. There are four recognised standard lung volumes: 1) Tidal volume: This is the amount of air that enters the lungs during normal breathing at rest. 2) Inspiratory Reserve Volume (IRV): This refers to the additional volume of air that can be inhaled beyond the normal tidal volume during

maximum inspiration. 3) Expiratory Reserve Volume (ERV): This is the extra volume of air that can be exhaled beyond the normal tidal volume during maximum expiration. 4) Residual Volume: This is the volume of air that remains in the lungs even after maximal expiration when the smallest airways are closed. In natural speech, individuals often seamlessly phonate during both high lung volume (IRV) and low lung volume (ERV) phases. Figure 1 illustrates the time-synchronised speech and breathing pattern of a speaker reading a phonetically balanced text. As observed in the figure, the speaker pronounces vowels both at inhalation peaks and during long exhalations. Inhalation peaks signify expanded lungs with a high volume of air, while the exhalation phase denotes the contraction of the lungs to expel air, indicating a low volume of air. As the speaker engages in speech production while reading, these phases exceed or fall short of tidal volume capacity and are thus termed IRV and ERV respectively.

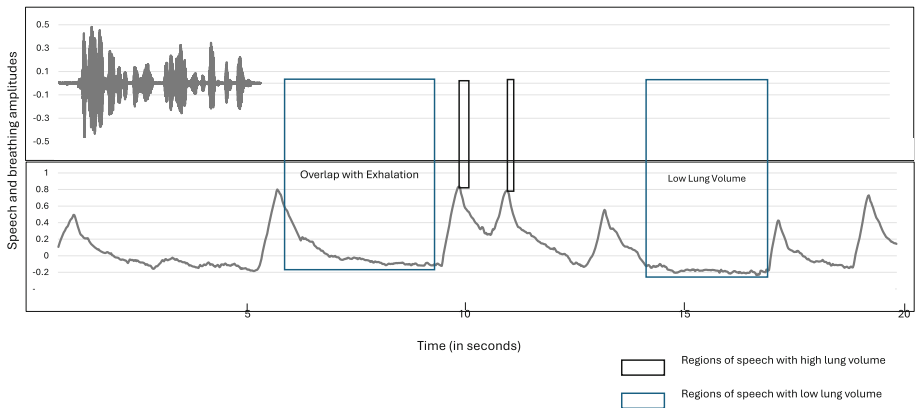


Fig. 1. The time-synchronised speech signal (above) and corresponding breathing pattern (below) depict a speaker reading a phonetically balanced text. This illustrates that the speaker utters vowels with both high and low breathing values, referring to high and low lung volumes, respectively

As outlined in [12] and [7], speech-breathing represents a cognitive phenomena. The natural manner in which individuals speak and breathe provides insight into their approach to planning breathing during speech. This, in turn, reflects underlying bio-psychological states such as emotions [6], confidence [3], detection of Parkinson’s disease [13] and pulmonary health. Therefore, it is crucial to comprehend the effects of fluctuating lung volume during continuous speech on vowel production.

1.1 Previous Work

Previous studies aimed to explore the effects of lung volumes typically involved subjects speaking deliberately with both high and low lung volumes. This type of

speech is termed “acted speech,” designed to simulate specific conditions, such as varying lung volume levels. However, these conditions may not accurately replicate those found in natural continuous speech, such as reading a paragraph or engaging in conversation, which are common real-world scenarios.

In a study conducted by Hoit et al. in [8], subjects are asked to consciously pronounce syllables during high and low lung volumes. It is reported that the voice onset time is longer at high lung volumes. Similarly, Iwarsson et al. report that the glottal adduction is shorter at higher lung volume in [10]. Further to this in [9], Iwarsson et al. reported a higher sub-glottal pressure, lower larynx, and higher closed quotient for higher lung volumes. Winkworth et al. in [14] observed higher lung volume for louder phonation and for those pronounced during sentence and paragraph boundaries. All these observations require a representation technique using statistical measures validated empirically on a larger population.

Below we highlight a comparison of earlier studies with ours with respect to number of subjects, methodology for collecting and analysing speech, and the parameters extracted:

1. Number of subjects: We have conducted our experiments on 94 subjects as compared to 5, 5, 6, and 24 subjects based experimental results presented in [8–10, 14] respectively.
2. Speech Types: In our experiments, the speech data is collected while participants read a phonetically balanced paragraph, ensuring natural variations in lung volume without any intentional alterations. In contrast, the studies by [8, 9], and [10] involved subjects intentionally producing speech with high and low lung volumes, which does not reflect a natural setting.
3. Measurement Approach: Higher lung volume is identified by longer voice onset time in [8] using hard-copy wide-band spectrograms. Iwarsson et al. In [10], and [9] measured the parameters employing several sensors such as: thin plastic tube based flow-mask, electroglottogram (EGG), and Glottal Enterprises filter (inverse filter). In [14] Winkworth et al. performed manual inspection to study the time-synchronous speech and respiratory signal. They paid special attention to aligning the speech and respiratory signals on the screen, discarding any data that did not meet the required alignment criteria. In contrast, our approach proposes extracting these parameters using data from a single sensor: speech.
4. Parameters : Voice onset time (VoT) is studied in [8] where a longer VoT is observed at higher lung volume. In [10], and [9], decreased glottal closure duration, and an increase in the values of sub-glottal pressure, peak-to-peak flow amplitude, and glottal leakage are observed at higher lung volume. Winkworth et al. observed higher lung volumes for louder phonation at sentence and paragraph boundaries. Our analysis includes 24 parameters extracted from speech signals using the corresponding glottal closure waveform. This set of parameters not only encompasses those from previous studies but also includes additional statistical measures. Additionally, we identify the

top three parameters that effectively differentiate between high and low lung volumes from speech signals collected from 94 individuals.

2 Current Study

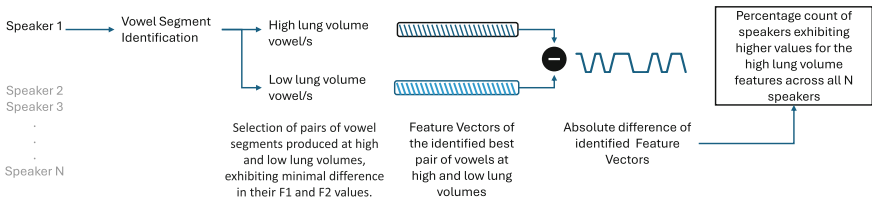


Fig. 2. The methodology employed to investigate the influence of lung volume on vowel production, utilising glottal parameters for a speaker, consists of the following stages: 1) vowel segment identification, 2) Determination of pairs of vowel segments pronounced at high and low lung volumes, with minimal discrepancy in their F1, F2, and F3 values, 3) Extraction of glottal parameters for the identified pair, and 4) Analysis of prominent disparities between the feature vectors

This paper aims to investigate the influence of lung volume on vowels produced in continuous natural speech signals. To achieve this, data from a dataset comprising 100 individuals reading phonetically balanced text is utilised. During this task, individuals engage in continuous speech for approximately 2–3 min while reading the text. As illustrated in Fig. 2, it begins with the detection of vowel segments from the speech. These segments are then processed to extract vector of audio features and glottal parameters. Feature vectors obtained from vowels pronounced at both high and low lung volumes are compared using subtraction operation. Positive differences in these features indicate higher values for vowels pronounced at high lung volumes. This analysis is conducted across all speakers to assess consistency in the difference values of the feature vectors. This approach helps identify prominent features that consistently exhibit higher or lower values in vowels pronounced at high lung volumes compared to those at low lung volumes. Consequently, the main contributions of this paper lie in comparing audio and glottal parameters for vowels pronounced at different lung volumes and identifying prominent features affected consistently across all speakers.

2.1 Data Details

As explained in [4], we gathered simultaneous speech and breathing data from a cohort of 100 healthy college students. Utilising ADInstruments' PowerLab equipment, we recorded time-synchronised speech and respiratory signals employing a microphone and respiratory transducer belt, respectively. Both speech and breathing patterns are sampled at a rate of 40 kHz. Subsequently,

speech data is down-sampled to 16 kHz, while breathing patterns to 50 Hz. The breathing values are normalised to a range between -1 and 1 by dividing by the maximum value. Breathing patterns encompass a series of time-varying signals corresponding to inhalation and exhalation phases. The time-span of high breathing values signifies the period of high lung volume, while the span of low breathing values denotes the phase of low lung volume. For further elaboration on the study, additional details can be found in [4].

The captured metadata encompasses the instantaneous pulse rate, blood pressure, height, weight, and a questionnaire consisting of six inquiries from the State and Trait Anxiety Inventory, current mental state, smoking habits, and any existing respiratory disorders or family history, with the aim of categorising participants as physically and mentally healthy. All participants fall within the 18 to 23 age group, comprising 31 females and 69 males. Each individual typically requires 2–3 min to read a phonetically balanced paragraph.

2.2 Data Processing

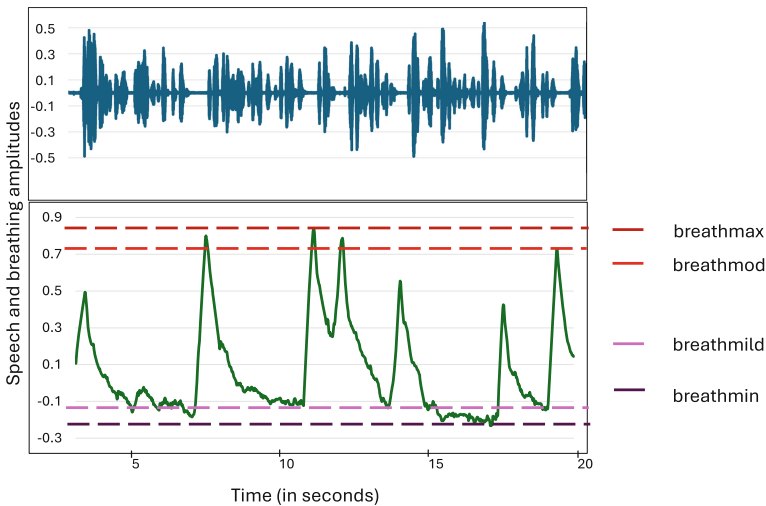


Fig. 3. The figure introduces naming conventions for the breathing values: the maximum and minimum breathing values are denoted as $-breathmax-$ and $-breathmin-$ respectively. Values that are 10% lower than the maximum and 10% higher than the minimum are termed $-breathmod-$ and $-breathmild-$ respectively

From the recorded speech signals, vowels are isolated using the Praat vocal toolkit [2]. Vowels that meet the following conditions are selected for subsequent analysis:

1. As seen in Fig. 3, the maximum breathing value is identified as `-breathmax-`. `-Breathmod-` represents the breathing value that is 10% below `-breathmax-`. Vowels pronounced with breathing values ranging between `-breathmod-` and `-breathmax-` are categorized as vowels articulated at high lung volume (Vhigh).
2. The minimum breathing value is identified as `-breathmin-`. The breathing value that is 10% below `-breathmin-` is represented as `-breathmild-`. Vowels pronounced with breathing values ranging between `-breathmin-` and `-breathmild-` are labelled as vowels articulated at low lung volume (Vlow). This is depicted in Fig. 3.
3. Vhigh and Vlow of a speaker with first formant frequency (F1) value surpassing 700 Hz, and
4. Vhigh and Vlow of a speaker with second formant frequency (F2) value falling below 2500 Hz.
5. A speaker may exhibit multiple instances of Vhigh and Vlow. The F1, F2 and the third formant frequency (F3) for the identified Vhigh and Vlow of a speaker are determined. Then, the difference between F1 values (F1diff), F2 values (F2diff), and F3 values (F3diff) among the same speaker's Vhigh and Vlow is computed. This difference is referred to as the Formant difference value (Fdiff). Finally, the pair of Vhigh and Vlow with the minimum Fdiff are selected for further analysis. This ensures that we compare the Vhigh and Vlow of a speaker having similar F1, F2, and F3 values. Similarity in formant values indicates that the vowels being produced share similar tongue positions or articulatory characteristics. Also, it suggests that the vowels have similar acoustic properties in terms of their height and front-back tongue position. This similarity often reflects similarities in vowel quality or perceived sound quality.

It is important to highlight that if a speaker lacks instances of either Vhigh or Vlow, the comparative analysis cannot be conducted for that speaker. Therefore, such cases are excluded from the current study. Among the 100 speakers, 94 have pronounced vowels meeting all five criteria. The designations of high and low lung volume serve as the ground truth for further speaker based analysis. Six features are derived from the identified vowel segments utilising Praat software [1]. The first three formants-F1, F2, and F3-reflect the resonant frequencies of the vocal tract, while the remaining three features encompass pitch, intensity, and duration.

2.3 Glottal Parameter Extraction

The segments acquired from vowel boundaries undergo processing to extract GCIs and their corresponding features. We have investigated two algorithms, SEDREAMS and SFF, for extracting GCIs from voiced speech.

SEDREAMS Algorithm: The Speech Event Detection using Residual Excitation and Mean-based Signal (SEDREAMS) algorithm, presented in [5], is designed for detecting GCIs and Glottal Open Instants (GOIs). Given our study's

emphasis on identifying GCIs, we will exclusively outline the steps for GCI detection. This algorithm comprises two primary steps: In the initial phase, a mean-based signal is computed to identify the brief intervals where GCIs are anticipated. These intervals, as depicted in Eq. 1, are derived by applying a Blackman window directly onto the original signal. This process is akin to convolving the original signal with an FIR filter exhibiting a frequency response similar to that of a low-pass filter.

$$y(n) = \frac{1}{2N + 1} \sum_{m=-N}^N w(m)s(n + m), \quad (1)$$

where $w(m)$ is the window and $s(n)$ is the original speech signal. The authors suggest that the method's reliability improves with the appropriate choice of the window length, denoted as $2N + 1$. Maximum reliability is achieved when the window length is between 1.5 to 2 times the average pitch period of the speaker under consideration. The signal $y(n)$ obtained after applying the window operation is termed the mean-based signal. Short intervals containing GCIs are derived from the minima of the mean-based signal. These intervals span from the minima to 0.35 times the local pitch period, where the pitch period is defined as the duration between two consecutive minima of the mean-based signal.

During the second step, the GCI location is further refined by employing the LP residual signal. By combining the intervals extracted from the mean-based signal with the LP residual, peaks within the identified intervals are selected, resulting in the determination of GCIs. This signal is termed the speech event detection (SED) signal, which is obtained by merging the LP residual and the GCI interval signal. The SED signal is used further for the extraction of glottal parameters.

SFF Algorithm: The Single Frequency Filtering (SFF) technique for GCI and GOI detection is presented in [11]. Given the primary focus of this paper on GCI-based analysis, we will exclusively address the GCI aspect discussed in [11]. This method of GCI detection relies on variations in the spectral characteristics throughout a glottal cycle. The algorithm consists of two stages: 1) SFF spectra calculation, and 2) calculation of spectral flatness from SFF spectra. The speech signal $s[n]$, as shown in Eq. 2, undergoes a difference operation to eliminate low-frequency variations:

$$x[n] = s[n] - s[n - 1]. \quad (2)$$

The signal $x[n]$, after differencing, undergoes multiplication by a complex exponential to generate a frequency-shifted signal, as illustrated in the Eq. 3:

$$x_k[n] = x[n]e^{j\bar{\omega}_k n} \quad (3)$$

$$\bar{\omega}_k = \pi - \omega_k = \pi - \frac{2\pi f_k}{f_s}. \quad (4)$$

$x_k[n]$ is the frequency shifted signal at any desired frequency k . This is further filtered using a single pole filter as per Eq. 5:

$$y_k[n] = -ry_k[n - 1] + x_k[n]. \quad (5)$$

To ensure filter stability, the r value is chosen to approximate 1. The amplitude envelope of the filtered signal $y_k[n]$ is derived by taking the square root of the squared sum of its real and imaginary components. These amplitude envelopes are computed for all frequencies at intervals of Δf , set at 10 Hz as recommended by the authors. The SFF spectrum exhibits flatter characteristics around the instances of glottal closure. Although temporal smearing occurs because of the Infinite Impulse Response (IIR) nature of the filter response, the impact of the impulse-like excitation remains consistent at the corresponding time points across all frequencies. The SFF spectra exhibit harmonics as a result of the sequence of impulse-like excitation.

The spectral flatness measure derived from the SFF spectra accentuates the impulse-like characteristics at the GCIs. Spectral flatness is computed by dividing the geometric mean of the spectral values by the arithmetic mean. Elevated spectral flatness signifies a uniform distribution of spectral values, which occurs at the GCI positions. The spectral flatness calculated from the SFF spectra offers a one-dimensional depiction of the excitation source features. The spectral flatness contour gradually decreases between the GCIs due to the temporal smearing of the SFF output. This SFF spectrum is further used for the extraction of glottal parameters.

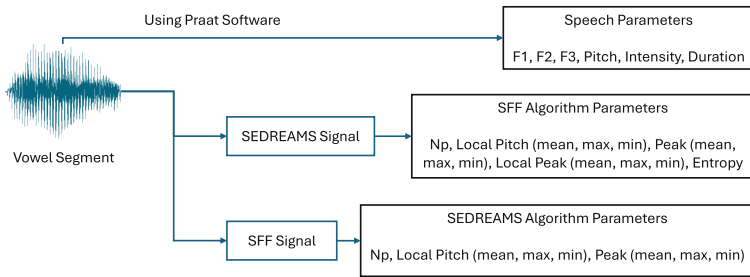


Fig. 4. The process for feature extraction involves three steps: 1) Extracting features from the raw speech signal, 2) Extracting features from the GCI signal acquired through the SFF algorithm, and 3) Extracting features from the GCI signal obtained with the SEDREAMS algorithm. In total, 24 features are extracted using these three methods

Feature Extraction: Figure 4 depicts the procedure followed for the extraction of speech and glottal features from the identified vowel segments. As explained in Sect. 2.2, 6 features are calculated directly from the speech signals (F1, F2, F3, pitch, intensity, and duration). Further, 11 features from the SFF spectrum, and 7 features from SED signal are extracted. The SED signal and SFF spectrum are 1-dimensional signals obtained from the SEDREAMS and SFF algorithms, respectively. All the 24 features extracted from the speech signal, SED signal, and spectral flatness are as described below. It should be noted that the features extracted from SED and SFF signals are referred to as glottal parameters.

1. Audio based features: Formants ($F1$, $F2$, and $F3$), Pitch, Intensity, and Duration: The first three formants, pitch, intensity, and duration are calculated directly from the speech signal using the Praat [1] software.
2. Number of peaks (SFF:Np and SED:Np): This is the count of peaks detected in the SFF spectrum and SED signal.
3. Peak-maxima, -minima, and -average (SFF:PeakMax, SFF:PeakMin, and SFF:PeakMean; SED:PeakMax, SED:PeakMin, and SED:PeakMean) : The maximum, minimum, and average of the amplitudes of the peaks identified in the SFF spectrum and SED signal.
4. Local pitch period-maxima, -minima, and -average (SFF: PitchMax, PitchMin, and PitchMean) and (SED: PitchMax, PitchMin, and PitchMean): The distance between two consecutive peaks (sample count) is the local pitch period. The maximum, minimum, and the average of the local pitch period is calculated for both SFF and SED.
5. Local pitch period amplitude-average, -maxima, and -minima (SFF:Pmean, SFF:Pmax, and SFF:Pmin): These features are calculated only from the SFF spectrum. They are the average, maximum, and minimum of the harmonics appearing between two consecutive peaks.
6. Power spectral entropy (SFF:Entropy): This feature is also calculated only from the SFF spectrum. It represents the entropy calculated for the regions between consecutive peaks. Entropy is calculated following the Eq. 6:

$$Entropy = - \sum_{i=1}^n p_i \ln(p_i), \quad (6)$$

where n represents the number of samples in the SFF spectrum and p_i is the amplitude of the sample at index i .

3 Results

Figure 5 displays the GCIs extracted through the SEDREAM and SFF algorithms for the speaker, designated with identity (ID) 6. Notably, the SFF signal throughout exhibits higher amplitude values at elevated lung volumes. Elevated values in the SFF spectra indicate greater spectral flatness, as discussed in Kadiri et al. [11]. This observation suggests that Speaker ID 6 exhibits higher spectral flatness during Vhigh compared to Vlow. The SED signal reveals more peaks, indicating additional glottal closure instances during vowel pronunciation at high lung volumes.

To assess the influence of lung volume on the glottal parameters extracted from GCI signals of Vhigh and Vlow, we compare their respective feature vectors. As detailed in Sect. ??, we extract 24 features from speech signals using Praat, and from SED and SFF signals using their respective algorithms. Since vowel segments are tagged with ground truth labels for high and low lung volume, their feature vectors inherit this labelling. As depicted in Fig. 2, we compare the feature vectors associated with high and low lung volume labels to analyse variations across the 24 feature values. We then calculate the difference feature vector

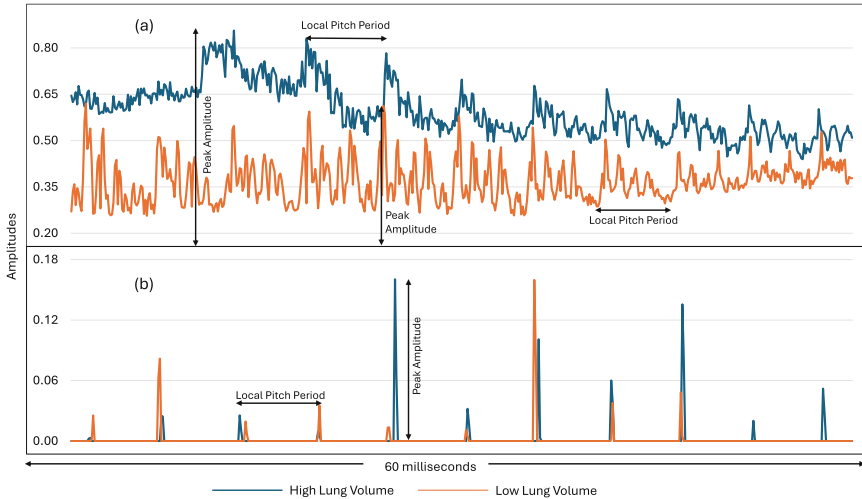


Fig. 5. Glottal closure instants extracted using SEDREAMS and SFF algorithms. (a) represents the SFF signals for the vowels pronounced at high and low lung volumes extracted using SFF algorithm. (b) represents the SED signal extracted using SEDREAMS algorithm for the same vowels articulated at high and low lung volume by the same speaker. Further glottal parameters are extracted from these SFF and SED signals

for all 94 speakers to examine the consistency of feature differences across the entire dataset. The criterion employed to gauge the importance of a particular feature is the count of speakers, out of the 94, for whom the disparity in feature values between V_{high} and V_{low} adheres to a discernible trend. The emphasis here lies on whether the trend in feature value difference is positive or negative.

As illustrated in Fig. 6, the absolute values of speaker percentages displaying elevated feature values for V_{high} is depicted. In our analysis, we find that speaker counts falling between 40% and 60% do not carry remarkable weight, as they closely resemble chance-level outcomes. This region is hence termed as plateau region as seen in Fig. 6.

Out of six audio features, the three formant frequencies are utilised to pinpoint suitable vowel segments that capture similar vowel articulation for analysing variations in glottal parameters. For this reason, the three formant frequencies are not compared and displayed in Fig. 6. While the speaker pitch values, although not exceeding 60%, are observed to be higher during high lung volumes for nearly 60% of the speakers. It is worth noting that intensity and vowel duration do not show any discernible pattern for distinguishing between V_{high} and V_{low} among the speakers. Notably, there are no speaker counts below 40%, suggesting that no audio parameter values are lower for V_{high} when compared to those of V_{low} .

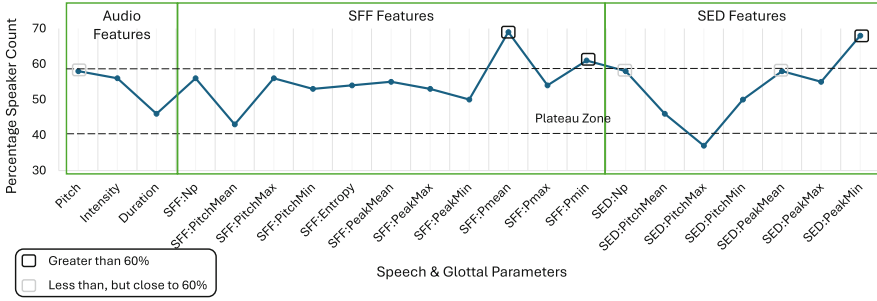


Fig. 6. This illustration shows the percentage of speakers exhibiting a higher feature value for high lung volume. Among the 94 speakers analysed, more than 60% speakers display a higher value of the three features marked in the figure: SFF:Pmean, SFF:Pmin, and SED:PeakMin

Among the 11 SFF features examined, SFF:Pmean shows elevated values for Vhigh in 69% of the speakers. This percentage represents the highest occurrence of a higher value for this glottal parameter in Vhigh instances, underscoring its important role in distinguishing between Vhigh and Vlow for each speaker. Additionally, SFF:Pmin remains elevated for over 60% of the speakers. Both SFF:Pmean and SFF:Pmin illustrate that vowel pronunciation at higher lung volumes elicits greater harmonics compared to those at lower lung volumes.

Out of the seven glottal parameters derived from the SED signal, the minimum amplitude values of peaks exhibit higher values in Vhigh for 68% of the speakers. Although falling below the 60% threshold, the number of peaks identified with SED signals surpasses those for Vlow in approximately 60% of speakers. Likewise, the average amplitude value of peaks identified in SED also exceeds for Vhigh in about 60% of speakers. A higher value for SED:Np indicates more frequent glottal closures during the pronunciation of vowels at higher lung volumes. An interesting trend is observed with SED:PitchMax, which denotes the maximum distance between two consecutive peaks identified in the SED signal. This is also referred to as maximum local pitch period. For 37% of speakers, this distance is higher in Vhigh instances, implying that 63% of speakers exhibit a low maximum local pitch value for Vhigh compared to Vlow. This suggests that the pitch, which is inversely related to the pitch period, does not decrease notably for vowels pronounced at high lung volume.

When examining the top three performers-SFF:Pmean, SFF:Pmin, and SED:PeakMin-it is noted that these three glottal parameters collectively enable differentiation between Vhigh and Vlow for 77% of speakers.

4 Conclusion

In conclusion, glottal parameters offer insights into an individual’s lung volume during vowel pronunciation within continuous speech. Augmented with a

broader array of features, they are poised to enhance classification performance considerably. Building upon the findings of previous studies such as Hoit et al. [8], Iwarsson et al. [9], and Winkworth et al. [14], we supplemented our empirical analysis of glottal parameters. Our findings suggested that higher harmonics and increased instances of glottal closure are additional indicators of vowels pronounced at higher lung volumes.

References

1. Boersma, P., Van Heuven, V.: Speak and unspeak with praat. *Glott Int.* **5**(9/10), 341–347 (2001)
2. Corrette, R.: Praat vocal toolkit: a praat plugin with automated scripts for voice processing (2012–2023). (<http://www.praatvocaltoolkit.com/index.html>)
3. Deshpande, G., Gudipalli, Y., Patel, S., Schuller, B.W.: Applying speech derived breathing patterns to automatically classify human confidence. In: 2023 31st European Signal Processing Conference (EUSIPCO), pp. 1335–1339. IEEE (2023)
4. Deshpande, G., Schuller, B.W., Deshpande, P., Joshi, A.R.: Automatic breathing pattern analysis from reading-speech signals. In: 2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 1–4. IEEE (2023)
5. Drugman, T., Dutoit, T.: Glottal closure and opening instant detection from speech signals. In: Proceedings of Interspeech 2009, pp. 2891–2894 (2009). <https://doi.org/10.21437/Interspeech.2009-47>
6. Goldman-Eisler, F.: Speech-breathing activity—a measure of tension and affect during interviews. *Br. J. Psychol.* **46**(1), 53 (1955)
7. Henderson, A., Goldman-Eisler, F., Skarbek, A.: Temporal patterns of cognitive activity and breath control in speech. *Lang. Speech* **8**(4), 236–242 (1965)
8. Hoit, J.D., Solomon, N.P., Hixon, T.J.: Effect of lung volume on voice onset time (vot). *J. Speech Lang. Hear. Res.* **36**(3), 516–520 (1993)
9. Iwarsson, J., Thomasson, M., Sundberg, J.: Lung volume and phonation: a methodological study. *Logoped. Phoniatr. Vocol.* **21**(1), 13–20 (1996)
10. Iwarsson, J., Thomasson, M., Sundberg, J.: Effects of lung volume on the glottal voice source. *J. Voice* **12**(4), 424–433 (1998)
11. Kadiri, S.R., Prasad, R., Yegnanarayana, B.: Detection of glottal closure instant and glottal open region from speech signals using spectral flatness measure. *Speech Commun.* **116**, 30–43 (2020)
12. Mitchell, H.L., Hoit, J.D., Watson, P.J.: Cognitive-linguistic demands and speech breathing. *J. Speech Lang. Hear. Res.* **39**(1), 93–104 (1996)
13. Solomon, N.P., Hixon, T.J.: Speech breathing in Parkinson’s disease. *J. Speech Lang. Hear. Res.* **36**(2), 294–310 (1993)
14. Winkworth, A.L., Davis, P.J., Ellis, E., Adams, R.D.: Variability and consistency in speech breathing during reading: lung volumes, speech intensity, and linguistic factors. *J. Speech Lang. Hear. Res.* **37**(3), 535–556 (1994)



Linear Frequency Residual Cepstral Features for Dysarthria Severity Classification

Aditya Pusuluri^(✉) and Hemant A. Patil

DAIICT, Gandhinagar, Gujarat, India
aditya_pss@daiict.ac.in

Abstract. The dysarthric severity-level classification system serves as a valuable diagnostic tool, which enables the assessment and monitoring of the condition progression in patients, and a selection of appropriate severity-specific models for recognizing dysarthric speech—an important assistive technology. Determining the severity of dysarthria presents a considerable challenge in clinical practice, given the heterogeneous nature of speech impairments associated with this motor speech disorder. This study investigates the application of Linear Frequency Residual Cepstral Coefficients (LFRCC), which are derived from the excitation source information captured via the Linear Prediction (LP) residual signal, for classification of dysarthria severity-levels. To our knowledge, this is the first work to utilize LFRCC for this purpose. Experimental assessments were conducted on two extensively employed datasets, namely, UA-Speech and TORGO. Validation of the results was carried out using a Convolutional Neural Network (CNN) with 5-fold cross-validation and test accuracies with MFCC, LFCC, and web-scale Supervised Pretraining for Speech Recognition (WSPSR), also known as Whisper, encoder module as the baseline features. Additionally, to ensure speaker-independence, Leave-One-Speaker-Out (LOSO) experiments were conducted. Furthermore, the robustness of LFRCC features against noise was explored, encompassing both stationary and non-stationary noises at varying Signal-to-Noise Ratio (SNR)-levels. Lastly, comparative analysis of latency periods with baseline feature sets suggests the potential applicability of LFRCC in real-world scenarios for severity-level classification systems.

Keywords: Dysarthria Severity-Level Classification · Linear Frequency Residual Cepstral Coefficients · Noise Robustness

1 Introduction

Dysarthria, a motor speech disorder characterized by impaired articulation, phonation, and prosody, poses significant challenges in clinical assessment and management due to its heterogeneous nature and varying severity-levels [1]. Several speech disorders, such as apraxia, dysarthria, and stuttering, affect an individual's ability to generate speech sounds. Accurate classification of dysarthria

severity is essential for guiding treatment planning and monitoring disease progression. Over the years, researchers have explored methodologies and feature sets to improve the reliability of dysarthria severity classification. The severity-level of the pathology speech can be identified using speech intelligibility, which is affected by various factors, such as articulation rate, audibility, prosody, etc. [2]. Hence, the severity of dysarthria can be identified by all these factors as they are related to the intelligibility of speech. The subjective analysis of the severity-level of dysarthria speech has proved to be exhaustive, expensive, and time-consuming, which brings the need for automation of the severity-level classification. The automation of severity-level classification also helps towards the betterment of automatic speech recognition (ASR) systems for dysarthric speech.

Early approaches focused on acoustic features derived from fundamental or pitch frequency (F_0), formant frequencies, and duration measures. However, these traditional features often lack the sensitivity and specificity required for precise severity classification, particularly in cases of subtle or nuanced speech impairments. Previous studies have demonstrated the efficacy of cepstral-based features for the classification of the severity-level of dysarthria, which captures the characteristics of vocal tract system. In [3], the study shows that measures obtained from fundamental or pitch frequency (F_0) and the second formant frequency (F_2) are highly correlated with the intelligibility of dysarthric speech. The Mel Frequency Cepstral Coefficients (MFCC) showed the ability for speech pathology classification more so for dysarthric speech [4]. In [5], MFCCs are encoded using a deep belief network and used for dysarthria classification using Multi-Layer Perceptron (MLP). Furthermore, the combination of MFCC with auditory features resulted in better results. Later, Linear Frequency Cepstral Coefficients (LFCC) are used to observe the information captured through the linear frequency scale. In [6], LFCC features are used to capture the speech intelligibility of dysarthric speech. One promising approach involves the extraction of cepstral features from the residual signal obtained through Linear Prediction (LP) analysis. Linear Frequency Residual Cepstral Coefficients (LFRCC) capture fine spectral details and excitation source information. By leveraging the LP residual signal, which represents the discrepancy between the actual speech signal and its linear prediction, LFRCC offers a novel perspective on dysarthric severity classification. This paper proposes application of LFRCC features, which contains the information of excitation source of the speech production mechanism, for the classification of dysarthria severity-level. Previously, the LFRCC features are used for various applications, such as speaker recognition [7], spoof detection [8], emotion recognition [9]. Furthermore, work represented in [10] indicates the importance of excitation source information for dysarthric speech and study reported in [11] indicated the importance of excitation information for nasalized speech. To the best of our knowledge and belief, no prior research has explored the potential of LFRCC features specifically for this task.

The rest of the paper is organized as follows: Sect. 2 represents a brief technical details about the proposed LFRCC feature extraction, whereas Sect. 3 gives the details about the database used, classifiers used, and the baseline features considered for this work. Section 4 consists of the motivation for the applica-

tion of LFRCC features for dysarthric severity-level classification task. Section 5 contains the experimental results and the work is concluded with results and conclusion along with potential future research direction.

2 Linear Frequency Residual Cesptral Coefficients (LFRCC)

Historically, the original idea of linear predictions is adopted from control and system identification literature to speech coding application [12]. Traditionally, in the framework of linear time-invariant (LTI) assumption, LP analysis is used to separate excitation source and vocal tract system information responsible for speech production. In LP analysis, each sample of a speech signal is represented by a linear combination of past “p” speech samples. The parameter “p” is called the order of linear prediction. The linear combination of the past speech samples are associated with weight parameter, which are called as *Linear Prediction Coefficients* (LPC). The predicted speech sample $\hat{x}(n)$, given the current speech sample $x(n)$ given by:

$$\hat{x}(n) = - \sum_{k=1}^p f_k x(n-k), \quad (1)$$

where f_k are the weights or LPC assigned for each of the previous speech samples. The difference between the original speech sample, $x(n)$ and predicted speech sample, $\hat{x}(n)$ is termed as *LP residual signal*, $r(n)$. In particular,:

$$r(n) = x(n) - \hat{x}(n) = x(n) + \sum_{k=1}^p f_k x(n-k). \quad (2)$$

Furthermore, an all-pole filtering is applied to the speech signal with LP analysis:

$$F(z) = 1 + \sum_{k=1}^p f_k z^{-k}, \quad (3)$$

$$H(z) = \frac{G}{1 + \sum_{k=1}^p f_k z^{-k}}, \quad (4)$$

where $F(z)$ denotes an inverse filter associated with the all-pole LP filter, $H(z)$. This system function is associated with the vocal tract system information, and the variable G is the gain term in LP model. In general, the system information is combined with excitation source information and ideally, the LP residual spectrum effectively captures the excitation source information by filtering out the system-related information. In particular, the peaks and valleys of the LP residual spectrum indicate the Glottal Closure Instant (GCI) and Glottal Open Instant (GOI), respectively, during the speech production mechanism [13]. The excitation source information is approximated either by a quasi-periodic train of impulses for voiced and noisy for unvoiced speech signal and a combination of both for voiced fricatives [10].

Furthermore, the LP residual is processed in the cepstral-domain using a linear filterbank to extract proposed LFRCC (as shown in Fig. 1) for the task of dysarthria severity classification because the linearity in the speech increases with the severity of the dysarthria [14]. The features are extracted using a pre-emphasis (high pass) filter with system function $[1-0.97z^{-1}]$ followed by a window duration of 25 ms and an overlap of 15 ms. We used 40 linearly-spaced subband filters followed by the logarithm of the filterbank output. Finally, Discrete Cosine Transform (DCT) is applied to obtain 20-D LFRCC features. DCT is used for feature decorrelation, energy compaction, and dimensionality reduction.

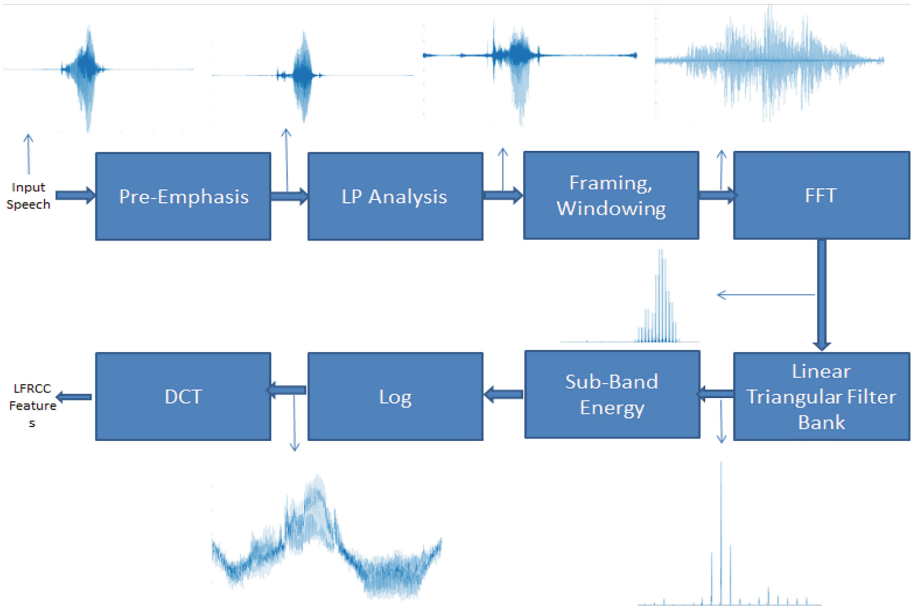


Fig. 1. Functional block diagram of proposed LFRCC extraction for dysarthric severity-level classification. After [8].

3 Experimental Setup

3.1 Dataset Used

In this work, two well-known dysarthria speech corpora, namely, the Universal Access Dysarthria Speech Corpus (UA-Speech) [15] and TORGO [16] are used, whose statistics are summarized in Table 1. Both corpora are primarily of the *spastic* dysarthric type, which is characterized by several factors, such as breathiness, hypernasality, harsh voice, and incorrect articulation that causes unintelligible speech. For UA-Speech corpus, 8 speakers (4 males, namely, M01, M05,

M07, and M09, and 4 females, namely, F02, F03, F04, and F05) is considered. From 735 utterances, 465 utterances per speaker as mentioned in [17] were used. The TORGO corpus consists of 1982 samples with 671 samples belonging to very low severity, 627 samples belonging to low severity, and 684 samples for medium severity. For both corpora, 80% of the data is used for training, and 20% is used for testing. The train-test split is performed in a manner such that both the splits consists of words, non-words, and sentences. The experiments conducted involve a 5-fold cross-validation (CV), where the training split is exclusively used for performance evaluation to understand the model’s robustness and has some speaker robustness (or independence). The classifier is further tested on the separate test split, ultimately leading to enhanced generalization capabilities and possible speaker independence. Further in order to ensure speaker-independency of the proposed system, we performed leave one speaker out (LOSO) on both corpora.

Table 1. Class-wise patient details for UA-Speech and TORGO. After [15,16].

Severity Level	UA-Speech	Dysarthria Type	TORGO	Dysarthria Type
Very Low (VL)	F05, M09	Spastic	F04, M03	Spastic
Low (L)	F04, M05	Mixed, Spastic	F01, M05	Spastic
Medium (M)	F02, M07	Spastic	M01, M04	Spastic
High (H)	F03, M01	Spastic	–	–

3.2 Classifier Used

The experiments were performed using Convolutional Neural Network (CNN) classifier. The work uses CNN as a classifier because of its ability to maintain translation invariant, and it can effectively capture relevant spectro-temporal patterns and variations of speech signals. Table 2 reports a detailed description of CNN architecture. The model is trained using stratified 5-fold cross validation (CV) strategy with a seed value and a train and validation split of 80% and 20% using *adam* optimizer, *categorical cross-entropy* as a loss function, and *accuracy* as the evaluation metric. The stratified method ensures the distribution of data in each fold to be similar to the distribution of the entire data. The algorithm was tuned using grid search to select the best learning rate, and the batch size for 80 epochs. Two activation functions are used, namely, *ReLU* and *softmax*. A *ReLU* is used in order to improve the learning speed while reducing the computational cost, and the softmax activation is used at the final layer for multiclass classification. A normalization layer is added along with a dropout layer after each convolutional layer in order to avoid the overfitting of CNN model. The fine-tuning of resulted in a learning rate, batch size, and epochs as 0.01, 128, and 70, respectively. The networks were implemented using the python library Keras v.2.24 using TensorFlow-GPU v.1.14.0 backend. The experiments are performed using GeForce GTX 1660 Ti graphic card.

3.3 Baseline Features

MFCC and LFCC features are considered as the baseline for this work. For a fair comparison, 20-D feature vectors were extracted using a 25 ms and a hop length of 10 ms using Librosa toolkit for all the feature sets used for this study. Additionally, whisper tiny model [18], as part of the whisper suite of models is selected as state-of-the-art baseline. Tiny model represents the smallest variant with relatively fewer trainable parameters and layers compared to its counterparts. With 4 layers, a width of 384, and 6 attention heads, the whisper tiny model provides a compact yet efficient solution for various speech-related tasks. The fixed-dimensional vectors obtained from the encoder module are of size $1 \times 1500 \times 384$, capturing temporal values of the input audio signal. Table 1 illustrates the specifications of whisper models, showcasing the different dimensions and complexities associated with varying model sizes [18] (Table 3).

Table 2. CNN Architecture

Output Size	Description
(20,2000,1)	LFRC
(20,2000,16)	convolution layer, 16 filters, BN, relu
(10,1000,16)	max-pooling, (2,2), dropout (0.25)
(10,1000,32)	convolution layer, 32 filters, BN, relu
(5,500,32)	max-pooling, (2,2), dropout (0.25)
(5,500,64)	convolution layer, 64 filters, BN, relu
(2,250,64)	max-pooling, (2,2), dropout (0.25)
(2,250,128)	convolution layer, 128 filters, BN, relu
(1,125,128)	max-pooling, (2,2), dropout (0.25)
(1,125,256)	convolution layer, 256 filters, BN, relu
(1,125,256)	dropout (0.25)
128	dense layer, ReLu
64	dense layer, ReLu
16	dense layer, ReLu
4	dense, softmax

Table 3. Parameter details of baseline features used.

Parameters	Whisper (Tiny)	MFCC	LFCC
Freq. Scale	-	Mel Scale	Linear Scale
Feat. Dimension	$1 \times 1500 \times 512$	20	20

4 Spectrographic Analysis

From Fig. 2, it can be observed that as the LP order increases, the noise content in the residual signal increases along with unwanted peaks and valleys at non-Glottal Closure Instance (GCI) and non-Glottal Open Instance (GOI) locations.

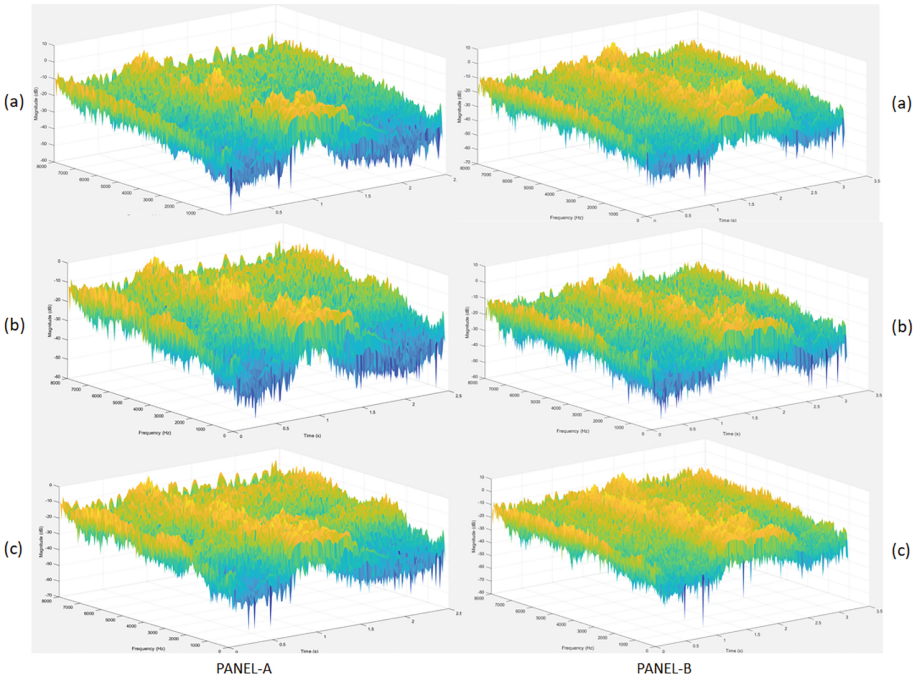


Fig. 2. Spectrogram of LP residual for (a) $p = 4$, (b) $p = 10$, (c) $p = 18$. Panel-A and Panel-B shows very low and high severity dysarthric signal, respectively.

For further analysis of residual signal for various severity-levels of dysarthria, vowels are selected because of their role in the speech production and their importance in estimating speech intelligibility [19]. In particular, vowels being typically longer carry prosodic components, such as rhythm and intonations (pp.96, Chap. 3, [20]). Vowels are generated by the quasi-periodic vibration of vocal folds in the larynx, which results in quasi-periodic sound waves. The shape of the vocal tract system results in distinct vowel sounds. *Vowel Onset Point* (VOP) marks the beginning of a vowel within a speech utterance. Since vowels are the major energy carriers in a speech signal, analyzing the VOP energy helps us understand how the source signal is affected for a dysarthric speaker. The time-varying changes in a speech signal are captured in the residual signal, however are smeared due to the peaks and valleys in the residual. In order to highlight these changes, the Hilbert envelope of the residual is considered [21].

From the Hilbert envelope of residual signal, for normal voiced speech segment, the signal consists of impulse-like excitation peaks at GCI locations. However, for a speaker with dysarthria, the residual signal becomes noisier due to the harshness introduced in the speech signal, resulting in random/unwanted peaks. This is because, the dysarthric speech (spastic dysarthria in particular) often consists of excessive nasalization. The residual signal for voiced speech segments is related with the glottal pulses and reflect strong impulse-like excitation peaks. However, since the spastic dysarthria speech is highly influenced by nasalization, the residual signal shows peaks at random instances instead of glottal excitation location as can be seen in Fig. 3.

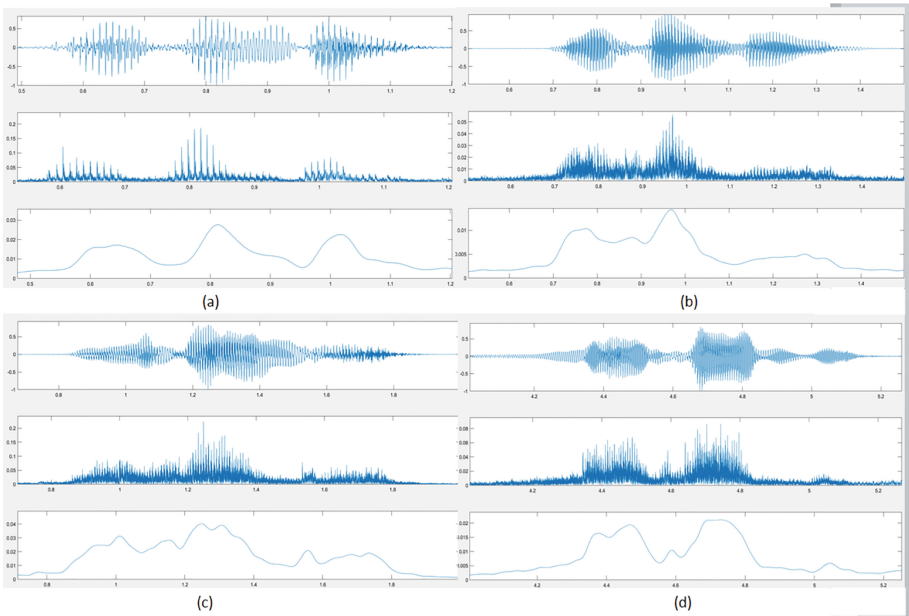


Fig. 3. Speech signal, Hilbert envelope of LP residual, and smoothed Hilbert envelope of LP residual of the word “November” for (a) normal speaker, (b) very low, (c) low, and (d) medium severity-level dysarthria speaker, where X-axis is time and Y-axis represents pitch (F_0) intensity.

Furthermore, to capture the VOP, the Hilbert envelope is smoothed by convolving the signal with a Hamming window of 50 ms duration [22]. This smoothed signal helps us to locate the VOP regions easily as compared to the Hilbert envelope. From Fig. 3, it can be observed that the energy content at the VOP region is lower for a dysarthria speaker when compared with a normal speaker. Additionally, it can be observed that the VOP peaks are merged for a dysarthria speaker, and unwanted peaks starts to occur as the severity increases.

5 Experimental Results

This section evaluates proposed LFRCC feature set w.r.t. several evaluation factors, such as fine-tuning of LP order, comparison with existing speech features (such as MFCC and LFCC), robustness under signal degradation conditions, and analysis of latency period.

5.1 Effect of LP Order

In this experiment, we determine the optimal LP order (p) by exploring values ranging from 4 to 18 with a step size of 2, considering a sampling frequency of 16 kHz. The evaluation employs a CNN classifier with 5-Fold cross-validation. An LP order of 10 achieves the highest fold (test) accuracy of **92.01 (94.91)** and **91.43 (94.20)** for UA-Speech and TORGO, respectively. Analysis from Fig. 4 reveals that higher values of p result in decreased classification accuracy compared to lower LP order.

This is likely due to the noisy nature of dysarthria speech, where higher order LP models tend to capture more noise components as the higher order has a tendency to capture finer spectral information, which can be observed from Fig. 5. Furthermore, given the reduced articulatory precision in dysarthric speech, lower order LP models may adequately capture severity-specific information.

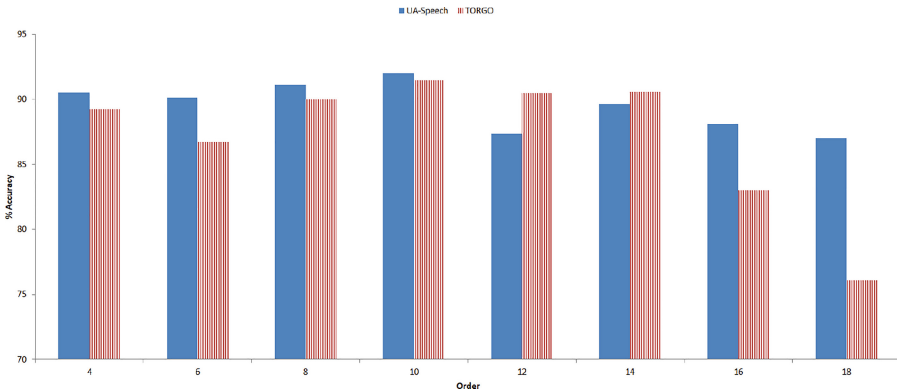


Fig. 4. Fine-Tuning of Order (P) for LFRCC on UA-Speech and TORGO.

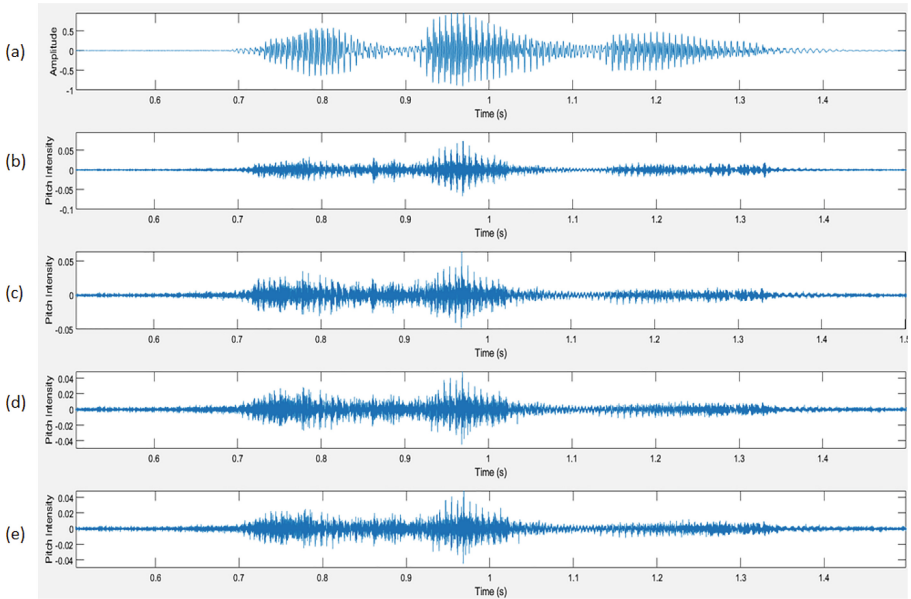


Fig. 5. (a) Speech signal of the word “November” for very low severity dysarthric speaker and its residual with LP order (b) $p = 8$, (c) $p = 10$, (d) $p = 14$, and (e) $p = 18$.

5.2 Comparison with Baseline Spectral Features

Table 4 shows the relative comparison of LFRCC feature set along with baseline spectral features, such as MFCC, LFCC, and state-of-the-art whisper model-based features using CNN classifier.

For both datasets, LFCC outperforms MFCC features, indicating the effectiveness of linear frequency scale for the task of dysarthria severity classification.

Table 4. Fold, Test, Precision (P), Recall (R), and F1-Score for various features using CNN classifier on UA-Speech and TORGO.

Data	Feature Set	Fold Acc.	Test Acc.	Precision	Recall	F1-Score
UA-Speech	MFCC	87.29	90.68	91.52	90.21	90.86
	LFCC	91.50	92.93	93.61	92.41	92.68
	Whisper (Tiny)	92.01	94.80	93.28	92.44	92.51
	LFRCC	92.01	94.91	95.06	94.85	94.83
TORGO	MFCC	85.71	88.62	89.40	88.64	89.01
	LFCC	90.58	91.82	91.51	91.02	91.26
	Whisper (Tiny)	90.97	94.10	93.47	94.00	94.00
	LFRCC	91.43	94.20	93.96	94.19	94.07

LF RCC outperforms both MFCC and LFCC baseline features by a fold (test) accuracy margin of 4.72% (4.23%), 0.51% (1.98%) for UA-Speech, and 5.72 % (2.81%), 5.58% (2.38%) for TORGO. Furthermore, state-of-the-art Web-scale Supervised Pretraining for Speech Recognition (WSPSR), also known as Whisper encoder module is used as a baseline for dysarthria severity level classification task. The proposed LF RCC features perform on par for UA-Speech and outperforms by fold (test) accuracy of 0.46% (0.2%) for TORGO, when compared with whisper model, which is an advance machine learning model that is trained using labelled data of 680,000 h. Table 5 indicate the classwise precision, recall, and F1-Score of LF RCC with $p = 10$ for UA-Speech and TORGO database. We notice the balanced F1-score across all the severity-levels reflects the model’s robustness and consistency in classification performance.

Table 5. Class-wise precision (P), recall (R), and F1 score for UA-Speech and TORGO.

Data	Class	Precision	Recall	F1-Score
UA	VL	94	93	93
	L	88	99	94
	M	100	94	97
	H	98	93	96
TORGO	VL	99	89	94
	L	92	95	94
	M	93	98	95

Due to computational limitations, further experiments are compared against MFCC and LFCC baseline features.

Additionally, Leave-One-Speaker-Out (LOSO) strategy was employed to evaluate the model’s performance across different speakers. For UA-Speech, a total of 8 speakers were considered, while 6 speakers were utilized for the TORGO corpus (Table 6).

Table 6. LOSO Based Speaker-Independent Results

Data	Features	Accuracy	Precision	Recall
UA-Speech	MFCC	31.55	23.89	33.33
	LFCC	32.90	20.77	36.90
	LF RCC	41.37	30.56	44.71
TORGO	MFCC	24.70	17.70	36.90
	LFCC	20.41	16.40	33.31
	LF RCC	40.10	31.59	39.72

This analysis was performed to investigate whether the classification model could generalize well to unseen speakers and thus, verifying its speaker independence. The results demonstrate that the LFRCC feature set outperforms both MFCC and LFCC features by a significant margin. Specifically, for the UA-Speech corpus, LFRCC features yield an improvement of 9.82% over MFCC and 8.47% over LFCC. Similarly, for the TORGO corpus, LFRCC features exhibit superiority with enhancements of 15.4% over MFCC and 19.69% over LFCC. The observed superior performance of LFRCC feature set over the baseline features suggests that the utilization of a lower LP order may contribute to capturing dysarthria-specific characteristics. This result may indicate that the lower LP order helps to capture dysarthria-specific information rather than speaker-specific characteristics. Furthermore, the LOSO experimentation highlights the robustness of the classification model across different speakers, underscoring the effectiveness of the LFRCC feature set in capturing generalizable dysarthric severity-specific acoustic cues while ignoring individual speaker-specific variations.

5.3 Noise Robustness of LFRCC

In order to test the practical applicability of the proposed LFRCC feature set, the robustness of LFRCC is analyzed using additive stationary (white) and non-stationary (babble) noises at different Signal-to-Noise Ratio (SNR) levels. From Table 7, it can be observed that the proposed LFRCC features outperforms the baseline (MFCC and LFCC) features for regions with higher noise energy, i.e., (at SNR -10 dB and -5 dB). Moreover, as illustrated in Fig. 6, the integrity of the excitation source characteristics within the residual signal remains intact even in the presence of -10 dB babble noise, as evidenced by the clear presence of Voice Onset Time (VOT) regions under noisy conditions.

Table 7. Accuracy for stationary and non-stationary noise types across various SNR levels using CNN classifier on UA-Speech.

Noise	Feature Set	SNR (dB) Level				
		-10	-5	0	5	10
White	MFCC	83.94	87	90.12	90.25	90.45
	LFCC	83.19	86.15	90.81	91.24	92.79
	LFRCC	90.52	92.37	93.91	94.07	95.02
Street	MFCC	83.92	84.91	91.66	93.92	94.42
	LFCC	81	81.04	91.79	92.14	94.62
	LFRCC	89.40	90	90.67	93.36	95.62

Furthermore, LP residual captures information about glottal activity by shooting a peak (positive peak for glottal closure instant and negative peak

for glottal open instant). Thus, the LP residual profile preserves the sequence of impulse-like excitation to vocal tract system. Moreover, the SNR is higher at open and closure instances and these regions are supposed to be more immune to noisy conditions [23]. Further, LFRCC represents the cepstral information extracted from LP residual in various sub-bands. On the other hand, MFCCs are known to be *notoriously* affected by noise due to its poor spectral representation at higher frequencies. Thus, it is the sequence of impulses that is initially captured by LP residual, which is generally occupied by noise energy, and thus, justifying noise robustness of proposed LFRCC feature set and then its Fourier transform (which is also expected to be sequence of impulses in the frequency domain) helps to maintain relatively better speech intelligibility under noisy conditions.

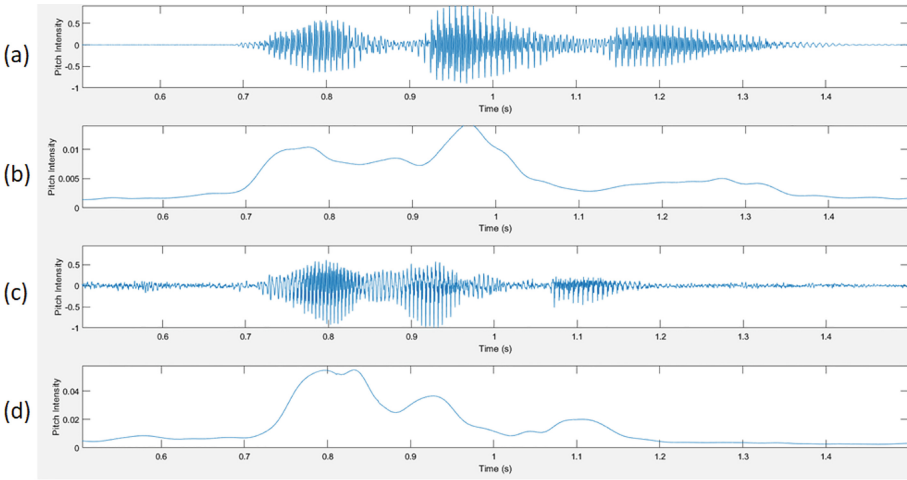


Fig. 6. (a), (b) represents clean speech and its smoothed Hilbert envelope, and (c), (d) represents speech signal with -10 dB babble noise, and its smoothed Hilbert envelope of the word “November” for very low severity-level

5.4 Analysis of Latency Period

The analysis of latency period helps us understand the minimum number of speech frames required to achieve maximum classification accuracy. This indicates the responsiveness of the proposed features. From Fig. 7, it can be observed that LFRCC achieves maximum accuracy with 750 ms (30 frames) of the input signal, and outperforms the baseline MFCC and LFCC features.

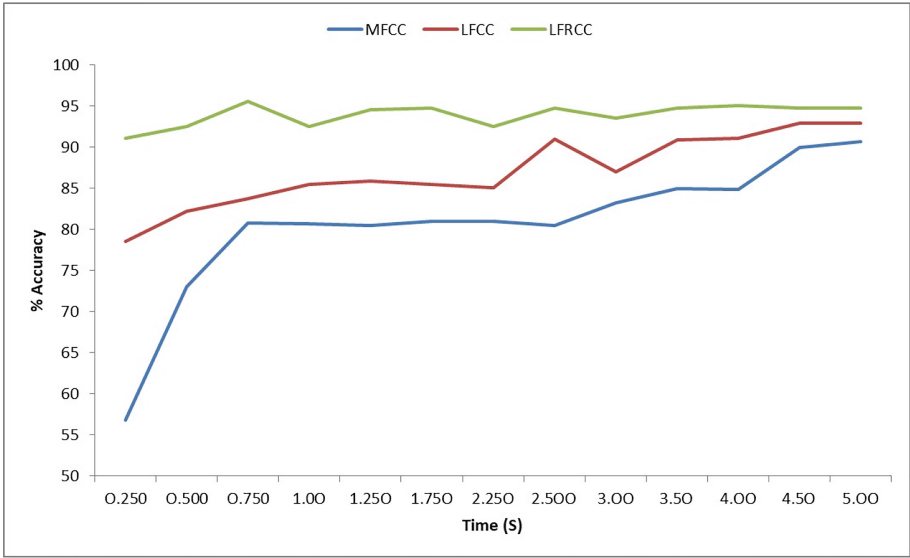


Fig. 7. Latency Period Analysis for MFCC, LFCC, and LFRCC features on UA-Speech.

6 Summary and Conclusions

The study proposed using LP residual-based LFRCC features for dysarthric severity-level classification task employing two widely recognized dysarthria speech corpora: the UA-Speech and TORGO. Through several experiments, we uncovered several key insights that showcase the utility of LFRCC features in dysarthria severity classification. Our exploration of the optimal LP order revealed that an order of 10 yielded the highest accuracy, maintaining a balance between capturing spectral information and minimizing noise interference. Notably, higher LP orders exhibited a tendency to incorporate more noise components, leading to poor classification accuracy, showcasing the importance of selecting the LP order for relatively accurate severity-level classification.

Comparative analysis against baseline features, namely, MFCC and LFCC, unveiled the superiority of LFRCC features in dysarthria severity-level classification. These findings were particularly significant as LFRCC features demonstrated comparable performance to state-of-the-art whisper model features, indicating their effectiveness in capturing dysarthria-specific characteristics. Moreover, the robustness of LFRCC features to noise emerged as a notable advantage, as they outperformed MFCC and LFCC features under both stationary and non-stationary noise conditions. Even at lower SNR, LFRCC features maintained high accuracy, preserving the integrity of excitation source characteristics within the residual, and thus, showcasing their suitability for real-world applications, where noise may be prevalent.

Further analysis of the latency period revealed the responsiveness of LFRCC features, achieving maximum accuracy with a relatively short input signal duration of 750 ms (equivalent to 30 frames). This responsiveness surpassed that of MFCC and LFCC features, highlighting the agility of LFRCC features in dysarthria severity-level classification tasks. In conclusion, the comprehensive evaluation of LFRCC features across multiple dimensions-LP order optimization, comparative analysis against baseline features, robustness to noise, and latency period analysis-underscored their efficacy and versatility in dysarthria severity-level classification. This work can further extended by feeding the residual based information to existing state-of-the-art deep learning models such as wav2vec 2.0, whisper model.

Acknowledgements. The authors sincerely thank the Ministry of Electronics and Information Technology (MeitY), New Delhi, Govt. of India, for sponsoring a consortium project titled ‘BHASHINI’, (Grant ID: 11(1)2022-HCC (TDIL)).

References

1. Lieberman, P.: Primate vocalizations and human linguistic ability. *J. Acoust. Soc. Am. (JASA)* **44**(6), 1574–1584 (1968)
2. Palmer, R., Enderby, P.: Methods of speech therapy treatment for stable dysarthria: a review. *Adv. Speech Lang. Pathol.* **9**(2), 140–153 (2007)
3. Kim, Y., Kent, R.D., Weismer, G.: An acoustic study of the relationships among neurologic disease, dysarthria type, and severity of dysarthria. *J. Speech Lang. Hear. Res. (JSLH)* **54**, 417–429 (2011)
4. Bhat, C., Vachhani, B., Kopparapu, S.K.: Automatic assessment of dysarthria severity-level using audio descriptors. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, New Orleans, LA, USA, 2017, pp. 5070–5074 (2017)
5. Farhadipour, A., Veisi, H., Asgari, M., Keyvanrad, M.A.: Dysarthric speaker identification with different degrees of dysarthria severity using deep belief networks. *ETRI J.* **40**(5), 643–652 (2018)
6. Kachhi, A., Therattil, A., Patil, A.T., Sailor, H.B., Patil, H.A.: Teager energy cepstral coefficients for classification of dysarthric speech severity-level. In: *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. Chiang Mai, Thailand, IEEE 2022, pp. 1462–1468 (2022)
7. Zhou, X., Garcia-Romero, D., Duraiswami, R., Espy-Wilson, C., Shamma, S.: Linear versus mel frequency cepstral coefficients for speaker recognition. In: *IEEE Workshop on Automatic Speech Recognition & Understanding 2011*, pp. 559–564 (2011). <https://doi.org/10.1109/ASRU.2011.6163888>
8. Gupta, P., Patil, H.A.: Linear frequency residual cepstral features for replay spoof detection on asvspoof 2019. In: *2022 30th European Signal Processing Conference (EUSIPCO)*, 2022, pp. 349–353 (2022)
9. Hora, B.S., Uthiraa, S., Patil, H.A.: Linear frequency residual cepstral coefficients for speech emotion recognition. In: Karpov, A., Samudravijaya, K., Deepak, K.T., Hegde, R.M., Agrawal, S.S., Prasanna, S.R.M. (eds.) *Speech and Computer. SPECOM 2023*. LNCS, vol. 14338, pp. 116–129. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-48309-7_10

10. Falk, T.H., Chan, W.Y., Shein, F.: Characterization of atypical vocal source excitation, temporal dynamics and prosody for objective measurement of dysarthric word intelligibility. *Speech Commun.* **54**(5), 622–631 (2012). advanced Voice Function Assessment. <https://doi.org/10.1016/j.specom.2011.03.007>, <https://www.sciencedirect.com/science/article/pii/S0167639311000513>
11. Vikram, C.M., Adiga, N., Prasanna, S.R.M.: Detection of nasalized voiced stops in cleft palate speech using epoch-synchronous features. *IEEE/ACM Trans. Audio Speech Lang. Process.* **27**(7), 1189–1200 (2019). <https://doi.org/10.1109/TASLP.2019.2913089>
12. Atal, B.S.: The history of linear prediction. *IEEE Signal Process. Mag.* **23**(2), 154–161 (2006)
13. Kadiri, S.R., Prasad, R., Yegnanarayana, B.: Detection of glottal closure instant and glottal open region from speech signals using spectral flatness measure. *Speech Commun.* **116**, 30–43 (2020). <https://doi.org/10.1016/j.specom.2019.11.004>, <https://www.sciencedirect.com/science/article/pii/S0167639318302292>
14. Kachhi, A., Therattil, A., Patil, A.T., Sailor, H.B., Patil, H.A.: Significance of energy features for severity classification of dysarthria. In: Prasanna, S.R.M., Karpov, A., Samudravijaya, K., Agrawal, S.S. (eds.) *Speech and Computer. SPECOM 2022. LNCS*, vol. 13721, pp. 325–337. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20980-2_28
15. Kim, H., Hasegawa-Johnson, M., Perlman, A., Gunderson, J.R., Watkin, K.L., Frame, S.: Dysarthric speech database for universal access research, pp. 1741–1744, September 2008. <https://doi.org/10.21437/Interspeech.2008-480>
16. Rudzicz, F., Namasivayam, A.K., Wolff, T.: The TORGO database of acoustic and articulatory speech from speakers with dysarthria. *Lang. Resour. Eval.* **46**, 523–541 (2012)
17. Gupta, S., et al.: Residual neural network precisely quantifies dysarthria severity-level based on short-duration speech segments. *Neural Netw.* **139**, 105–117 (2021)
18. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I., : Robust speech recognition via large-scale weak supervision (2022). [arXiv:2212.04356](https://arxiv.org/abs/2212.04356)
19. Yunusova, Y., Weismer, G., Westbury, J.R., Lindstrom, M.J.: Articulatory movements during vowels in speakers with dysarthria and healthy controls. *J. Speech Lang. Hear. Res.* **51**(3), 596–611 (2008)
20. Quatieri, T.F.: *Discrete-Time Speech Signal Processing: Principles and Practice*, Pearson Education, 3rd edition, India, 2006
21. Prasanna, S.R.M., Yegnanarayana, B.: Detection of vowel onset point events using excitation information. In: *Interspeech*, 2005. <https://api.semanticscholar.org/CorpusID:14413800>
22. Mahadeva Prasanna, S.R., Sandeep Reddy, B.V., Krishnamoorthy, P.: Vowel onset point detection using source, spectral peaks, and modulation spectrum energies. *IEEE Trans. Audio, Speech Lang. Process.* **17**(4), 556–565 (2009). <https://doi.org/10.1109/TASL.2008.2010884>
23. Murty, K.S.R., Yegnanarayana, B.: Combining evidence from residual phase and MFCC features for speaker recognition. *IEEE Signal Process. Lett.* **13**(1), 52–55 (2005)



Generating High-Quality Symbolic Music Using Fine-Grained Discriminators

Zhedong Zhang¹, Liang Li²(✉), Jiehua Zhang³, Zhenghui Hu⁴, Hongkui Wang¹, Chenggang Yan¹(✉), Jian Yang⁵, and Yuankai Qi⁵

¹ Hangzhou Dianzi University, Hangzhou, Zhejiang, China

² Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

³ Xi'an Jiaotong University, Xi'an, Shaanxi, China

⁴ Hangzhou Innovation Institute, Beihang University, Hangzhou, Zhejiang, China

⁵ Macquarie University, Sydney, NSW, Australia

Abstract. Existing symbolic music generation methods usually utilize discriminator to improve the quality of generated music via global perception of music. However, considering the complexity of information in music, such as rhythm and melody, a single discriminator cannot fully reflect the differences in these two primary dimensions of music. In this work, we propose to decouple the melody and rhythm from music, and design corresponding fine-grained discriminators to tackle the aforementioned issues. Specifically, equipped with a pitch augmentation strategy, the melody discriminator discerns the melody variations presented by the generated samples. By contrast, the rhythm discriminator, enhanced with bar-level relative positional encoding, focuses on the velocity of generated notes. Such a design allows the generator to be more explicitly aware of which aspects should be adjusted in the generated music, making it easier to mimic human-composed music. Experimental results on the POP909 benchmark demonstrate the favorable performance of the proposed method compared to several state-of-the-art methods in terms of both objective and subjective metrics. The source code and more demos are available at <https://github.com/ZZDoog/fine-grained-music-discriminators>.

1 Introduction

Due to the high-level representation of music based on Musical Instrument Digital Interface (MIDI) and its variants, symbolic music generation models do not need to learn how to create the sounds of various instruments so that they can focus more on the music itself [4, 14, 21]. Since the high-level discrete tokens of music are similar to words of text, transformer-based models have been widely applied in symbolic music generation, and towards the goal of generating high-quality music in recent years [2, 3, 10–12, 17, 18, 22, 25, 27, 31, 33]. Most symbolic music generation models are trained to maximize the likelihood of observed sequences. These methods can learn the patterns of discrete token sequences

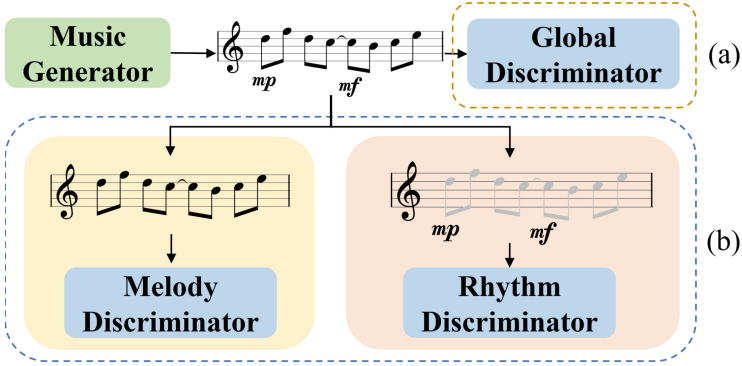


Fig. 1. (a) Main structure of conventional GAN-based method with coarse-grained global discriminator. (b) The structure of proposed fine-grained discriminators architecture.

and ensure statistical consistency, but they may suffer from noticeable quality degradation when generating complex music sequences due to exposure bias [20].

Some studies [6, 20, 32] have attempted to address the aforementioned issues by introducing adversarial loss [8]. They enhance the generative model by utilizing the feedback from the discriminator based on the discriminator’s discernment on generated and real music. Despite the progress, their discriminators cannot explicitly reflect the defects in terms of two important music properties: melody and rhythm, due to the lack of corresponding designs. According to music perception theory, melody and rhythm are two primary dimensions of music [7]. They respectively represent the arrangement of musical pitches in a particular order and the progression patterns of notes, which constitute the core of music composition [13]. Well-sounding music should feature a stable melody with rich variation, supported by a rhythmic framework that maintains smooth and varied progressions [7]. Lacking an effective targeted model, the quality of music generated by existing methods is limited.

To address the above problems, we propose a novel architecture with fine-grained discriminators for symbolic music generation, as shown in Fig. 1. Aiming to provide fine-grained adversarial feedback to the generator, we first design a decoupling module to well disentangle the melody and rhythm information from music. Specifically, we mask all the note velocity and note pitch tokens with the same token [Mask] in the sequence respectfully to extract melody and rhythm information from the original music sequence. After decoupling, we design the corresponding fine-grained melody and rhythm discriminator for the generator. To discriminate whether the melodies of generated music closely resemble real data, a pitch augmentation strategy is used in the melody discriminator to reduce the impact of the absolute pitch. Correspondingly, we design a fine-grained rhythm discriminator elaborately. By devising a bar-level relative positional encoding to enhance the discriminator to better capture the rhythm pattern within the local structure.

The contributions of this paper are summarized below:

- We propose a fine-grained discriminators architecture for melody and rhythm respectfully in symbolic music generation domain, which is more aligned with music perception theories.
- We design a melody-rhythm decoupling module for symbolic music and incorporate pitch enhancement strategies and bar-level relative position encoding to enhance the corresponding fine-grained discriminators, providing elaborate feedback to the generator.
- Extensive experiments show the favorable performance of our method in terms of both objective metrics and subjective listening tests. More generated examples are available at materials.

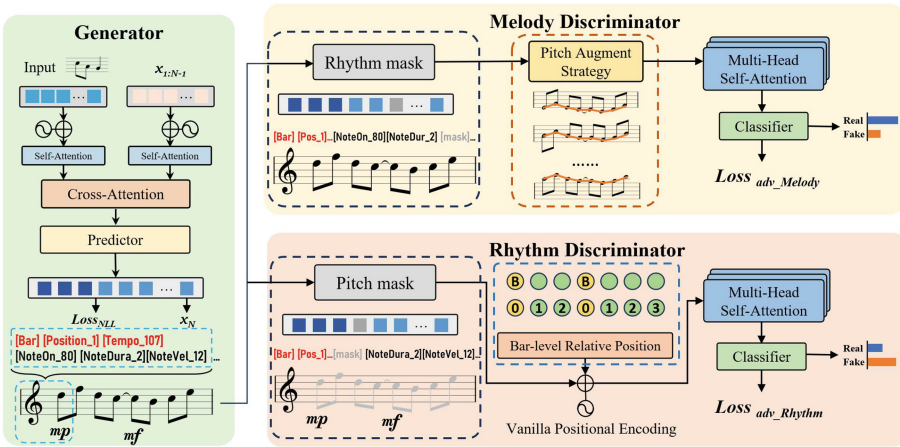


Fig. 2. Main framework of the proposed symbolic music generation model, consists of three main components: a music generator and two fine-grained discriminators—rhythm discriminator and melody discriminator.

2 Methodology

The proposed model consists of an auto-regressive symbolic music generator and two fine-grained discriminators as shown in Fig. 2. First, the generator takes a representative condition music sequence c as the input, and attempts to generate whole music sequence align with the condition. Then, during the generator optimization, the fine-grained melody and rhythm discriminators provide more precise feedback to the generator by decoupling and analyzing the output of the generator. Simultaneously, the fine-grained discriminators continually enhance their discriminatory abilities relying on samples generated by the evolving generator to provide further feedback to the generator. The value function of the

generator and fine-grained melody and rhythm discriminators are defined as follows:

$$\begin{aligned} \min_{G_\theta} \max_{D_m, D_r} V = & \{\mathbb{E}[\log D_m(s_r)] + \mathbb{E}[\log D_r(s_r)] \\ & + \mathbb{E}[\log(1 - D_m(G_\theta(c)))] + \mathbb{E}[\log(1 - D_r(G_\theta(c)))]\}, \end{aligned} \quad (1)$$

where the G , D_m , and D_r denote the generator, melody discriminator, and rhythm discriminator respectively. θ and s_r denote the parameter of the generator and real sample from the dataset respectively.

2.1 Generator

We adopt the seq2seq symbolic music generation transformer model [23] as our generator. It takes condition music sequence as input and generates a complete and harmonious music composition that aligns with the input. The condition music sequence is the thematic material of each music composition, implies the main idea of the whole composition, retrieved from the complete music by clustering algorithm [23]. The overall loss function of the generator as follows:

$$\mathcal{L}_G = \mathcal{L}_{NLL} + \alpha \cdot \mathcal{L}_{adv_Melody} + \beta \cdot \mathcal{L}_{adv_Rhythm}, \quad (2)$$

$$\mathcal{L}_{NLL} = \sum_{n=1}^N -\log P(x_n | \theta, x_{1:n-1}, c), \quad (3)$$

where the α and β are pre-defined hyper-parameters. Details of the two adversarial losses are in the following sections. Note that our fine-grained discriminators architecture applies equally to other state-of-the-art music generation methods.

2.2 Melody Discriminator with Pitch Augmentation Strategy

Melody is one of the primary properties of music. It provides a tuneful and recognizable musical line that serves as a focal point for listeners. The arrangement of pitches in a particular order and duration forms the melody [19]. Traditional NLL-trained models perform poorly in generating long and harmonious melodies due to the lack of specific guidance. To deal with this issue, we propose a melody discriminator with a pitch augmentation strategy to facilitate the discrimination of the melody in generated music.

First, we decouple the melody information from symbolic music by replacing all the [Note-Velocity] tokens with the [mask] token. Then, to enhance our melody discriminator, we augment the original data via uniformly raising or decreasing the absolute pitch of all original notes to simulate the melody in different voice parts, as shown in Fig. 2 top right. All these decoupled melody data are fed into the melody discriminator which uses an encoder-only transformer with a multi-head self-attention mechanism as backbone [26]. During the

adversarial training process, the adversarial loss from the melody discriminator and back-propagation gradients to the melody discriminator are formulated as:

$$\begin{aligned} \mathcal{L}_{adv_Melody} &= \frac{1}{N} \sum_{i=1}^N [\log(1 - D_m(G_\theta(c^{(i)})))] \\ \nabla_{\theta_m} \frac{1}{N} \sum_{i=1}^N &[\log(D(s_r^{(i)}) + \log(1 - D_m(G_\theta(c^{(i)})))] \end{aligned} \tag{4}$$

where θ_m denotes the parameter of melody discriminator, $s_r^{(i)}$ and $c_r^{(i)}$ denote as i -th ground truth and conditional input.

2.3 Rhythm Discriminator with Bar-Level Relative Positional Encoding

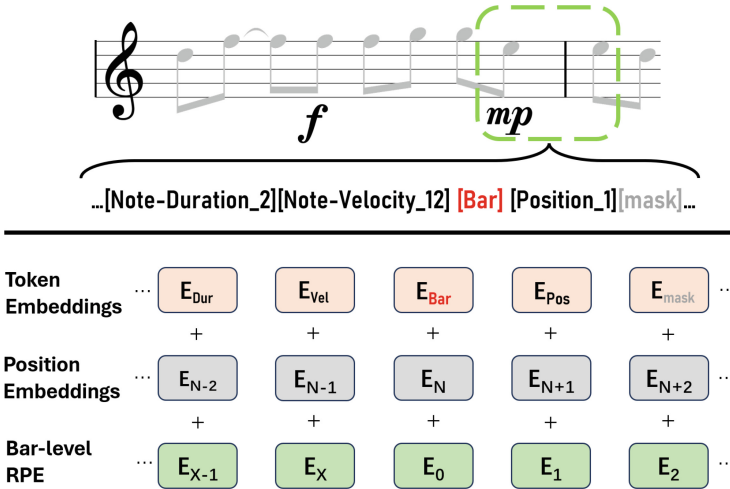


Fig. 3. Illustration of the proposed bar-level relative positional encoding (RPE). The relative position accumulates from the previous [Bar] token to the next [Bar] token, implemented by learnable embedding, and then added to the token embedding with the vanilla positional embedding.

In addition to melody, rhythm is another crucial property of music, as it reflects the progression of notes and variations in velocity, governing the dynamics of music [9]. To improve the quality in terms of rhythm, we design a fine-grained rhythm discriminator.

To facilitate the discriminator to focus on rhythms instead of other music elements, we decouple rhythm information from the music by replacing the [Note-On-Pitch] token with the [Mask] token. Apart from that, we observe

that the symbol “bar” plays a fundamental role in organizing and structuring music, which therefore can help establish the rhythmic framework of the music [16]. Based on this observation, we introduce a bar-level relative positional encoding as shown in Fig. 3. It accumulates position starting from the beginning of each bar and resets at the beginning of the next bar, *i.e.*, from [Bar] token to next [Bar] token, embeds the bar-level relative position information into the decoupled music rhythm sequence. Like other relative position embedding implementations, our bar-level position embedding is also learnable. The general position encoding of a symbolic music token, e.g., t -th in the whole sequence and x -th within the current bar, is defined as follows:

$$A_{t,x} = \cos/\sin(t/1000^{2i/d}) + W_{BRPE}[\delta(x, 1), \delta(x, 2), \dots], \quad (5)$$

where the first part is traditional sine and cosine position encoding in the Transformer, the W_{BRPE} is a learnable matrix, and $\delta(\cdot)$ is dirichlet function. The rhythm discriminator shares similar back-propagation and adversarial loss to the generator as the melody discriminator in Eq. (4).

3 Experiments

3.1 Experimental Setting

Dataset and Preprocess. We employ the POP909 dataset [28] for performance evaluation. There are three separate tracks in each arrangement in the dataset: MELODY, BRIDGE and PIANO. To encode a MIDI file into a sequence of discrete tokens, we adopt the REMI-like [12] encoding method. In detail, we use metric-related tokens [Bar], [Tempo], [Position] and note-related tokens [Note-On-Pitch], [Note-Duration] and [Note-Velocity] to represent music, as shown in the generator part of Fig. 2. For fair comparisons, we retrain all the baseline models using the same data as ours, and reserve 4% of them only for evaluation where all models take the same music piece as the condition or the prefix sequence.

Implementation Details. The proposed melody and rhythm discriminators use a 6-layer encoder-only Transformer as the backbone. Both of them have 8 heads for multi-head attention, 256 hidden dimensions, 1,024-dim feed-forward layers, and ReLU as the activation function. In the first stage, we pre-train the generator along with all baseline models using Adam optimizer ($\beta_1 = 0.9$ and $\beta_2 = 0.99$) [15] until the training NLL loss model below 0.55. Afterward, we pre-train the melody and rhythm discriminator using the dataset and the output of the trained generator for 120 epochs. During adversarial training, both α and β are set to 0.15, and using the same optimizer in the first stage to train the generator for 100 epochs.

Baselines. 1) **GT** [28]: the above-mentioned 4% of the dataset which is not included in the training set or validation set. 2) **Music Transformer (MT)** [11]: pioneer algorithm that successfully applied the transformer model to

the domain of symbolic music generation. **3) Theme Transformer (TT)** [23]: a theme-conditioned music generation model optimized by NLL loss only. **4) Anticipatory Music Transformer (AMT)** [24]: the current state-of-art model for piano music generation based on transformer. **5) WGAN** [32]: music generation model that utilizes a conventional global discriminator which will primarily serve to validate the effectiveness of our proposed fine-grained discriminator approach. **6,7) Ours (wRo) & Ours (wMo)**: our model uses only rhythm discriminator or melody discriminator in the adversarial training. **8) Ours**: the complete fine-grained discriminator model.

3.2 Objective Evaluation

Evaluation Metrics. We employ various metrics to demonstrate the comprehensive performance of the models. First, following [5, 29], we adopt **1) pitch class entropy**, **2) scale consistency**, and **3) groove consistency** to evaluate entropy of the normalized note pitch class histogram, largest pitch-in-scale rate over all major and minor scales, and mean hamming distance of the neighboring measures. Then, we calculate the **4) pitch** and **5) velocity divergence** between the generated and real music to measure the distribution similarity in melody and rhythm respectively. Furthermore, we utilize a pre-trained music understanding model MIDI-BERT [1] and transform the music into feature vectors. The cosine **6) MIDI-BERT similarity** between generated and real music can measure the proximity of generated music to real music in a higher-level feature.

Table 1. The results of objective evaluation. For the pitch class entropy, groove consistency and scale consistency, a closer value to that of ground truth is considered better. Mean values and 95% confidence intervals are reported. Red and blue fonts denote the best and second-best performance, respectively.

	Pitch Class Entropy	Groove Consistency	Scale Consistency	Pitch Divergence↓	Velocity Divergence↓	MIDI-BERT Similarity↑
GT [28]	2.7726 ± 0.0012	0.9889 ± 0.0023	0.9799 ± 0.0029	-	-	-
MT [11]	2.5907 ± 0.0035	0.9876 ± 0.0029	0.9634 ± 0.0039	0.7092 ± 0.0085	0.3529 ± 0.0089	0.3073 ± 0.0046
TT [23]	2.6749 ± 0.0073	0.9572 ± 0.0038	0.9706 ± 0.0020	0.1470 ± 0.0007	0.0904 ± 0.0011	0.2809 ± 0.0027
AMT [24]	2.7133 ± 0.0094	0.9165 ± 0.0043	0.9792 ± 0.0033	1.3645 ± 0.0165	0.6346 ± 0.0132	0.2921 ± 0.0025
WGAN [20]	2.6437 ± 0.0129	0.9575 ± 0.0064	0.9739 ± 0.0074	0.1516 ± 0.0012	0.0824 ± 0.0010	0.2733 ± 0.0012
Ours (wRo)	2.7123 ± 0.0088	0.9579 ± 0.0020	0.9735 ± 0.0047	0.1598 ± 0.0028	0.0625 ± 0.0012	0.3103 ± 0.0030
Ours (wMo)	2.7590 ± 0.0021	0.9553 ± 0.0045	0.9743 ± 0.0035	0.1368 ± 0.0014	0.0726 ± 0.0031	0.3205 ± 0.0028
Ours	2.7164 ± 0.0024	0.9583 ± 0.0022	0.9763 ± 0.0026	0.1282 ± 0.0013	0.0675 ± 0.0021	0.3239 ± 0.0026

Compared with SOTA Methods. Table 1 shows the comparison results. We can observe that the introduction of the fine-grained melody discriminator makes our model closer to the ground truth on pitch class entropy, scale consistency, and pitch distribution divergence. Regarding rhythm, we can see that our model

Table 2. The results of the subjective evaluation. Mean values and 95% confidence intervals are reported.

	Coherence	Richness	Overall
GT [28]	3.98 ± 0.18	4.15 ± 0.23	4.06 ± 0.10
MT [11]	3.56 ± 0.24	2.94 ± 0.18	3.37 ± 0.33
TT [23]	3.21 ± 0.20	3.48 ± 0.16	3.44 ± 0.32
AMT [24]	3.43 ± 0.34	3.32 ± 0.27	3.39 ± 0.24
WGAN [20]	3.60 ± 0.35	3.57 ± 0.29	3.59 ± 0.26
Ours	3.68 ± 0.28	3.81 ± 0.25	3.71 ± 0.26

and Music Transformer [11] outperform other models in groove consistency, suggesting better rhythm control and more stable groove. With the inclusion of the rhythm discriminator, music generated by our model is also closer to ground truth in note velocity distribution. In terms of MIDI-BERT similarity, which largely reflects the overall quality of the generated music, our complete model achieves the highest average similarity. This suggests that according to the pre-trained music understanding model, music generated by our model exhibits a closer resemblance to human music compositions, both in style and musical quality.

Overall, compared to the conventional GAN-based baseline model WAGN [20] and other benchmark models, our model achieves superior performance in objective performance metrics attributable to the fine-grained tuning of the generator by melody and rhythm discriminators.

3.3 Subjective Evaluation

To assess the quality of music samples generated by our model, we conduct a listening test with 17 survey participants. Ten of them can play at least one musical instrument and understand basic music theory. We provide 6 sets of 30 music samples for participants, consisting of ground truth and samples generated by each model. All generated MIDI files are rendered to audio using MuseScore General SoundFont [5]. In the questionnaire, each participant is asked to listen to all 30 samples and then rate them on a scale of 1 to 5 according to three criteria—*coherence*, *richness*, and *overall*. Results are reported in Table 2.

The results show that our model achieves higher scores across all criteria than other models. It’s worth noticing that while Music Transformer [11] surpasses our model in terms of the groove consistency metric in objective evaluation, it is less favorable than our model in terms of coherence and richness in subjective listening tests, especially richness. Based on the feedback from survey participants, we find that the music generated by Music Transformer contains a larger amount of repetition, leading to a monotonous listening experience. Benefiting from the fine-grained adversarial optimization, our model outperforms the SOTA single

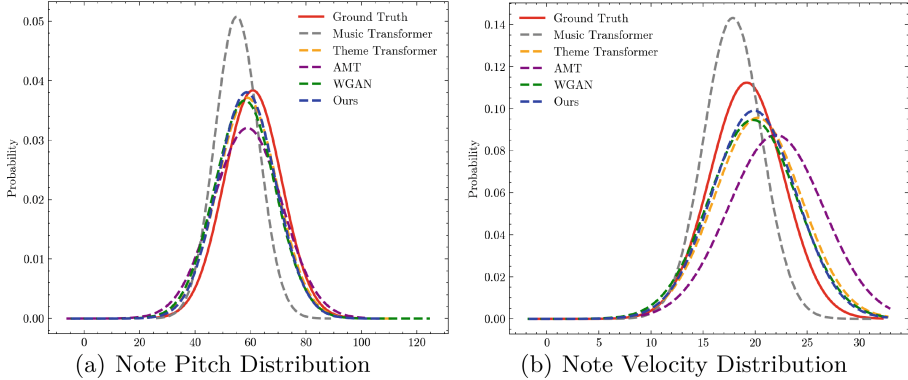


Fig. 4. Quantitative analysis. (a) & (b) Visualization of the note pitch and note velocity distribution of music generated by different models and the Ground Truth.

discriminator method WGAN. The performance improvements demonstrate the effectiveness of our method on both coherence and richness aspects and overall quality.

3.4 Ablation Studies and Qualitative Evaluation

As shown in Table 1, when using only the fine-grained melody discriminator, our method has shown a significant improvement compared to other baseline models in metrics strongly related to melody such as pitch class entropy, scale consistency, and pitch divergence, reaching a closer level to real music. Moreover, since melody and pitch are the core components of music expression [30], the melody discriminator enables the model to generate more realistic music, as indicated by the outstanding MIDI-BERT similarity. When solely using the fine-grained rhythm discriminator, our method also achieves better performance than baseline models in velocity divergence and MIDI-BERT similarity, proving the effectiveness of both fine-grained discriminators.

Figure 4(a) and (b) visualize the pitch and velocity distribution between the music generated by different models and the ground truth. We approximate each distribution to a normal distribution for better comparison. It can be observed that benefiting from the fine-grained melody and rhythm discriminators, our model is closer to real music in both attributes.

To better evaluate the effectiveness of our method, we compute the feature vectors of music generated by each model along with the ground truth using the pre-trained music understanding model MIDI-BERT [1]. We utilize PCA algorithm to reduce the high-dimensional feature vectors to 2 dimensions and visualize the feature vectors in Fig. 5. It is evident that the music generated by other baseline models exhibits a noticeable domain gap compared to the ground truth, while MIDI-BERT can effectively distinguish whether they are real music or not. Music Transformer [11] and AMT [24] are trained solely using maxi-

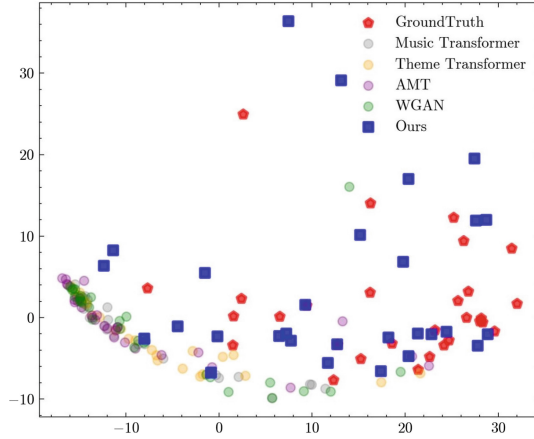


Fig. 5. The PCA visualization results of music feature obtained from MIDI-BERT [1].



Fig. 6. WGAN [20], AMT [24], and our model’s generated examples and their corresponding ground truth music piece.

mum likelihood as a training objective function, thus they suffer from quality degradation caused by exposure bias. Therefore, music generated by them has a certain degree of uniformity, with their feature vector distributions being highly concentrated.

WGAN [20] alleviates quality degradation through adversarial loss, resulting in a more dispersed distribution of its feature vectors compared to the previous two, but still exists a considerable gap from the ground truth. Equipped with fine-grained discriminators, our model’s style vectors exhibit a distribution that closely resembles the ground truth and also diversifies itself.

Figure 6 illustrates examples generated by both our model, WGAN [20] and AMT [24] when giving the same condition. Compared to other models, our generated example exhibits a closer resemblance to the ground truth in terms of melody design and transitions (highlighted by the green bounding boxes). The example from AMT [24] and WGAN [20] exhibits some disharmony caused by abnormal notes and discrepant rhythm patterns (highlighted by the red bounding box).

4 Conclusion

This work proposes a fine-grained discriminators architecture for the symbolic music generation task. We decouple the music into melody and rhythm for independent discrimination, which provides the generator with more specific feedback. We also devise a pitch augment strategy and a bar-level relative positional encoding scheme to enhance the learning of melody discriminator and rhythm discriminator, respectively. Extensive objective and subjective results demonstrate the effectiveness of the proposed method.

Acknowledgement. This work was supported by the “Leading Goose” R&D Program of Zhejiang Province under grants (2024C01107, 2023YFB4502800, 2023YFB4502803), National Natural Science Foundation of China: 62322211, 62336008, “Pioneer” and “Leading Goose” R&D Program of Zhejiang Province (2024C01023), Zhejiang Provincial Natural Science Foundation of China under Grant No. LD24F020007, Zhejiang Provincial Natural Science Foundation of China(LDT23F01011F01, LDT23F01015F01, LDT23F01014F01), Key Laboratory of Intelligent Processing Technology for Digital Music (Zhejiang Conservatory of Music), Ministry of Culture and Tourism (2023DMKLB004). Yuankai Qi and Jian Yang are not supported by the aforementioned fundings.

References

1. Chou, Y.H., Chen, I., Chang, C.J., Ching, J., Yang, Y.H., et al.: MidiBERT-piano: large-scale pre-training for symbolic music understanding. arXiv preprint [arXiv:2107.05223](https://arxiv.org/abs/2107.05223) (2021)
2. Cong, G., et al.: Learning to dub movies via hierarchical prosody models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14687–14697 (2023)
3. Cong, G., et al.: Styledubber: towards multi-scale style learning for movie dubbing. arXiv preprint [arXiv:2402.12636](https://arxiv.org/abs/2402.12636) (2024)
4. Dhariwal, P., Jun, H., Payne, C., Kim, J.W., Radford, A., Sutskever, I.: Jukebox: a generative model for music. CoRR abs/2005.00341 (2020)
5. Dong, H.W., Chen, K., Dubnov, S., McAuley, J., Berg-Kirkpatrick, T.: Multitrack music transformer. In: ICASSP, pp. 1–5 (2023)
6. Dong, H., Hsiao, W., Yang, L., Yang, Y.: MuseGAN: multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In: AAAI, pp. 34–41 (2018)

7. Dowling, W.J., Tighe, T.J.: *Psychology and music: the understanding of melody and rhythm*. Psychology Press (2014)
8. Goodfellow, I., et al.: Generative adversarial networks. *Commun. ACM* **63**(11), 139–144 (2020)
9. Honing, H.: Structure and interpretation of rhythm in music. *Psychol. Music* **3**, 369–404 (2013)
10. Hsiao, W., Liu, J., Yeh, Y., Yang, Y.: Compound word transformer: learning to compose full-song music over dynamic directed hypergraphs. In: *AAAI*, pp. 178–186 (2021)
11. Huang, C.A., et al.: Music transformer: generating music with long-term structure. In: *ICLR* (2019)
12. Huang, Y., Yang, Y.: Pop music transformer: beat-based modeling and generation of expressive pop piano compositions. In: *ACM MM*, pp. 1180–1188 (2020)
13. Jeong, D., Kwon, T., Kim, Y., Nam, J.: Score and performance features for rendering expressive music performances. In: *Music Encoding Conference*, pp. 1–6 (2019)
14. Ji, S., Luo, J., Yang, X.: A comprehensive survey on deep music generation: multi-level representations, algorithms, evaluations, and future directions. *CoRR abs/2011.06801* (2020)
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings* (2015)
16. Levitin, D.J., Grahn, J.A., London, J.: The psychology of music: rhythm and movement. *Annu. Rev. Psychol.* **69**, 51–75 (2018)
17. Li, L., Gao, X., Deng, J., Tu, Y., Zha, Z.J., Huang, Q.: Long short-term relation transformer with global gating for video captioning. *IEEE Trans. Image Process.* **31**, 2726–2738 (2022)
18. Liu, X., et al.: Entity-enhanced adaptive reconstruction network for weakly supervised referring expression grounding. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(3), 3003–3018 (2022)
19. McDermott, J.H., Oxenham, A.J.: Music perception, pitch, and the auditory system. *Curr. Opin. Neurobiol.* **18**(4), 452–463 (2008)
20. Muhamed, A., et al.: Symbolic music generation with transformer-GANs. In: *AAAI*, pp. 408–417 (2021)
21. van den Oord, A., et al.: Wavenet: a generative model for raw audio. In: *The 9th ISCA Speech Synthesis Workshop*, p. 125 (2016)
22. Ren, Y., He, J., Tan, X., Qin, T., Zhao, Z., Liu, T.: Popmag: pop music accompaniment generation. In: *ACM MM*, pp. 1198–1206 (2020)
23. Shih, Y., Wu, S., Zalkow, F., Müller, M., Yang, Y.: Theme transformer: symbolic music generation with theme-conditioned transformer. *TMM Early Access* 1–14 (2022)
24. Thickstun, J., Hall, D., Donahue, C., Liang, P.: Anticipatory music transformer. *CoRR abs/2306.08620* (2023)
25. Tu, Y., Li, L., Su, L., Zha, Z.J., Huang, Q.: Smart: syntax-calibrated multi-aspect relation transformer for change captioning. *IEEE Trans. Pattern Anal. Mach. Intell.* (2024)
26. Vaswani, A., et al.: Attention is all you need. In: *NeurIPS*, pp. 5998–6008 (2017)
27. Wang, H., Zha, Z.J., Li, L., Chen, X., Luo, J.: Semantic and relation modulation for audio-visual event localization. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(6), 7711–7725 (2022)

28. Wang, Z., et al.: POP909: a pop-song dataset for music arrangement generation. In: ISMIR, pp. 38–45 (2020)
29. Wu, S., Yang, Y.: The jazz transformer on the front line: exploring the shortcomings of AI-composed music through quantitative measures. In: ISMIR, pp. 142–149 (2020)
30. Yang, R., Wang, D., Wang, Z., Chen, T., Jiang, J., Xia, G.: Deep music analogy via latent representation disentanglement. In: Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, pp. 596–603 (2019)
31. Zhang, B., et al.: Inductive state-relabeling adversarial active learning with heuristic clique rescaling. *IEEE Trans. Pattern Anal. Mach. Intell.* (2024)
32. Zhang, N.: Learning adversarial transformer for symbolic music generation. *IEEE Trans. Neural Netw. Learn. Syst.* **34**(4), 1754–1763 (2023)
33. Zhang, Z., et al.: From speaker to dubber: movie dubbing with prosody and duration consistency learning. In: *ACM Multimedia 2024* (2024)



Enduring Memory Self-learning Multi-level Transformer Network for Remote Sensing Image Super-Resolution

Peishan Li, Yonghong Zhang^(✉), Junfei Wang, and Guangyi Ma

Nanjing University of Information Science and Technology, Nanjing, Jiangsu, China
{zyh, jf_wang, gyma}@nuist.edu.cn

Abstract. High-resolution (HR) remote sensing is essential for remote sensing image interpretation, but challenges in super-resolution (SR) stem from scale and texture differences within images, neglecting high-dimensional detail extraction and long-range dependencies among multi-dimensional features. Addressing this, we propose TDSNet, an enduring memory self-learning multi-level Transformer Network for remote sensing image super-resolution (RISSR). The utilization of the similarity balancing module and memory gated group establishes connections between mixed-scale information, while also possessing enduring memory across various receptive fields. Shallow and deep-level data fuse in the transformer, employing a dual learning strategy, enhancing reconstruction through a constrained mapping process with a loss function. This transforms the ill-posed problem into a well-posed one. Attribution analysis with the LAM method reveals TDSNet's efficacy in capturing content information. Experiments on NWPU-RESISC45 and AID datasets demonstrate TDSNet's superior performance in remote sensing image super-resolution compared to other methods.

Keywords: Dual learning · transformer · remote sensing image super-resolution (RSISR) · super-resolution (SR)

1 Introduction

Super-resolution (SR) is a technique that enhances the resolution of images by processing low-resolution (LR) images, finding applications in diverse fields such as computer vision, image and video compression, medical imaging, and satellite remote sensing [1]. In satellite remote sensing, crucial for various domains like agriculture, meteorology, and military, the spatial and spectral resolution often falls short due to limitations of Earth observation satellites in optical and sensor equipment, hindering high-resolution remote sensing images acquisition. Super-resolution algorithms reconstruct high-resolution images from low-resolution counterparts, addressing challenges related to expensive sensors [2] and improving recognizability by recovering lost information. This enhances the accuracy of tasks like image recognition and detection [3].

Traditional image super-resolution methods, like interpolation [4], and reconstruction [5], have limitations. Interpolation, rearranging pixels, struggles with texture and

high-frequency details, resulting in jagged contours. Reconstruction methods, involving complex mathematical operations, offer superior texture and detail restoration but grapple with nonlinear noise, inability to model complex relationships, and substantial computational requirements. Recent attention has shifted to deep learning-based super-resolution algorithms [6–8], leveraging deep learning to predict missing high-frequency information in LR images by learning the mapping between LR and high-resolution (HR) image spaces. These techniques automatically extract relevant features from images, offering a time-efficient solution.

To address the challenges outlined above and extract intricate structures and texture details efficiently from extensive remote sensing images, this paper introduces the Enduring Memory Self-Learning Multi-Level Transformer Network with Dual Regression Mechanism (**TDSNET**). The network, combining a dual-learning mechanism with a Transformer framework, considers mixed-scale texture features and incorporates long-term memory capabilities. In the feature extraction phase, the model utilizes a combination of aggregated self-similar mixed-scale feature blocks and memory blocks with a gating mechanism. This enhances the network's discriminative capabilities, resulting in more precise extraction of both shallow and deep features. During the reconstruction phase, the model synergizes the strengths of the Swin Transformer and convolutional layers to effectively explore the correlation between high-dimensional and low-dimensional features. Additionally, a dual-learning mechanism is implemented to restrict the reverse mapping of loss function, effectively optimizing the solution domain for reconstruction and mitigating the ill-posed nature inherent in the super-resolution process. The primary contributions are as follows:

- We introduce a multi-level Transformer-enhanced network utilizing a dual-regression mechanism. This network utilizes a dual-learning approach to govern the opposite mapping process, integrating multi-scale high-dimensional and low-dimensional information from remote sensing data. The primary focus is on reconstructing finer structures and textures, leading to the generation of more accurate super-resolution images.
- We design a Utilize similarity balancing module that capitalizes on the internal recursive nature of information in remote sensing images. This module establishes skip connections between mixed-scale information, simulating interdependencies among different channels and spatial positions. Additionally, we introduce a Memory gated group to learn increasingly abstract feature representations and explore persistent memory through an adaptive learning process.
- The proposed network is evaluated on two publicly available remote sensing benchmark datasets, NWPU-RESISC45 and AID, and compared with eleven state-of-the-art methods. Experimental results highlight the method's effectiveness in achieving superior super-resolution effects in both accuracy and visual performance.

2 Related Work

Local image patterns exhibit recurring small patches at various scales, showcasing similar contours and textures [9]. This self-similarity, indicative of internal data redundancy, proves crucial for tasks like image denoising [10], deblurring [11], and super-resolution

reconstruction [12]. Early work by Glasner et al. [13] proposed a unified framework that integrates multi-image super-resolution (SR) and example-based SR, leveraging repeated patches within and across image scales. Recently, there has been a growing interest in enhancing the resolution of remote sensing images, captured from altitudes of hundreds of kilometers, facing challenges from diverse scenes and variable weather conditions. In remote sensing super-resolution (SR), sparsity and self-similarity play pivotal roles. While sparse representation methods initially dominated, with Dong et al. [14] introducing a coupled sparse autoencoder, recent deep learning methods [15, 16] surpass sparse representation-based approaches. Xia et al. [17] integrates meta-learning with MCMC to optimize kernel priors and employs a plug-and-play framework for unsupervised blind super-resolution. LGCNet [18], the first CNN-based model, incorporates multiple CNN layers for image residuals. HSENet [12] explores mixed-scale self-similarity in remote sensing images using non-local attention. Lei et al. [34] integrates multi-scale features through convolutional extraction and hierarchical super-resolution reconstruction. Liu et al. [19] utilize a dual-learning graph neural network, constraining the mapping for more accurate remote sensing SR results.

3 Methodology

3.1 Overview of TDSNET

Unlike natural images, remote sensing images display intricate textures and structural details, complicating the learning of mapping relationships between Low-Resolution (LR) and High-Resolution (HR) images. Moreover, the scale range of objects in remote sensing images is extensive, surpassing the capabilities of single-level features in fulfilling the requirements of super-resolution (SR) tasks. Thus, considering mixed-scale features, persistent memory, and long-range dependencies becomes crucial. Achieving an optimal solution is paramount. As a typical ill-posed problem, traditional deep learning-based methods often produce images with artifacts due to the vast solution space. To tackle these challenges, this paper introduces TDSNET, an innovative remote sensing SR algorithm building upon the classical SR model RCAN [20]. Illustrated in Fig. 1, TDSNET comprises four main sections: shallow feature extraction, deep feature extraction, super-resolution reconstruction, and dual-learning (indicated by the red solid line).

The shallow feature extraction section involves three primary operations: downsampling, the initial three layers of VGG19, and the CLC module (Conv-LeakyReLU-Conv), extracting partial semantic information and shallow features. The deep feature extraction section consists of two main operations: the Utilize similarity balancing module (USBM) and the Memory gated group (MGG). The former utilizes channel and spatial attention, establishing skip connections among mixed-scale information, while the latter learns deep feature representations and forms long-term memory through an adaptive learning process. Subsequently, the super-resolution reconstruction section includes three main operations: upsampling, the Residual.

Conv-swintransformer structure (RCSTS), and iterative upsampling-downsampling. Leveraging internal features for detailed reconstruction, this section transfers shallow information from the network's front to later stages through shortcuts, producing more

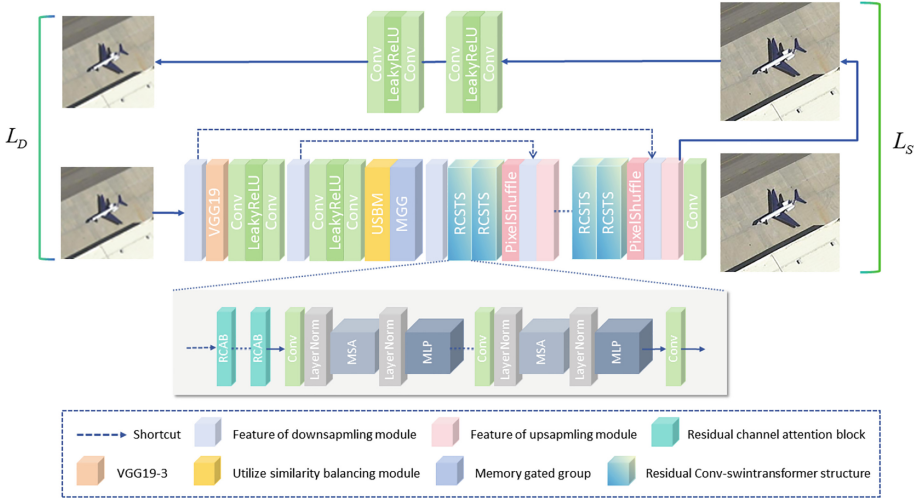


Fig. 1. Framework of the proposed TDSNet.

details with HR features. In RCSTS, several Residual channel attention blocks (RCAB) [20] are used to construct each basic block to improve the module's capacity. Finally, the dual-learning section employs simple CLC modules with lower computational costs compared to the original model, allowing the network to learn the transformation from SR images to the original LR images, thereby reducing the solution space for SR. The image degradation process is described as follows: $I_{LR} = I_{HR} \downarrow_{bic}$ where represents the LR image I_{LR} , represents the HR image I_{HR} , and represents bicubic downsampling \downarrow_{bic} . We simplify the SR reconstruction process with the following equation: $I_{SR} = G(I_{LR})$ where represents the SR image I_{SR} , and $G(\cdot)$ is the proposed method in this paper.

3.2 Utilize Similarity Balancing Module

In Fig. 2, we devised the USBM module, integrating the Sub-balanced attention module (SBAM) to enhance spatial attention (SA) and channel attention (CA) mechanisms. Operating on an intermediate feature map, the SBAM block acquires SA and CA mappings, optimizing the input feature map adaptively through attention-based mappings. The mapping function of SBAM is represented as follows:

$$\begin{aligned}
 T_{CA} &= f_{\text{sigmoid}} \left(f_{\text{conv}}^{k1s1} \left(f_{\text{ReLU}} \left(f_{\text{conv}}^{k1s1} \left(f_{\text{AvePool}}(F_0) \right) \right) \right) \right) \\
 M_{\text{SBAM}}(F_0) &= f_{\text{sigmoid}} \left(f_{\text{conv}}^{k7s1} \left(f_{\text{MaxPool}}(T_{CA}) \right) \right)
 \end{aligned} \tag{1}$$

The input shallow information is denoted as F_0 , where f_{AvePool} represents the average pooling layer mapping function, f_{MaxPool} represents the max pooling layer mapping function, f_{conv}^{k1s1} denotes the convolution operation with a kernel size of 1 and a stride of 1, f_{conv}^{k7s1} denotes the convolution operation with a kernel size of 7 and a stride of 1, and f_{sigmoid} is the sigmoid mapping function.

Next, we exploit both single-scale and cross-scale similarity information in remote sensing images through a series of operations. Let F_S^b symbolize the input denoted as Single similarity group (SSG), considered the feature derived from the fundamental scale. To exploit the internal recursion of information at different scales, features at the downsampled scale F_S^d can be acquired through downsampling operation $F_S^d = D_S(F_S^b)$, where D_S represents the downsampling operation with a s scale factor. Subsequently, we extract potent feature representations at two different scales, F_S^b and F_S^d , where the output of the downsampled scale is further upsampled by the same scale factor. The outputs through the basic scale and downsampling scale are denoted as X^b and X^d ,

$$\begin{aligned} X^b &= M_{SSG}(F_S^b) \\ X^d &= U_S M_{SSG}(F_S^d) \end{aligned} \quad (2)$$

where U_S represents the up-sampling operation, and its scale factor is s .

Afterward, we utilize the self-similarity information extracted by SSG as attention to guide the main high-frequency feature extraction. We also use Across similarity groups (ASG) to exploit the correlation between the two remote sensing image scales. Specifically, in the main branch, convolution layers extract higher-level features, while in the attention branch, SSG is used to adaptively rescale features generated by the main branch. The non-local operation is expressed as:

$$Y_i = \left(\sum_{\forall j} f(X_i, X_j) g(X_j) \right) / \sum_{\forall j} f(X_i, X_j) \quad (3)$$

Here, i denotes the index of the output position, j enumerates all positions, X and Y are the input and output of the operation, f functions use embedded Gaussian functions to compute correlations among all positions, resulting in a single-scale self-similarity feature representation.

Moreover, ASG can integrate features at multiple scales and exploit their similarities, differing mainly from SSG in the input structure, while the subsequent workflow for self-similarity computation remains similar. For ANLB, the computation $\sum_{\forall j} f(X_i^b, X_j^d)$ is denoted as $\exp(\theta^T(X_i^d)\varphi(X_j^b))$, X_i^b and X_j^d represent the outputs through SSMs for the basic scale and downsampling scale. At the end, Conv-ReLU block (CRB) is applied to further map the fused feature output.

3.3 Memory Gated Group

Recent findings [21] highlight the significant contribution of second-order statistics in deep CNNs to discriminative representation. To leverage this, we introduce a Memory gated group (MGG) in Fig. 3 for capturing feature interdependence through adaptive learning, emphasizing persistent memory. Given a C-feature map of size $H \times W$, we reconstruct it into a feature matrix A . The sample covariance matrix is calculated as $\Sigma = A\bar{I}A^T$, where I is the identity matrix. Covariance normalization plays a crucial role in achieving more discriminative representations. Therefore, we utilize a normalized covariance matrix that characterizes the correlation of channel features through

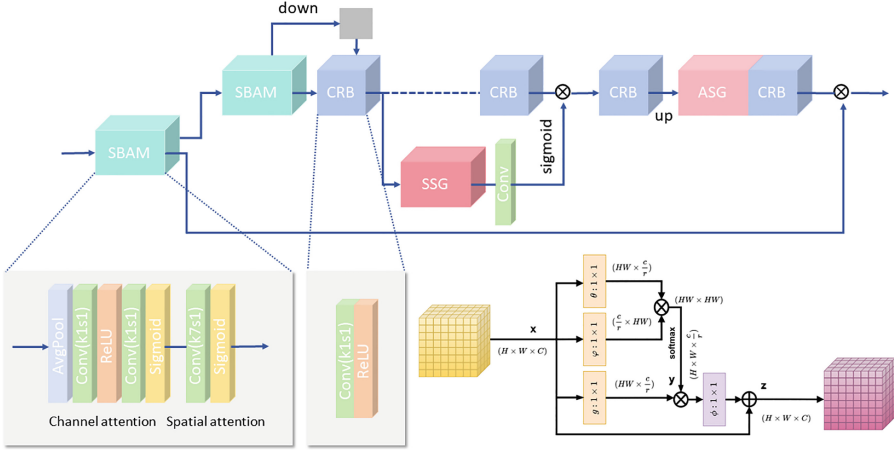


Fig. 2. Structure of Utilize similarity balancing module. Details of the SSG are in the lower right of the figure.

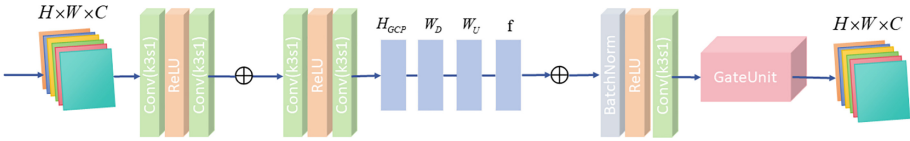


Fig. 3. Overview of the Memory gated group.

global covariance pooling. This approach fully exploits the interdependence of features in aggregated information, treating this normalized covariance matrix as the channel descriptor.

$$\mathbf{w} = f(\mathbf{W}_U \delta(\mathbf{W}_D(H_{GCP}(\mathbf{y})))) \tag{4}$$

Let $H_{GCP}(\cdot)$ represent the global covariance pooling function. Applying a simple gating mechanism with sigmoid. \mathbf{W}_D and \mathbf{W}_U as the convolutional layer’s weight set and $f(\cdot)$ and $\delta(\cdot)$ as the Sigmoid and ReLU functions, respectively, we obtain channel attention mapping, achieving adaptive scaling of residual components.

A memory block of recursive and gating units establishes long-term memory of deep features. Some Research [22] suggest recursive units can learn multi-level representations under different receptive fields, generating short-term memory. Assuming R recursions in the unit, the r -th recursion’s representation is expressed as

$$Q_m^r = \underbrace{\mathcal{R}_m(\mathcal{R}_m(\dots(\mathcal{R}_m(B_{m-1}))))}_{r} \tag{5}$$

where $\{Q_m^r\}_{r=1}^R$ is the multi-level representation. These representations concatenate to form short-term memory $O_m^{short} = [Q_m^1, Q_m^2, \dots, Q_m^R]$. Additionally, long-term memory from the previous block is constructed as $O_m^{long} = [O_0, O_1, \dots, O_{m-1}]$. Both memory

types concatenate as inputs to the gating unit.

$$O_m^{gate} = [O_m^{short}, O_m^{long}] \quad (6)$$

Representations from the previous block and output are sent to the gating unit, which learns adaptive weights for different memories. In this paper, a 1×1 convolutional layer implements the gating mechanism.

$$O_m = f_m^{gate}(O_m^{gate}) \quad (7)$$

Here, f_m^{gate} and O_m represent the convolutional layer and the m -th memory block's output functions. The weight control of long-term memory determines how much of the previous state to retain, while short-term memory weight control decides how much of the current state to store.

3.4 Loss Function

The entirety of the network is composed of a Super-Resolution (SR) process and a Dual-learning (DL) process, with the primary goal of establishing a two-way mapping between Low-Resolution (LR) and High-Resolution (HR) images. The loss function comprises three components: pixel loss between I_{SR} and I_{HR} , pixel loss between I_{LR} and $I_{LR'}$. TDSNet aims to simultaneously update the SR and DL operations by minimizing pixel-level losses (between L_S and L_D).

$$\begin{aligned} L_S &= L_1(S(I_{LR}^i), I_{HR}^i) \\ L_D &= L_1(D(S(I_{LR}^i)), I_{LR}^i) \end{aligned} \quad (8)$$

Due to the stronger robustness of L_1 loss over L_2 loss, we apply L_1 in our loss function for pixel-level losses in both the SR and DL processes. Learning the mapping from HR to LR images introduces constraints into the network, alleviating ill-posed problems. The overall loss of TDSNet is expressed as a weighted sum of the mentioned losses.

$$\begin{aligned} \text{Loss} &= \sum_{i=1}^N \lambda_1 L_S + \lambda_2 L_D \\ &= \sum_{i=1}^N \lambda_1 L_1(S(I_{HR}^i), I_{HR}^i) + \lambda_2 L_1(D(S(I_{LR}^i)), I_{LR}^i) \\ &= \sum_{i=1}^N \lambda_1 L_1(I_{SR}^i, I_{HR}^i) + \lambda_2 L_1(I_{LR}^i, I_{LR}^i) \end{aligned} \quad (9)$$

Here, λ_1 and λ_2 expressing the weighted coefficients in the loss function, set λ_1, λ_2 to 1 and 0.1, respectively. The lower index i denotes the sequential order of image set, with N representing the overall quantity of image sets.

4 Experimental Analysis

4.1 Experimental Dataset and Metrics

This paper validates the proposed method using two publicly available remote sensing datasets: NWPU-RESISC45 [23] and AID [24]. Each dataset is partitioned into training, testing, and validation sets at a ratio of 6:3:1.

1. NWPU-RESISC45 Dataset: A remote sensing image dataset with a pixel size of 256×256 , comprising a total of 31,500 images across 45 scene categories, with 700 images per category.
2. AID Dataset: A remote sensing image dataset with a pixel size of 600×600 , featuring 30 scene categories and approximately 220–420 images per category. The dataset contains a total of 10,000 images, with resolutions ranging from 8 m to 0.5 m.

Building upon the foundations of HAUNet [25], we employ the original images from each dataset as authentic HR references. LR images are generated through bicubic interpolation, forming HR/LR image pairs for both training and evaluation. Building on this foundation, quantitative assessments are undertaken, encompassing Peak Signal-to-Noise Ratio (PSNR) [26], Structural Similarity Index (SSIM) [26], Spatial Correlation Coefficient (SCC) [27], and Spectral Angle Mapper (SAM) [28]. Higher PSNR, SSIM, SCC, and lower SAM values signify improved image quality. To gain deeper insights into the workings of the super-resolution (SR) network and understand its behavior, we leverage Local Attribution Map (LAM) [29]. LAM helps identify which input pixels contribute significantly to the overall performance. For instance, in Fig. 4 (b), pixels marked in red are pivotal for the reconstruction process. Furthermore, the varying Degree of Importance (DI) indicates the extent of pixel involvement, with higher DI reflecting a broader attention range. Intuitively, superior network performance is achieved when more informative pixels are utilized.

4.2 Implementation Details

In this study, we focus on scale ratios 2, 3, and 4, adjusting the upsampling operations in the reconstruction based on the specific scale factor. The quantities of Utilize similarity balancing module and Memory gated group in the proposed TDSNet are set to 2, and the number of transformer layers in the Residual Conv-swintransformer structure is set to 4. Regarding the number of transformer heads in the RCSTS, if the magnification factor is 4, the number of heads for the first upsampling is set to 4, and for the second upsampling, it is set to 8, with a uniform window size set to 8. In the training process, the initial learning rate is set to $1e-4$, and the number of iterations is set to 200. The training model is optimized using Adaptive Moment Estimation (ADAM) [30] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 1e-8$. All experiments were performed using the PyTorch framework on an NVIDIA GeForce RTX 2080Ti graphics processing unit (GPU).

4.3 Comparisons with State-Of-The-Art SR Methods

In Fig. 4 (a) and (b), we present super-resolution results for various magnification factors on a general test set, highlighting the detailed reconstruction effects of super-resolution images through subjective visual analysis. Figure 4 (a) compares super-resolution outcomes at $2\times$ and $4\times$ magnification rates using images from the NWPURESISC45. Low-Resolution (LR) images are derived by downsampling test set images, and the proposed algorithm, along with comparison methods, is employed for image reconstruction. Images produced by TDSNet exhibit clearer shapes and edges compared to other

methods, which often result in excessively smooth and somewhat blurry outputs. Particularly, details in the locally zoomed-in region (highlighted in the red box) reveal that the bicubic method produces overly blurry details, and images from deep learning-based algorithms like FSRCNN [7] and DRN [8] exhibit less clear and sharp edges than our proposed algorithm. In comparison to deep learning-based remote sensing image super-resolution algorithms such as LGCNet [18] and HSENet [12], our proposed algorithm provides clearer and sharper reconstructions, preserving high-frequency information and enhancing visual results in texture, edges, and similar content.

In Fig. 4 (b), qualitative comparisons of Local Attention Map (LAM) and Super-Resolution (SR) results for different networks on the NWPURESISC45 test dataset at a 4x magnification factor are conducted. Remarkably, the LAM result images of FENET [15] and SRDD [16] contain fewer informative pixels for reconstruction, resulting in less detailed structural information and an inability to recover clear white lines. In contrast, TDSNet’s attention extends along the texture direction, distributing more widely across the scene, resulting in superior performance in the SR task and a more accurate understanding and reconstruction of detailed information. TDSNet, featuring USBM and MGG, not only retains local detailed information but also captures more global and meaningful attention to context and content, maintaining competitive performance.

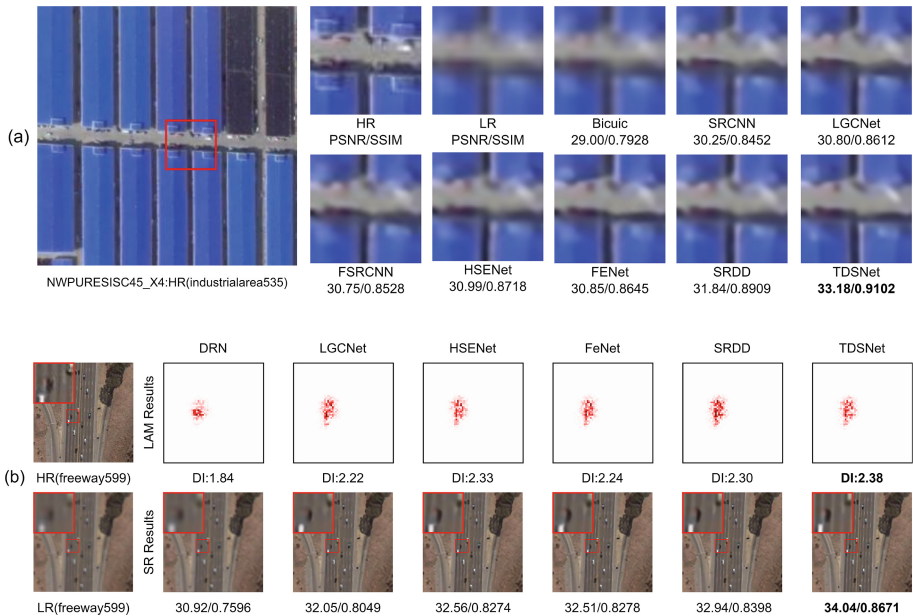


Fig. 4. (a) Comparisons of results using various methods on NWPURE-RESISC45 datasets. (b) Evaluating 4x Super-Resolution (SR) results and LAM attribution results from different SR networks on NWPURE-RESISC4. The LAM outcomes visually represent the significance of individual pixels.

Beyond visual assessments, quantitative comparisons using PSNR, SSIM, SCC, and SAM values are presented in Tables 1 and 2. The best and second-best performances

are highlighted in red and blue, respectively. These values represent the average PSNR, SSIM, SCC, and SAM values of the NWPU-RESISC45 and AID test sets after undergoing super-resolution enlargement through various algorithms. For the NWPU-RESISC45 dataset at a 4x magnification factor, TDSNet outperforms the classical dual-regression SR method DRN [8], improving PSNR and SSIM values by 2.06 dB and 0.0941, respectively. Compared to SRDD [16], our proposed method enhances PSNR and SSIM values by 0.25 dB and 0.0083, respectively. In Table 2, compared to TransENet [34], the proposed method improves PSNR and SSIM values by 0.27 dB and 0.0064, respectively. The results underscore TDSNet’s higher performance, showcasing the practicality of the proposed method in edge and fine detail recovery. The USBM, MGG, and RCSTS are emphasized as practical tools for achieving more accurate recovery.

Table 1. Experimental results on NWPURESISC45 dataset. Red color indicates the best performance, blue color indicates the second best performance and green color indicates the third best performance.

Method	NWPURESISC45							
	×2				×4			
	PSNR	SSIM	SCC	SAM	PSNR	SSIM	SCC	SAM
BICUIC	32.12	0.8801	0.5375	0.0730	27.61	0.6967	0.1483	0.1192
SRCNN[6]	34.06	0.9202	0.6050	0.0587	28.59	0.7431	0.2073	0.1069
LGCNET[18]	34.26	0.9227	0.6080	0.0574	28.74	0.7519	0.2124	0.1052
FSRCNN[7]	34.16	0.9219	0.6116	0.0581	28.82	0.7554	0.2222	0.1044
DRN[8]	32.39	0.8878	0.4917	0.0709	27.47	0.6882	0.1249	0.1210
HSENET[12]	34.62	0.9284	0.6650	0.0551	29.20	0.7709	0.2575	0.1000
FENET[15]	34.55	0.9272	0.6340	0.0555	29.16	0.7694	0.2527	0.1006
SRDD[16]	34.68	0.9289	0.6401	0.0546	29.28	0.7740	0.2666	0.0991
OMNISR[31]	34.51	0.9266	0.5964	0.0552	29.44	0.7800	0.2810	0.0973
Ours	34.86	0.9329	0.6912	0.0536	29.53	0.7823	0.2920	0.0967

4.4 Ablation Studies

To evaluate the impact of key modules in TDSNet, we conducted ablation studies by removing specific components: 1) Utilize similarity balancing module, 2) Memory gated group, 3) Residual Conv-swintransformer structure, and 4) Dual Learning (DL) module. The study maintained consistent datasets and experimental settings across different variations. Table 3 and Fig. 5 present both the quantitative and qualitative results of the ablation study conducted on the AID dataset, showcasing the best-performing outcomes. It is crucial to note that the removal of any essential component leads to a significant decrease in evaluation metrics and visual quality. Specifically, the elimination of RCSTS results in a substantial decrease in PSNR (-1.97dB), and consequently, the depiction of ship lines in the image becomes notably blurred, thus emphasizing the Transformer’s powerful performance in enhancing SR results. Additionally, omitting the dual learning module not only leads to a decrease in PSNR of 0.14 dB on the AID dataset but also results in

Table 2. Comparison results of PSNR, SSIM, SCC and SAM on AID dataset with a scale factor of 3 and 4.

Method	AID							
	×3				×4			
	PSNR	SSIM	SCC	SAM	PSNR	SSIM	SCC	SAM
BICUIC	29.82	0.753 5	0.169 7	0.094 1	28.69	0.733 4	0.158 6	0.105 1
SRCNN	31.85	0.859 6	0.349 6	0.072 4	29.76	0.778 8	0.217 3	0.092 8
FSRCNN	32.03	0.860 3	0.349 9	0.072 2	29.80	0.779 8	0.207 4	0.092 9
VDSR[32]	32.14	0.860 7	0.351 4	0.071 3	30.35	0.797 6	0.249 1	0.087 1
DRN	30.21	0.742 1	0.1198	0.097 1	28.48	0.720 3	0.092 7	0.107 2
HSENet	32.69	0.877 5	0.398 0	0.064 3	30.44	0.8011	0.260 3	0.086 3
DCM[33]	32.52	0.874 4	0.399 5	0.063 7	30.50	0.803 2	0.268 5	0.085 7
TransENet[34]	32.80	0.881 2	0.402 5	0.662 8	30.53	0.804 8	0.265 5	0.085 3
Ours	32.94	0.882 1	0.404 2	0.062 4	30.80	0.8112	0.288 9	0.083 0

insufficient recovery of terrain edges, underscoring its importance. Consequently, this study continues to employ this approach. In conclusion, through comprehensive considerations, we achieved further generalization, affirming the indispensability of each design element for attaining satisfactory SR results in the proposed network.

Table 3. Ablation study with different components combinations (× 4)

Method				NWPURESISC45			
USBM	RCSTS	MGG	DL	PSNR	SSIM	SCC	SAM
×	√	√	√	29.41	0.7781	0.2809	0.0979
√	×	√	√	27.56	0.6921	0.1251	0.1199
√	√	×	√	29.47	0.7794	0.2868	0.0973
√	√	√	×	29.39	0.7772	0.2801	0.0981
√	√	√	√	29.53	0.7823	0.2920	0.0967

To gain a better understanding of the model's complexity, we also present a comparison between PSNR and network parameters (Params) on the AID (×4) dataset in Fig. 6. As observed, our TDSNet achieves higher PSNR compared to existing re-remote sensing



Fig. 5. Ablation results figure. From top to bottom, left to right: original high-resolution image, results from our proposed network, results without USBM, results without RCSTS, results without MGG, and results without dual loss.

super-resolution methods while utilizing fewer parameters. Our approach effectively strikes the optimal balance between reconstruction performance and computation.

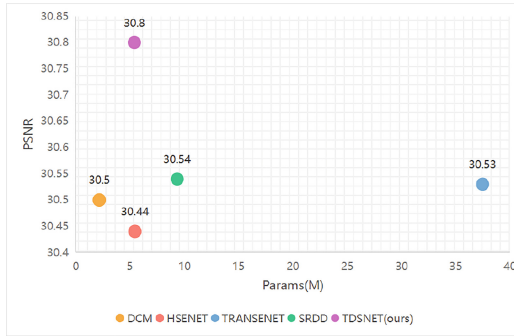


Fig. 6. PSNR vs. Params

5 Conclusion

In this paper, we present a novel remote sensing image super-resolution network, a multi-level transformer-enhanced model with a dual-regression mechanism. The proposed algorithm incorporates a Utilize similarity balancing module and a Memory gated group, enabling the integration of valuable information from analogous patches throughout the entirety of the remote sensing image and providing long-term memory for accurate feature extraction. The model effectively combines features from CNN and SwinTransformer, allowing the integration of high-dimensional and low-dimensional features and modeling distant relationships between pixels. Additionally, a dual-regression approach is introduced to govern the mapping from low-resolution (LR) images to high-resolution (HR) images, as well as from super-resolved (SR) images to LR images. To assess TDSNet’s generalization capability, training and testing are conducted on two public

datasets. Finally, the proposed method is compared with traditional bicubic interpolation and various deep learning-based super-resolution methods for both natural and remote sensing images. The experimental results demonstrate the superiority and effectiveness of this method, showcasing improved quantitative and qualitative performance.

References

1. Wang, P., Bayram, B., Sertel, E.: A comprehensive review on deep learning based remote sensing image super-resolution methods. *Earth Sci. Rev.* **232**, 104110 (2022)
2. Wang, X., Yi, J., Guo, J., et al.: A review of image super-resolution approaches based on deep learning and applications in remote sensing. *Remote Sens.* **14**(21), 5423 (2022)
3. Girshick, R., Donahue, J., Darrell, T., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587 (2014)
4. Lu, Z., Wu, C., Yu, X., et al.: Single image super resolution based on multi-scale structural self similarity and neighborhood regression. In: *Tenth International Conference on Digital Image Processing (ICDIP 2018)*, SPIE, pp. 10806: 876–880 (2018)
5. Yang, J., Wright, J., Huang, T.S., et al.: Image super-resolution via sparse representation. *IEEE Trans. Image Process.* **19**(11), 2861–2873 (2010)
6. Dong, C., Loy, C.C., He, K., et al.: Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(2), 295–307 (2015)
7. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016, Proceedings, Part II 14*, Springer, pp. 391–407 (2016)
8. Guo, Y., Chen, J., Wang, J., et al.: Closed-loop matters: dual regression networks for single image super-resolution. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5407–5416 (2020)
9. Zontak, M., Irani, M.: Internal statistics of a single natural image. In: *CVPR 2011*, pp. 977–984. IEEE (2011)
10. Xu, J., Zhang, L., Zuo, W., et al.: Patch group based nonlocal self-similarity prior learning for image denoising. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 244–252 (2015)
11. Michaeli, T., Irani, M.: Blind deblurring using internal patch recurrence. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014, Proceedings, Part III 13*, Springer pp. 783–798 (2014)
12. Lei, S., Shi, Z.: Hybrid-scale self-similarity exploitation for remote sensing image super-resolution. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–10 (2021)
13. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: *2009 IEEE 12th International Conference on Computer Vision*, pp. 349–356. IEEE (2009)
14. Dong, X., Wang, L., Sun, X., et al.: Remote sensing image super-resolution using second-order multi-scale networks. *IEEE Trans. Geosci. Remote Sens.* **59**(4), 3473–3485 (2020)
15. Wang, Z., Li, L., Xue, Y., et al.: FeNet: Feature enhancement network for lightweight remote-sensing image super-resolution. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–12 (2022)
16. Maeda, S.: Image super-resolution with deep dictionary. In: *European Conference on Computer Vision*, Springer, Cham, pp. 464–480 (2022)
17. Xia, J., Yang, Z., Li, S., et al.: Blind super-resolution via meta-learning and Markov chain Monte Carlo simulation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024)

18. Lei, S., Shi, Z., Zou, Z.: Super-resolution for remote sensing images via local–global combined network. *IEEE Geosci. Remote Sens. Lett.* **14**(8), 1243–1247 (2017)
19. Liu, Z., Feng, R., Wang, L., et al.: Dual learning-based graph neural network for remote sensing image super-resolution. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–14 (2022)
20. Zhang, Y., Li, K., Li, K., et al.: Image super-resolution using very deep residual channel attention networks. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 286–301 (2018)
21. Dai, T., Cai, J., Zhang, Y., et al.: Second-order attention network for single image super-resolution. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11065–11074 (2019)
22. Tai, Y., Yang, J., Liu, X., et al.: Memnet: a persistent memory network for image restoration. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4539–4547 (2017)
23. Cheng, G., Han, J., Lu, X.: Remote sensing image scene classification: benchmark and state of the art. *Proc. IEEE* **105**(10), 1865–1883 (2017)
24. Xia, G.S., Hu, J., Hu, F., et al.: AID: a benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* **55**(7), 3965–3981 (2017)
25. Wang, J., Wang, B., Wang, X., et al.: Hybrid attention based u-shaped network for remote sensing image super-resolution. *IEEE Trans. Geosci. Remote Sens.* (2023)
26. Wang, Z., Bovik, A.C., Sheikh, H.R., et al.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
27. Zhou, J., Civco, D.L., Silander, J.A.: A wavelet transform method to merge Landsat TM and SPOT panchromatic data. *Int. J. Remote Sens.* **19**(4), 743–757 (1998)
28. Yuhas, R.H., Goetz, A.F.H., Boardman, J.W.: Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm. In: *JPL, Summaries of the Third Annual JPL Airborne Geoscience Workshop*, vol. 1, AVIRIS Workshop (1992)
29. Gu, J., Dong, C.: Interpreting super-resolution networks with local attribution maps. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9199–9208 (2021)
30. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)* (2014)
31. Wang, H., Chen, X., Ni, B., et al.: Omni aggregation networks for lightweight image super-resolution. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22378–22387 (2023)
32. 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)
33. Haut, J.M., Paoletti, M.E., Fernández-Beltrán, R., et al.: Remote sensing single-image super-resolution based on a deep compendium model. *IEEE Geosci. Remote Sens. Lett.* **16**(9), 1432–1436 (2019)
34. Lei, S., Shi, Z., Mo, W.: Transformer-based multistage enhancement for remote sensing image super-resolution. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–11 (2021)



Local and Global Features Fusion for No-Reference Quality Assessment of Super-Resolution Images

Yun Liu^{1,2,3}, Tong Tang^{1,2,3}(✉), Zhiyuan Zhu^{1,2,3}, and Jun Ying^{1,2,3}

¹ School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China

² Advanced Network and Intelligent Interconnection Technology Key Laboratory of Chongqing Education Commission of China, Chongqing, China

³ Chongqing Key Laboratory of Ubiquitous Sensing and Networking, Chongqing, China

tangtong@cqupt.edu.cn

Abstract. Image super-resolution (SR) technology aims to enhance the resolution and improve the quality of images, and it has been widely used in face recognition, small target detection, medical imaging and remote sensing image analysis. Image quality assessment (IQA) is important for optimizing SR algorithms. At present, the main challenge lies in how to comprehensively learn features to characterize perceptual properties of human visual characteristics. Therefore, in this paper, we propose a no-reference quality assessment method for SR images based on local and global features fusion. First, a two-branch feature extractor is proposed, which uses convolutional neural network and vision transformer respectively to extract local features and global features. Then, considering the perception properties of the human visual system (HVS), local and global features are fused and adaptive weight strategy is applied to predict the quality score. Finally, Experimental results show that the proposed method outperforms the state-of-the-art methods in terms of prediction accuracy and generalization capability on benchmark datasets.

Keywords: Image quality assessment · Super-resolution · Feature fusion

1 Introduction

Image super-resolution (SR) reconstruction technology is used to reconstruct blurred low-resolution images into high-resolution images containing rich details, which can improve the visual perception of low-quality images [9]. In the past few decades, SR technology has developed rapidly and been widely used in face recognition, small target detection, medical imaging and remote sensing image analysis [10, 18]. With a large number of SR algorithms being proposed, how to

effectively assess the quality of SR reconstructed images has become an urgent problem to be solved.

Image quality assessment(IQA) can be classified into subjective quality assessment and objective quality assessment [28]. Although subjective quality assessment is considered to be the most direct and reliable way, it is time-consuming, expensive, and difficult to be embedded into practical application. Therefore, objective quality assessment plays a more effective role in evaluating image quality. According to the amount of reference information, these objective models can be classified into three categories: full-reference image quality assessment (FR-IQA) [15, 19], reduced-reference quality assessment (RR-IQA) [4] and no-reference quality assessment (NR-IQA) [11, 14]. FR-IQA and RR-IQA methods can achieve good performance, but they both require information of the reference image [26]. In practical SR applications, it's almost impossible to obtain the original reference image, thus NR-IQA appears more important for the quality assessment of SR images. However, most existing NR-IQA are designed for generic distorted images rather than SR images [22]. Different from the general natural image distortion, the distortion of SR images is mainly due to the blurring, sawtooth and checkerboard distortion introduced by the SR reconstruction algorithms [29]. Therefore, it's not suitable to simply apply generic NR-IQA methods to assess the quality of SR images [1, 23, 27].

Researchers have presented some no-reference quality assessment methods for SR images (NR-SRIQA), including traditional methods [6, 31] and deep learning-based methods [3, 13, 24, 30]. Traditional NR-SRIQA methods are usually based on manual feature extraction, thus the performance is limited by the accuracy of hand crafted descriptors. In recent years, the NR-SRIQA method based on convolutional neural network (CNN) has been gradually developed. However, due to the limited local perception ability of CNN, these methods may ignore the context information and lose the global perception information of the SR image, degrading performance of these NR-SRIQA methods.

In this paper, to comprehensively learn features to characterize human visual characteristics, we propose a local and global features fusion based no-reference quality assessment method for SR images, named as LAGFBN. Our contributions are summarized below:

- We propose a two-branch feature extractor based NR-SRIQA framework. The two branches are complementary, using CNN and vision transformer respectively to capture local information and global information of super-resolution images.
- We fuse local and global features and apply adaptive weight strategy to predict the quality score, which could better simulate the perceptual properties of the human visual system.
- Extensive experiments on two benchmark SRIQA datasets show that our proposed method could achieve more accurate predicted scores compared with the state-of-the-art methods.

2 Related Work

No-Reference Quality Assessment for General Natural Images. The current NR-IQA methods can be roughly classified into the traditional method based on manual extraction of image features and the method based on deep learning. Traditional methods aim to design suitable manual feature extractors to quantify image distortion. For example, Saad et al. [14] used discrete cosine transform (DCT) coefficients to design perceptual features suitable for quality scores, and used a simple Bayesian model to predict quality scores. Liu et al. [11] developed an NR-IQA index called SSEQ by using the local space and spectral entropy features of distorted images. In recent years, NR-IQA methods based on deep learning have been gradually developed. These methods have achieved better performance than traditional methods by building deep neural networks to automatically learn the mapping relationship between input images and quality scores. For example, Kang et al. [7] proposed a groundbreaking CNN-based NR-IQA method to achieve end-to-end NR-IQA method. Considering that the dual-stream network is more effective to extracting rich quality perception features, Yan et al. [20] used two identical branches to extract the features of RGB distorted images and gradient images respectively for image quality assessment.

Full-Reference and Reduced-Reference Quality Assessment for Super-Resolution Images. According to the amount of reference information, SR-IQA methods can be divided into FR-SRIQA, RR-SRIQA and NR-SRIQA. FR-SRIQA requires original high-resolution images as reference information. For example, Zhou et al. [29] proposed a FR-SRIQA method by considering both structure components and texture components in SR images. This method gives the final quality score by calculating the variation of texture distribution and the similarity of structural components. Zhou and Wang [31] proposed to measure the distortion of SR images by developing fidelity (DF) and statistical fidelity (SF) models, and use content-adaptive weighting scheme to combine SF and DF into an overall quality predictor. For RR-SRIQA, it needs part of the original image information as reference. For example, Yeganeh et al. [21] proposed a NSS-based distortion measurement method using LR images as reference information. This method established the statistical model based on frequency energy falloff, dominant orientation and spatial continuity. Zhao et al. [28] designed an end-to-end RR-SRIQA method named DISQ, which utilized a dual-flow deep neural network to extract LR and HR image features respectively.

No-Reference Quality Assessment for Super-Resolution Images. NR-SRIQA can predict image quality without reference information. Therefore, it is more practical in actual SR problems compared with FR-SRIQA and RR-SRIQA. Ma et al. [12] proposed to extract the frequency and spatial features of SR images, and use random forest regression and ridge regression to establish the mapping relationship between perception features and quality scores. Zhou et al. [30] proposed to use two parallel AlexNet networks to extract discriminant

features from structure maps and texture maps, and use fully connected layers to predict quality scores. Zhang et al. [24] quantified the distortion of SR images by using two different subnetworks to learn the low-level features and middle-level features of SR images, where the CBAM attention module in the middle-level feature extraction subnetwork could better extract the prominent features of space and channel. Quan et al. [13] used the triple attention mechanism to learn the association between spatial dimension and channel dimension in SR images, and constructed multi-scale modules to capture features at different scales, thus further improving the prediction accuracy of image quality scores.

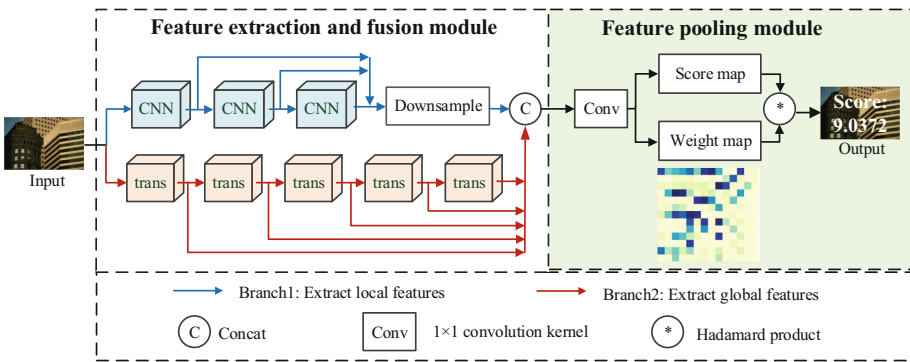


Fig. 1. The overall framework of proposed LAGFBN, where “CNN” represents the residual block in ResNet50, “trans” represents the encoding block in vision transformer.

3 Proposed Method

The framework of proposed LAGFBN is shown in Fig. 1, consists of two modules: feature extraction and fusion module and feature pooling module. In the feature extraction and fusion module, two-branch feature extraction network is designed to comprehensively learn features of SR images. Branch 1 extracts local features, Branch 2 extracts global features, and features from two branches are fused. In the feature pooling module, adaptive weighting strategy is applied to predict the final quality scores. The overall learning algorithm of proposed LAGFBN could be summarized as Algorithm 1.

3.1 Feature Extraction and Fusion

As shown in Fig. 1, we use two branches to extract the features of the input image. Branch 1 uses residual blocks in ResNet50 [5] to capture local information such as texture details. Considering that human beings are not only affected by the local information, but also by the global information when evaluating the

Algorithm 1. The learning algorithm of proposed LAGFBN

Input: Super-resolution images, and their corresponding subjective quality scores Q_{gt}

Output: Predicted quality scores Q_{pred}

Number of epoch: N_{epoch}

Number of batches: N_{batch}

```

1: Initialize ViT and ResNet50 with pre-trained parameters
2: for  $1 \leq e_i \leq N_{epoch}$  do
3:   for  $1 \leq e_j \leq N_{batch}$  do
4:     Extract the features  $F_{local}$  and  $F_{global}$ 
5:     Concat( $F_{local}, F_{global}$ )
6:     Generate score map(S) and weight map(W)
7:     Use formula(8) to calculate  $Q_{pred}$ 
8:   end for
9: end for
10:  $\mathcal{L} = \mathcal{L2\_loss}(Q_{pred}, Q_{gt})$ 
11: Backpropagation with Adam optimizer
12: Updating the model parameters  $\theta$ 

```

quality of the image. Branch 2 captures remote dependencies between image patches through encoding blocks in vision transformer [2]. The self-attention mechanism contained in each encoding block can spatially model the spatial dependencies between image patches, thus capturing global information and high-level semantic representation of SR images. Furthermore, deep neural networks exhibit superior performance only when applied to large-scale datasets. However, the currently available datasets for SR-IQA are limited in size. Therefore, we use pre-trained parameters to initialize ResNet50 and vision transformer, and then feed the obtained features into the pooling model to get better experimental results.

Local Feature Extraction. Branch 1 extracts the local features by three basic residual blocks in ResNet50. The internal detail of the residual block is shown in Fig. 2, which works based on the idea of residual connection. With the residual connection, the network is able to learn subtle changes and details in the image more intently, which in turn better captures complex visual perception features in the input image. In addition, the skip connection in the residual block enables the model to bypass unnecessary CNN weight layers, effectively reducing the risk of overfitting the training set. We represent the output features obtained by each residual block with f_i ($i=1,2,3$). By concatenating them in the channel dimension, we can get the output F_{local} of branch 1. The formula is as follows, where C is equal to 256.

$$F_{local} = Concat(f_1, f_2, f_3), F_{local} \in \mathbb{R}^{56 \times 56 \times 3C} \quad (1)$$

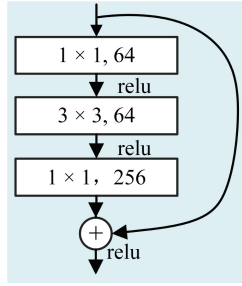


Fig. 2. Residual block of stage 1 in ResNet50.

Global Feature Extraction. We use the first five encoding blocks from the vision transformer to capture the global features of the SR image. The internal structure of the encoding block in vision transformer is shown in Fig. 3.

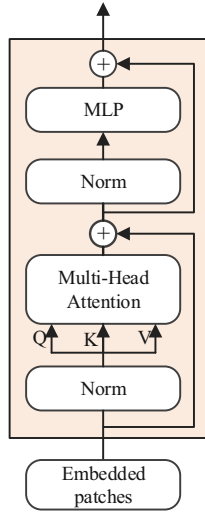


Fig. 3. Internal detail of encoding block in vision transformer.

The input SR image first passes through a large convolution kernel to obtain multiple image patches, then these patches are converted into vector representation by linear mapping. Next, the position vectors are added to the feature vectors of image patches by position coding to retain the position information of image patches. Finally, the obtained embedded vectors are used as the input of the encoder. The encoder consists of multi-head self-attention and MLP blocks, calculated by the following formula:

$$f'_i = \text{MSA}(\text{LN}(f_{i-1})) + f_{i-1} \tag{2}$$

$$f_i = \text{MLP}(\text{LN}(f'_i)) + f'_i \quad (3)$$

In the above formula, $f_i(i=1,2,3,4,5)$ represents the output features obtained by each encoding block, MSA represents multi-head self-attention mechanism, LN represents layer normalization, and MLP is a multi-layer perceptron composed of feedforward neural networks. The ability of vision transformer to capture global features of images is mainly due to the self-attention mechanism. The self-attention mechanism allows each image patch to interact with other image patches and calculate the similarity between them. This similarity is determined based on the semantic correlation between image patches, and is not limited by specific location or distance. The input sequence is first transformed linearly to obtain three feature maps: query (Q), key (K), and value (V), and then the inner product of the query and key is used to calculate the attention weight. Finally, the values are weighted and summed according to the attention weight to obtain the final feature representation. Therefore, the self-attention mechanism allows different areas in the image to interact with each other, thus capturing the context dependencies of the image.

$$F'_{global} = \text{Concat}(\hat{f}_1, \hat{f}_2, \hat{f}_3, \hat{f}_4, \hat{f}_5), F'_{global} \in \mathbb{R}^{N \times 5C} \quad (4)$$

$$F_{global} = \text{Reshape}(F'_{global}), F_{global} \in \mathbb{R}^{\sqrt{N} \times \sqrt{N} \times 5C} \quad (5)$$

In addition, considering that the extracted quality perception features may gradually disappear in the process of deep network propagation, we concatenate the output features of the first five encoding blocks, instead of only using the output features of the last encoding block in vision transformer. It can be seen in Eq. (4), where \hat{f}_i stands for the encoding block output feature dropping the class token, $C=768$, $N=196$. N represents the number of image patches obtained through convolution operation before the input SR image enters the encoding blocks, and C represents the vector length obtained after flattening the input image patches. Then we reconstructed the feature sequence F'_{global} obtained after concatenating into F_{global} , which is given by Eq. (5).

Feature Fusion. To enable the model learn both local and global information of the input image, F_{local} from branch 1 and F_{global} from branch 2 are concatenated in channel dimension. However, it should be noted that the width and height of F_{local} are four times that of F_{global} . Therefore, F_{local} needs to be downsampled before concatenation, which is shown in Eq. (6), where F_{local}^* represents the feature representation obtained after downsampling. Then we concatenate the downsampled local features and global features to achieve feature fusion, which is expressed by Eq. (7).

$$F_{local}^* = \text{Conv2}(\text{ReLU}(\text{Conv2}(F_{local}))) \quad (6)$$

$$F_{all} = \text{Concat}(F_{local}^*, F_{global}) \quad (7)$$

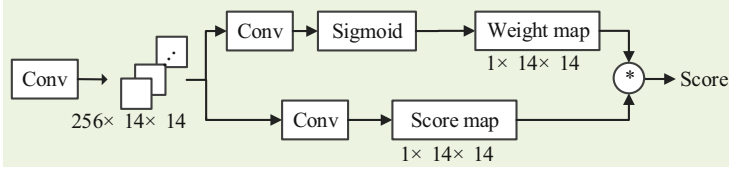


Fig. 4. Feature pooling module.

3.2 Feature Pooling

Our feature pooling module is shown in Fig. 4. We apply an adaptive weighting strategy for two-branch architecture to establish a mapping relationship between visual perception features and quality scores, so as to better simulate the visual characteristics of human eyes. The module obtains a score map (S) and a weight map (W) of shape $C \times H \times W$ by two independent linear projections, where $C=1$, H and W are both 14. The values contained in the score map represent the quality scores of elements in different regions, and the weights of the corresponding fractions are represented in the weight map. The weighted summation of the two feature maps can obtain the final image quality prediction score, and the formula is calculated as shown in Eq. (8). In addition, we optimize the parameters of the network model by minimizing MSE losses during the training process. It can be expressed as the following Eq. (9):

$$Q_{pred} = \frac{\mathbf{s} * \mathbf{w}}{\sum \mathbf{w}} \tag{8}$$

$$Loss = \frac{1}{N} \sum_{i=1}^N (Q_{pred,i} - Q_{gt,i})^2 \tag{9}$$

where N represents the batch size during the training process, $Q_{pred,i}$ and $Q_{gt,i}$ represents the predicted score and subjective quality score label of the i-th image in each batch, respectively.

4 Experiments

4.1 Datasets and Metrics

We use two SR-IQA datasets named CVIU-17 [12] and QADS [29] for training and testing. In addition, we also use the small-scale dataset called SISRSet [16] to conduct cross-dataset experiments to further verify the generalization capability of our method. CVIU-17 is composed of 1620 images generated by 8 SR algorithms and bicubic interpolation with 6 scaling factors. QADS is composed of 980 SR images generated by 21 methods with 3 scaling factors. SISRSet consists of 360 SR images generated by 8 SR algorithms with 3 scaling factors. We randomly selected 80% of the images for training and 20% for testing, and there was no image overlap between the training set and the test set.

We use four quantitative indicators including Spearman Rank Order Correlation Coefficient (SROCC), Kendall Rank Order Correlation Coefficient (KROCC), Pearson Linear Correlation Coefficient (PLCC) and Root Mean Square Error (RMSE) to evaluate the performance of the comparison methods [13, 24].

4.2 Training Details

As ResNet50 and ViT-B/16 (pre-trained on ImageNet) are used as the backbone in our network, we need to randomly crop the input image into the size of 224×224 in the image preprocessing stage. In the training process, we use random horizontal flip to enhance the generalization ability of the model. In the test process, we randomly crop 20 image patches of 224×224 size from the input image, and use their average score as the final image quality prediction score. In the implementation process, we use AdamW algorithm to optimize the network parameters, and set the initial learning rate to 10^{-4} , weight attenuation to 10^{-5} . We set the batch size as 8, and the number of training epochs as 200. The network model is trained on an NVIDIA GeForce RTX 3090 GPU.

Table 1. Performance comparison on CVIU-17 and QADS

Methods	CVIU-17				QADS			
	SROCC	PLCC	KROCC	RMSE	SROCC	PLCC	KROCC	RMSE
BLINDS-II	0.8983	0.8921	0.7252	1.2621	0.8889	0.8838	0.7100	0.1437
SSEQ	0.8854	0.8832	0.7013	1.1087	0.8679	0.8643	0.6887	0.1502
Ma	0.9139	0.9258	0.7531	0.8833	0.8954	0.8964	0.7280	0.1174
CNN	0.9226	0.9364	0.7599	0.9494	0.9533	0.9502	0.8114	0.0943
CNN++	0.9312	0.9307	0.7710	0.9874	0.9525	0.9509	0.8160	0.0932
Two-Stream	0.9424	0.9423	0.7931	0.9645	0.9541	0.9584	0.8192	0.1047
DBCNN	0.9465	0.9509	0.8021	0.8389	0.9575	0.9449	0.8200	0.0921
HyperIQA	0.9327	0.9284	0.7717	1.0166	0.9541	0.9568	0.8151	0.0986
JCSAN	0.9490	0.9565	0.8080	0.7769	0.9705	0.9734	0.8576	0.0646
TADSRNet	0.9516	0.9585	0.8120	0.7966	0.9720	0.9742	0.8616	0.0671
LAGFBN(Pro.)	0.9784	0.9808	0.8762	0.5355	0.9776	0.9777	0.8771	0.0684

4.3 Comparison with State-of-the-Art Methods

Ten NR-IQA methods are compared with proposed LAGFBN, including NR-IQA methods for generic distorted images (BLINDS-II [14], SSEQ [11], CNN [7], CNN++ [8], Two-Stream [20], DBCNN [25], HyperIQA [17]) and NR-SRIQA methods specifically designed for SR images (Ma [12], JCSAN [24] and TADSRNet [13]).

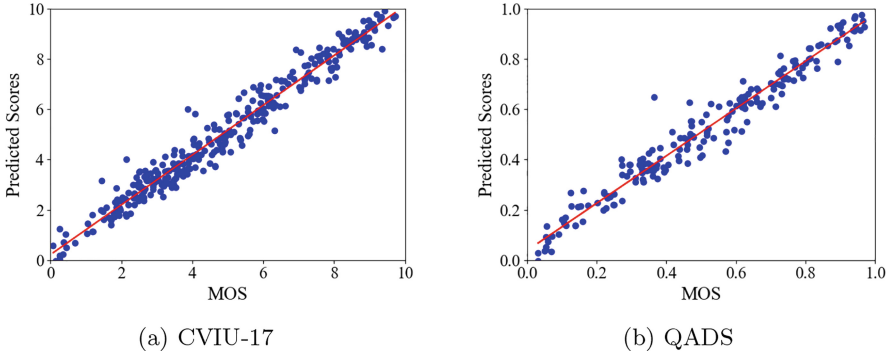


Fig. 5. Scatter plots and fitted straight lines of the MOS against the scores generated by proposed LAGFBN on different datasets

Table 1 shows the performance comparisons of all NR-IQA methods on CVIU-17 and QADS. It can be seen that, firstly, the proposed LAGFBN almost achieves the best performance on both two datasets CVIU-17 and QADS in terms of all the metrics (SROCC, KROCC, PLCC and RMSE). For example, in terms of SROCC, the proposed LAGFBN has a performance advantage of 0.268 compared to the second best method TADSRNet on CVIU-17. Secondly, NR-SRIQA methods (e.g. JCSAN and TADSRNet) have significant performance advantages over generic NR-IQA methods (e.g. CNN++ and DBCNN), which verifies that it’s not suitable to simply apply generic IQA methods to assess quality of SR images. Thirdly, the proposed LAGFBN has significant performance advantages over CNN-based methods (e.g. HyperIQA, JCSAN and TADSRNet), because our method jointly learns local and global features of SR image, while CNN-based NR-IQA methods only consider local features.

Figure 5 shows the scatter plots and fitted straight lines of the MOS against the scores generated by proposed LAGFBN on CVIU-17 and QADS, where the

Table 2. Cross-dataset evaluation: trained on CVIU-17 and tested on two complete datasets SISRSset and QADS

Methods	SISRSset		QADS	
	SROCC	KROCC	SROCC	KROCC
CNN++	/	/	0.7282	0.5432
Two-Stream	0.7952	0.5974	0.6619	0.4750
DBCNN	0.8372	0.6430	0.8087	0.6160
HyperIQA	0.7749	0.5756	0.7687	0.5730
JCSAN	0.7549	0.5733	0.7020	0.5298
TADSRNet	0.8183	0.6265	0.7390	0.5465
LAGFBN(Pro.)	0.8495	0.6528	0.8456	0.6571

Table 3. Cross-dataset evaluation: trained on QADS and tested on two complete datasets SISRSset and CVIU-17

Methods	SISRSset		CVIU-17	
	SROCC	KROCC	SROCC	KROCC
CNN++	/	/	0.6287	0.4460
Two-Stream	0.9010	0.7296	0.5819	0.4165
DBCNN	0.6769	0.4876	0.6533	0.4595
HyperIQA	0.8075	0.6160	0.8095	0.6235
JCSAN	0.9036	0.7345	0.6305	0.4629
TADSRNet	0.9225	0.7583	0.6574	0.4894
LAGFBN(Pro.)	0.9118	0.7383	0.8399	0.6498

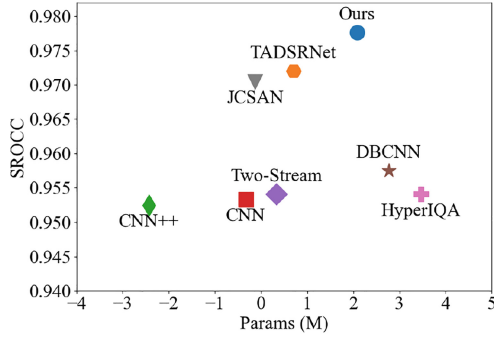
horizontal axis represents the mean opinion score (MOS) values obtained by human subjective tests, and the vertical axis represents the predicted quality score obtained by proposed LAGFBN. As can be seen from Fig. 5, our distribution is relatively concentrated and fitted straight lines are close to diagonal. In other words, the proposed LAGFBN matches well with the human visual system.

Figure 6 intuitively shows the model parameters (Params) and floating point operations (FLOPs) versus the performance (SROCC) value on QADS. Where Fig. 6(a) represents model parameters versus the SROCC, and Fig. 6(b) represents FLOPs versus the SROCC. It should be noted that, for better visualization, the x-axis is scaled using logarithmic coordinates. From the results in Fig. 6, we can see that our method achieves the best performance with a slight increase in Params and FLOPs.

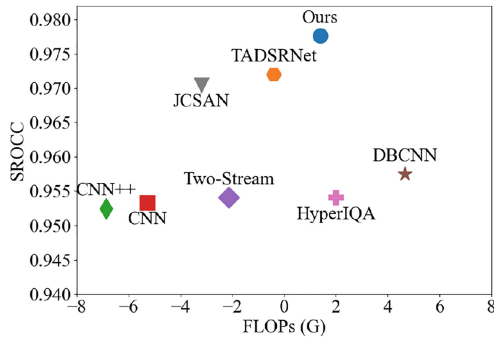
In order to further verify the generalization capability of proposed LAGFBN, we conducted cross-dataset experiments on CVIU-17, QADS and SISRSset, and the experimental results are shown in Table 2 and Table 3. Table 2 shows the results of different methods trained on CVIU-17 and tested on SISRSset and QADS. The results reveal that proposed LAGFBN achieves the highest SROCC value and KROCC value on both two datasets. Table 3 shows the results of different methods trained on QADS and tested on SISRSset and CVIU-17. Similarly, the proposed method still achieves impressive results. The experimental results show that the proposed LAGFBN achieves stable generalization ability, which may be due to the results of pre-trained models used in our network framework.

4.4 Visualization

In this part, we visualize the attention map to show important roles of the two branches within the feature extraction module. As depicted in Fig. 7, the redder color indicates that the network model pays more attention to this region. Specifically, Fig. 7(a) corresponds to the input super-resolution image, while Fig. 7(b) and Fig. 7(c) represent the attention maps of branch 1 and branch 2, respectively. We can see that the attention map of branch 1 in Fig. 7 (b) verifies that



(a) Params vs. performance(SROCC)



(b) FLOPs vs. performance(SROCC)

Fig. 6. Params and FLOPs versus performance (SROCC) on QADS

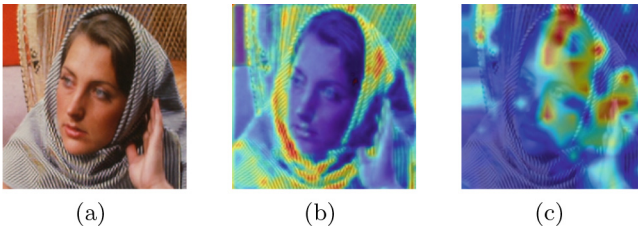


Fig. 7. Visualization of attention map, where (a) represents the input SR image, (b) and (c) represent the attention map of branch 1 and branch 2, respectively

the convolutional blocks could effectively capture local texture features, and the attention map of branch 2 in Fig. 7 (c) exhibits a stronger emphasis on extracting global semantic features of the image.

4.5 Ablation Experiment

To prove the complementarity of local features and global features in the feature extraction and fusion module, we test the network performance of a single branch on QADS as ablation experiment, and the results are shown in Table 4. From the Table 4, we note that the model using both branch 1 and branch 2 achieves the best performance metrics than other methods. Compared with method 1 that only uses branch 1 to extract local features, the SROCC value, PLCC value and KROCC value of the two-branch method are increased by 0.0116, 0.0105 and 0.0312, respectively, and the RMSE value is decreased by 0.025. Compared with method 2 that only uses branch 2 to extract global features, the SROCC value, PLCC value and KROCC value of the two-branch method are increased by 0.003, 0.0024 and 0.0078, respectively, and the RMSE value is decreased by 0.0042. Therefore, it can be concluded that the multi-source information combined with local and global features is more beneficial to the quality prediction task of SR images.

Table 4. Ablation experiment on QADS

Branch1	Branch2	SROCC	PLCC	KROCC	RMSE
1✓		0.9660	0.9665	0.8459	0.0934
2	✓	0.9746	0.9753	0.8693	0.0726
3✓	✓	0.9776	0.9777	0.8771	0.0684

5 Conclusion

In this paper, we propose a new dual-stream NR-SRIQA network named LAGFBN. Branch 1 uses convolutional neural network to capture local features, and branch 2 utilizes the self-attention mechanism in vision transformer to capture global features. In order to better simulate the visual characteristics of human visual system, we feed the fused features into the feature pooling module of two-branch architecture, and the final prediction score can be obtained by the weighted summation of the score map and the weight map. The experimental results show that proposed LAGFBN achieves better performance on benchmark datasets.

Acknowledgements. This work was supported by National Natural Science Foundation of China (62402074, 62271096, U20A20157), Natural Science Foundation of Chongqing, China (CSTB2023NSCQ-LZX0134), University Innovation Research Group of Chongqing (CXQT20017), Youth Innovation Group Support Program of ICE Discipline of CQUPT (SCIE-QN-2022-04), the Science and Technology Research Program of Chongqing Municipal Education Commission (KJQN202300632), and the Chongqing Postdoctoral Special Funding Project(2022CQBSHTB2057).

References

1. Bare, B., Li, K., Yan, B., Feng, B., Yao, C.: A deep learning based no-reference image quality assessment model for single-image super-resolution. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1223–1227. IEEE (2018)
2. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
3. Fang, Y., Zhang, C., Yang, W., Liu, J., Guo, Z.: Blind visual quality assessment for image super-resolution by convolutional neural network. *Multimedia Tools Appl.* **77**, 29829–29846 (2018)
4. Golestaneh, S., Karam, L.J.: Reduced-reference quality assessment based on the entropy of dwt coefficients of locally weighted gradient magnitudes. *IEEE Trans. Image Process.* **25**(11), 5293–5303 (2016)
5. 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)
6. Jiang, Q., et al.: Single image super-resolution quality assessment: a real-world dataset, subjective studies, and an objective metric. *IEEE Trans. Image Process.* **31**, 2279–2294 (2022)
7. Kang, L., Ye, P., Li, Y., Doermann, D.: Convolutional neural networks for no-reference image quality assessment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1733–1740 (2014)
8. Kang, L., Ye, P., Li, Y., Doermann, D.: Simultaneous estimation of image quality and distortion via multi-task convolutional neural networks. In: 2015 IEEE International Conference on Image Processing (ICIP), pp. 2791–2795. IEEE (2015)
9. Lepcha, D.C., Goyal, B., Dogra, A., Goyal, V.: Image super-resolution: a comprehensive review, recent trends, challenges and applications. *Inf. Fusion* **91**, 230–260 (2023)
10. Li, H., Zhang, K., Niu, Z., Shi, H.: C²mt: a credible and class-aware multi-task transformer for SR-IQA. *IEEE Signal Process. Lett.* **29**, 2662–2666 (2022)
11. Liu, L., Liu, B., Huang, H., Bovik, A.C.: No-reference image quality assessment based on spatial and spectral entropies. *Sig. Process. Image Commun.* **29**(8), 856–863 (2014)
12. Ma, C., Yang, C.Y., Yang, X., Yang, M.H.: Learning a no-reference quality metric for single-image super-resolution. *Comput. Vis. Image Underst.* **158**, 1–16 (2017)
13. Quan, X., Zhang, K., Li, H., Fan, D., Hu, Y., Chen, J.: Tadsrnet: a triple-attention dual-scale residual network for super-resolution image quality assessment. *Appl. Intell.* **53**(22), 26708–26724 (2023)
14. Saad, M.A., Bovik, A.C., Charrier, C.: Blind image quality assessment: a natural scene statistics approach in the DCT domain. *IEEE Trans. Image Process.* **21**(8), 3339–3352 (2012)
15. Sheikh, H.R., Bovik, A.C., De Veciana, G.: An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Trans. Image Process.* **14**(12), 2117–2128 (2005)
16. Shi, G., Wan, W., Wu, J., Xie, X., Dong, W., Wu, H.R.: Sisrset: single image super-resolution subjective evaluation test and objective quality assessment. *Neurocomputing* **360**, 37–51 (2019)
17. Su, S., et al.: Blindly assess image quality in the wild guided by a self-adaptive hyper network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3667–3676 (2020)

18. Wang, Z., Chen, J., Hoi, S.C.: Deep learning for image super-resolution: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(10), 3365–3387 (2020)
19. Xue, W., Zhang, L., Mou, X., Bovik, A.C.: Gradient magnitude similarity deviation: a highly efficient perceptual image quality index. *IEEE Trans. Image Process.* **23**(2), 684–695 (2013)
20. Yan, Q., Gong, D., Zhang, Y.: Two-stream convolutional networks for blind image quality assessment. *IEEE Trans. Image Process.* **28**(5), 2200–2211 (2018)
21. Yeganeh, H., Rostami, M., Wang, Z.: Objective quality assessment of interpolated natural images. *IEEE Trans. Image Process.* **24**(11), 4651–4663 (2015)
22. Zhang, H., Su, S., Zhu, Y., Sun, J., Zhang, Y.: Boosting no-reference super-resolution image quality assessment with knowledge distillation and extension. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE (2023)
23. Zhang, K., Zhu, D., Li, J., Gao, X., Gao, F., Lu, J.: Learning stacking regression for no-reference super-resolution image quality assessment. *Signal Process.* **178**, 107771 (2021)
24. Zhang, T., Zhang, K., Xiao, C., Xiong, Z., Lu, J.: Joint channel-spatial attention network for super-resolution image quality assessment. *Appl. Intell.* **52**(15), 17118–17132 (2022)
25. Zhang, W., Ma, K., Yan, J., Deng, D., Wang, Z.: Blind image quality assessment using a deep bilinear convolutional neural network. *IEEE Trans. Circuits Syst. Video Technol.* **30**(1), 36–47 (2020)
26. Zhang, Z., et al.: A no-reference evaluation metric for low-light image enhancement. In: *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6 (2021)
27. Zhang, Z., et al.: A no-reference deep learning quality assessment method for super-resolution images based on frequency maps. In: *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3170–3174. IEEE (2022)
28. Zhao, T., Lin, Y., Xu, Y., Chen, W., Wang, Z.: Learning-based quality assessment for image super-resolution. *IEEE Trans. Multimedia* **24**, 3570–3581 (2021)
29. Zhou, F., Yao, R., Liu, B., Qiu, G.: Visual quality assessment for super-resolved images: database and method. *IEEE Trans. Image Process.* **28**(7), 3528–3541 (2019)
30. Zhou, W., Jiang, Q., Wang, Y., Chen, Z., Li, W.: Blind quality assessment for image superresolution using deep two-stream convolutional networks. *Inf. Sci.* **528**, 205–218 (2020)
31. Zhou, W., Wang, Z.: Quality assessment of image super-resolution: balancing deterministic and statistical fidelity. In: *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 934–942 (2022)



Image Super-Resolution with Multi-scale Hybrid Attention

Ningzhi Wang^{1,2}, Hanyi Shi³(✉), Wenna Ruan^{1,2}, and Lingbin Zeng⁴

¹ School of Artificial Intelligence, Anhui University, Hefei, China

² Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China

³ Field Engineering College, Army Engineering University of PLA, Nanjing, China
spororange3187@gmail.com, shihanyi12@alumni.nudt.edu.cn

⁴ National University of Defense Technology, Changsha, China

Abstract. Super-resolution reconstruction stands as a critical task within the domain of computer vision. To enhance texture information extraction and improve visual perception, we introduce the Multi-Scale Hybrid Attention Network (MSHA), a novel single-image super-resolution model engineered to augment image global processing capabilities while accentuating detail reconstruction. The MSHA architecture seamlessly integrates a Multi-Scale Feature Block (MFB) for comprehensive feature extraction and a Parallel Hybrid Attention Module (PHA) for refining detail reconstruction capabilities. Through extensive experimentation and comparative analyses against state-of-the-art models, we substantiate the superior performance of the MSHA network in producing high-quality super-resolution images. Our methodology effectively addresses the shortcomings of existing approaches by emphasizing multi-scale feature extraction and detail reconstruction, thereby significantly advancing the field of image super-resolution.

Keywords: Multi-scale Feature Extraction · Parallel Hybrid Attention · Single Image Super-Resolution

1 Introduction

Image super-resolution is a pivotal technique in image processing, dedicated to reinstating intricate details in high-resolution images from their low-resolution counterparts. Within the domain of image super-resolution, methodologies can be classified into two distinct categories based on the quantity of input images: single-image approaches and multi-image approaches. Single-image techniques predominantly adopt two strategies: interpolation-based and learning-based methodologies [1]. Interpolation-based approaches elevate the resolution of low-resolution images through interpolation, often leading to blurred and distorted outputs [2]. Conversely, learning-based methodologies leverage deep learning models to assimilate image mapping relationships from extensive high-resolution

image datasets [3], subsequently applying these learned relationships to low-resolution inputs to yield clearer and more authentic high-resolution reconstructions. Prominent examples of such methodologies encompass the Super-Resolution Convolutional Neural Network (SRCNN) [4], Efficient Sub-Pixel Convolutional Neural Network (ESPCN) [5], and Super-Resolution Using a Generative Adversarial Network (SRGAN) [6], which have demonstrated remarkable performance in advancing image super-resolution techniques. Conversely, multi-image methodologies harness multiple low-resolution images to generate corresponding high-resolution reconstructions. Among these approaches, motion estimation-based methodologies exploit motion data among multiple low-resolution images to recuperate detailed information in the resulting high-resolution reconstruction, whereas self-similarity-based methodologies capitalize on self-similarity patterns within multiple low-resolution images to extract detailed information in the high-resolution reconstruction. By amalgamating information from multiple low-resolution inputs, these methodologies yield more precise and accurate high-resolution outputs.

With the proliferation of deep learning methodologies in recent years, learning-based single-image techniques have emerged as potent means to generate high-fidelity super-resolution imagery. These approaches excel by adeptly capturing intricate image mapping relationships through the utilization of sophisticated deep learning architectures. This trend, coupled with inherent drawbacks associated with multi-image methodologies, notably high data processing and computational expenses, has established single-image techniques as the predominant approach adopted across both industrial and academic domains.

In 2014, Dong et al. introduced the Super-Resolution Convolutional Neural Network (SRCNN), a seminal contribution that revolutionized single-image super-resolution (SISR) by leveraging a series of convolutional layers, rectified linear units, and up-sampling layers to learn the mapping from low-resolution to high-resolution images. SRCNN represented a significant breakthrough in the application of convolutional neural network models to SISR tasks. The efficacy of image super-resolution reconstruction employing convolutional neural network models primarily stems from their depth and incorporation of residual connections. Numerous scholars have subsequently proposed representative SISR methodologies, focusing on leveraging the depth and residual connectivity of convolutional neural networks to advance the field.

Despite the considerable success achieved by existing SISR methods, there persists a noticeable oversight in their design, particularly regarding the preservation of intricate details such as edge segments and dense textures. This limitation poses a barrier to the generation of high-quality super-resolution (SR) images. The introduction of attention mechanisms by previous researchers has brought attention to the potential to prioritize more effective feature components during the training process [7]. However, it is worth noting that these mechanisms, including the channel attention mechanism, primarily focus on amplifying or attenuating the influence of individual channels based on correlation informa-

tion. While such mechanisms have shown promise in improving metrics like peak signal-to-noise ratio (PSNR), they still fall short in addressing the finer details of image boundaries and textures.

In light of these limitations, we propose the development of the Multi-Scale Hybrid Attention (MSHA) network. This innovative approach incorporates a multi-scale feature extraction mechanism, enabling a holistic analysis of images from a global perspective. Central to this framework is the Multi-Scale Feature Extraction Block (MFB), which consists of an unconventional convolutional layer, an activation layer, and a Multi-Feature Hybrid Extraction (MFHE) process. The MFHE component utilizes null convolution following an expanded receptive field, thereby facilitating the capture of features across a broader spectrum while maintaining model simplicity.

Furthermore, our proposed MSHA network leverages the Parallel Hybrid Attention (PHA) module to refine the detail reconstruction capabilities of the model. By generating feature maps at various scales and inputting them into the PHA module, we ensure a comprehensive enhancement of the model's ability to reconstruct fine details. This strategic integration of multi-scale feature extraction and parallel hybrid attention mechanisms serves to overcome the limitations of existing methodologies and facilitate the production of high-fidelity SR images.

The main contributions of this article are as follows:

1. We propose the Multi-scale Hybrid Attention Network (MSHA), a novel single-image super-resolution model designed to enhance image global processing capabilities while prioritizing detail reconstruction.
2. We introduce the Multi-Scale Feature Block (MFB) to comprehensively capture essential features through multi-branch atrous convolution within the Multi-Feature Hybrid Extraction (MFHE) process. Subsequently, the corresponding feature image is fed into the Parallel Hybrid Attention Module (PHA) to augment feature expression across both channel space dimensions. Ultimately, this facilitates detailed reconstruction.
3. We conduct multiple rounds of experiments and the results show that our model outperforms other SOTA SISRs on multiple dimensions of multiple metric datasets.

2 Related Work

2.1 Deep CNN-Based Networks

Dong et al. pioneered the use of Convolutional Neural Networks (CNNs) for image super-resolution with their proposal of the SRCNN, marking a significant milestone in CNN-based SR tasks. Following this breakthrough, researchers embarked on optimizing network depth to further enhance performance. Kim et al. [8] introduced residual connections to construct deeper networks, leading to the development of the Very Deeply Structured Convolutional Network (VDSR),

which achieved superior performance in image super-resolution. Shi et al. [reference] introduced an efficient sub-pixel convolutional layer in the ESPCN, facilitating the upscaling of low-resolution feature maps to high-resolution images at the conclusion of training. This sub-pixel convolution has since emerged as a prominent architectural choice in deep network design for super-resolution tasks. Lim et al. [9] proposed the Enhanced Deep Residual Network (EDSR), refining the residual network architecture for super-resolution by eliminating unnecessary modules like batch normalization, thereby enhancing efficiency. Ahn [10] introduced the Cascaded Residual Network (CARN), a neural network architecture featuring cascaded modules and multiple shortcut connections, effectively leveraging multilevel representations to boost performance in image super-resolution tasks.

2.2 Enhanced Deep Residual Networks

The Enhanced Deep Residual Network (EDSR) stands as a cornerstone in image super-resolution tasks, elevating the performance of deep convolutional networks to new heights. Its significance is underscored by its selection as the baseline model in our study. In the Super-Resolution Residual Network (SRResNet) [6], the residual block comprises two batch normalization layers, two convolutional layers, a Rectified Linear Unit (ReLU) activation function, and residual connections. While batch normalization serves to expedite model convergence and mitigate overfitting—a widely acknowledged benefit across various deep learning tasks—it can inadvertently smooth out reconstructed super-resolution images, leading to texture loss. Thus, EDSR strategically omits batch normalization from its ResBlock. Moreover, to enhance model complexity, EDSR augments the number of output channels in the ResBlock to 256 and stacks 32 blocks for profound feature extraction. This architectural refinement underscores EDSR’s capability to generate reconstruction results surpassing those of competing algorithms. It substantiates the notion that a judiciously chosen network depth is pivotal in achieving superior performance in image super-resolution tasks.

2.3 Multi-scale Feature Fusion

Multi-scale feature fusion has become key to improving the performance of computer vision models, enabling the integration of features at different scales to capture both fine and coarse details. For example, Chen et al. (2021) [27] enhanced the detection capability of target objects by using multiple inputs to improve the extraction of effective information through multi-scale feature fusion. Similarly, Zhu et al. (2024) [28] proposed ConvNeXtFF, which employs multi-level down-sampling to obtain contextual information at different scales, thereby improving segmentation performance. In medical image analysis, Xie et al. (2023) [23] introduced a structure using dilated convolution to capture multi-scale features, demonstrating its effectiveness in image super-resolution tasks. Building upon the structure proposed by Xie et al., our work introduces the MSHA model,

which fully utilizes and integrates the information obtained from multi-scale feature processing.

3 Methodology

3.1 Network Architecture

As depicted in Fig. 1, the entire network architecture is delineated into three distinct components: shallow feature extraction, deep feature extraction, and super-resolution reconstruction. This architectural configuration, as evidenced by prior research [11, 12], has proven highly efficacious for image super-resolution (SR) tasks. In the shallow feature extraction stage, a single-layer convolution is employed to extract low-frequency information directly from the input image [3, 9]. This extracted structure serves as the foundation for subsequent deep feature extraction, ensuring the preservation of essential low-frequency components inherent to the image.

The deep feature extraction network is structured with multiple Residual Convolution Blocks (ResBlocks) and Multiscale Feature Blocks (MFBs). The MFBs are crucial for recalibrating feature weights and enhancing attention to detailed image components. This mechanism significantly enhances the network’s capacity to capture fine-grained information essential for high-quality super-resolution reconstruction. Subsequently, the deeply extracted features are propagated through the reconstruction network to produce the high-resolution (HR) image. The reconstruction network consists of a varying number of convolutions and sub-pixel convolutions, tailored to meet the specific requirements of the desired reconstruction size. This modular approach ensures adaptability and scalability in generating HR images across diverse resolution specifications.

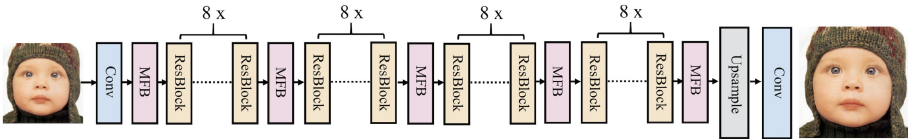


Fig. 1. General structure of our proposed Multi-Scale Hybrid Attention (MSHA) model

The data undergoes processing within the network as follows: the low-resolution image (LR) $I_{LR} \in \mathbb{R}^{H \times W \times C_{in}}$ is initially inputted into the network and forwarded to the shallow feature extraction network. Here, it undergoes a single layer of convolutional processing to derive shallow features $F_0 \in \mathbb{R}^{H \times W \times C}$.

$$F_0 = C_{SF}(I_{LR}) \tag{1}$$

$C_{SF}(\bullet)$ represents the convolution operation that performs shallow feature extraction. Subsequently the shallow features F_0 are fed into the deep feature extraction network for processing.

$$F_{DF} = C_{MR}(F_0) \quad (2)$$

The multi-scale and residual operations within the deep feature extraction network are denoted as $C_{MR}(\bullet)$ executed through MFBs and ResBlocks. Subsequently, the features F_{DF} , obtained after full image extraction, progress to the final stage of the model. Here, they are transmitted to the super-resolution reconstruction network, where a high-quality SR image $I_{LR} \in \mathbb{R}^{H \times W \times C_{out}}$ is generated through amplification via sub-pixel convolution.

$$I_{SR} = C_{UP}(F_{DF}) \quad (3)$$

Subsequently, our MSHA model employs the L_1 loss function to optimize its parameters.

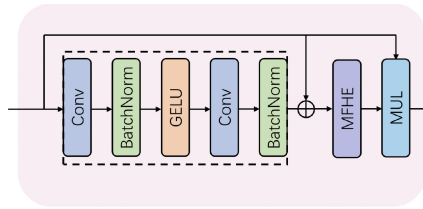


Fig. 2. The composition of Multi-scale Feature Block (MFB).

3.2 Multi-Scale Feature Block

The structure of our Multi-Scale Feature Module is illustrated in Fig. 2. The input feature initially undergoes convolutional processing, as depicted by the contents enclosed within the dotted box in the figure. We employ the Gaussian Error Linear Unit (GELU) activation function, which has been shown to exhibit exceptional performance in various image processing tasks [22, 23]. Unlike the baseline ResBlock, which omits Batch Normalization (BN), we reintroduce BN in our convolutional group. This choice is driven by the subsequent weighting operation on the features. BN enhances the regularization of feature representations and helps adjust input distributions to maintain stability. Consequently, this fosters more effective learning of feature relationships by the attention mechanism, ensuring robustness against distribution fluctuations.

$$F'_0 = C_G(F_0) \quad (4)$$

After the convolution group processing, $F'_0 \in \mathbb{R}^{H \times W \times C}$ and F_0 exchange information through the Local Feature Fusion (LFF) mechanism and then enter the multi-feature hybrid extraction module.

$$F''_0 = C_{LF}(F_0 + F'_0) \quad (5)$$

Through iterative learning processes within the Multi-Feature Hybrid Extraction Module, intricate features are comprehensively captured, and the feature weights of each channel are judiciously allocated. Once fine-grained processing is accomplished, the next step involves acquiring low-level global features to complement these details. To facilitate the exchange of high-level and low-level feature information, we employ Global Feature Fusion (GFF). Notably, rather than directly adding features, a multiplication operation is conducted. This meticulous approach enables precise control over information flow, minimizing information loss, and empowering the model to discern the relative importance of features.

$$F_1 = MUL \left(MFHE \left(F_0'' \right) + F_0 \right) \tag{6}$$

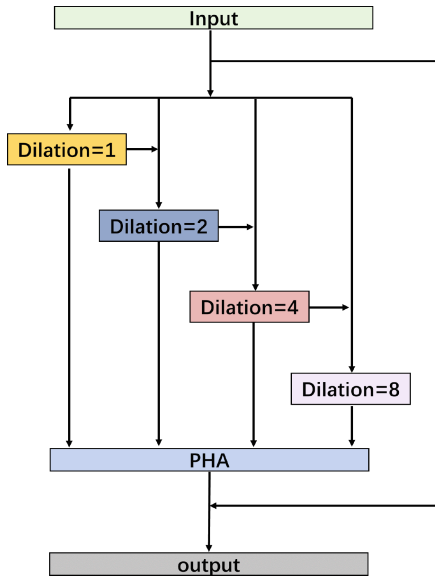


Fig. 3. Multi-Feature Hybrid Extraction(MFHE) in MFB.

3.3 Multi-Feature Hybrid Extraction

As implied by its name, the Multi-Feature Hybrid Extraction module encompasses both extraction and fusion processes. As illustrated in Fig. 3, the input feature F_n undergoes four 3×3 atrous convolution layers with varying expansion rates, enabling the capture of multi-scale image features. Following each branch’s convolution operation, the processed features are shared across adjacent branches to facilitate information exchange across different scales. This feature extraction segment not only maintains network depth in a parallel manner

but also enhances network width, thereby improving the network’s capability to extract both fine local textures and broad global semantic information.

After completing the feature extraction operation in each branch, the resulting feature maps are passed to the PHA module for hybrid weighting operations. Subsequently, the input feature F_n and the features obtained through the hybrid extraction process are fused via residual connections to produce the output feature F'_n .

$$F'_n = PHA(C_{d=1}(F_{n_1})) + \sum_{j=(1,2,3)} PHA\left(C_{d=i}\left(F'_{n_j} + F_{n_{j+1}}\right)\right)_{i=(2,4,8)} + F_n \quad (7)$$

3.4 Parallel Hybrid Attention

The PHA mechanism extends from channel attention principles by integrating channel and spatial dimensions to extract comprehensive and detailed key features. It also enables the fusion of features across various scales. Detailed illustrations of PHA are provided in Fig. 4.

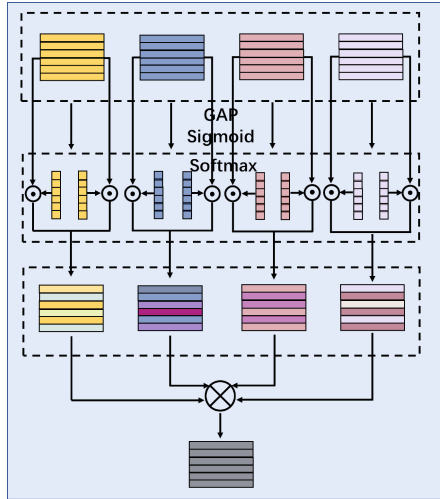


Fig. 4. Illustration of Parallel Hybrid Attention (PHA) module.

Given M feature representations $[F'_{n_1}, F'_{n_2}, \dots, F'_{n_m}] \in \mathbb{R}^{C \times H \times W}$ of identical sizes, the global feature representation is initially derived through global average pooling, capturing long-distance dependencies in both horizontal and vertical directions while preserving positional relationships. Subsequently, 1×1 convolutional operation models inter-channel correlations, generating channel descriptors via a sigmoid activation function.

$$X_{n_i} \in \mathbb{R}^{1 \times 1 \times W} = \text{sigmoid}\left(\text{Conv}\left(\text{AvgPool}_H\left(F'_{n_1}\right)\right)\right) \quad i = (1, \dots, m) \quad (8)$$

$$Y_{n_j} \in \mathbb{R}^{1 \times H \times 1} = \text{sigmoid} \left(\text{Conv} \left(\text{AvgPool}_W \left(F'_{n_1} \right) \right) \right) \quad j = (1, \dots, m) \quad (9)$$

Following this, two sets of orthogonal global features, each corresponding to M scales, are concatenated along the second dimension, representing the channel dimension, resulting in feature sets $X' \in \mathbb{R}^{M \times 1 \times W}$ and $Y' \in \mathbb{R}^{M \times H \times 1}$. Subsequently, multi-scale channel weights are derived through the application of the *Softmax* function along the channel dimension. Finally, the features are element-wise multiplied by the corresponding normalized weights, and the processed features are summed to obtain the new multi-scale mixed feature F'_n .

$$W_x[:, i, :] \in \mathbb{R}^{1 \times 1 \times W} \quad W_y[:, i, :] \in \mathbb{R}^{1 \times H \times 1} \quad (10)$$

$$F'_n = F_n + W_x[:, i, :] \odot F'_{n_i} + W_y[:, j, :] \odot F'_{n_j} \quad i=(1, \dots, m) \quad j=(1, \dots, m) \quad (11)$$

4 Experiments

4.1 Datasets and Metrics

In our experiments, we employed the DIV2K dataset for training, which encompasses 800 high-quality images. This dataset offers a wide range of image types, including natural landscapes, portraits, animals, and more, making it well-suited for assessing the generalizability of image super-resolution algorithms. For testing purposes, we utilized standard datasets such as Set5, Set14, B100, Urban100, and Manga109. To evaluate the results, we measured the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) metrics specifically on the Y channel within the transformed YCbCr color space.

4.2 Experiment Details

For Multi-Features Extraction, we utilize atrous rates of (2, 4, 8), which have been validated through extensive experiments to exhibit exceptional performance for the MFEB. The complete model comprises 32 ResBlocks and 5 MFBs. Moreover, all ResBlock intermediate feature channels align with those in EDSR, set to 256. Regarding input data, we utilize LR images with a batch size of 64×4 alongside their corresponding HR counterparts. The learning rate is initialized to $1e-4$ and is halved every 200 epochs, ultimately reaching the final model after 1000 epochs. We employ the ADAM optimizer with parameters $\beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The experiments are conducted using the PyTorch framework and trained on an Nvidia 3090 GPU for efficient computation.

4.3 Ablation Study

In this section, we undertake multiple sets of ablation experiments on the proposed Multi-scale Feature Block to assess the efficiency of related modules. Initially, we address the necessity of normalization operations within the MFB.

Table 1. Effectiveness of relevant components in MFB in the Set14 when scale is 2.

Module	PSNR/SSIM
Pure	33.65/0.9172
Pure+BN	33.72/0.9183
Pure+LFF	33.70/0.9173
Pure+GLF	33.76/0.9189
Pure+BN+LFF+GLF(MFB)	34.04/0.9228

While normalization operations are omitted in many super-resolution reconstruction networks, our MFB requires normalization to facilitate subsequent hybrid extraction processes. Simultaneously, we hypothesize that both the Local Feature Fusion within the convolution group and the overall Global Feature Fusion play corresponding roles in capturing effective features, as delineated in Table 1.

Next, we delve into the investigation of the number of MFBs. While the performance of deep convolutional neural networks is partly attributed to the depth of their stacking, in super-resolution tasks, it’s known that deeper networks may not always yield better results; the optimal depth is crucial. Accordingly, we insert 5 ResBlocks into the baseline for each MFB interval. The exploration results regarding the number of MFBs employed will be presented in Table 2.

Table 2. Results of different usage of MFB in the network for each batchmarks at x4 scale.

Number of Blocks	Set5		Set14		B100		Urban100		Mange109	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
3	32.31	0.8956	28.70	0.7834	27.63	0.7386	26.20	0.7875	30.70	0.9107
4	32.43	0.8953	28.78	0.7863	27.70	0.7415	26.72	0.8060	31.14	0.9169
5	32.58	0.9000	28.85	0.7875	27.75	0.7423	26.89	0.8072	31.28	0.9181
6	32.40	0.8946	28.78	0.7861	27.71	0.7415	26.70	0.8060	31.02	0.9154
7	32.37	0.8942	28.76	0.7858	27.64	0.7392	26.53	0.8042	30.87	0.9130

Finally, we selected several models and conducted a comparative analysis of FLOPs, parameter count, and running time with the proposed MSHA, as presented in Table 3. Our findings indicate that, in order to comprehensively capture effective features, our model exhibits an increased computational workload and parameter count relative to the baseline. However, through optimization of running time and overall performance, we have achieved superior results.

The results substantiate our hypothesis that blindly increasing network depth does not necessarily lead to improved performance. Additionally, it underscores the notion that the attention mechanism should not be overused. Improper usage not only escalates computational costs and reduces model efficiency but also

Table 3. Comparing the complexity of different Methods, Complexity metrics and PSNR are evaluated on Manga109 (x2)

Method \ Metrics	MSRN	HRAN	EDSR	RDN	MSHA
FLOPs(G)	297.6	388.5	2043.7	1109.8	2089.2
Param(M)	5.9	7.9	40.7	20.3	52.1
PSNR(dB)	38.82	39.12	39.10	39.18	39.47
Runtime(s)	0.06	0.1565	0.09	0.1132	0.0987

risks overfitting irrelevant information in the data, consequently diminishing the model’s generalization ability and overall performance.

4.4 Comparisons with State-of-the-Arts

To assess the performance of MSHA, we conduct comparative evaluations against 10 state-of-the-art methods, namely, WMRN [14], MSRN [15], SeaNet [16], EDSR [9], RDN [3], SRFBN [17], DBPN [18], WDRN [19], MGAN [20], HRAN [21], DDistill [24], TPCNN [25], and ESRT [26]. The datasets and visualization results utilized in the comparisons are sourced from the respective authors of each paper or reproduced independently.

Quantitative Evaluation. As depicted in Table 4, a clear comparison between our model and other SOTA methods is presented. It’s evident that due to MSHA’s multi-scale fusion functionality, its adeptness at capturing global information and collecting detailed textures surpasses that of other SOTAs. This superiority is corroborated by both qualitative observations and numerical indicators. Across all benchmarks, our MSHA consistently outperforms other methods in terms of PSNR and SSIM at nearly every scale, affirming its efficacy in enhancing super-resolution image quality.

Qualitative Evaluation. To experience the actual perceptual impact of MSHA-processed images, we present the reconstruction results after applying MSHA alongside the results of other methods on each benchmark dataset. As illustrated in Figs. 5, we evaluate the image reconstruction effects of different methods across four benchmarks, each at a scale factor of x4. From these results, it’s evident that most methods struggle to achieve satisfactory completion in the detailed texture areas, yielding only vague results. Conversely, MSHA consistently produces clearer and more natural textures. This compelling outcome further underscores the efficacy of our multi-scale hybrid attention mechanism in image super-resolution tasks.

Table 4. Quantitative evaluation of the average values of PSNR and SSIM of SOTA methods on several benchmarks with scale factor $\times 2$, $\times 3$, and $\times 4$. Red and blue suggest optimal and suboptimal results.

Method	Scale	Set5		Set14		B100		Urban100		Mange109		
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
WMRN[14]	x2	37.83	0.9599	33.41	0.9162	32.08	0.8984	31.68	0.9241	38.27	0.9763	
MSRN[15]		38.08	0.9605	33.74	0.9170	32.23	0.9013	32.22	0.9326	38.82	0.9868	
SeaNet[16]		38.15	0.9611	33.86	0.9198	32.31	0.9013	32.68	0.9332	38.76	0.9774	
EDSR[9]		38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	39.10	0.9773	
RDN[13]		38.24	0.9614	34.01	0.9212	32.34	0.9017	32.89	0.9353	39.18	0.9780	
SRFBN[17]		38.11	0.9609	33.82	0.9196	32.29	0.9010	32.62	0.9328	39.08	0.9779	
DBPN[18]		38.09	0.9600	33.85	0.9190	32.27	0.9000	32.55	0.9324	38.89	0.9775	
WDRN[19]		38.19	0.9631	33.39	0.9212	32.27	0.9014	32.64	0.9372	-	-	
MGAN[20]		38.16	0.9612	33.83	0.9198	32.28	0.9009	32.75	0.9340	39.11	0.9778	
HRAN[21]		38.32	0.9613	33.85	0.9200	32.34	0.9016	32.95	0.9357	39.12	0.9780	
DDistill[24]		38.08	0.9608	33.73	0.9195	32.35	0.9007	32.39	0.9301	39.16	0.9781	
TPCNN[25]		38.03	0.9613	33.67	0.9187	32.25	0.9014	31.76	0.9257	-	-	
ESRT[26]		-	-	-	-	-	-	-	-	-	-	
MSHA(Ours)		x2	38.39	0.9623	34.04	0.9228	32.40	0.9023	33.24	0.9376	39.47	0.9787
WMRN[14]	x3	34.11	0.9251	30.17	0.8390	28.98	0.8021	27.80	0.8448	33.07	0.9413	
MSRN[15]		34.38	0.9262	30.34	0.8395	29.08	0.8041	28.08	0.8554	33.44	0.9427	
SeaNet[16]		34.65	0.9290	30.53	0.8461	29.23	0.8081	28.68	0.8620	33.73	0.9463	
EDSR[9]		34.65	0.9280	30.52	0.8462	29.25	0.8093	28.80	0.8653	34.17	0.9476	
RDN[13]		34.71	0.9296	30.57	0.8468	29.26	0.8093	28.80	0.8653	34.13	0.9484	
SRFBN[17]		34.70	0.9292	30.51	0.8461	29.24	0.8084	28.73	0.8461	34.18	0.9481	
WDRN[19]		34.62	0.9292	30.50	0.8454	29.20	0.8085	28.59	0.8625	-	-	
MGAN[20]		34.65	0.9292	30.51	0.8460	29.22	0.8086	28.61	0.8621	34.00	0.9474	
HRAN[21]		34.69	0.9292	30.54	0.8463	29.25	0.8089	28.76	0.8645	34.08	0.9479	
DDistill[24]		34.43	0.9276	30.39	0.8432	29.16	0.8070	28.31	0.8546	33.97	0.9465	
TPCNN[25]		34.43	0.9281	30.48	0.8451	29.16	0.8085	28.59	0.8625	-	-	
ESRT[26]		34.42	0.9268	30.43	0.8433	29.15	0.8063	28.46	0.8574	33.95	0.9455	
MSHA(Ours)		x3	34.79	0.9308	30.67	0.8479	29.29	0.8102	28.95	0.8678	34.41	0.9493
WMRN[14]		x4	32.00	0.8925	28.47	0.7786	27.49	0.7328	25.89	0.7789	30.11	0.9040
MSRN[15]	32.07		0.8903	28.60	0.7751	27.52	0.7273	26.04	0.7896	30.17	0.9034	
SeaNet[16]	32.44		0.8981	28.81	0.7872	27.70	0.7399	26.50	0.7976	30.74	0.9129	
EDSR[9]	32.46		0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033	31.02	0.9148	
RDN[13]	32.47		0.8990	28.81	0.7871	27.72	0.7419	26.61	0.8028	31.00	0.9151	
SRFBN[17]	32.47		0.8983	28.81	0.7868	27.72	0.7409	26.60	0.8015	31.15	0.9160	
DBPN [18]	32.47		0.8980	28.82	0.7860	27.72	0.7400	26.38	0.7946	30.91	0.9137	
WDRN[19]	32.43		0.8985	28.75	0.7862	27.65	0.7384	26.41	0.7975	-	-	
MGAN[20]	32.45		0.8980	28.74	0.7852	27.68	0.7400	26.74	0.7981	30.81	0.9131	
HRAN[21]	32.43		0.8976	28.76	0.7863	27.70	0.7407	26.55	0.8006	30.94	0.9143	
DDistill[24]	32.29		0.8961	28.69	0.7833	27.65	0.7385	26.25	0.7893	30.79	0.9098	
TPCNN[25]	32.14		0.8957	28.72	0.7846	27.62	0.7381	26.00	0.7835	-	-	
ESRT[26]	32.19		0.8947	28.69	0.7833	27.69	0.7379	26.39	0.7962	30.75	0.9100	
MSHA(Ours)	x4		32.58	0.9000	28.85	0.7875	27.75	0.7423	26.89	0.8072	31.28	0.9181

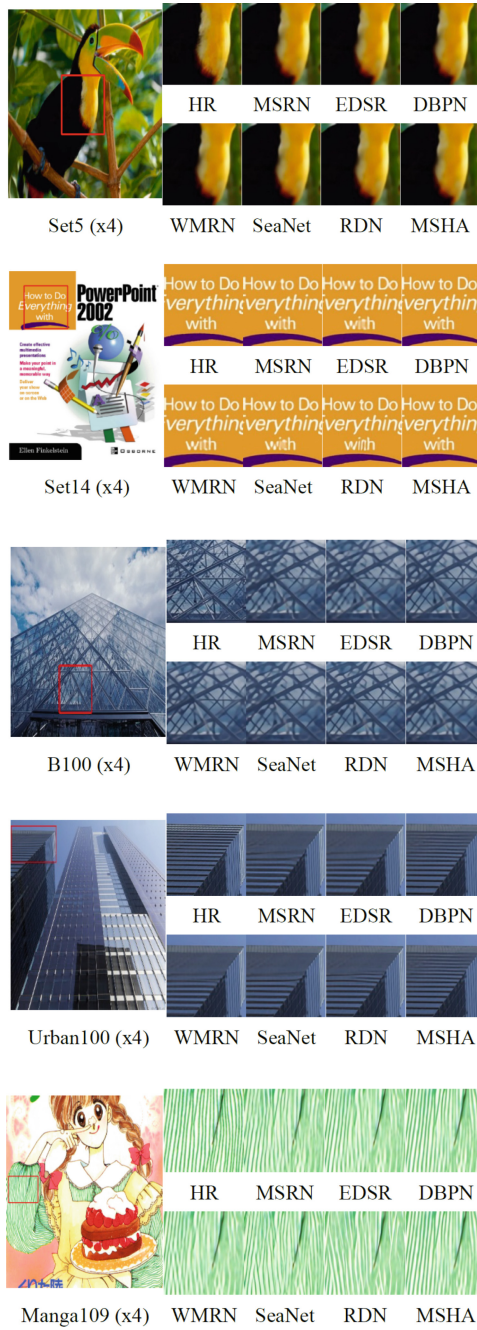


Fig. 5. Visualizing the results of MSHA and SOTAs at a scale of $\times 4$, it is clear to perceive that the quality of MSHA’s reconstruction for details is higher than that of most of the other models.

5 Conclusion

To address the limitations of existing SISR methods in recognizing detailed textures and processing images with larger field of views, we introduce a novel SISR approach called the Multi-Scale Hybrid Attention Network (MSHA). MSHA leverages a larger receptive field at multiple scales during feature extraction, enabling better inference based on adjacent information. Additionally, MSHA prioritizes detail enhancement, leading to significant improvements in the perceptual quality of the reconstructed images.

References

1. Freeman, W.T.: Learning low-level vision. *Int. J. Comput. Vision* **40**, 25–47 (2000)
2. Zhang, L.: An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE Trans. Image Process.* **15**(8), 2226–2238 (2006)
3. Zhang Y., Tian Y.: Residual dense network for image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2472–2481 (2018)
4. Dong, C.: Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(2), 295–307 (2015)
5. Shi, W., Caballero, J.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1874–1883 (2016)
6. Ledig, C., Theis, L.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4681–4690 (2017)
7. Mnih, V.: Recurrent models of visual attention. In: *Advances in Neural Information Processing Systems*, vol. 27 (2014)
8. Kim, J., Lee, J.K.: 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)
9. Lim, B., Son, S.: Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 136–144 (2017)
10. Ahn, N., Kang, B.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: Proceedings of the European Conference on Computer Vision, pp. 252–268 (2018)
11. He, K., Zhang, X.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
12. Zhang, Y., Li, K.: Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European Conference on Computer Vision, pp. 286–301 (2018)
13. Hendrycks, D., Gimpel, K.: Learning low-level vision. *Int. J. Comput. Vision* **40**, 25–47 (2000)
14. Sun, L.: Lightweight image super-resolution via weighted multi-scale residual network. *IEEE/CAA J. Autom. Sinica* **8**(7), 1271–1280 (2021)
15. Li, J., Fang, F.: Multi-scale residual network for image super-resolution. In: Proceedings of the European Conference on Computer Vision, pp. 517–532 (2018)

16. Fang, F.: Soft-edge assisted network for single image super-resolution. *IEEE Trans. Image Process.* **29**, 4656–4668 (2020)
17. Li Z., Yang, J.: Feedback network for image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3867–3876 (2019)
18. Haris, M., Shakhnarovich, G.: Deep back-projection networks for super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1664–1673 (2018)
19. Xin, J.: Wavelet-based dual recursive network for image super-resolution. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(2), 707–720 (2020)
20. Wu, H.: Multi-grained attention networks for single image super-resolution. *IEEE Trans. Circuits Syst. Video Technol.* **31**(2), 512–522 (2020)
21. Muqet, A.: HRAN: hybrid residual attention network for single image super-resolution. *IEEE Access* **7**, 137020–137029 (2019)
22. Hendrycks, D.: Gaussian error linear units. arXiv preprint [arXiv:1606.08415](https://arxiv.org/abs/1606.08415), (2016)
23. Xie, L., Li, C.: SHSRCNet: super-resolution and classification network for low-resolution breast cancer histopathology image. In: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 23–32 (2023)
24. Wang, Y.: Ddistill-SR: Reparameterized dynamic distillation network for light-weight image super-resolution. *IEEE Trans. Multimedia* **33**, 7222–7234 (2023)
25. Alireza, E.: Ultralight-weight three-prior convolutional neural network for single image super resolution. *IEEE Trans. Artif. Intell.* **4**(6), 1724–1738 (2023)
26. Zhisheng, L.: Transformer for single image super-resolution. In: Proceedings of the 35th IEEE/CVF Conference on Computer Vision and Pattern Recognition workshops, pp. 456–465 (2022)
27. Qiang, C.: You only look one-level feature. In: Proceedings of the 34th IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13034–13043 (2021)
28. HaoChen, Z.: Effective image tampering localization with multi-scale ConvNeXt feature fusion. *J. Vis. Commun. Image Represent.* **98**, 1724–1738 (2024)



A Sinkhorn Regularized Adversarial Network for Image Guided DEM Super-resolution Using Frequency Selective Hybrid Graph Transformer

Subhajit Paul^(✉) and Ashutosh Gupta

Space Applications Centre (SAC), ISRO, Ahmedabad, India
{subhajitpaul, ashutoshg}@sac.isro.gov.in

Abstract. Digital Elevation Model (DEM) is an essential aspect in the remote sensing (RS) domain to analyze various applications related to surface elevations. Here, we address the generation of high-resolution (HR) DEMs using HR multi-spectral (MX) satellite imagery as a guide by introducing a novel hybrid transformer model consisting of Densely connected Multi-Residual Block (DMRB) and multi-headed Frequency Selective Graph Attention (M-FSGA). To promptly regulate this process, we utilize the notion of discriminator spatial maps as the conditional attention to the MX guide. Further, we present a novel adversarial objective related to optimizing Sinkhorn distance with classical GAN. In this regard, we provide both theoretical and empirical substantiation of better performance in terms of vanishing gradient issues and numerical convergence. Based on our experiments on 4 different DEM datasets, we demonstrate both qualitative and quantitative comparisons with available baseline methods and show that the performance of our proposed model is superior to others with sharper details and minimal errors.

Keywords: Sinkhorn loss · Graph Attention · Adversarial learning

1 Introduction

The Digital Elevation Model (DEM) is a digital representation of any three-dimensional surface. It is immensely useful in precision satellite data processing, geographic information systems, hydrological studies, urban planning [27], and many other key applications. The main sources of DEM generation are terrestrial, airborne, or spaceborne, depending on the platform used for data acquisition. However, each of these scenarios has its own set of advantages and disadvantages. While elevation models generated using terrestrial and airborne systems have a high spatial resolution, their coverage is quite restricted and

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78498-9_27.

they typically suffer from several issues and systematic errors [22]. Space-borne missions such as SRTM, and ASTER [1, 10], on the other hand, have almost global coverage but lack the spatial resolution. Due to the emerging significance and diverse applications of DEM, both its accuracy and resolution have a substantial impact in different fields of operation [18]. However, HR DEM products are expensive, as they require special acquisition and processing techniques. As an alternative to generating HR DEM from scratch, enhancing the resolution (super-resolution) of existing DEM datasets can be seen as the most optimal strategy to address the shortfall. Hence, we intend to take a step in this direction to generate HR DEM and, to make it more tractable, we formulate this problem in an image super-resolution (SR) setting. As shown in Fig. 1, our primary objective is to synthesize HR DEM provided its coarser resolution and existing False Colour Composite (FCC) of HR MX imagery.

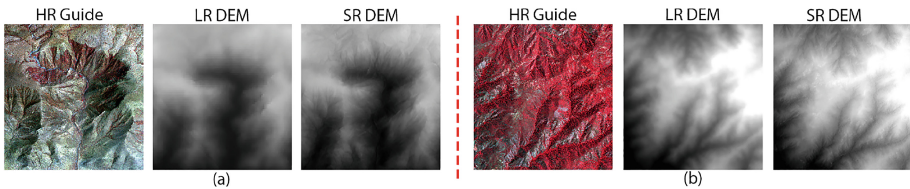


Fig. 1. Two sample results of DEM SR consisting HR FCC of NIR(R), R(G), and G(B), Bicubic interpolated LR DEM, and Generated HR DEM, respectively.

Recent advances in deep learning (DL) show compelling progress over conventional approaches for various computer vision applications like image or video SR. However, we found that very few methods approach the problem of DEM SR, especially, for real-world datasets. We propose a novel framework, which effectively addresses this problem. Our key contributions can be summarized as

1. We propose a novel architecture for DEM SR based on a hybrid transformer block consisting of a Densely connected Multi-Residual Block (DMRB) and multi-headed Frequency Selective Graph Attention (M-FSGA), which effectively utilizes information from an HR MX image as a guide by conditioning it with a discriminative spatial self-attention (DSA).
2. We develop and demonstrate SiRAN, a framework based on Sinkhorn regularized adversarial learning. We provide theoretical and empirical justification for its effectiveness in resolving the vanishing gradient issue while leveraging tighter iteration complexity.
3. We generate our own dataset where we take realistic coarse resolution data instead of considering bicubic downsampled HR image as input.
4. We perform experiments to assess the performance of our model along with ablation studies to show the impact of the different configuration choices.

2 Related Work

Traditional DEM super-resolution (SR) methods include interpolation-based techniques like linear, and bicubic, but they under-perform at high-frequency regions producing smoothed outputs. To preserve edge information, multiple reconstruction-based methods like steering kernel regression (SKR) [34] or non-local means (NLM) [28], have also been proposed. Though they can fulfill their primary objective, they cannot produce SR DEM at a large magnification factor.

DEM is an essential component for RS applications, but research on DEM SR is still limited. After the introduction of SR using Convolutional Neural Network (SRCNN) in the category of single image SR (SISR), its variant D-SRCNN was proposed by [5] to address the DEM SR problem. Later, Xu *et al.* [38] uses the concept of transfer learning where an EDSR (Enhanced Deep SR) [20], pre-trained over natural images, is taken to obtain an HR gradient map which is fine-tuned to generate HR DEM. After the introduction of Generative Adversarial Network (GAN), a substantial number of methods have evolved in the field of SR like Super-resolution using GANs (SRGAN). Based on this recently, Benkir *et al.* [8] proposed a DEM SR model, namely D-SRGAN, and later they suggested another model based on EfficientNetV2 [7] for DEM SISR. Although D-SRGAN produces good perceptual SR DEMs, it usually results in noisy predicted samples. They also suffer from issues of conventional GAN, mode collapse, and vanishing gradients. To resolve this, Wasserstein GAN (WGAN) [2] and its other variants [14] have been introduced. However, these methods are computationally expensive, which can be untangled by introducing an entropic regularization term [6]. In this study, we explore the efficacy of sinkhorn distance [13] in DEM SR, which is one of the forms of entropic optimal transport (EOT).

Recently, Li *et al.* [15, 24] proposed DEM SR algorithms using a global Kriging interpolation based information supplement module and a CNN based local feature generation module. It results preferably as a SISR technique, but, in practical scenarios, it generates artifacts near boundary regions and are unable to reproduce the very fine ground truth (GT) details in the predicted SR. Hence, here we propose a guided SR technique which is a key research area in computer vision, especially for depth estimation. One of the pioneering works in this domain is [17], where Kim *et al.* proposes Deformable Kernel Networks (DKN) and Faster DKN (FDKN) which learn sparse and spatially invariant filter kernels. Later, He *et al.* [16] exerts a high-frequency guided module to embed the guide details in the depth map. Recently, Metzger *et al.* [25] has achieved baseline performance by adapting the concept of guided anisotropic diffusion with CNNs. Our proposed method aligns with such depth SR methods as we leverage important HR MX features to generate SR DEM. To address this promptly, we incorporate a graph-based attention due to their efficacy in representation learning for image restoration tasks [23, 32]. However, these works are extended versions of graph neural networks (GNNs) which suffer from over-smoothing problems. To resolve this, [39, 40] utilizes GNN based on filtering in the frequency domain. Despite its efficacy in different DL tasks, it is not properly explored for vision

tasks. Hence, here we design our graph attention based on its selected frequencies.

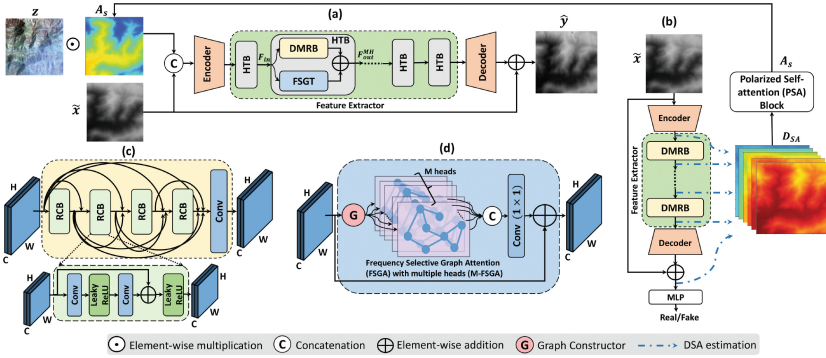


Fig. 2. Overview of proposed framework. (a) The generator G have multiple HTBs with parallely connected (c) DMRB and (d) FSGT. Given guide \mathbf{z} and upsampled LR DEM $\tilde{\mathbf{x}}$ to G , each HTB extracts global selective frequency information by FSGT and dense local features via DMRBs in latent space. (b) The discriminator D consists of only DMRBs. Besides classifying predicted $\hat{\mathbf{y}}$ and GT \mathbf{y} as real or fake, D also estimates DSA \mathbf{D}_{SA} with input $\tilde{\mathbf{x}}$. \mathbf{D}_{SA} is passed through a PSA [21] block to estimate \mathbf{A}_s which acted as spatial attention for HR guide \mathbf{z} during passing it to G along $\tilde{\mathbf{x}}$.

3 Methodology

In Fig. 2, we have illustrated the architecture of our framework. The generator G takes upsampled low-resolution (LR) DEM $\tilde{\mathbf{x}}$, and HR MX image guide \mathbf{z} , consisting FCC of NIR, red and green bands as input. Let $\mathbf{z} \sim \mathbb{P}_Z$, where $\mathbf{z} \in \mathbb{R}^{H \times W \times 3}$ with \mathbb{P}_Z being the joint distribution of FCC composition and $\tilde{\mathbf{x}} \sim \mathbb{P}_{\tilde{\mathbf{x}}}$, where $\mathbb{P}_{\tilde{\mathbf{x}}}$ constitute of upsampled LR DEM with $\tilde{\mathbf{x}} \in \mathbb{R}^{H \times W}$. Let $\hat{\mathbf{y}} \sim \mathbb{P}_{G_\theta}$ be the predicted SR DEM where \mathbb{P}_{G_θ} is the generator distribution parameterized by $\theta \in \Theta$, parameters of set of all possible generators. Let $\mathbf{y} \sim \mathbb{P}_y$ with \mathbb{P}_y represents the target HR DEM distribution. The discriminator D classifies \mathbf{y} and $\hat{\mathbf{y}}$ as real or fake, and is assumed to be parameterized by $\psi \in \Psi$, parameters of a set of all possible discriminators. Our D is also designed to estimate spatial attention \mathbf{D}_{SA} from its latent space features with LR DEM $\tilde{\mathbf{x}}$ as input as shown in Fig. 2. Since \mathbf{D}_{SA} contains discriminative information of HR DEM, it acts as spatial attention for \mathbf{z} allowing the model to focus on salient parts of it and avoid generating out-of-distribution (OOD) image information in the predicted SR DEM. To ensure this further, we process \mathbf{D}_{SA} through a self-attention (SA) block PSA [21] to remove redundant semantics, resulting in an enhanced representative attention map \mathbf{A}_s as demonstrated in Fig. 2. Therefore, the predicted SR DEM ($\hat{\mathbf{y}}$) is estimated as $\hat{\mathbf{y}} = G(\tilde{\mathbf{x}}, \mathbf{z} \odot \mathbf{A}_s)$, where \odot denotes element-wise multiplication.

3.1 Network Architecture

As shown in Fig. 2, G is designed based on a novel hybrid transformer block (HTB) [41, 43] due to their effectiveness in capturing both long-distance as well as local relations in image restoration tasks. Our HTB consists of a DMRB and a FSGT block. DMRB is developed based on ResNet and DenseNet by using both skip and dense connections. Each DMRB block is constituted of multiple densely connected Residual Convolution Blocks (RCBs). DMRB enables efficient context propagation and also stable gradient flow throughout the network while allowing local dense feature extraction. We introduce FSGT to leverage the extraction of global structural and positional relationships between spatially distant but semantically related regions. We use similar design for D . Both incorporate an encoder followed by a feature extractor and finally, a decoder. The feature extractor in G consists of six HTBs while for D , it only consists of six DMRBs to extract dense discriminative latent space features, which are used as spatial attention to the HR MX guide. D also adds a Multi-Layer Perceptron (MLP) layer to map its latent features into the required shape. We avoid using batch normalization as it degrades the performance and gives sub-optimal results for image SR [36] tasks. Next, we discuss the functionality of FSGT and DSA.

3.2 Frequency Selective Graph Transformer (FSGT) Module

To exploit high-frequency sharp details from HR guide and enhance latent feature representations, we propose a novel graph transformer, FSGT. As shown in Fig. 3, for a given input $\mathbf{F}_{\text{in}} \in \mathbb{R}^{H \times W \times C}$, FSGT extracts N patches using the patch generation method in W-MSA to construct the graph followed by a FSGA block for graph processing. A graph is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $\mathcal{V} = \{v_i | v_i \in \mathbb{R}^{h \times w \times c}, i = 1, \dots, N\}$, where h , w and c denotes height, width and channels for each patch represented as node and \mathcal{E} is the set of all the edges connecting these nodes. The edge weights are defined by an adjacency matrix $\mathcal{A} \in \mathbb{R}^{N \times N}$. The value of N is decided by the shape of each patch (h, w).

As shown in Fig. 3 (a), we build the graph connections by computing the similarities [44] between the nodes after the linear transformation as $\mathcal{A}_{i,j} = \langle f_1(v_i), f_2(v_j) \rangle$, where $\langle \cdot, \cdot \rangle$ is the inner product, v_i and v_j are i -th and j -th node, and f_1 and f_2 corresponds to 1×1 convolution. However, the generated graph \mathcal{G} is dense connecting every node to every other node. Thus, low similarities between some nodes confuse the model on how close different nodes are in the graph. This redundant information will hamper the objective and quality of graph reconstruction. To tackle this, we design FSGA to focus on high-frequency features and also generate a sparse representative graph.

Fig. 3(b) shows the detailed workflow of FSGA. Initially, the nodes \mathcal{V} are flattened out and converted to a matrix $\mathbf{X} \in \mathbb{R}^{N \times hwc}$ as shown in Fig. 3(a). It is later projected to query (\mathbf{Q}), key (\mathbf{K}) and value (\mathbf{V}) matrices with $\mathbf{Q} = \mathbf{X}\mathbf{W}_q$, $\mathbf{K} = \mathbf{X}\mathbf{W}_k$ and $\mathbf{V} = \mathbf{X}\mathbf{W}_v$, with \mathbf{W}_q , \mathbf{W}_k , and \mathbf{W}_v being learnable projection weights. However, instead of using \mathbf{K} directly, we filter out certain nodes in \mathbf{X} based on graph Fourier transform (GFT) to generate filtered graph matrix as $\bar{\mathbf{X}}$.

From this the updated key matrix is computed as $\hat{\mathbf{K}} = \bar{\mathbf{X}}\mathbf{W}_k$ which is used to get the attention as $\mathbf{A} = \text{Softmax}(\mathbf{Q}\hat{\mathbf{K}}^T)/\sqrt{d}$.

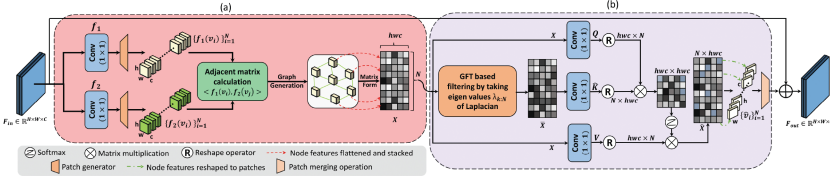


Fig. 3. Workflow of FSGT, (a) graph construction mechanism, (b) FSGA block

Graph signals can be analyzed in the frequency domain [31] by using normalized Laplacian $\mathcal{L} = \mathbf{I} - \mathcal{D}^{-\frac{1}{2}}\mathcal{A}\mathcal{D}^{-\frac{1}{2}}$, where \mathbf{I} is the identity matrix and \mathcal{D} is the diagonal matrix with $\mathcal{D}_{ii} = \sum_j \mathcal{A}_{ij}$. Taking the eigen-decomposition of \mathcal{L} , we get: $\mathcal{L} = \mathcal{P}\mathbf{\Lambda}\mathcal{P}^{-1}$, where \mathcal{P} is the eigen-vector matrix and $\mathbf{\Lambda} = \text{diag}([\lambda_1, \dots, \lambda_N])$ is the diagonal eigen-value matrix with eigen values $\lambda_i \forall i \in \{1, \dots, N\}$ ordered in an ascending order. Then, the GFT of \mathbf{X} is defined as $\tilde{\mathbf{X}} = \mathcal{F}_g(\mathbf{X}) = \mathcal{P}^T\mathbf{X}$, where $\mathcal{P} \in \mathbb{R}^{hwc \times N}$ (for this section, we use tilde for frequency domain signal). Similarly, the inverse GFT (IGFT) is written as, $\mathbf{X} = \mathcal{F}_g^{-1}(\tilde{\mathbf{X}}) = \mathcal{P}\tilde{\mathbf{X}}$. $\mathcal{F}_g(\cdot)$ and $\mathcal{F}_g^{-1}(\cdot)$ denotes GFT and IGFT operation. Hence in GFT, the time domain is graph space while the frequency domain is the eigen values $[\lambda_1, \dots, \lambda_N]$ with each λ_i being related to a particular frequency. To estimate the high-frequency, we consider only higher-order eigen values as $\lambda_1 < \lambda_2 \leq \dots \leq \lambda_N$. It results in a sparse graph representation with significant frequency elements by blacking out low-weighted edges as they result in lower eigen values. Hence, we define a vector $\tilde{\mathbf{h}} = [\mathbf{0} \ \mathbf{1}]^T$ to act as a filter in frequency domain, where $\mathbf{0} = \{0\}^{k \times hwc}$ is all-zero matrix, $\mathbf{1} = \{1\}^{(N-k) \times hwc}$ is all-one matrix and k is related to cut-off eigen value λ_k . The final filtered graph matrix is obtained as Eq. 1.

$$\bar{\mathbf{X}} = \mathcal{F}_g^{-1}(\tilde{\mathbf{h}} \odot \mathcal{F}_g(\mathbf{X})) = \bar{\mathcal{P}}\bar{\mathcal{P}}^T\mathbf{X}, \quad (1)$$

where, $\bar{\mathcal{P}} = \mathcal{P}_{:,k:N}$ are first k eigen vectors. Hence, the node feature aggregation occurs by taking a sparse representative version of \mathcal{A} . It also reduces the computational complexity of our attention module. As we are blacking out k insignificant patches during key estimation, the effective complexity of our overall attention module is $\mathcal{O}((N-k)hwc)$ while it is $\mathcal{O}(Nh^2w^2c)$ for regular MSA.

Using $\bar{\mathbf{X}}$, we estimate the attention weights as $\hat{\mathbf{X}}$ as shown in Fig. 3 (b), from which the updated node feature patches are generated as $\hat{\mathbf{V}} = \{\hat{v}_i | \hat{v}_i \in \mathbb{R}^{hwc}\}$ by reshaping each node \hat{v}_i . The output of a FSGA is computed as $\mathbf{F}_{\text{out}} = \mathbf{F}_{\text{in}} + \text{patch_merger}(\{\hat{v}_i\}_{i=1}^N)$. For patch merging, we adapt the method used in W-MSA. We also employ multi-headed attention (M-FSGA) and to stabilize our training process, we dynamically select the value of $k \in \{\lfloor \frac{N}{2} \rfloor, \dots, N-1\}$ for different heads to ensure not to miss out significant features at different

frequencies. The outcomes of M-FSGA ($\{\mathbf{F}_{\text{out}}^j\}_{j=1}^M$) are passed through a Feed Forward Network (FFN) consisting of a concatenate and 1×1 convolution block to aggregate them and project them to a desired shape as shown in Fig. 2 (d).

3.3 Discriminator Spatial Attention (DSA)

The feature maps from the latent space of D can be viewed as spatial attention to the HR guide \mathbf{z} . Since D performs binary classification, apparently, it captures the discriminative features in latent space. [9] introduced the concept of transferring these domain-specific latent features as attention to G . We use this similar notion to help G focus on the salient parts of the HR guide while also helping to avoid the generation of redundant image features in SR DEM.

Therefore, besides classification, D has another major functional branch, D_{SA} , to approximate spatial attention maps. For any input \mathbf{m} , D_{SA} is used to estimate the normalized spatial feature maps, $D_{SA} : \mathbb{R}^{H \times W} \rightarrow [0, 1]^{H \times W}$. Let D consist of t DMRBs and a_i be the activation maps after i^{th} DMRB with c channels, such that $a_i \in \mathbb{R}^{H \times W \times c}$. We select t different attention maps after t DMRBs since at different depths, D focuses on different features. Eventually, we calculate these attention coefficients according to [9], $D_{SA}(\mathbf{m}) = \sum_{i=1}^t \sum_{j=1}^c |a_{ij}(\mathbf{m})|$.

To estimate the attention, we use upsampled LR DEM $\tilde{\mathbf{x}}$ as unlike image-to-image translation in [9], we do not have HR samples in the target domain during testing. Hence, we use domain adaptation loss from [30] to estimate sharper latent features. The final attention maps \mathbf{A}_s are derived by passing D_{SA} through a PSA [21] to exclude redundant features while highlighting key areas. It is chosen because of its ability to retain a high internal resolution compared to other SA modules. Next, we discuss the theoretical framework for optimizing our model.

3.4 Theoretical Framework

We train our model with SiRAN, a novel framework regularizing traditional GAN with Sinkhorn distance. Compared to WGAN and its variants which are designed to solve the Kantorovich formulation of OT problems to minimize the Wasserstein distance, SiRAN showcases favourable sample complexity of $\mathcal{O}(n^{-1/2})$ [11] (for WGAN, it is $\mathcal{O}(n^{-2/d})$ [37]), given a sample size n with a dimension d . This is because Sinkhorn is estimated based on entropic regularization. Another key issue with WGANs is the vanishing gradient problem near the optimal point resulting in a suboptimal solution. SiRAN avoids such scenarios, as it provides better convergence and tighter iteration complexity as we derive later.

Let $\mu_\theta \in \mathbb{P}_{G_\theta}$ and $\nu \in \mathbb{P}_y$ be the measure of generated and true distribution with support included in a compact bounded set $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^d$, respectively. Therefore, the EOT [3] between the said measures can be defined using Kantorovich formulation as shown in Eq. 2 where we assume $\hat{\mathbf{y}} = G(\tilde{\mathbf{x}}, \mathbf{z} \odot A_s(\tilde{\mathbf{x}}))$.

$$\mathcal{W}_{C,\varepsilon}(\mu_\theta, \nu) = \inf_{\pi \in \Pi(\mu_\theta, \nu)} \mathbb{E}_\pi[C(\hat{\mathbf{y}}, \mathbf{y})] + \varepsilon I_\pi(\hat{\mathbf{y}}, \mathbf{y}), I_\pi(\hat{\mathbf{y}}, \mathbf{y}) = \mathbb{E}_\pi[\log(\frac{\pi(\hat{\mathbf{y}}, \mathbf{y})}{\mu_\theta(\hat{\mathbf{y}})\nu(\mathbf{y})})], \quad (2)$$

where, $\Pi(\mu_\theta, \nu)$ is the set of all joint distribution on $\mathcal{X} \times \mathcal{Y}$ with marginals μ_θ and ν , $C : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is the cost of transferring unit mass between locations $\hat{\mathbf{y}} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$, and the regularization $I_\pi(\cdot)$ is the mutual information between two measures [12] with ε as its weight. When $C(\cdot)$ is distance-metric the solution of Eq. 2 is referred to entropic Wasserstein distance between two probability measures. To fit μ_θ to ν , $\mathcal{W}_{C,\varepsilon}(\mu_\theta, \nu)$ is to be minimized which can be treated as loss function for G [2]. However, it has one major issue of being strictly larger than zero, i.e. $\mathcal{W}_{C,\varepsilon}(\nu, \nu) \neq 0$ which is resolved by adding normalizing terms to Eq. 2 leading to the Sinkhorn loss [13] as defined below.

$$\mathcal{S}_{C,\varepsilon} = \mathcal{W}_{C,\varepsilon}(\mu_\theta, \nu) - \frac{1}{2}\mathcal{W}_{C,\varepsilon}(\mu_\theta, \mu_\theta) - \frac{1}{2}\mathcal{W}_{C,\varepsilon}(\nu, \nu). \tag{3}$$

Based on the value of ε , Eq. 3 shows asymptotic behaviour [13]. When $\varepsilon \rightarrow 0$, it recovers the conventional OT problem, while $\varepsilon \rightarrow \infty$, it converges to maximum mean discrepancy (MMD). Therefore, the Sinkhorn loss interpolates between OT loss and MMD loss as ε varies from 0 to ∞ leveraging the concurrent advantage of non-flat geometric properties of OT loss and, high dimensional rigidity and energy distance properties of MMD loss (when $C = \|\cdot\|_p$ with $1 < p < 2$). Apart from this, the selection of ε also affects the overall gradients of G , which eventually results in preventing vanishing gradient problems near the optimal point. This can be established from the smoothness property of $\mathcal{S}_{C,\varepsilon}(\mu_\theta, \nu)$ with respect to θ . In this context, we propose Theorem 1, where we derive a formulation to estimate the smoothness of Sinkhorn loss.

Theorem 1 (Smoothness of Sinkhorn loss). *Consider $\mathcal{S}_{C,\varepsilon}(\mu_\theta, \nu)$ be the Sinkhorn loss between measures μ_θ and ν on \mathcal{X} and \mathcal{Y} , two bounded subsets of \mathbb{R}^d , with a C^∞ , L_0 -Lipschitz, and L_1 -smooth cost function C . Then, for $(\theta_1, \theta_2) \in \Theta$,*

$$\mathbb{E}\|\nabla_\theta \mathcal{S}_{C,\varepsilon}(\mu_{\theta_1}, \nu) - \nabla_\theta \mathcal{S}_{C,\varepsilon}(\mu_{\theta_2}, \nu)\| = \mathcal{O}\left(L\left(L_1 + \frac{2L_0^2 L}{\varepsilon(1 + Be^{\frac{\kappa}{\varepsilon}})}\right)\|\theta_1 - \theta_2\|\right), \tag{4}$$

where L is the Lipschitz in θ , $\kappa = 2(L_0|\mathcal{X}| + \|C\|_\infty)$, $B = d \cdot \max(\|m\|, \|M\|)$ with m and M being the minimum and maximum in set \mathcal{X} . Let Γ_ε be the smoothness mentioned above, then we get the following asymptotic behavior in ε :

$$1. \text{ as } \varepsilon \rightarrow 0, \Gamma_\varepsilon \rightarrow \mathcal{O}\left(\frac{2L_0^2 L^2}{B\varepsilon e^{\frac{\kappa}{\varepsilon}}}\right), \quad \text{and}, \quad 2. \text{ as } \varepsilon \rightarrow \infty, \Gamma_\varepsilon \rightarrow \mathcal{O}(LL_1).$$

Proof. Refer to Appendix B in supplementary (supp.).

Theorem 1. shows the variation of smoothness of $\mathcal{S}_{C,\varepsilon}(\mu_\theta, \nu)$ with respect to ε . Using this, we can estimate the upper bound of the overall expected gradient of our proposed adversarial set-up. Hence, to formulate this upper bound, we present Proposition 1. Here, we assume $\mathbf{x} = \text{concat}(\tilde{\mathbf{x}}, \mathbf{z} \odot A_s(\tilde{\mathbf{x}}))$.

Proposition 1. *Let $l(\cdot)$, $g(\cdot)$ and $\mathcal{S}_{C,\varepsilon}(\cdot)$ be the objective functions related to supervised losses, adversarial loss and Sinkhorn loss with smoothness Γ_ε , and*

θ^* and ψ^* be the parameters of optimal G and D . Let us suppose $l(\hat{\mathbf{y}}, \mathbf{y})$, where $\hat{\mathbf{y}} = G_\theta(\mathbf{x})$ is β -smooth in $\hat{\mathbf{y}}$ for some input \mathbf{x} . If $\|\theta - \theta^*\| \leq \epsilon$ and $\|\psi - \psi^*\| \leq \delta$, then $\|\nabla_\theta \mathbb{E}_{(x,y) \sim \mathcal{X} \times \mathcal{Y}}[l(\hat{\mathbf{y}}, \mathbf{y}) + \mathcal{S}_{C,\epsilon}(\mu_\theta(\hat{\mathbf{y}}), \nu(\mathbf{y})) - g(\psi; \hat{\mathbf{y}})]\| \leq L^2\epsilon(\beta + \Gamma_\epsilon) + L\delta$.

Proof. Refer to Appendix C in supp.

In GAN setups as mentioned in [29], $\epsilon \rightarrow 0$ leads to a vanishing gradient near the optimal region due to reductions in δ . However, regularizing with Sinkhorn introduces an upper bound dependent on Γ_ϵ , which varies exponentially with ϵ (see Proposition 1). Choosing an appropriate ϵ mitigates the vanishing gradient and enhances performance. Additionally, Sinkhorn regularization improves iteration complexity [29], resulting in faster convergence as established in Proposition 2.

Proposition 2. *Suppose the supervised loss $l(\theta)$ is lower bounded by $l^* > \infty$ and it is twice differentiable. For some arbitrarily small $\zeta > 0$, $\eta > 0$ and $\epsilon_1 > 0$, let $\|\nabla g(\psi; \hat{\mathbf{y}})\| \geq \zeta$, $\|\nabla \mathcal{S}_{C,\epsilon}(\mu_\theta, \nu)\| \geq \eta$ and $\|\nabla l(\hat{\mathbf{y}}, \mathbf{y})\| \geq \epsilon_1$, with $\delta \leq \frac{\sqrt{2\epsilon_1\zeta}}{L}$, and $\Gamma_\epsilon < \frac{\sqrt{2\epsilon_1\eta}}{L^2\epsilon}$, then the iteration complexity in Sinkhorn regularization is upper bounded by $\mathcal{O}\left(\frac{(l(\theta_0) - l^*)\beta_1}{\epsilon_1^2 + 2\epsilon_1(\zeta + \eta) - L^2(\delta^2 + L^2\Gamma_\epsilon^2\epsilon^2)}\right)$, assuming $\|\nabla^2 l(\theta)\| \leq \beta_1$.*

Proof. Refer to Appendix D in supp.

Corollary 1. *Using first order Taylor series, the upper bound in Proposition 2 becomes $\mathcal{O}\left(\frac{l(\theta_0) - l^*}{\epsilon_1^2 + \epsilon_1(\zeta + \eta)}\right)$.*

Proof. Refer to Appendix D.1 in supp.

When $\Gamma_\epsilon < \frac{\sqrt{2\epsilon_1\eta}}{L^2\epsilon}$, the denominator of the derived upper bound in Proposition 2 is greater than the same in Theorem 3 of [29]. This is true for almost all valid ϵ as we experimentally verify in Appendix E in supp. Therefore, SiRAN has tighter iteration complexity compared to the regular GAN set-ups. Corollary 1 also verifies this using a simpler setup, as it increases the convergence rate from $\mathcal{O}((\epsilon_1^2 + \epsilon_1\zeta)^{-1})$ [29] to $\mathcal{O}((\epsilon_1^2 + \epsilon_1(\zeta + \eta))^{-1})$. Due to these advantages, we regularize the generator loss with Sinkhorn distance as defined below,

$$\mathcal{L}_{OT} = \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_{\tilde{x}}, \mathbf{z} \sim \mathbb{P}_{Z}, \mathbf{y} \sim \mathbb{P}_y} \mathcal{S}_{C,\epsilon}(\mu(\hat{\mathbf{y}}), \nu(\mathbf{y})), \quad (5)$$

where μ and ν is the measure of generated and true distributions. \mathcal{L}_{OT} is estimated according to [13] which utilizes ϵ and the Sinkhorn iterations T as the major parameters. As Sinkhorn loss also minimizes the Wasserstein distance, it serves the purpose of WGAN to resolve the issues of the original GAN more effectively. Hence, we use original GAN objective function (\mathcal{L}_{ADV}) while regularized with Sinkhorn loss. We also regularize the objective function of G with pixel loss (\mathcal{L}_P) and SSIM loss (\mathcal{L}_{SSIM}) to generate samples close to GT in terms of minimizing the pixel-wise differences while preserving the perceptual quality and structural information. Therefore, the overall generator loss is defined as

$$\lambda_P \mathcal{L}_P + \lambda_{SSIM} \mathcal{L}_{SSIM} + \lambda_{ADV} \mathcal{L}_{ADV} + \lambda_{OT} \mathcal{L}_{OT}, \quad (6)$$

where λ_P , λ_{SSIM} , λ_{ADV} and λ_{OT} represent the weight assigned to pixel loss, SSIM loss, adversarial loss, and Sinkhorn loss respectively.

Similarly, the objective function of D is designed based on the original GAN. In addition, we include domain adaptation loss [30] (\mathcal{L}_{DA}) to enforce the D to mimic the latent features of the HR DEM and sharpen spatial attention maps provided an upsampled LR DEM data. The final objective function of D becomes

$$\min_D -\mathbb{E}_{y \sim \mathbb{P}_y}[\log(D(\mathbf{y}))] - \mathbb{E}_{\hat{y} \sim \mathbb{P}_{G_\theta}}[\log(1 - D(\hat{\mathbf{y}}))] + \lambda_{DA}\mathcal{L}_{DA}, \quad (7)$$

where λ_{DA} is the assigned weight for \mathcal{L}_{DA} in the discriminator objective. The details of \mathcal{L}_{ADV} , \mathcal{L}_P , \mathcal{L}_{SSIM} , and \mathcal{L}_{DA} are discussed in Appendix A in supp.

4 Experiments

Here, we discuss the necessary experiments and datasets for DEM SR.

4.1 Datasets

DEM SR is a relatively unexplored area that suffers from a lack of realistic datasets. Hence, we generate our own DEM SR dataset for this study. From the real-world application point of view, we use real coarse resolution SRTM DEM with a ground sampling distance (GSD) of 30 m as input instead of conventional bicubic downsampled while taking Indian HR DEM (GSD=10 m) generated from Cartosat-1 stereoscopic satellite as the GT. For the guide, we take the HR MX data (GSD=1.6 m) from the Cartosat-2S satellite. The DEMs are upsampled to the resolution of MX images using bicubic interpolation to generate a paired dataset. This helps in increasing the training samples and also assists the model in learning dense HR features from the guide. The dataset consists of 72,000 patches of size (128, 128) including various signatures such as vegetation, mountains, and, water regions. We use 40,000 samples for training, 20,000 for cross-validation, and 12,000 for testing, where 10,000 patches belong to the Indian region and the rest outside India. As GT is only available for Indian regions, our model is trained on limited landscape areas. To check its generalization ability, we test our model on data from the Fallbrook region, US, where Cartosat DEM data is unavailable. For these cases, we validate our result based on available 10 m DEM data of 3DEP [35]. We further test our trained model by taking other available 30 m DEM like ASTER [1] and AW3D30 [33]. In these cases, we have taken 5000 samples each from different parts of the India for testing.

4.2 Implementation Details

All the experiments are conducted under identical environments. We use 3×3 convolution kernel and leaky ReLU activation except in the last layer where 1×1 kernel is used without any activation. Each DMRB has 64 convolution

Table 1. Quantitative comparison SOTA methods. Testing is performed with the trained model on generated training dataset using LR SRTM DEM. First and second methods are highlighted in **red** and **green**.

Dataset	Inside India (SRTM)				Outside India (SRTM)				ASTER				AW3D30				Params (M)	Avg. Runtime
	RMSE	MAE	PSNR	SSIM	RMSE	MAE	PSNR	SSIM	RMSE	MAE	PSNR	SSIM	RMSE	MAE	PSNR	SSIM		
Method	(m)	(m)	(dB)	(%)	(m)	(m)	(dB)	(%)	(m)	(m)	(dB)	(%)	(m)	(m)	(dB)	(%)		
Bicubic	14.25	13.42	30.37	71.27	14.79	13.86	30.07	70.49	25.24	23.19	27.37	68.47	18.24	17.16	30.05	70.57	-	0.25s
ENetV2 [6]	20.35	18.72	29.74	70.63	30.53	28.36	25.58	69.63	35.62	32.71	25.63	69.49	26.34	25.13	28.44	71.49	24	1.43s
DKN [16]	12.98	11.18	32.16	73.59	21.16	19.78	28.02	68.45	28.43	27.24	26.24	73.04	23.26	21.74	29.15	76.29	1.15	0.625s
FDKN [16]	13.05	11.34	32.09	74.43	21.93	20.41	27.86	66.83	30.49	28.68	26.05	72.27	23.94	22.42	29.06	76.54	0.69s	0.54s
DADA [23]	37.49	32.17	27.94	73.32	40.89	37.74	25.59	69.86	39.66	37.84	25.79	68.46	34.49	33.43	27.59	71.26	-	22.80s
FEN [22]	12.15	11.06	32.23	76.49	20.96	19.06	28.42	73.49	26.27	24.91	27.09	71.67	25.28	23.82	28.63	77.63	1.31	1.23s
GISR [14]	13.18	12.34	32.49	78.49	20.15	18.84	28.61	76.34	27.59	26.13	26.48	70.14	24.49	23.17	28.96	78.56	1.49	1.27s
D-SRGAN [7]	21.33	19.56	29.88	85.68	20.45	18.34	29.55	80.48	24.29	22.62	28.79	75.19	21.28	20.31	29.87	81.49	40	1.57s
FDSR [15]	12.89	10.87	33.07	86.49	21.57	20.22	29.09	79.81	22.08	20.87	29.13	78.84	19.26	18.43	30.26	80.28	0.61	0.51s
SiRAN (ours)	9.28	8.51	34.55	89.36	14.74	12.25	31.56	83.90	20.28	18.59	30.16	82.42	16.52	14.87	31.14	84.79	7.41	0.92s

operations. For FSGT in HTB, we select patch size as 7×7 and the number of heads in the attention block as $M = 16$. We use an ADAM optimizer with a fixed learning rate of 0.0001. During adversarial training, we update the critic once every single update in the generator. We set $\lambda_{DA} = 0.1$, $\lambda_P = 100$, $\lambda_{str} = 1$, $\lambda_{ADV} = 1$ and $\lambda_{OT} = 0.01$. For estimating \mathcal{L}_{OT} , we set $T = 10$ and $\varepsilon = 0.1$. The entire framework is developed using PyTorch. All the experiments are performed on 2 Nvidia V100 GPUs. We compare our method with traditional bicubic as well as other learning-based state-of-the-art (SOTA) DEM SR methods [7, 15, 24]. For a fair comparison, we also include recent baseline models for image-guided depth SR [8, 16, 17, 25]. All the learning-based methods are trained on our dataset from scratch according to the respective authors' guidelines. Among them, we train [7, 15, 24] without any guide as there is no provision in including an image guide in these methods, whereas, [8, 16, 17, 25] are trained on our dataset in the presence of the guide due to their similar set-up for guided SR.

5 Result Analysis

Here, we analyze both qualitatively and quantitatively, the quality of generated HR DEM by our proposed method.

5.1 Quantitative Analysis

To quantitatively analyze the performance, we use RMSE, MAE, PSNR, and SSIM as the evaluation metrics. Our proposed method outperforms other SOTA methods for 4 different datasets, as shown in Table 1. For both inside and outside India images, SiRAN achieves more than 24% improvement in RMSE and MAE, 8% in SSIM, and 1.2 dB in PSNR with respect to the second best. Despite having different source domains for reference DEM for outside India cases, SiRAN generates SR DEM closer to GT as depicted in Table 1 suggesting better generalization capability of other baseline methods. This also can be depicted by analyzing on test cases for other LR DEM data like ASTER and AW3D30 as

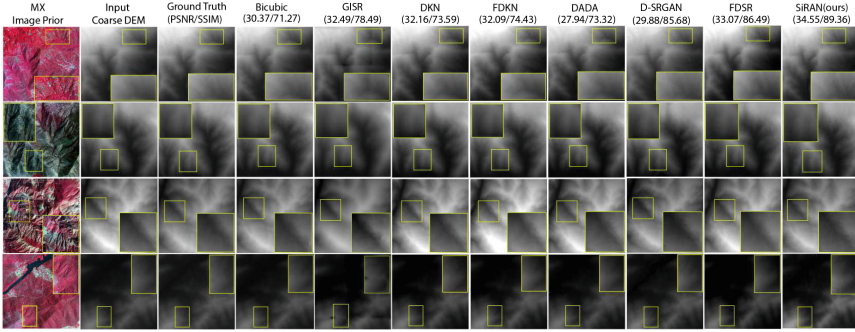


Fig. 4. Test results (inside India) for DEM super-resolution (better viewed at 200%) and comparisons with other baseline methods.

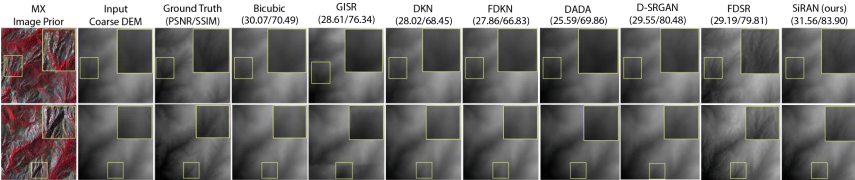


Fig. 5. Test results (outside India) for DEM super-resolution (better viewed at 200%) and comparisons with other baseline methods.

shown in Table 1. In these cases, SiRAN gains more than 10–18% improvement in RMSE, 11–27% in MAE, 4% in SSIM, and ~ 1 dB in PSNR. Among others, FDSR [16] performs close to our model for Indian patches as well as for other LR DEM samples. However, for outside patches, it performs poorly. Although D-SRGAN captures structural details, it has poor RMSE and MAE. Figure 7 shows the line profiles of SiRAN and other baselines with respect to GT. Comparatively SiRAN has the lowest bias and follows the true elevation values most closely. This supports the error analysis in Table 1. Table 1 shows a comparison of number of parameters and average runtime for 512×512 patches. Despite having larger parameters, our model takes comparable inference time due to its effective complexity as discussed in Sect. 3.2.

5.2 Qualitative Analysis

Figure 4 demonstrates the qualitative comparison of DEM SR for patches of India. Clearly, SiRAN highlights key features and comparatively retain more the perceptual quality with respect to GT. D-SRGAN also captures major structural information in its outcomes, however, it tends to produce artifacts and noise in the generated DEM which is depicted in Table 1 and Fig. 7. In Fig. 5, we have compared the outcomes for outside India cases. Here also compared to other

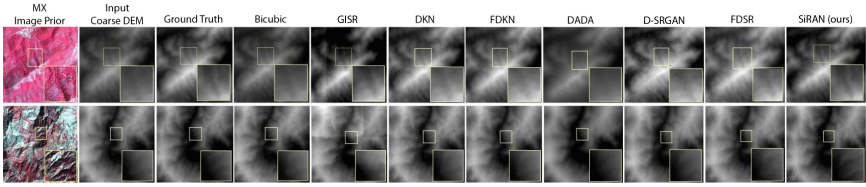


Fig. 6. Test results on ASTER (top row) and AW3D30 (bottom row) dataset for DEM super-resolution (better viewed at 200%) and comparisons with other baseline methods.

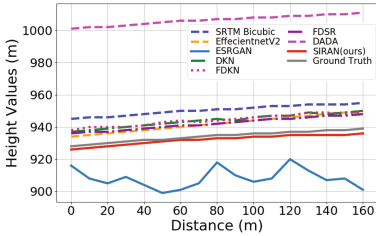


Fig. 7. Line profile analysis of SiRAN and other baselines.

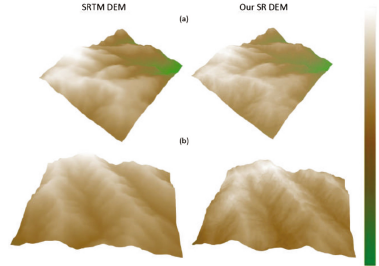


Fig. 8. Illustration of 3-D visualization of Super-resolved and SRTM DEM

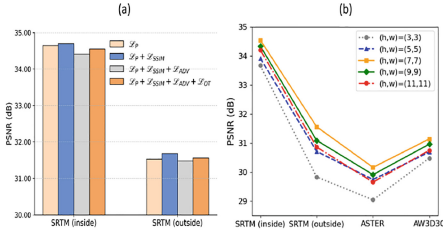
SOTA methods, SiRAN is able to generate higher resolution DEM in close proximity to the GT despite having a different source domain. Although FDSR [16] performed well for Indian patches, due to a lack of generalization capability it introduces image details prominently in the generated DEM for test patches outside India. The generalization ability of these models can also be visualized from 6 where we demonstrate visual test cases for LR DEMs of ASTER and AW3D30 datasets. Clearly, SiRAN captures the high-frequency details most effectively in the predicted SR DEM followed by FDSR and D-SRGAN. Among the other models, while DKN and FDKN try to incorporate HR guide details in the SR output, DADA blurs out important features resulting in outputs similar to bicubic interpolation. GISR model also showcases similar results, however, it generates boundary artifacts in their predictions. In Fig. 8, we show 3-D visualization of generated DEMs for a region, where GT is unavailable. We compare it with available SRTM DEM, and clearly, our topographic view of generated DEM captures sharper features in mountainous regions and in the tributaries of the water basin area as shown in Fig. 8.

5.3 Ablation Study

We discuss different configuration choices we have taken in our designed model for optimal performance in DEM SR in our dataset.

Table 2. Quantitative analysis on effect of different modules for DEM SR.

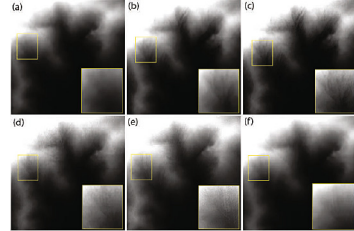
Image Guide	DSA	PSA	FSGT	RMSE (m)	MAE (m)	SSIM (%)	PSNR
✓	✓	✓	✓	20.04	17.63	75.27	30.27
✓	✓	✓	✓	20.32	18.41	82.92	30.57
✓	✓	✓	✓	16.06	13.62	85.68	32.08
✓	✓	✓	✓	13.43	11.31	87.04	32.71
✓	✓	✓	✓	9.28	8.51	89.36	34.55

**Fig. 9.** Quantitative ablation study for: (a) introducing different loss functions, and (b) different values of patch size (h, w) on various test dataset.**Table 3.** Ablation of No. of heads.

Number of heads	Params (M)	PSNR	SSIM
4	5.29	34.34	89.04
8	7.41	34.55	89.36
12	16.37	34.59	89.64
16	21.36	34.61	90.09
24	30.22	34.72	90.13

Table 4. Model size comparison.

Model	Params (M)	FLOPs (G)	PSNR (dB)
SwinIR [19]	11.90	215.3	34.41
CAT [4]	16.60	360.7	34.16
HAN [26]	16.07	269.1	33.94
ARF [42]	11.87	278.3	34.25
FSGT (ours)	21.36	189.4	34.55

**Fig. 10.** Loss ablation: (a) LR DEM, (b) GT; predicted SR DEM of (c) all losses, (d) $\mathcal{L}_P + \mathcal{L}_{SSIM} + \mathcal{L}_{ADV}$, (e) $\mathcal{L}_P + \mathcal{L}_{SSIM}$, and (f) \mathcal{L}_P .

Choice of Different Architectural Designs: Table 2 shows the performance comparisons in terms of different proposed modules. Introducing FSGT brings about the best performance of our framework for DEM SR. However, the utilization of the image guide improves the SSIM only due to its tendency to prominently capture HR MX features in SR DEM. Introducing discriminator spatial attention (DSA) and PSA controls the imitation of guide features phenomenon which results in performance gain in terms of all the metrics. This can also be visualized from Fig. 11 and 12 where we show how D focuses on different features at different depths and also how PSA highlights certain features to give more weight. FSGT further enhances this performance. In this regard, we have also tested with constant $k = \lfloor \frac{3N}{4} \rfloor$, and we have seen more than 0.75 dB performance drop in terms of PSNR and 1.34% in SSIM.

Choice of Different Loss Functions: Figure 9 (a) shows the performance of our model with different combinations of loss functions. Introducing \mathcal{L}_{ADV} decreases the PSNR by 0.2-0.3 dB, while adding \mathcal{L}_{OT} improves it by 0.1 dB. Although, it is still less by 0.15 dB compared with $\mathcal{L}_P + \mathcal{L}_{SSIM}$ loss combination, the major reason for using \mathcal{L}_{ADV} and \mathcal{L}_{OT} is to improve the overall perceptual quality of SR DEM as shown in Fig. 10. However, as depicted in **Proposition 2**, it provides faster convergence as shown in Fig. 13. More experiments are carried out in Appendix E to justify these claims.

Different Patch Sizes in FSGT: Figure 9 (b) shows the performance of our model for different patch sizes in FSGT layers. In our case of DEM SR, patch size 7×7 performs the best in terms of PSNR for all of the four datasets.

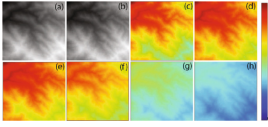


Fig. 11. (a) Source, (b) Target, (c)-(h) Discriminator spatial attention after each DMRB.

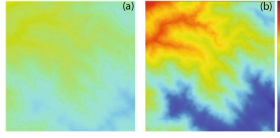


Fig. 12. Weights of (a) mean DSA (\mathbf{D}_{SA}), and (b) after passing it through PSA block.

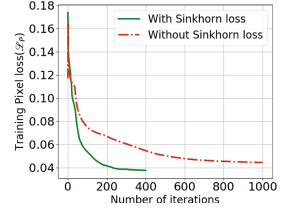


Fig. 13. Effect of Sinkhorn loss in training convergence.

Different Numbers of Heads in M-FSGA: Table 3 shows the performance of our model for different numbers of heads (M) in proposed M-FSGA. As shown in the table, $M = 8$ is the optimal choice in our case. $M = 24$ improves the performance by 0.13 dB PSNR but at the cost of 40% more parameters.

Model Size Comparison: Table 4 shows the comparison of model size, computational complexity, and performance for DEM SR with respect to popular benchmark transformer models. Clearly, FSGT provides excellent performance while having the least number of FLOPs with competitive model size.

6 Conclusion

In this paper, we present an effective approach for DEM SR using realistic coarse data samples in the presence of an HR MX guide. We propose a novel hybrid transformer model based on FSGT and DMRB. In particular, FSGT is constructed to capture the HR features based on dynamically selected frequencies in a graph attention layer. This also reduces the overall complexity from $\mathcal{O}(Nh^2w^2c)$ to $\mathcal{O}((N-k)hwc)$. To control the in-painting of HR guide features in SR DEM, we also introduce DSA, and through an intense ablation study, we validate the performance of each of these proposed modules. We also present a new adversarial set-up, SiRAN based on Sinkhorn loss optimization. We provided theoretical and empirical evidence to show its efficiency in improving the convergence and speed of training our model. We perform quantitative and qualitative analysis by generating and comparing DEMs related to different signatures for four different datasets which includes not only the generated inside and outside India test cases corresponding to LR SRTM DEM but also includes LR test samples corresponding to other DEM datasets, ASTER and AW3D30. In all these cases, our model performs preferably by generating close-to-ground truth SR predictions compared to other baseline methods, which showcases its efficiency in capturing high-frequency details as well as better generalization capability.

References

1. Abrams, M., Crippen, R., Fujisada, H.: Aster global digital elevation model (GDEM) and aster global water body dataset (ASTWBD). *Remote Sens.* **12**(7) (2020)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan (2017)
3. Aude, G., Cuturi, M., Peyré, G., Bach, F.: Stochastic optimization for large-scale optimal transport (2016)
4. Chen, Z., Zhang, Y., Gu, J., Kong, L., Yuan, X., et al.: Cross aggregation transformer for image restoration. In: *NeurIPS*, vol. 35, pp. 25478–25490 (2022)
5. Chen, Z., Wang, X., Xu, Z., Wenguang, H.: Convolutional neural network based dem super resolution. *ISPRS Int. Arch. Photogrammetry, Remote Sens. Spat. Inf. Sci.* **XLI-B3**, 247–250 (2016)
6. Cuturi, M.: Sinkhorn distances: lightspeed computation of optimal transportation distances. In: *Advances in Neural Information Processing Systems*, vol. 26, June 2013
7. Demiray, B.Z., Sit, M., Demir, I.: Dem super-resolution with efficientnetv2 (2021)
8. Demiray, B.Z., Sit, M.A., Demir, I.: D-SRGAN: DEM super-resolution with generative adversarial networks. *CoRR abs/2004.04788* (2020)
9. Emami, H., Aliabadi, M.M., Dong, M., Chinnam, R.B.: SPA-GAN: spatial attention GAN for image-to-image translation. *CoRR abs/1908.06616* (2019)
10. Farr, T.G., Kobrick, M.: Shuttle radar topography mission produces a wealth of data. *Eos. Trans. AGU* **81**, 583–583 (2000)
11. Genevay, A., Chizat, L., Bach, F., Cuturi, M., Peyré, G.: Sample complexity of sinkhorn divergences (2019)
12. Genevay, A., Cuturi, M., Peyré, G., Bach, F.R.: Stochastic optimization for large-scale optimal transport. *ArXiv abs/1605.08527* (2016)
13. Genevay, A., Peyre, G., Cuturi, M.: Learning generative models with sinkhorn divergences. In: *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, vol. 84, pp. 1608–1617. PMLR (2018)
14. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. *CoRR abs/1704.00028* (2017)
15. Han, X., Ma, X., Li, H., Chen, Z.: A global-information-constrained deep learning network for digital elevation model super-resolution. *Remote Sens.* **15**(2) (2023)
16. He, L., et al.: Towards fast and accurate real-world depth super-resolution: benchmark dataset and baseline. In: *2021 IEEE/CVF CVPR*, pp. 9225–9234 (2021)
17. Kim, B., Ponce, J., Ham, B.: Deformable kernel networks for joint image filtering. *Int. J. Comput. Vis.* **129** (2021)
18. Kim, D.E., Gourbesville, P., Liang, S.Y.: Overcoming data scarcity in flood hazard assessment using remote sensing and artificial neural network. *Smart Water* (2019)
19. Liang, J., Cao, J., Sun, G., Zhang, K., Gool, L.V., Timofte, R.: Swinir: image restoration using swin transformer. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1833–1844 (2021)
20. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. *CoRR abs/1707.02921* (2017)
21. Liu, H., Liu, F., Fan, X., Huang, D.: Polarized self-attention: towards high-quality pixel-wise regression. *CoRR abs/2107.00782* (2021)
22. Liu, X.: Airborne lidar for dem generation: some critical issues. *progress in physical geography. Prog. Phys. Geography - PROG PHYS GEOG* **32** (2008)
23. Liu, Z., Li, L., Wu, Y., Zhang, C.: Facial expression restoration based on improved graph convolutional networks. In: *Conference on Multimedia Modeling* (2019)

24. Ma, X., Li, H., Chen, Z.: Feature-enhanced deep learning network for digital elevation model super-resolution. *IEEE J. Sel. Top. Appl. Earth Obser. Remote Sens.* **PP**, 1–17 (2023)
25. Metzger, N., Daudt, R.C., Schindler, K.: Guided depth super-resolution by deep anisotropic diffusion. In: 2023 IEEE CVPR (2023)
26. Niu, B., et al.: Single image super-resolution via a holistic attention network. In: European Conference on Computer Vision (ECCV) (2020)
27. Priestnall, G., Jaafar, J., Duncan, A.: Extracting urban features from lidar digital surface models. *Comput. Environ. Urban Syst.* **24**, 65–78 (2000)
28. Protter, M., Elad, M., Takeda, H., Milanfar, P.: Generalizing the nonlocal-means to super-resolution reconstruction. *IEEE Trans. Image Process.* **18**(1), 36–51 (2009). <https://doi.org/10.1109/TIP.2008.2008067>
29. Rout, L.: Understanding the role of adversarial regularization in supervised learning. *CoRR* **abs/2010.00522** (2020)
30. Rout, L., Misra, I., Moorthi, S.M., Dhar, D.: S2a: wasserstein gan with spatio-spectral laplacian attention for multi-spectral band synthesis (2020)
31. Sandryhaila, A., Moura, J.M.F.: Discrete signal processing on graphs. *IEEE Trans. Signal Process.* **61**(7), 1644–1656 (2013)
32. Simonovsky, M., Komodakis, N.: Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: 2017 IEEE CVPR, pp. 29–38 (2017)
33. Tadono, T., et al.: Generation of the 30 m-mesh global digital surface model by alos prism. In: ISPRS, pp. 157–162, June 2016. <https://doi.org/10.5194/isprs-archives-XLI-B4-157-2016>
34. Takeda, H., Farsiu, S., Milanfar, P.: Kernel regression for image processing and reconstruction. *IEEE Trans. Image Process.* **16**(2), 349–366 (2007)
35. (USGS), U.G.S.: 1/3rd arc-second dems- usgs national map 3dep downloadable data collection (2019). <https://www.usgs.gov/the-national-map-data-delivery>
36. Wang, Y., Perazzi, F., McWilliams, B., Sorkine-Hornung, A., Sorkine-Hornung, O., Schroers, C.: A fully progressive approach to single-image super-resolution. *CoRR* **abs/1804.02900** (2018). <http://arxiv.org/abs/1804.02900>
37. Weed, J., Bach, F.: Sharp asymptotic and finite-sample rates of convergence of empirical measures in wasserstein distance (2017)
38. Xu, Z., Chen, Z., Yi, W., Gui, Q., Wenguang, H., Ding, M.: Deep gradient prior network for dem super-resolution: transfer learning from image to dem. *ISPRS J. Photogrammetry Remote Sens.* **150**, 80–90 (2019)
39. Yu, W., Zhang, Z., Qin, Z.: Low-pass graph convolutional network for recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 8954–8961 (2022). <https://doi.org/10.1609/aaai.v36i8.20878>
40. Yu, W., Qin, Z.: Graph convolutional network for recommendation with low-pass collaborative filters. *CoRR* **abs/2006.15516** (2020)
41. Zhang, D., Ouyang, J., Liu, G., Wang, X., Kong, X., Jin, Z.: Ff-former: swin fourier transformer for nighttime flare removal. In: 2023 IEEE/CVF CVPRW, pp. 2824–2832 (2023). <https://doi.org/10.1109/CVPRW59228.2023.00283>
42. Zhang, J., Zhang, Y., Gu, J., Zhang, Y., Kong, L., Yuan, X.: Accurate image restoration with attention retractable transformer. In: ICLR (2023)
43. Zhou, Z., Li, G., Wang, G.: A hybrid of transformer and cnn for efficient single image super-resolution via multi-level distillation. *Displays* **76**, 102352 (2023)
44. Zhu, X., Guo, K., Fang, H., Ding, R., Wu, Z., Schaefer, G.: Gradient-based graph attention for scene text image super-resolution. *Proc. AAAI Conf. Artif. Intell.* **37**(3), 3861–3869 (2023)



Long-Wave Infrared Non-Line-of-Sight Imaging with Visible Conversion

Shaohui Jin¹, Wenhao Zhang¹, Hao Liu^{1,2}, Huimin Wang¹, Shuang Cui¹,
and Mingliang Xu^{1,2}(✉)

¹ Zhengzhou University, Zhengzhou 450000, China

² College of Advanced interdisciplinary Studies, National University of Defense
Technology, Changsha 410073, China
haoliu1989@hotmail.com, iexumingliang@zzu.edu.cn

Abstract. The challenge of non-line-of-sight (NLOS) imaging lies in the multiple reflections of light paths, causing a significant drop in signal-to-noise ratio. Visible light is easily affected by illumination conditions, making passive NLOS reconstruction algorithms based on visible light difficult to achieve clear results. However, long-wave infrared (LWIR) light provides stronger specular reflections compared to visible light, improving the signal-to-noise ratio when imaging obscured targets. While LWIR can enhance the quality of NLOS reconstructions, it typically lacks the chromatic details present in visible light. In this study, we make an unprecedented attempt to combine LWIR for high-quality reconstruction with methods to preserve color information, which is crucial for passive NLOS imaging. We introduce NLOS-I2V, an innovative end-to-end training framework. NLOS-I2V reconstructs two-dimensional images of thermal radiation captured on a relay wall and converts these blurred infrared domain images into the clear visible light domain using a generative adversarial network (GAN). This method allows for the synthesis of high-quality, long-range reconstructions while preserving color information. Extensive experiments on a custom-built LWIR NLOS dataset demonstrate exemplary performance in both quantitative metrics and subjective visual representation. The code is available at <https://github.com/codeMakerZWH/NLOS-I2V>.

Keywords: Non-line-of-sight · Long-wave infrared · Infrared to visible

1 Introduction

NLOS imaging is an innovative technique that extends beyond the limitations of traditional perception, allowing for the detection of hidden targets through scat-

This work is supported by the National Natural Science Foundation of China under Grant No. 62272421, Hunan Natural Science Fund for Distinguish Young Scholars (No. 2024JJ4044) and in part supported by the No. U22B2051, No. 62172371 and No. 62272422.

teredoptical signals on the relay surface. This technology is of significant importance in various application scenarios, such as disaster rescue, assisted driving, and industrial patrol.

Early studies on NLOS imaging mainly rely on external encodable light sources, high-resolution time detectors, and the analysis of photon responses at different times and locations [1, 4, 11, 17]. However, this results in a considerably large and expensive system design. In contrast, some NLOS imaging works are carried out by estimating the motion trajectory [28] of hidden objects from the shadows produced by the reflection of ambient light or its own light on the objects, or by conducting 2D imaging, a method referred to as passive NLOS imaging. The passive NLOS imaging method struggles to reconstruct high-quality two-dimensional images due to the high complexity of the light transmission matrix. To enhance the available imaging data, prior studies attempt to stabilize the linear problem by utilizing the prior knowledge of the scene [21], coherence [3] and polarization [22] to enhance the conditions of the light transmission matrix.

The emergence of deep learning has catalyzed the advancement of sophisticated passive NLOS imaging techniques. These data-driven methodologies consider passive NLOS imaging tasks as a more challenging form of image translation, which refers to the process of converting an image from one domain to another while preserving essential features and details. A significant portion of this work has been conducted using conventional cameras within the visible spectrum [2, 7, 8, 23, 25, 30]. Visible light, with its inherent color information, is readily interpretable by the human eye, leading to a focus on the accurate reconstruction of occluded visible-light scenes. However, the propagation of visible light in NLOS environments is marred by extensive scattering and absorption, resulting in signal attenuation and color information loss. Some passive NLOS imaging circumvent this by employing luminous screens to project images of obscured objects within a distance of one meter. This approach enhances signal quality, albeit at the cost of practical applicability in real-world scenarios.

In the LWIR spectrum, common materials ranging from coarse metallic surfaces to colored acrylic exhibit more pronounced mirror-like reflections compared to visible light [16]. LWIR has begun to play an active role in the field of NLOS imaging [6, 12, 18, 20]. Subsequent research combining polarization with LWIR has shown promising results [14, 15]. Although NLOS imaging with LWIR is generally easier to reconstruct, it is limited by the absence of color information, as it only captures thermal radiation.

Specifically, we make the following contributions:

- (1) We introduce a novel passive NLOS imaging method that leverages LWIR cameras to capture thermal radiation on relay surfaces. This approach establishes a mapping between the target and its color, compensating for the lack of color while maintaining the high signal-to-noise ratio advantage of long-wave infrared at long distances.
- (2) We develop NLOS-I2V, an end-to-end training framework that simultaneously addresses passive NLOS two-dimensional reconstruction and infrared

image colorization. Based on a GAN architecture, NLOS-I2V effectively transforms blurry infrared images into clear visible light images.

- (3) To further improve the quality of reconstruction, a novel Efficient Multi-Scale Attention (EMA) [19] module is implemented, which is coupled with enhancer module, to refine the embedded GAN.

2 Related Work

2.1 Data-Driven Passive NLOS Imaging

Recent advancements in NLOS imaging have leveraged sophisticated techniques and deep learning to detect and track objects concealed around corners. Tancik et al. [23] have integrated geometric processing with deep learning to identify, locate, and track hidden objects in 2D spaces. Maeda et al. [16] have employed LWIR for human pose estimation of NLOS concealed targets. Aittala et al. [2] have utilized Convolutional Neural Networks (CNNs) to reconstruct videos of hidden scenes. Wang et al. [27] have demonstrated the superiority of CNN-trained NLOS recognition models compared to traditional methods. Zhou et al. [30] have developed deep neural networks based on Phong reflection theory. Geng et al. [7] have proposed a novel imaging framework that incorporates manifold embedding and optimal transport theory. Wang et al. [25] have introduced a passive, event-based method for NLOS imaging. He et al. [8] have proposed a deep learning framework, R-UNet, for simultaneous imaging and tracking using standard RGB cameras. Liu et al. [14] have developed PI-NLOS, which utilizes polarized infrared for NLOS imaging, providing enhanced image quality by reducing noise and improving clarity. Additionally, Liu et al. [14, 15] have developed PI-NLOS and DFAR-Net, utilizing polarized infrared and a dual-input, three-branch attention fusion reconstruction network to enhance image quality, reduce noise, and improve clarity in NLOS imaging.

Despite its potential, data-driven passive NLOS imaging continues to confront a multitude of challenges, such as limited imaging distance and subpar image quality. These issues primarily stem from the severe attenuation of useful information caused by the diffusive properties of common surfaces and the multiple reflections of visible light. While LWIR can simplify the problem by reducing multiple reflections to primarily single-bounce reflections, accurate image reconstruction remains challenging due to the scattering and absorption properties of the materials.

2.2 Infrared Image Coloring

Infrared image coloring converts infrared images into a format visible to the human eye. Color is one of the essential features of the human visual system, and introducing color information can make the reconstructed scenes more realistic and recognizable, enhancing the ability to understand the scenes. However, the issue of NLOS has not been considered by anyone. This includes image-to-image translation and video-to-video translation. The Pix2pix framework, as

introduced by Isola et al. [10], delves into the realm of image-to-image translation, necessitating the use of paired datasets. To alleviate the constraints imposed by the need for paired data, Zhu et al.’s CycleGAN [31] employs cycle consistency to preserve content fidelity throughout the training process. Nonetheless, its bidirectional mapping strategy incurs additional computational overhead due to the requisite supplementary generators and discriminators. To address this, certain frameworks have been proposed that implicitly maintain structural coherence by leveraging high-level semantic information [5]. Extending the concept of image-to-image translation, Li et al. [13] have broadened the scope to include video-to-video transformations. The technique of infrared image coloring has been instrumental in compensating for the shortcomings of long-wave infrared NLOS reconstruction, thereby enriching the semantic information, which holds significant implications for the field.

3 Method

3.1 Passive NLOS Imaging In The LWIR Spectrum

In the current experimental scenario, as depicted in Fig. 1, let’s assume a concealed target in the scene. This target, under the thermodynamic temperature T_s (measured in K or R), can emit radiation at a rate which is connected to its temperature T_s and surface emission rate A_s . This rate E can be expressed as follows:

$$E(T_s) = A_s \sigma T_s^4 \tag{1}$$

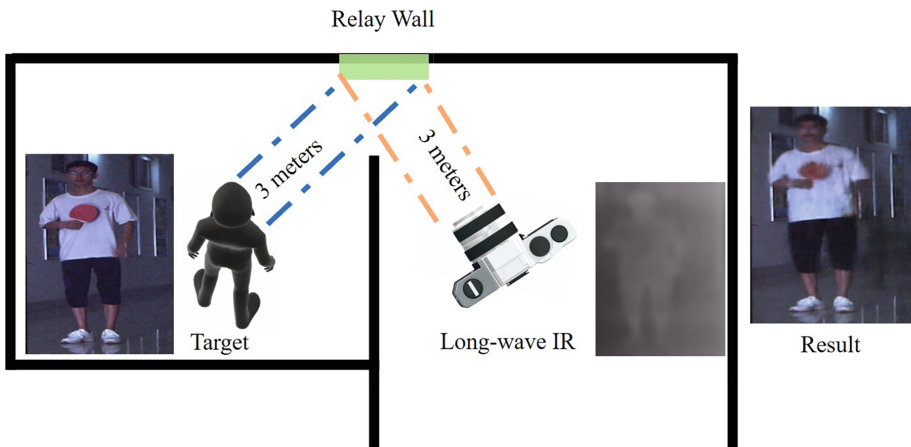


Fig. 1. Passive NLOS Imaging. The image on the far left portrays an individual wielding a ping-pong paddle, representing the original scene. The image on the right exhibits the reconstructed image obtained using the proposed method. The middle image illustrates an NLOS image captured through LWIR at a distance of 6 m.

where, σ symbolizes the Stefan-Boltzmann constant. Kirchoff’s law of radiation illustrates that at certain temperature and wavelength, a surface’s emission rate equals its absorption rate. In most practical scenarios, the surface temperature and incident radiation source temperature are of the same order of magnitude.

Assuming the reflection of the relay surface (a wall) conforms to a specific bidirectional reflectance distribution function (BRDF), the intensity of the thermal radiation obtained by the camera can be represented as follows:

$$I = \int_{\Omega} E(\omega_i) f_r(\omega_i, \omega_o) (\omega_i \cdot \mathbf{n}) d\omega_i \quad (2)$$

where, $E(\omega_i)$ is the incident radiation intensity, $f_r(\omega_i, \omega_o)$ is the Bidirectional Reflectance Distribution Function (BRDF), ω_i and ω_o are the incident and exit directions respectively, \mathbf{n} is the normal vector of the relay surface, and Ω is the hemisphere space.

Based on Phong theory, NLOS imaging works have demonstrated that the specular reflection component is crucial for non-line-of-sight reconstruction [30]. Therefore, we consider the specular reflection BRDF of micro-surfaces to simplify and understand the optical process:

$$f_r(\omega_i, \omega_o) = \frac{\rho}{\pi} \delta(\omega_i - \omega_r) \quad (3)$$

where ρ is the reflectance of the relay surface, ω_r denotes the direction of specular reflection, and δ represents the Dirac delta function, ensuring that there is a non-zero reflection component only when the incident direction ω_i exactly matches the reflection direction ω_r .

Substituting the BRDF into the equation for I Hence, we get:

$$I = \frac{\rho}{\pi} \int_{\Omega} E(\omega_i) \delta(\omega_i - \omega_r) (\omega_i \cdot \mathbf{n}) d\omega_i \quad (4)$$

The main objective of the NLOS-I2V is to reconstruct a two-dimensional image by inverting the transformation from the hidden scene to the projection image. This inverse mapping process is a key component of the reconstruction task that NLOS-I2V aims to achieve.

3.2 Network Structure

In this study, we introduce an innovative end-to-end network, NLOS-I2V, that originates from GANs [26]. A notable feature of our method is that it is specifically designed to handle both NLOS imaging and infrared image colorization tasks simultaneously, rather than performing these tasks in a cascading manner. This is achieved through self-constructed NLOS paired datasets, where each NLOS infrared image corresponds to a ground truth visible-light image.

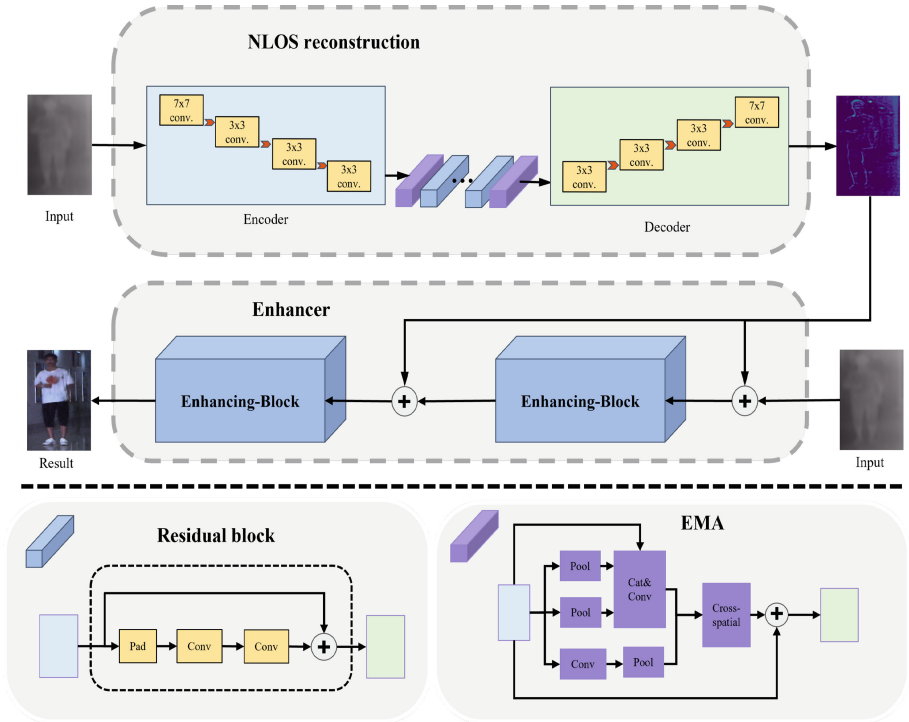


Fig. 2. The image depicts the generator, comprised of two components, namely NLOS reconstruction and the enhancer. The bottom part shows the network components, including Residual Blocks [9] and EMA [19].

Generator Structure. In the context of the NLOS-I2V framework, as depicted in Fig. 2, the generator assumes a pivotal role in achieving high-fidelity reconstructions and colorization of LWIR. Meticulously designed with two integral components, namely the NLOS reconstruction and the enhancer, the generator is tailored to address the unique challenges presented by passive NLOS imaging.

The NLOS reconstruction component is the core of the generator, featuring an encoder that delves into the input image to extract high-dimensional features. These features are intricately processed by the EMA (Efficient Multi-Scale Attention) [19] module, which employs an attention mechanism to discern and accentuate contextually significant information across various scales. This is particularly advantageous for NLOS imaging, where discerning obscured details is paramount. The residual blocks within this segment further aid in preserving and refining the feature information, ensuring that the quality of the image is not compromised through the layers of the network.

The incorporation of group normalization and convolutions within the EMA module enables the network to dynamically focus on salient features, a critical aspect for reconstructing images within the context of NLOS scenarios, which

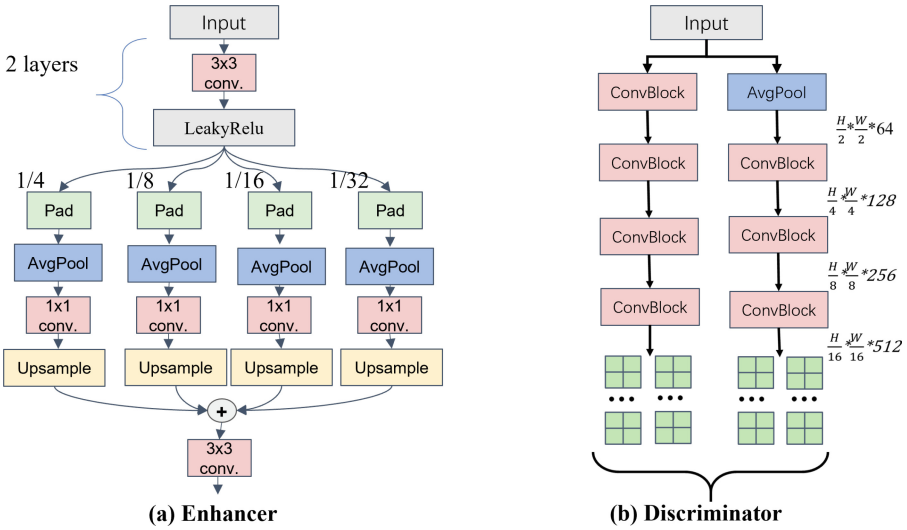


Fig. 3. Architecture of the Enhancer and Discriminator. (a) The network architecture of the Enhancer constructs a four-scale pyramid for the detailed processing of features at different scales. (b) The Discriminator discriminates the generator at two scales through average pooling layers.

are frequently hindered by the inherent limitations of optical phenomena. The softmax normalization of attention weights ensures a concentrated and efficient integration of features, thereby augmenting the clarity and detail of the reconstructed image.

Mirroring the encoder, the decoder employs Deconvolutional Blocks to meticulously upsample the feature maps, restoring them to their original image size. This symmetrical design facilitates the reconstruction of a high-resolution image from the condensed feature representation, integrating the nuanced details refined by the EMA module and residual blocks.

The enhancer module complements NLOS reconstruction by refining the visual clarity of the reconstructed image, as depicted in Fig. 3(a). Employing convolutional layers, it systematically refines feature maps and captures spatial information across different scales. These multi-scale features are combined to form a comprehensive representation, further refined by a final convolutional layer to yield enhanced output.

Within the enhancer block, two 3×3 front-end convolutional layers are utilized. Their outputs are downsampled by factors of 4x, 8x, 16x, and 32x to construct a four-scale pyramid, providing varying receptive fields for image reconstruction. Subsequently, a 1×1 convolution with an adaptive weighted channel attention mechanism reduces dimensionality. Upscaling the feature maps and concatenating with the original size, a 3×3 convolution is applied. In NLOS-I2V, the enhancer consists of two enhancement blocks. The first block receives input from the original image’s concavity and generator feature maps, which are also provided to the second block.

Discriminator Structure. The discriminator Fig. 3(b), has been enhanced in this study to enable it to operate efficiently across various image scales. This is achieved by resizing the real and reconstructed images by a factor of 2, which results in a pyramid of scaled images. Consequently, the discriminator can distinguish between the real and reconstructed images at two different scales. Though the discriminator’s structure remains unchanged, the coarsest-scale discriminator has the broadest receptive field. It offers a holistic view of the image, thus guiding the generator towards achieving globally consistent image production. Conversely, the fine-grained discriminator encourages the generator to produce more elaborate details, such as colors and finer information.

3.3 Loss Function

Our loss function comprises adversarial loss L_{adv} , fidelity loss L_{L1} , perceptual loss L_{vgg} and color loss L_{color} [24].

$$\mathcal{L}_{GAN} = L_{adv}(G, D) + \lambda_{L1}L_{L1}(G) + \lambda_{vgg}L_{vgg}(G) + \lambda_{color}L_{color}(G). \quad (5)$$

In the equation above, λ_{L1} , λ_{vgg} , and λ_{color} are hyperparameters used to balance the contributions of the corresponding loss components in the total loss function.

L_{adv} facilitates the training of the generator and discriminator within an adversarial game context. For the generator, the goal is to minimize the discrepancy between the generated image and the real image within the discriminator’s purview.

L_{L1} ensures the uniformity of the generated image with the actual image at the pixel level.

L_{vgg} leverages a pre-trained VGGNet, renowned for image classification, to distill the activation layers, which are construed as perceptual features. The pixel-wise Euclidean distance serves as the metric for quantifying discrepancies. To preserve the perceptual and semantic integrity, the perceptual loss function is employed to gauge high-level differences.

L_{color} calculates the discrete cosine similarity between the reconstructed image and the original within the three channels of RGB at the pixel level. Here, k represents each pixel value.

$$L_{color} = \frac{1}{HWC} \sum_{i \in \xi} \sum_{k=1}^K \angle(I_i, I'_i), \xi = \{R, G, B\} \quad (6)$$

where $\angle(\cdot)$ denotes the pixel-wise computation of the discrete cosine similarity between the output image and the ground truth visible image across the R, G, B channels. The variable K signifies the total number of pixels present in the image, while i represents the elements of the R, G, B channels. This compensates for the color information deficit left by the infrared presence, thereby enabling the reconstructed image to obtain more detailed information.

4 Experiment

4.1 Experimental Setup

This experiment was carried out under a real-world scenario replicated from Fig. 4 schematic diagram, which ensures the complexity of the real-world experimental conditions. The relay wall is regular plywood with a surface roughness of approximately $30\ \mu\text{m}$, which is larger than the wavelength of LWIR. Therefore, LWIR scattering on the relay wall includes both specular and diffuse reflections. The LWIR camera used has a resolution of 640×512 pixels, each pixel measuring $17\ \mu\text{m}$, a NETD less than $25\ \text{mK}$, and a response band of $8\text{--}14\ \mu\text{m}$. The distance between the hidden target and the camera is approximately $6\ \text{m}$, with both the hidden target and the camera being $3\ \text{m}$ away from the relay surface. Due to the angle β being around 85 milliradians, we ignore the minor differences caused by slight perspective deviations.

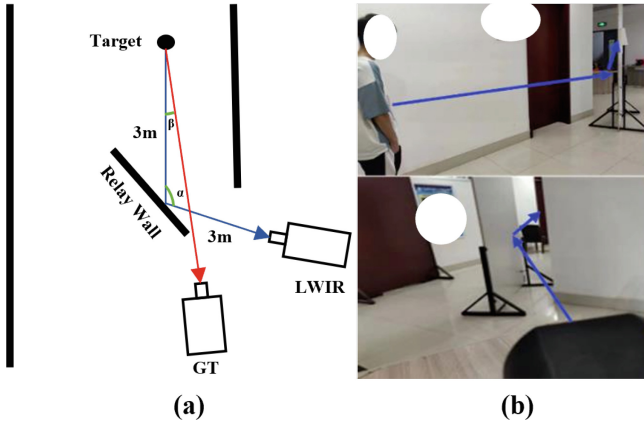


Fig. 4. (a) The overall data collection setup is illustrated. (b) It includes the actual relay surface and the data collection scenarios.

The dataset is collected in various scenarios, including different lighting conditions and temperature environments. It encompasses a wide range of human poses, such as waving limbs and holding objects. Each group contains ground truth images and grayscale images of the target in invisible scenarios, with 3,500 groups for indoor lighting, 8,000 groups for indoor dimming, and 6,000 groups for natural light.

Variations in ambient temperature, influenced by different lighting conditions, lead to differences in thermal radiation from the objects being imaged and their surroundings, indirectly influencing the imaging outcome. The greater the temperature difference, the higher the image contrast, enabling our model to better distinguish and reconstruct target objects in different scenes. Thus, the different lighting effects shown in Fig. 5 (Input) are actually a result of thermal radiation differences caused by changes in ambient temperature.

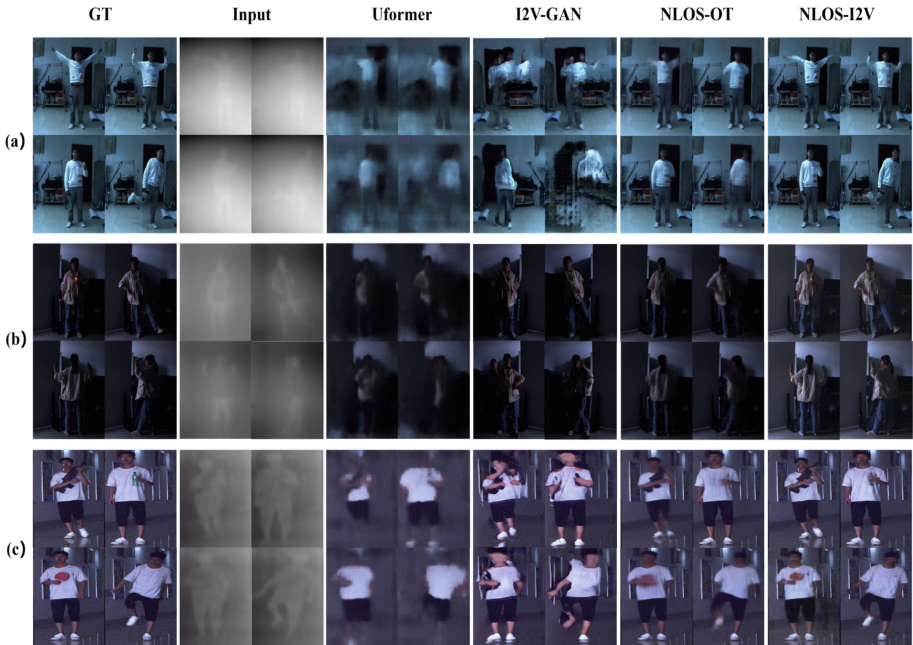


Fig. 5. Results under different kinds of hidden images.(a), (b), and (c) which illustrate Indoor lighting, Indoor dimming, and Natural light conditions respectively.

For color transformation, our goal is to establish a mapping between objects and their colors from infrared images that lack color information. Therefore, in addition to clothing, the dataset includes various objects such as red ping pong paddles, green water bottles, and black toy guns.

We utilized PyTorch 2.0.0+cu117 framework and NVIDIA RTX A4000 GPU for model training. We implemented ADAM optimizer with momentum parameters β_1 and β_2 , preset at 0.5 and 0.999, respectively. The learning rate and batch size were respectively started off with 0.0002 and 16. A total of 200 epochs of training were performed to ensure effective model convergence during training.

4.2 Experimental Results

In our study, we utilized the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) as the evaluative metrics. Our proposed method, NLOS-I2V, was compared against a selection of high-performing methods. These include Uformer [29], a universal image restoration network; NLOS-OT [7], which integrates variational autoencoder (VAE) and optimal transmission theory and is currently the state-of-the-art in passive NLOS; and the I2V-GAN [13] model, specifically engineered for the transformation of imagery from the infrared spectrum to the visible range.

Table 1. Quantitative Comparison with SSIM and PSNR(dB).

Method	Params(M)	Indoor lighting		Indoor dimming		Natural light	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Uformer [29]	50.88	17.6727	0.5781	24.3446	0.7231	18.1554	0.5004
I2V-GAN [13]	49.67	12.6293	0.3134	18.7249	0.445	15.2168	0.3374
NLOS-OT [7]	49.89	24.6828	0.8537	27.1549	0.8649	23.4984	0.7459
NLOS-I2V	45.63	25.4158	0.8676	28.6245	0.8741	25.485	0.7849

As depicted in Fig. 5, the datasets presented in sections (a), (b), and (c) are utilized to draw comparisons under varying conditions of ambient light. Among the current methodologies, Uformer and I2V-GAN are impeded by the intrinsic constraints of NLOS imaging tasks, culminating in suboptimal image fidelity. In contrast, NLOS-OT has been specifically engineered to tackle the more challenging aspects of image translation tasks within NLOS contexts, thereby outperforming the aforementioned methods. The NLOS-I2V, with its generative adversarial architecture featuring multi-scale enhancement, achieves superior feature representation. This is evidenced by the more comprehensive limb reconstruction in sections (a) and (b), as well as the detailed depiction of objects held by the model in section (c), such as the black toy gun, green bottle, and red ping-pong paddle. Furthermore, despite the differing lighting conditions across sections (a), (b), and (c), the outcomes are not significantly impacted. This is attributed to the LWIR imaging’s ability to capture thermal radiation. The thermal radiation from the human body contrasts with that of the relay surfaces, endowing LWIR with robust illumination robustness. This attribute demonstrably affirms the superiority of LWIR in the NLOS imaging. Additionally, environmental temperature changes caused by different lighting conditions significantly impact NLOS imaging in the LWIR band. The quality of the input data is closely related to the difference in thermal radiation between the hidden target and the environment. Consequently, the image quality of inputs (a), (b), and (c) varies distinctly. We can clearly observe that I2V-GAN and NLOS-OT perform worse in (a), characterized by more artifacts and less detailed information. However, our proposed NLOS-I2V method effectively overcomes this, demonstrating its robustness and superiority.

Table 1 provides a quantitative comparison of our NLOS-I2V approach with existing methods across different datasets. The results reveal notable improvements achieved by our NLOS-I2V method compared to other approaches. Specifically, in scenarios involving indoor lighting conditions (low thermal radiation difference), our method achieves a PSNR of 25.4158 dB and an SSIM of 0.8676, outperforming Uformer [29], I2V-GAN [13], and NLOS-OT [7]. Moreover, under indoor dimming and natural light conditions (higher thermal radiation difference), our method continues to demonstrate superior performance, with PSNR values of 28.6245 dB and 25.485 dB, respectively, accompanied by SSIM scores of 0.8741 and 0.7849.

The qualitative and quantitative evaluations fully demonstrate the superiority of the NLOS-I2V method in NLOS imaging tasks. This method not only significantly enhances reconstruction quality but also excels in preserving color information. These findings validate the feasibility of combining NLOS reconstruction with color conversion, providing a novel approach for future NLOS imaging technologies.

4.3 Ablation Study

In the ablation experiment, we conducted a comprehensive analysis of the different components of the NLOS-I2V framework and their contributions to the overall performance. The experiment was designed to isolate the effects of the EMA module and the Enhancer modules on the image translation quality.

Figure 6 illustrates the qualitative advancements across different configurations. The visual progression from the baseline to the fully enhanced model clearly demonstrates the efficacy of each added module. Incorporating EMA has resulted in a more accurate alignment with the ground truth in terms of overall background coloration and luminance. The addition of the Enhancer modules has been instrumental in mitigating artifacts and bolstering detail enhancement. Furthermore, a deeper network architecture has facilitated a richer feature representation, successfully mapping objects such as the red ping-pong paddle, the green bottle, and the black toy gun back to their authentic color spaces.

Table 2. Results of the ablation experiments.

EMA	Enhancer	Enhancer	PSNR (dB)	SSIM
			18.5486	0.6239
✓			19.6545	0.7286
✓	✓		23.0265	0.7528
✓	✓	✓	25.485	0.7849

The results, as shown in Table 2, demonstrate a clear trend of improvement in both PSNR and SSIM with the incremental addition of these components.

The baseline model, without any additional modules, achieved a PSNR of 18.5486 dB and an SSIM of 0.6239. The integration of the EMA module resulted in a PSNR of 19.6545 dB and an SSIM of 0.7286, indicating a significant enhancement in image quality. The addition of a single Enhancer module further improved the PSNR to 23.0265 dB and the SSIM to 0.7528. The incorporation of two Enhancer modules, along with the EMA module, yielded the best results with a PSNR of 25.485 dB and an SSIM of 0.7849, marked in bold to signify the optimal performance achieved by the proposed NLOS-I2V approach.

These results underscore the effectiveness of the EMA module in providing a robust initial enhancement, while the Enhancer modules contribute to further

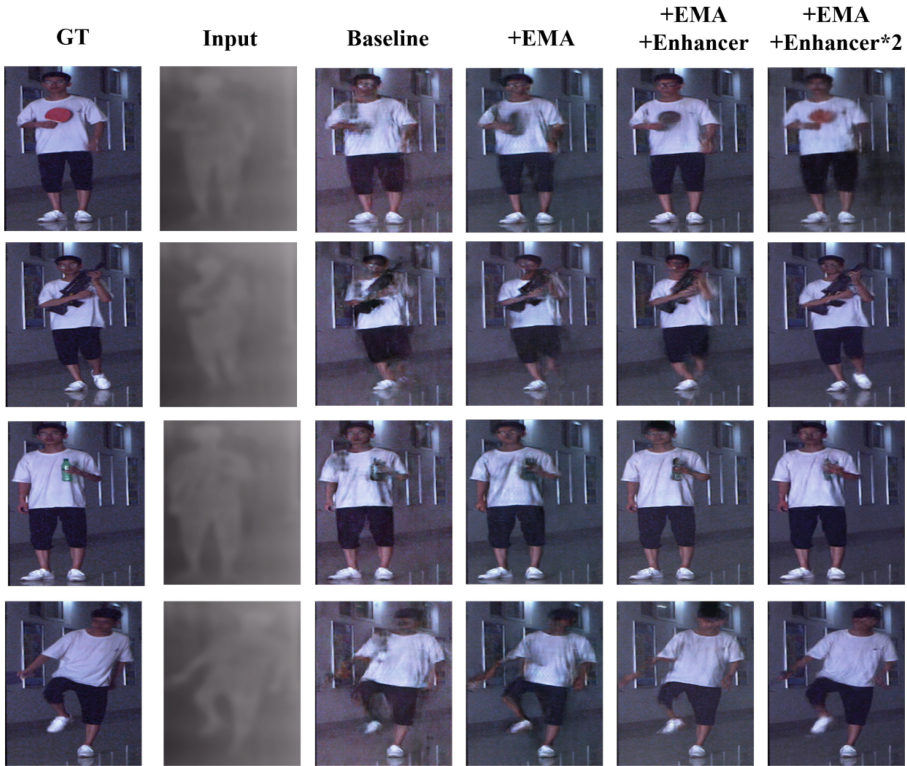


Fig. 6. Ablation experiment results. The baseline configuration does not include the EMA and Enhancer modules. ‘+EMA’ denotes the addition of the EMA module to the baseline. ‘+EMA+Enhancer’ indicates the inclusion of one Enhancer module on top of the EMA module. ‘+EMA+Enhancer*2’ signifies the addition of two Enhancer modules to the EMA module, which constitutes the proposed NLOS-I2V approach in this paper.

refinement of the image details. The progressive increase in both PSNR and SSIM with the addition of each module confirms their synergistic effect, leading to superior image translation outcomes.

5 Conclusion

This paper presents a long-range, high-quality passive non-line-of-sight (NLOS) imaging method and introduces an exemplary model, NLOS-I2V. This model employs a generative adversarial network combined with an efficient multi-scale attention and multi-scale enhancement modules. It leverages the advantages of long-wave infrared (LWIR), including its robustness to lighting conditions and stronger specular reflections, while addressing the lack of chromatic information.

Extensive experimentation on a custom-built LWIR NLOS dataset has demonstrated that NLOS-I2V significantly enhances image clarity and color fidelity compared to existing methods. However, it is important to note that while LWIR facilitates easier NLOS reconstruction, it also presents certain limitations, such as the material properties of relay surfaces and the thermal radiation characteristics of hidden objects.

Future work will focus on incorporating a wider range of optical conditions, such as polarization information [14,15], to capture more effective data and enhance the retrieval of finer details, such as hands, feet, and facial features. In terms of infrared coloring, more advanced coloring methods will be used, such as unpaired and video2video, to enhance the generalization ability of the model. Additionally, efforts will be directed towards expanding the dataset to encompass a wider array of scenarios, thereby enriching the robustness and applicability of NLOS-I2V.

References

1. Ahn, B., Dave, A., Veeraraghavan, A., Gkioulekas, I., Sankaranarayanan, A.C.: Convolutional approximations to the general non-line-of-sight imaging operator. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7889–7899 (2019)
2. Aittala, M., et al.: Computational mirrors: blind inverse light transport by deep matrix factorization. *Adv. Neural Inf. Process. Syst.* **32** (2019)
3. Beckus, A., Tamasan, A., Atia, G.K.: Multi-modal non-line-of-sight passive imaging. *IEEE Trans. Image Process.* **28**(7), 3372–3382 (2019)
4. Cao, R., de Goumoens, F., Blochet, B., Xu, J., Yang, C.: High-resolution non-line-of-sight imaging employing active focusing. *Nat. Photonics* **16**(6), 462–468 (2022)
5. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1511–1520 (2017)
6. Divitt, S., Gardner, D.F., Watnik, A.T.: Passive, thermal, reference-free, non-line-of-sight imaging. In: CLEO: QELS Fundamental Science, pp. FW4Q–5. Optica Publishing Group (2020)
7. Geng, R., et al.: Passive non-line-of-sight imaging using optimal transport. *IEEE Trans. Image Process.* **31**, 110–124 (2021)
8. He, J., Wu, S., Wei, R., Zhang, Y.: Non-line-of-sight imaging and tracking of moving objects based on deep learning. *Opt. Express* **30**(10), 16758–16772 (2022)
9. 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)
10. 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)
11. Jin, S., Xu, Z., Xu, M., Liu, H.: Time-gated imaging through dense fog via physics-driven swin transformer. *Opt. Express* **32**(11), 18812–18830 (2024)
12. Kaga, M., Kushida, T., Takatani, T., Tanaka, K., Funatomi, T., Mukaigawa, Y.: Thermal non-line-of-sight imaging from specular and diffuse reflections. *IPSIJ Trans. Comput. Vis. Appl.* **11**(1), 1–6 (2019). <https://doi.org/10.1186/s41074-019-0060-4>

13. Li, S., Han, B., Yu, Z., Liu, C.H., Chen, K., Wang, S.: I2v-gan: unpaired infrared-to-visible video translation. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 3061–3069 (2021)
14. Liu, H., et al.: Pi-nlos: polarized infrared non-line-of-sight imaging. *Opt. Express* **31**(26), 44113–44126 (2023)
15. Liu, H., et al., Xu, M.: Dfar-net: dual-input three-branch attention fusion reconstruction network for polarized non-line-of-sight imaging. In: Chinese Conference on Pattern Recognition and Computer Vision (PRCV), pp. 41–52. Springer (2023)
16. Maeda, T., Wang, Y., Raskar, R., Kadambi, A.: Thermal non-line-of-sight imaging. In: 2019 IEEE International Conference on Computational Photography (ICCP), pp. 1–11. IEEE (2019)
17. Mu, F., et al.: Physics to the rescue: deep non-line-of-sight reconstruction for high-speed imaging. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022)
18. Nagase, Y., Kushida, T., Tanaka, K., Funatomi, T., Mukaigawa, Y.: Shape from thermal radiation: passive ranging using multi-spectral lwir measurements. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12661–12671 (2022)
19. Ouyang, D., et al.: Efficient multi-scale attention module with cross-spatial learning. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE (2023)
20. Sasaki, T., Hashemi, C., Leger, J.R.: Passive 3d location estimation of non-line-of-sight objects from a scattered thermal infrared light field. *Opt. Express* **29**(26), 43642–43661 (2021)
21. Saunders, C., Murray-Bruce, J., Goyal, V.K.: Computational periscopy with an ordinary digital camera. *Nature* **565**(7740), 472–475 (2019)
22. Tanaka, K., Mukaigawa, Y., Kadambi, A.: Polarized non-line-of-sight imaging. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2136–2145 (2020)
23. Tancik, M., Satat, G., Raskar, R.: Flash photography for data-driven hidden scene recovery. arXiv preprint [arXiv:1810.11710](https://arxiv.org/abs/1810.11710) (2018)
24. Tang, L., Xiang, X., Zhang, H., Gong, M., Ma, J.: Divfusion: darkness-free infrared and visible image fusion. *Inf. Fusion* **91**, 477–493 (2023)
25. Wang, C., et al.: Passive non-line-of-sight imaging for moving targets with an event camera. *Chin. Opt. Lett.* **21**(6), 061103 (2023)
26. 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)
27. Wang, Y., et al.: Accurate but fragile passive non-line-of-sight recognition. *Commun. Phys.* **4**(1), 88 (2021)
28. Wang, Y., Wang, Z., Zhao, B., Wang, D., Chen, M., Li, X.: Propagate and calibrate: real-time passive non-line-of-sight tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 972–981 (2023)
29. Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., Li, H.: Uformer: a general u-shaped transformer for image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17683–17693 (2022)
30. Zhou, C., Wang, C.Y., Liu, Z.: Non-line-of-sight imaging off a phong surface through deep learning. arXiv preprint [arXiv:2005.00007](https://arxiv.org/abs/2005.00007) (2020)
31. 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)



Re-Identification Based on the Spatial-Temporal Fusion Network

Hye-Geun Kim¹, You-Kyoung Na¹, Hae-Won Joe¹, Yong-Hyuk Moon²,
and Yeong-Jun Cho¹(✉)

¹ Chonnam National University, Gwangju, Republic of Korea
{hyegeunkim, youkyoung, haewon716, yj.cho}@jnu.ac.kr

² Sungshin Women's University, Seoul, Republic of Korea
yhmoon@sungshin.ac.kr

Abstract. Re-identification (ReID) in a large-scale camera network is critical in public safety, traffic control, and security. However, due to the ambiguous appearance of objects, the previous appearance-based ReID methods often fail to track objects across multiple cameras. To overcome this challenge, we propose a ReID based on a spatial-temporal fusion network that estimates a reliable camera network topology based on the adaptive Parzen window method and optimally combines the appearance and spatial-temporal similarities through a fusion network. The proposed methods demonstrated the best performance on the public vehicle dataset (VeRi776) with 99.7% rank-1 accuracy and on the person dataset (Market1501) with 99.11% rank-1 accuracy. The experimental results support that using spatial and temporal information for ReID can leverage the accuracy of appearance-based methods and effectively manage appearance ambiguities.

Keywords: Re-identification · adaptive Parzen window · Fusion Network

1 Introduction

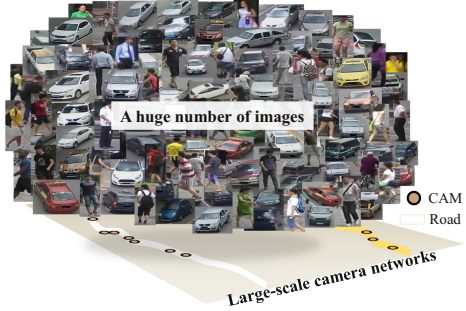
Recently, a large number of surveillance cameras have been installed in public places for safety, traffic monitoring, and security. However, monitoring all cameras requires substantial human effort and resources. To reduce human efforts, re-identification (ReID), which automatically tracks targets across multiple non overlapping cameras, can be applied. In general, most studies have focused on the visual appearances of the targets to perform ReID. For example, many studies [1, 6] have proposed feature learning methods to represent target appearances. Similarly, metric learning methods [23, 43] have also been proposed to measure feature distances effectively between query and gallery images. Recently, developments in deep learning have led to higher performance improvements by training visual features and distance metrics [11, 13]. These appearance-based methods are robust to target pose variations, viewpoint changes, and illumination changes.

Nevertheless, appearance ambiguity caused by objects with similar appearances is still not alleviated. As shown in Fig. 1 (a), vehicles can have the same appearance due to

the same model types, making it difficult to match the correct target. Compared to vehicle, people have relatively distinctive features, but they show similar appearances due to the same clothes. Furthermore, the appearance of numerous objects in multiple cameras causes high computational complexity and low identification performance (Fig. 1 (b)) because the number of objects with a similar appearance to the target increases. Thus, relying only on the target appearance is ineffective for the object ReID problem.



(a) Appearance ambiguity in ReID



(b) High computational complexity

Fig. 1. Challenges in object re-identification

Recently, ReID studies that use additional spatial and temporal information have been proposed to alleviate appearance ambiguity [8, 18, 35, 39]. Researchers have built a camera-network topology explaining spatial and temporal relationships between cameras and used the topology to reduce redundant searching time ranges for queries. While the methods exhibit the potential for improving appearance-based ReID models [11, 13], they still have limitations. Their camera network topology modeling approaches are too simple, and the integration of the appearance model with spatial-temporal information lacks optimization.

In this work, we propose ReID based on the proposed spatial-temporal fusion network (FusionNet) to overcome the limitations. The proposed ReID framework consists of two main parts: 1) camera network topology estimation, 2) fusing appearance similarity and spatial-temporal probabilities. For the camera network topology estimation, we newly propose an adaptive Parzen window that is robust to outliers and sparse responses between camera pairs (Sect. 4.1). It can effectively manage different connection strengths of camera pairs for reliable camera network topology estimation. After estimating the topology, we train a FusionNet that can optimally combine appearance similarity and spatial-temporal probabilities (Sect. 4.2).

To evaluate the proposed methods, we tested the *VeRi776* [25] vehicle ReID dataset and *Market1501* [42] person ReID dataset. In the experiments, we evaluated the effectiveness of the proposed methods, and achieved the best performances in both datasets for the rank-1 accuracy (99.7%, 99.11%) and mean average precision (mAP) (91.71%, 95.5%). The results support that our methods can significantly improve the re-id performance regardless of the data domain (e.g., vehicle, person).

The main contributions of this work are as follows:

- We estimated a reliable camera network topology based on the proposed adaptive Parzen window.
- We trained the FusionNet to combine two different similarities optimally (appearance and spatial-temporal).
- We achieved superior performance on the both vehicle and person ReID tasks.

To the best of our knowledge, this is the first attempt to train a network to fuse the appearance and spatial-temporal similarities. In addition, the proposed framework is very flexible because any appearance-based model can be used as a baseline of the framework.

2 Related Works

2.1 Appearance-Based ReID

Most ReID studies have focused on learning visual representations of images to distinguish their appearance. To this end, metric learning and feature learning methods have been widely studied. For metric learning, learning the Mahalanobis distance [23,43] has been widely studied. Particularly, optimizing the triplet loss for deep metric learning [7, 13, 17] has exhibited superior performance in ReID tasks. Ghosh *et al.* [11] proposed Relation Preserving Triplet Mining (RPTM), a feature matching guided triplet mining scheme that ensures that triplets preserve natural subgroupings in object IDs.

For the feature learning method, Ahmed *et al.* [1] initially used deep convolutional neural network (CNN) architecture that captures local relationships between two input images based on mid-level features. Chen *et al.* [6] improved the CNN-based ReID and proposed a deep pyramid feature learning CNN, which can learn scale-specific discriminative features. Similarly, studies [7, 34] have tried to extract robust local features. To capture more appearance details, Khamis *et al.* [21] employed the attributes of targets for ReID. Recently, He *et al.* [16] proposed TransReID, which is the first attempt to use a transformer to learn robust features from the image patches and Chen *et al.* [4] used Swin Transformer [27] as a backbone to downstream to the ReID task. For a better visual representation, Li *et al.* [24] proposed CLIP-ReID, which fine-tunes the initialized visual model using the image encoder in CLIP. Some studies [41, 46] have focused on the pre-training methods that can overcome the domain gap in ReID tasks.

Moreover, several methods [9, 14, 30, 36] have employed additional cues, such as human pose and body parts, to handle pose variations and occlusion problems. Similarly, studies [19, 45] have tried to improve visual representation quality based on additional identity-guided human semantic parsing and multi-head attention. However, alleviating appearance ambiguity is still challenging when relying only on appearance for ReID.

2.2 Spatial-Temporal ReID

Many studies have used spatial-temporal information from cameras and target objects to overcome the limitations of appearance-based ReID. Generally, they have employed

an appearance-based ReID model as the baseline and exploited the spatial and temporal information. In spatial-temporal ReID, there are two main problems: 1) estimating spatial-temporal information (the camera network topology) in given camera networks, 2) using the estimated camera network topology for ReID.

To estimate the camera network topology, many studies have attempted to design accurate transition time distributions of targets (e.g., person, vehicle). For example, Huang *et al.* [18] modeled a spatial-temporal model leveraging vehicle pose view embedding. Wang *et al.* [39] proposed the Histogram–Parzen method to estimate spatial-temporal probability distributions. Liu *et al.* [25,26] proposed a progressive vehicle ReID that partially applies simple spatial-temporal information. Similarly, studies [8,28,33,44] have estimated spatial-temporal information to filter out irrelevant gallery images. Moreover, Shen *et al.* [35] proposed a Siamese-CNN+Path-LSTM network to predict the path through visual feature information and spatial-temporal information.

While numerous spatial-temporal ReID methods have been proposed, there are still some limitations. First, methodologies for estimating spatial-temporal models are simple. For example, many methods [8,25,26,28] have built object transition time distributions based on the positive responses between cameras. However, noisy and sparse responses make the estimated distributions unreliable. Second, the usage of the spatial-temporal information is not optimized. For example, studies [18,33,39] have merged both probabilities (i.e., appearance and spatial-temporal) with the same importance to obtain the joint probability. Similarly, many methods [8,28,33,44] have applied spatial-temporal information to reduce the search range or perform re-ranking of the initial ReID results.

3 Motivation and Main Ideas

To address the challenges in ReID, we analyzed the characteristics of objects in camera networks. First, there are many objects showing similar or the same appearances in large-scale camera networks. For example, people can show similar appearances due to the same clothes (e.g., uniforms, belongings). Especially, in the vehicle ReID task, vehicles can look exactly the same according to their same model types. Second, movements of objects between non-overlapping cameras are predictable. E.g., vehicles can only move along roads and highways, rarely deviating from existing roads. Compared to vehicles, people show more complex transition patterns across cameras, but people paths are also generally established along the sidewalks and aisles. To summarize, objects show high appearance ambiguities but predictable movements. Therefore, relying only on appearance differences between objects is not effective for the ReID tasks.

Based on these observations, we additionally exploited spatial and temporal relationships between cameras, called a camera network topology. As shown in Fig. 2, the proposed ReID framework consists of two parts: 1) camera network topology estimation, 2) the fusion of the appearance and spatial-temporal similarities. We first built the topology based on the proposed adaptive Parzen window (Sect. 4.1). Then, we trained a FusionNet that optimally combines visual similarity and the camera network topology information for the final ReID prediction (Section 4.2).

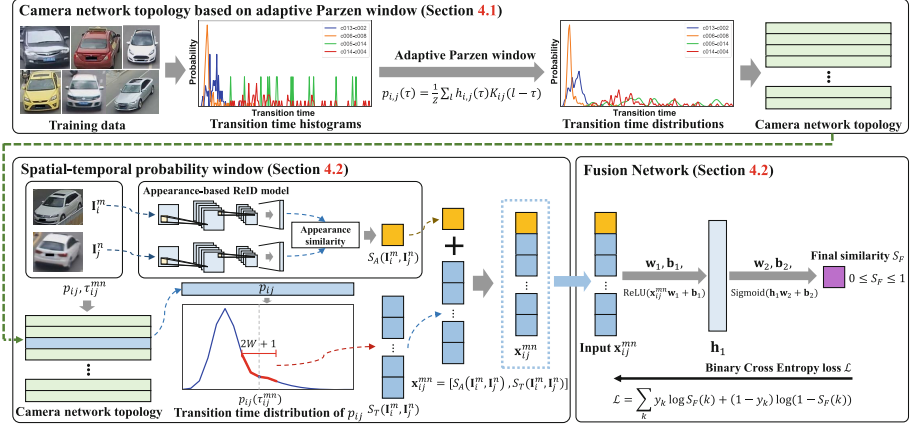


Fig. 2. The overall framework for re-identification with spatial-temporal information.

4 Proposed Methods

4.1 Adaptive Parzen Window for Camera Network Topology Estimation

The camera network topology represents spatial-temporal relationships and connections between cameras that can be represented by a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. The vertices \mathbf{V} denote the cameras, and the edges \mathbf{E} denote the distribution of the object transition time. If N_{cam} cameras exist in the camera networks, the topology is represented as follows:

$$\begin{aligned} \mathbf{V} &\in \{c_i | 1 \leq i \leq N_{cam}\}, \\ \mathbf{E} &\in \{p_{ij} | 1 \leq i \leq N_{cam}, 1 \leq j \leq N_{cam}, i \neq j\}, \end{aligned} \quad (1)$$

where c denotes a camera, and p_{ij} denotes the object transition time distribution between camera pairs c_i and c_j .

To build the transition time distribution p_{ij} , we used positive pairs between all camera pairs in the training dataset. Based on the multiple time differences (Δt) of positive pairs, we generated an initial histogram of the transition time h_{ij} , as depicted in the cyan vertical lines (—) in Fig. 3. Cho *et al.* [8] proposed connectivity checking criteria regarding whether a pair of cameras is connected by fitting a Gaussian model to the histogram h_{ij} . However, this parametric method followed strong assumptions and had difficulty handling outliers and the sparsity of the histogram. Inspired by [39], a Parzen window method can be applied to the initial histograms, and we estimated the probability density function (PDF) of the transition time in a non-parametric manner as follows:

$$p_{ij}(\tau) = \frac{1}{Z} \sum_l h_{ij}(l) K(l - \tau), \quad (2)$$

where τ is an index of the distribution, $Z = \sum_l p_{ij}(\tau)$ represents a normalized factor, and $K(\cdot)$ is a kernel function. For the kernel K , we used the Gaussian function, as

follows:

$$K(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-x^2}{2\sigma^2}\right), \quad (3)$$

where σ is a standard deviation.

While the Parzen window method efficiently estimates continuous PDFs from discrete histograms, employing a single kernel across diverse histograms from various camera pairs is unreasonable. The strength of the spatial-temporal connection between cameras can be determined by how many objects pass through those cameras during a certain period [8]. For example, few positive pairs between two cameras indicate weak connectivity. Nevertheless, the Parzen window method extremely enlarges those small responses with a small σ value, as depicted in the orange line (—) in Fig. 3 (c). In that case, it is better to use a large σ value to avoid overfitting the distribution for noise and outliers.

In contrast, if many positive pairs occur between the cameras, then the connectivity should be strong. However, with a large σ value, the resulting distribution becomes uniform, failing to capture any meaningful spatial and temporal relationships between the cameras, as depicted in the green line (—) in Fig. 3 (a). In that case, it is better to use a relatively small σ value to reflect temporal information between cameras. Thus, selecting the proper σ value is important to the estimated distribution (p_{ij}) quality.

To overcome the limitation of the original Parzen window method [39], we newly propose an adaptive Parzen window by setting various σ_{ij} values for the camera pairs (c_i, c_j). To this end, we designed an adaptive standard deviation according to the different strengths of the camera connectivity as follows:

$$\sigma_{ij} = \max\left(\alpha \exp\left(\frac{-N_{ij}}{\beta}\right), 1\right), \quad (4)$$

where N_{ij} denotes the number of positive object pairs between camera c_i and c_j . In addition, α is a scale factor determining the maximum range of σ_{ij} , and β is a smoothness factor that adjusts the sensitivity of σ_{ij} . The minimum value of σ_{ij} cannot be less than 1 unit of the histogram. Then, the values of σ_{ij} lie on $[1, \alpha]$.

By considering the camera indexes, Eq. 2 and Eq. 3 are reformulated as follows:

$$p_{ij}(\tau) = \frac{1}{Z} \sum_l h_{ij}(l) K_{ij}(l - \tau), \quad (5)$$

$$K_{ij}(x) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp\left(\frac{-x^2}{2\sigma_{ij}^2}\right). \quad (6)$$

Therefore, we can estimate reliable distributions (p_{ij}) from the initial discrete histograms (h_{ij}) by considering the connectivity between cameras. The blue lines (—) in Fig. 3 are our results based on the adaptive Parzen window.

4.2 Fusion Network

The proposed ReID framework can employ any appearance-based ReID method as its baseline to estimate appearance similarities between images. Images from each camera (c_i, c_j) are denoted by $\mathbf{I}_i^m, \mathbf{I}_j^n$, where m and n are indexes of the images. Then, the

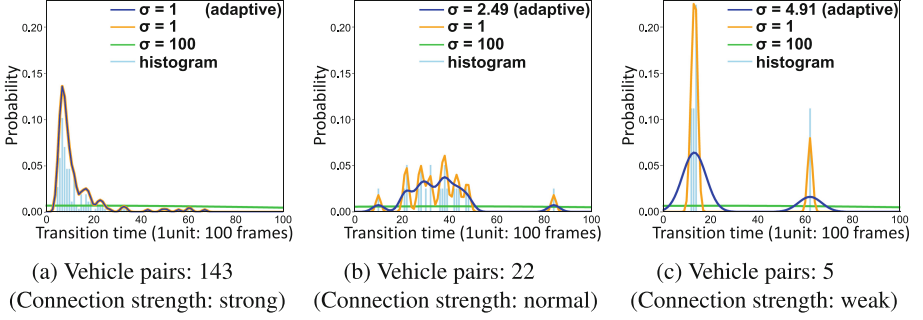


Fig. 3. Examples of estimated transition time distributions between camera pairs. Each bin covers 100 frame ranges. Solid blue lines (—) mark the estimated distribution (p_{ij}) from the histogram (h_{ij}) by the proposed adaptive Parzen window. (Color figure online)

appearance-based ReID methods estimate the visual similarity between two images as $S_A(\mathbf{I}_i^m, \mathbf{I}_j^n)$ that lies on $[0, 1]$. The proposed framework does not depend on the types of appearance-based models.

To perform spatial-temporal ReID, Cho *et al.* [8] used only camera network topology to restrict the search range of the gallery, which is effective in reducing the complexity of ReID. However the spatial-temporal probability does not affect the final similarity. In contrast, previous studies [18, 33, 39] have merged both probabilities (i.e., appearance and spatial-temporal) with the same importance to obtain the joint probability. However, they neglected two points. First, the domain of each probability is not the same. Second, both appearance and spatial-temporal probabilities can be imperfect. Therefore, it is unreasonable to simply merge these probabilities.

In this work, we optimally combined visual similarities $S_A(\mathbf{I}_i^m, \mathbf{I}_j^n)$ and estimated spatial-temporal probability distributions $p_{ij}(\tau)$ through the fusion network named FusionNet. An input vector of the network for two images ($\mathbf{I}_i^m, \mathbf{I}_j^n$) in the camera pair (i, j) can be represented by

$$\mathbf{x}_{ij}^{mn} = [S_A(\mathbf{I}_i^m, \mathbf{I}_j^n), S_T(\mathbf{I}_i^m, \mathbf{I}_j^n)], \quad (7)$$

where S_A is an appearance similarity, and S_T is a spatial-temporal vector. The S_T vector between the images is defined by

$$S_T(\mathbf{I}_i^m, \mathbf{I}_j^n) = \left[p_{ij}(\tau_{ij}^{mn} - W), \dots, p_{ij}(\tau_{ij}^{mn}), \dots, p_{ij}(\tau_{ij}^{mn} + W) \right], \quad (8)$$

where τ_{ij}^{mn} denotes the time difference between two images \mathbf{I}_i^m and \mathbf{I}_j^n . W is the size of a time window. According to W , the range of the S_T vector is determined around distributions of $p_{ij}(\tau_{ij}^{mn})$. For example, when we set $W = 0$, the S_T becomes a scalar value as $S_T(\mathbf{I}_i^m, \mathbf{I}_j^n) = p_{ij}(\tau_{ij}^{mn})$. When we set $W > 0$, the S_T vector has a $2W + 1$ dimensional vector. By adjusting the value of W , we can determine how much spatial-temporal information to provide for the FusionNet. Then, the dimension of the input vector \mathbf{x}_{ij}^{mn} for FusionNet is $2W + 2$.

We designed the FusionNet based on the simple multi-layer perceptron. We empirically found that the FusionNet does not require a sophisticated deep neural network structure to estimate the final similarity. The network has one hidden layer with several nodes and a one-dimensional output layer, as shown in Fig. 2. For the activation function, we used Rectified Linear Unit (ReLU) for nodes in the hidden layer, and used the sigmoid function for the output node. Then, the final output of the FusionNet $S_F(\mathbf{I}_i^m, \mathbf{I}_j^n)$ lies on $[0, 1]$. To train the network, we optimized the binary cross-entropy loss defined by

$$\mathcal{L} = \sum_k y_k \log S_F(k) + (1 - y_k) \log(1 - S_F(k)), \quad (9)$$

where k represents the index of the training image pair, and $y_k \in [0, 1]$ denotes the ground truth of the k th image pair.

5 Experimental Results

5.1 Dataset and Settings

For the experiments, we used the `VeRi776` [25] vehicle re-identification (ReID) dataset and `Market1501` [42] person ReID dataset. The `VeRi776` [25] dataset contains over 49,000 images of 776 different vehicle identities (IDs) captured by 20 non-overlapping synchronized cameras. `Market1501` [42] contains over 32,000 images of 1,501 different people IDs captured by 6 non-overlapping synchronized cameras. In both datasets, each image contains object IDs, timestamps (frame No.), and camera IDs.

To estimate the camera network topology $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, we used training datasets (576 IDs in `VeRi776` and 751 IDs in `Market1501`). Among them, 90% of the training data were used for appearance-based ReID model training and the remaining 10% were used for FusionNet training. Note that the object identities (IDs) were completely separated for each training task. In `VeRi776` with 20 cameras ($N_{cam} = 20$), the estimated camera network topology contains 400 object transition time distributions (p_{ij}). Among them, 380 distributions are between different camera pairs (i.e., p_{ij} , where $c_i \neq c_j$), and each distribution has 300 bins, each covering 100 frame ranges. All distributions were estimated based on the proposed adaptive Parzen window. By performing the same processes, we estimated the camera network topology of `Market1501` as well.

For the appearance-based ReID model in our framework, we trained `FastReID` [13] and `SOLIDER` [4] to extract the appearance similarity. The hyper parameters of `FastReID` in both datasets are as follows: epoch – 60, batch size – 64. For vehicle ReID task, a simple `ResNet-50` [12] was utilized as the backbone network structure of `FastReID`. Meanwhile, for the Person ReID task, `ResNet-101` with `MGN` [40] were utilized as the backbone network structure of `FastReID`.

The proposed FusionNet has a single hidden layer, and we designed the number of nodes in the hidden layer to be around 65% of the size of the input vector by rounding $(2(2W + 2)/3 + 1)$. Training parameters of FusionNet are as follows: epoch – 40, batch size – 128, learning rate – 0.001, optimizer – Adam. We evaluated the top- k

accuracy ($k = 1, 5$) and mean average precision (mAP) for the test images. `VeRi776` and `Market1501` have 200 and 750 different IDs for the performance validation of methods.

5.2 Effects of FusionNet

In this experiment, we tested `VeRi776` [25] dataset according to W values in FusionNet. Additionally, we compared other methods such as *FastReID* [13] using only appearance similarities (S_A) between images. *FastReID* + *Wang*'s estimated the appearance similarity (S_A) based on *FastReID* [13] and simply combined the spatial-temporal similarity by $S_F = S_A(\mathbf{I}_i^m, \mathbf{I}_j^n) \cdot p_{ij}(\tau_{ij}^{mn})$, as in [39].

As shown in Table 1, *FastReID* [13] using only appearance information achieved a rank-1 accuracy of 96.96% and mAP of 81.91%. On the other hand, *FastReID* + *Wang*'s, which used additional spatial-temporal similarity, improved the mAP of the appearance-based *ReID* [13] by 3.56%. However, simple combining of two different similarities by $S_A(\mathbf{I}_i^m, \mathbf{I}_j^n) \cdot p_{ij}(\tau_{ij}^{mn})$ is not yet optimized. The rank-1 and rank-5 performances were degraded after adopting *wang*'s approach in *FastReID* [13].

Based on the proposed FusionNet, we significantly improved the *ReID* performance. For fair comparisons, we employed *FastReID* [13] as the appearance model for our framework and utilized 526 IDs for appearance model and 50 IDs for FusionNet training. When W was set to 10, FusionNet achieved the best performance in rank-1 accuracy of 99.70%, rank-5 accuracy of 99.82%, and mAP of 91.71%. Except at $W = 0$, FusionNet outperformed other methods in all evaluation metrics (rank-1, -5, and mAP). This result supports that the proposed FusionNet can optimally combine different types of information, such as appearance similarity and spatial-temporal information. In addition, even if the training image for the appearance model was less utilized, it achieved superior performance compared to other methods.

Figure 4 (a) illustrates the trained weight vector (\mathbf{w}_1) between input and hidden layer (\mathbf{h}_1). The row numbers (0–14) denote the index of the nodes in \mathbf{h}_1 , and column numbers (0–21) denote the index of the input vector \mathbf{x}_{ij}^{mn} . The first column (0 index) shows the weights for appearance similarity (S_A). As we can see, the magnitudes of the weights are relatively bigger than those of other weights. It means that the FusionNet properly trained the importance of the appearance similarity (S_A). The other columns (from 1 to 21 index) are the weights for spatial-temporal distribution (S_T). Interestingly, the weights from the 10th to 12th columns showed large magnitudes. It implies that the spatial-temporal information around the time difference (τ_{ij}^{mn}) between two images plays a key role in *ReID*. In addition, FusionNet has a lightweight structure but effective.

5.3 Effects of Adaptive Parzen Window

We evaluated various factors for the proposed adaptive Parzen window method, such as the scale factor α and smoothness factor β in Eq. 4. In all experiments, we set $W = 10$ for FusionNet. Figure 4 (b) illustrates the mAP according to the α and β factors. When the scale factor $\alpha = 6$ and the smoothness factor $\beta = 25$, the proposed framework

Table 1. ReID performances of VeRi776 [25] dataset according to W values in FusionNet. The method *FastReID* [13] does not exploit spatial-temporal information S_T . *FastReID+Wang's* estimates the final similarity by $S_F = S_A(\mathbf{I}_i^m, \mathbf{I}_j^n) \cdot p_{ij}(\tau_{ij}^{mn})$, as in [39] and employed *FastReID* [13] for its appearance model.

Methods	No. of IDs	Rank-1	Rank-5	mAP
<i>FastReID</i> [13]	576 for	96.96	98.45	81.91
<i>FastReID</i> [13]+Wang's [39]	Appearance	95.77	97.74	85.47
FusionNet ($W = 0$)		97.08	98.57	80.58
FusionNet ($W = 2$)	526 for	98.09	98.87	84.68
FusionNet ($W = 4$)	Appearance	99.52	99.70	90.77
FusionNet ($W = 6$)	————	99.64	99.82	91.45
FusionNet ($W = 8$)	50 for	99.64	99.82	91.55
FusionNet ($W = 10$)	FusionNet	99.70	99.82	91.71
FusionNet ($W = 12$)		99.64	99.82	91.57

Table 2. ReID performance according to the values of σ for the Parzen window method.

Methods	VeRi776 [25]			Market1501 [42]		
	Rank-1	Rank-5	mAP	Rank-1	Rank-5	mAP
Fixed $\sigma = 1$	99.70	99.82	91.57	99.02	99.58	93.56
Fixed $\sigma = 5$	99.52	99.70	91.43	98.81	99.55	93.34
Fixed $\sigma = 10$	99.28	99.76	91.28	98.25	99.41	92.12
Fixed $\sigma = 100$	98.81	99.46	86.62	97.25	99.08	91.91
Adaptive σ (ours)	99.70	99.82	91.71	99.11	99.58	93.80

achieved the best mAP performance. Note that ReID performances do not fluctuate significantly due to the factor values (min mAP: 91.62 – max mAP: 91.71).

We further compared the ReID performances for the fixed and adaptive σ values. The fixed $\sigma = 1, 5, 10, 100$, and the adaptive σ as in Eq. 2 were tested. As summarized in Table 2, the fixed $\sigma = 100$ performed the worst because a too-large σ leads to smoothed distributions close to a uniform distribution and reduces the influence of spatial-temporal information. On the other hand, $\sigma = 1$, $\sigma = 5$ and $\sigma = 10$ performed relatively well, achieving higher than a 98% rank-1 accuracy. Compared to using fixed σ values, the proposed adaptive Parzen window with the adaptive σ performed the best in all evaluation metrics. It achieved 99.70% rank-1 accuracy, 99.82% rank-5 accuracy, and 91.71% mAP on the VeRi776 [25] dataset and 99.11% rank-1 accuracy, 99.58% rank-5 accuracy, and 93.80% mAP on the Market1501 [42] dataset. As explained in Sect. 4.1 and Fig. 3, a fixed value of σ has difficulty handling various types of initial histograms h_{ij} . This result implies that setting different σ_{ij} values by considering the various connection strengths of camera pairs is effective and improves ReID performance on both vehicle and person ReID.

5.4 Comparison with State-of-the-Art Methods

In this section, we compare the proposed method with state-of-the-art re-identification methods using the *VeRi776* [25] vehicle ReID dataset and *Market1501* [42] person ReID dataset. The methods are primarily categorized into two approaches: 1) only appearance-based, 2) using additional spatial-temporal information (marked by †). The methods marked by a ‘*’ performed re-ranking post-processing for the final ReID results.

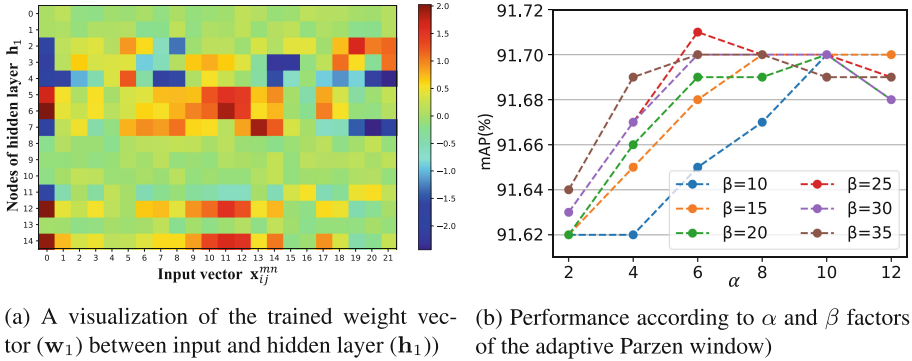


Fig. 4. Illustrations for effects of FusionNet and adaptive Parzen window.

Table 3 summarizes the vehicle and person ReID results. In *VeRi776* [25] dataset, our framework achieved the best performance with 99.7% rank-1 accuracy and 91.71% mAP. Although the spatial-temporal approaches [25, 26, 35] have improved their baseline methods, the performance is relatively worse than other appearance-based methods. That is because their appearance models were quite older methods, such as the SIFT, bag-of-words, and Siamese-CNN for ReID. Furthermore, the methods did not provide a direct estimation of the camera network topology or optimize the use of spatial-temporal information.

As deep learning models have developed, many appearance based methods have improved ReID performance. For example, FastReID [13], which is our baseline appearance model, achieved a 96.96% rank-1 accuracy and 81.91% mAP. Especially, RPTM [11] used the GMS [2] feature matcher and employed the ResNet-101 [12] structure, demonstrating the superior performance with 97.3% rank-1 accuracy and an 88.0% mAP. Our methods used a lightweight structure (ResNet-50) for the appearance model, but it outperformed the sophisticated RPTM [11] method by the rank-1 accuracy of 2.4%, rank-5 accuracy of 1.42%, and mAP score of 3.71%. In addition, the our methods did not perform any re-ranking processes for post-processing.

In *Market1501* [42] dataset, st-ReID [39] using spatial-temporal information achieved reasonable rank-1 accuracy of 98.1%, but the mAP of 87.6% was quite lower than the other state-of-the-art methods. SOLIDER [4] with Swin Transformer [27] achieved the good performance in mAP (93.9%), but the rank-1 performance of 96.9%

was relatively low. In this dataset, we selected two different appearance-based models such as FastReID [13] and SOLIDER [4] as the baseline of our framework. As a result, our frameworks achieved the best rank-1 accuracy (99.11%) with the FastReID [13], and the best rank-5 accuracy (99.6%) and mAP (95.5%) with the SOLIDER [4]. These results imply that the proposed spatial-temporal framework improves both vehicle and person ReID tasks. In addition, it has the potential to achieve higher performance when it employs better appearance-based model as its baseline.

Figure 5 illustrates the qualitative ReID results. We compared the proposed method (Ours) with the baseline method (FastReID [13]), which used only appearance information. The baseline method shows numerous false matches, where the appearance closely resembles that of the query image. In particular, it rarely matched the correct images of the 662-nd vehicle query image due to many similar black cars. In contrast, the proposed method perfectly matched the correct images at rank-1 to rank-10 under the challenging query and gallery pairs thanks to the spatial-temporal information.

In the person ReID task, the baseline model easily failed to match correct pairs in the gallery. For example, ID 342 wearing a stripe yellow shirt, gray shorts and carrying a black backpack is easy to confuse with others wearing very similar outfits. ID 1083 wearing red dress and carrying a shoulder bag is also very confusing with other person wearing similar red dress with a hand bag. These results support that the proposed ReID (Ours) based on the spatial-temporal fusion network can effectively manage the appearance ambiguity problems and overcome the limitations of the previous ReID methods.

Table 3. Performance comparisons on the Veri776 [25] and Market1501 [42] datasets. † and * indicate the spatial-temporal approach and re-ranking, respectively. The best and second best performances are marked in **bold** and underline.

Models	Veri776 [25]			Models	Market1501 [42]		
	Rank-1	Rank-5	mAP		Rank-1	Rank-5	mAP
†Siamese-CNN+Path-LSTM [35]	83.49	90.04	58.27	†st-ReID [39]	98.1	99.3	87.6
†PROVID [26]	81.56	95.11	53.42	GCP [31]	95.2	–	88.9
†KPGST [18]	92.35	93.92	68.73	TransReID [16]	95.2	–	89.5
†FastReID [13] + Wang’s [39]	95.77	97.74	85.47	ISP [45]	95.3	98.6	88.6
PAMTRI [38]	92.86	96.97	71.88	GASM [14]	95.3	–	84.7
CAL [32]	95.40	97.90	74.30	ABDNET [3]	95.6	–	88.28
PVEN [29]	95.60	98.40	79.50	SCSN [5]	95.7	–	88.5
TBE [37]	96.00	<u>98.50</u>	79.50	CLIP-ReID [24]	95.7	–	89.8
VehicleNet* [44]	96.78	–	83.41	SAN [20]	96.1	–	88.0
SAVER* [22]	96.90	97.70	82.00	FastReID [13]	96.35	–	90.77
DMT* [15]	96.90	–	82.00	PASS [46]	96.9	–	93.3
FastReID [13]	96.96	98.45	81.91	SOLIDER [4]	96.9	–	<u>93.9</u>
CLIP-ReID [24]	<u>97.30</u>	–	84.50	Unsupervised Pre-training [10]	97.0	–	92.0
Strong Baseline* [19]	97.00	–	87.10	UP-ReID [41]	97.1	–	91.1
RPTM* [11]	<u>97.30</u>	98.40	<u>88.00</u>	†Ours ($S_A = \text{FastReID [13]}$)	99.11	<u>99.58</u>	93.8
†Ours ($S_A = \text{FastReID [13]}$)	99.70	99.82	91.71	†Ours ($S_A = \text{SOLIDER [4]}$)	<u>99.0</u>	99.6	95.5

Query ID	Model	Query	Rank-1	Rank-2	Rank-3	Rank-4	Rank-5	Rank-6	Rank-7	Rank-8	Rank-9	Rank-10	
VeRi776 [25]	ID 30	Baseline											
		Ours											
	ID 150	Baseline											
		Ours											
	ID 662	Baseline											
		Ours											
Market1501 [42]	ID 342	Baseline											
		Ours											
	ID 1083	Baseline											
		Ours											

Fig. 5. Qualitative ReID results of the baseline and proposed methods (Ours). The baseline model [13] is an appearance-based method. Green and red boxes denote true and false matching. Compared to the baseline method, the proposed method can match true positive pairs despite similar appearances and overcome the appearance ambiguities due to the spatial-temporal information (best viewed in color).

6 Conclusion

In this work, we proposed a ReID framework based on spatial-temporal fusion network that can estimate camera network topology and combines appearance and spatial-temporal similarities to alleviate appearance ambiguity. To this end, we proposed an adaptive Parzen window for reliable topology estimation and FusionNet for optimal similarity aggregation. The proposed framework achieved the best performance for vehicle ReID with a rank-1 accuracy of 99.70% and mAP of 91.71% on VeRi776 dataset and achieved the best performance for person ReID with a rank-1 accuracy of 99.11% and mAP of 95.5% on Market1501 dataset as well. The results support that using spatial and temporal information for ReID can leverage the accuracy of appearance-based methods and effectively deal with appearance ambiguity. Although the proposed framework has a lightweight networks, it can effectively perform ReID task. In addition, the proposed framework is flexible, so it has the potential to achieve high performance by utilizing other appearance-based methods as its baseline.

Acknowledgements. This work was supported in part by (90%) Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2021-0-00907, Development of Adaptive and Lightweight Edge-Collaborative

Analysis Technology for Enabling Proactively Immediate Response and Rapid Learning); and in part by (10%) Institute of Information & communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2023-RS-2023-00256629) grant funded by the Korea government(MSIT).

References

1. Ahmed, E., Jones, M., Marks, T.K.: An improved deep learning architecture for person re-identification. In: CVPR, pp. 3908–3916 (2015)
2. Bian, J., Lin, W.Y., Matsushita, Y., Yeung, S.K., Nguyen, T.D., Cheng, M.M.: GMS: grid-based motion statistics for fast, ultra-robust feature correspondence. In: CVPR, pp. 4181–4190 (2017)
3. Chen, T., et al.: Abd-net: attentive but diverse person re-identification. In: ICCV, pp. 8351–8361 (2019)
4. Chen, W., et al.: Beyond appearance: a semantic controllable self-supervised learning framework for human-centric visual tasks. In: CVPR, pp. 15050–15061 (2023)
5. Chen, X., et al.: Saliency-guided cascaded suppression network for person re-identification. In: CVPR, pp. 3300–3310 (2020)
6. Chen, Y., Zhu, X., Gong, S.: Person re-identification by deep learning multi-scale representations. In: ICCV Workshops, pp. 2590–2600 (2017)
7. Cheng, D., Gong, Y., Zhou, S., Wang, J., Zheng, N.: Person re-identification by multi-channel parts-based CNN with improved triplet loss function. In: CVPR, pp. 1335–1344 (2016)
8. Cho, Y.J., Kim, S.A., Park, J.H., Lee, K., Yoon, K.J.: Joint person re-identification and camera network topology inference in multiple cameras. *Comput. Vis. Image Underst.* **180**, 34–46 (2019)
9. Cho, Y.J., Yoon, K.J.: Improving person re-identification via pose-aware multi-shot matching. In: CVPR, pp. 1354–1362 (2016)
10. Fu, D., et al.: Unsupervised pre-training for person re-identification. In: CVPR, pp. 14750–14759 (2021)
11. Ghosh, A., Shanmugalingam, K., Lin, W.Y.: Relation preserving triplet mining for stabilising the triplet loss in re-identification systems. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 4840–4849 (2023)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
13. He, L., Liao, X., Liu, W., Liu, X., Cheng, P., Mei, T.: Fastreid: a pytorch toolbox for general instance re-identification. *arXiv preprint arXiv:2006.02631* (2020)
14. He, L., Liu, W.: Guided saliency feature learning for person re-identification in crowded scenes. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12373, pp. 357–373. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58604-1_22
15. He, S., et al.: Multi-domain learning and identity mining for vehicle re-identification. In: CVPR Workshops, pp. 582–583 (2020)
16. He, S., Luo, H., Wang, P., Wang, F., Li, H., Jiang, W.: Transreid: transformer-based object re-identification. In: ICCV, pp. 15013–15022 (2021)
17. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737* (2017)
18. Huang, W., et al.: Vehicle re-identification with spatio-temporal model leveraging by pose view embedding. *Electronics* **11**(9), 1354 (2022)
19. Huynh, S.V.: A strong baseline for vehicle re-identification. In: CVPR, pp. 4147–4154 (2021)

20. Jin, X., Lan, C., Zeng, W., Wei, G., Chen, Z.: Semantics-aligned representation learning for person re-identification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11173–11180 (2020)
21. Khamis, S., Kuo, C.-H., Singh, V.K., Shet, V.D., Davis, L.S.: Joint learning for attribute-consistent person re-identification. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014. LNCS, vol. 8927, pp. 134–146. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16199-0_10
22. Khorramshahi, P., Peri, N., Chen, J., Chellappa, R.: The devil is in the details: self-supervised attention for vehicle re-identification. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12359, pp. 369–386. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58568-6_22
23. Koestinger, M., Hirzer, M., Wohlhart, P., Roth, P.M., Bischof, H.: Large scale metric learning from equivalence constraints. In: CVPR, pp. 2288–2295. IEEE (2012)
24. Li, S., Sun, L., Li, Q.: Clip-reid: exploiting vision-language model for image re-identification without concrete text labels. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 1405–1413 (2023)
25. Liu, X., Liu, W., Mei, T., Ma, H.: A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 869–884. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_53
26. Liu, X., Liu, W., Mei, T., Ma, H.: PROVID: progressive and multimodal vehicle reidentification for large-scale urban surveillance. *IEEE Trans. Multimed.* **20**(3), 645–658 (2017)
27. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: ICCV, pp. 10012–10022 (2021)
28. Lv, K., et al.: Vehicle re-identification with location and time stamps. In: CVPR Workshops, pp. 399–406 (2019)
29. Meng, D., et al.: Parsing-based view-aware embedding network for vehicle re-identification. In: CVPR, pp. 7103–7112 (2020)
30. Miao, J., Wu, Y., Liu, P., Ding, Y., Yang, Y.: Pose-guided feature alignment for occluded person re-identification. In: ICCV, pp. 542–551 (2019)
31. Park, H., Ham, B.: Relation network for person re-identification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11839–11847 (2020)
32. Rao, Y., Chen, G., Lu, J., Zhou, J.: Counterfactual attention learning for fine-grained visual categorization and re-identification. In: ICCV, pp. 1025–1034 (2021)
33. Ren, M., He, L., Liao, X., Liu, W., Wang, Y., Tan, T.: Learning instance-level spatial-temporal patterns for person re-identification. In: ICCV, pp. 14930–14939 (2021)
34. Rong, L., Xu, Y., Zhou, X., Han, L., Li, L., Pan, X.: A vehicle re-identification framework based on the improved multi-branch feature fusion network. *Sci. Rep.* **11**(1), 20210 (2021)
35. Shen, Y., Xiao, T., Li, H., Yi, S., Wang, X.: Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In: ICCV, pp. 1900–1909 (2017)
36. Su, C., Li, J., Zhang, S., Xing, J., Gao, W., Tian, Q.: Pose-driven deep convolutional model for person re-identification. In: ICCV, pp. 3960–3969 (2017)
37. Sun, W., Dai, G., Zhang, X., He, X., Chen, X.: TBE-Net: a three-branch embedding network with part-aware ability and feature complementary learning for vehicle re-identification. *IEEE Trans. Intell. Transp. Syst.* **23**(9), 14557–14569 (2021)
38. Tang, Z., et al.: Pamtri: pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data. In: ICCV, pp. 211–220 (2019)
39. Wang, G., Lai, J., Huang, P., Xie, X.: Spatial-temporal person re-identification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 8933–8940 (2019)

40. Wang, G., Yuan, Y., Chen, X., Li, J., Zhou, X.: Learning discriminative features with multiple granularities for person re-identification. In: Proceedings of the 26th ACM International Conference on Multimedia, pp. 274–282 (2018)
41. Yang, Z., Jin, X., Zheng, K., Zhao, F.: Unleashing potential of unsupervised pre-training with intra-identity regularization for person re-identification. In: CVPR, pp. 14298–14307 (2022)
42. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q.: Scalable person re-identification: a benchmark. In: Computer Vision, IEEE International Conference on (2015)
43. Zheng, W.S., Gong, S., Xiang, T.: Person re-identification by probabilistic relative distance comparison. In: CVPR, pp. 649–656. IEEE (2011)
44. Zheng, Z., Ruan, T., Wei, Y., Yang, Y., Mei, T.: Vehiclenet: learning robust visual representation for vehicle re-identification. *IEEE Trans. Multimed.* **23**, 2683–2693 (2020)
45. Zhu, K., Guo, H., Liu, Z., Tang, M., Wang, J.: Identity-guided human semantic parsing for person re-identification. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12348, pp. 346–363. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58580-8_21
46. Zhu, K., Guo, H., Yan, T., Zhu, Y., Wang, J., Tang, M.: Pass: part-aware self-supervised pre-training for person re-identification. In: European Conference on Computer Vision, pp. 198–214. Springer (2022)



CMAEH: Contrastive Masked Autoencoder Based Hashing for Efficient Image Retrieval

Mehul Kumar¹, Aditya Sharma¹, Prerana Mukherjee^{1(✉)},
and Koteswar Rao Jerripothula^{2,3}

¹ Jawaharlal Nehru University, New Delhi, India

{mehulk43_soe, aditya35_soe, prerana}@jnu.ac.in

² Indian Institute of Technology Kanpur, Kanpur, India
kotesrj@iitk.ac.in

³ Indraprastha Institute of Information Technology, Delhi, New Delhi, India

Abstract. In recent years, the intersection of deep learning, hashing, and retrieval systems has witnessed significant advancements, particularly in the realm of image processing and information retrieval. In this paper, we present a novel end-to-end trainable network using contrastive masked autoencoder (CMAE) for efficient image retrieval. We comprehensively investigate the integration of contrastive masked autoencoders with hashing techniques, coupled with various types of losses, namely HashNet loss, Deep Supervised Hash (DSH) loss, and Greedy Hash loss, for enhancing retrieval performance. We delve into the efficacy of different masking methods in conjunction with these techniques to facilitate efficient representation learning. We investigate HashNet loss, a novel objective function tailored for learning hash functions directly from data, and contrastive loss, which encourages similar items to have similar hash codes while pushing dissimilar items apart. First, we introduce the concept of masked autoencoders, a variant of traditional autoencoders designed to learn robust representations from partially observed input data. We explore various masking strategies, such as attention masking, random masking, and patch masking, elucidating their effects on the encoding process and subsequent retrieval performance. Furthermore, we present a comparative analysis of different retrieval methods, including cosine similarity, knn approach and content-based retrieval approach, within the context of contrastive masked autoencoder method on different benchmark datasets like CIFAR10, ImageNet, MS-COCO and NUS-WIDE in terms of retrieval accuracy (mAP). The code is available at <https://github.com/Mehulk43/CMAEH>.

Keywords: Masked autoencoder · Contrastive learning · Hashing · Image retrieval

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78498-9_30.

1 Introduction

Effective image retrieval is crucial in several domains, such as computer vision, multimedia databases, and recommendation systems. Conventional techniques often face challenges when dealing with complex and multi-dimensional characteristics of image data, which calls for efficient feature representations. Recent progress in deep learning has led to substantial improvements in image retrieval tasks.

Contrastive learning, shown by recent improvements like SimCLR [4] and MoCo [10], has proven highly effective in acquiring resilient representations from unlabeled data. Contrastive learning methods have shown greater performance in tasks such as image classification, object detection, and semantic segmentation by focusing on maximizing similarity between positive pairs and minimizing similarity between negative pairs. When it comes to image retrieval, the importance of contrastive learning principles cannot be underestimated in finding semantically similar images from a vast database. Furthermore, autoencoders have been commonly used to reduce dimensionality and learning features. Autoencoders learn to extract important features from input images by encoding them into a latent space and then reconstructing them in the original space while filtering out noise and extraneous details. In vision tasks, autoencoders can be used for various applications such as denoising, image compression, and image generation. Similarly, in anomaly detection, autoencoders excel at accurately reconstructing normal data instances. However, when confronted with anomalous data, they typically exhibit higher reconstruction errors, allowing anomalies to be detected based on deviations from the normal data. The adaptability and versatility of autoencoders make them valuable tools in various domains. Their ability to learn representations directly from data without the need for explicit labels makes them particularly well-suited for tasks where labeled data is scarce or expensive to obtain.

Hashing-based methods are used to convert high-dimensional data, such as images, into compact binary codes. This allows for quick and effective similarity search by utilizing the Hamming distance metric. By transforming images into concise hash codes, these methods simplify storage and retrieval processes, leading to a significant decrease in memory usage and computing costs. Finding hash functions that maintain semantic similarity is still a difficult task. Masked Autoencoders (MAEs) [9] have become a potent deep-learning architecture for acquiring concise and semantically significant image representations. MAEs utilize the reconstruction loss of autoencoders and include masked regions in the input images during training. This encourages the model to concentrate on relevant image areas, boosting its capacity to collect distinct features and enhance retrieval effectiveness.

The Contrastive Masked Autoencoder (CMAE) [12, 21] is yet another variant of MAEs that combines contrastive learning principles with a masked autoencoder framework. In this paper, CMAE attempts to tackle the issues of efficient image retrieval by merging the powers of masked autoencoders in learning concise representations with the discriminative abilities of contrastive learning. Further-

more, we explore Contrastive Masked Autoencoder’s theoretical underpinnings and practical applications for efficient image retrieval. Here, we augment the hashing layer with the Contrastive Masked Autoencoder to ensure similarity-preserving learning, which facilitates efficient image retrieval. Furthermore, we utilize a joint loss optimization strategy to combine the deep supervised contrastive loss and hash loss. Previous research has been done on masked autoencoders for classification tasks, and we are doing masked autoencoder hashing for retrieval tasks. Wang et al. [21] have proposed new methods combining masked autoencoders and contrastive learning for video hashing in a self-supervised manner. In this paper, the authors take two random samples from a given video frame, then apply the hash layer to both samples and apply the contrastive loss. They have not used any masking ratio. Mishra et al. [15] proposed a new architecture combining Contrastive Learning, Masked Autoencoders, and Noise Prediction. In this paper, the authors implemented the two views of the masked autoencoder model, added noise after masking the image, and regenerated the image after denoising the model for the classification task. In [20], authors proposed a novel deep hashing method with minimal-distance separated hash centers that are used to represent image classes. In this paper, the goal is to learn the hash center along with the hashing function for efficient image retrieval tasks. However, in our model, we have done this by just applying the random masking after patch generation, adding the hash layer at last for generating the hash code, and then applying the supervised contrastive loss. Bao et al. [12] presented BEIT (Bidirectional Encoder representation from image Transformer) using masked image modeling (MIM) similar to BERT. This method works on two views: discrete representations of patches (visual tokens) and image patches. Random patches are masked during the training, and BEIT predicts the original visual tokens based on the remaining patches and visual tokens. Cao et al. [2] proposed Deep Cauchy hashing (DCH) architecture that leverages deep learning and hamming space retrieval. DCH used a deep network to encode images into compact binary codes for hamming space retrieval. After that, a Cauchy distribution-based loss function is used. However, our work presents MAE (Masked autoencoder) architecture focusing on reconstructing the original image from masked patches of images.

The novel contributions are as follows:

- Using the pre-trained MAE-ViT models, we present a novel end-to-end trainable Contrastive Masked Autoencoder Hashing (CMAEH) model for image retrieval.
- The proposed CMAEH model uses the MAE model’s encoder as a feature extractor, adds a hashing module and leverages existing hashing frameworks for training. Finally, we utilize training within a joint loss optimization framework.
- The CMAEH model has demonstrated outstanding performance on benchmark CIFAR10, ImageNet, NUS-Wide, and MS-COCO datasets in hashing frameworks. We provide exhaustive experiments to exhibit the proposed

model’s performance with varying masking ratios, different masking strategies, and different hashing techniques.

2 Related Works

Deep Supervised Hashing (DSH) [14] represents an early endeavor leveraging Convolutional Neural Networks (CNN) by converting network outputs into binary hash codes through quantization. DSH integrates a regularizer on the real-valued network outputs to produce binary outputs. HashNet [3], on the other hand, introduces a Tanh function-based continuation method to facilitate a seamless transition from real-valued features to binary codes. HashNet employs weighted cross-entropy loss to effectively learn sparse data while preserving similarity. To mitigate the issue of vanishing gradients encountered in the GreedyHash method [16], HashNet employs an identity mapping through which gradients are propagated. GreedyHash utilizes the Sign function on the hash layer. These models utilize CNN as the backbone network architecture.

The transformer [17] architecture, first employed for sequence-to-sequence learning in NLP, has been a great success based primarily on the attention mechanism, which permits it to identify long-range interdependencies within input sequences. Vision Transformer (ViTs) [6], which received a big boost from the success of transformer architectures in NLP, represent image data as sequences of patches which allow such models to represent global spatial relationships effectively. Through this hierarchical process, ViTs learn to find the features from images that can be generalized for image retrieval purposes. These representations not only encode features at low-level and high-level embeddings but also recognize and retain semantically similar images. ViTs have unique abilities that include adaptation to different input types, scalability, and parallelization to process large amounts of data. Consequently, this creates the need for further research and developments involved in these image retrieval systems.

Autoencoder, as a classical method, brings forth the issue of representation learning. It utilizes an encoder that converts an input to a low-dimensional representation and a decoder that replicates the input, as in PCA and k-means [11]. Denoising autoencoders [18] are autoencoder variants that take the corrupted input signal and learn to reconstruct the original uncorrupted version [19]. In [8], the authors introduced an unsupervised image hashing method designed to compress images into binary codes without supervision. It is essential to retain significant data while compressing information. It provides a simple method to fine-tune the Vision Transformer (ViT) for unsupervised hashing, based on the success of ViT as a large-scale vision pre-training model. The suggested method consists of two primary components: a module for storing features and a module for storing hashes. The feature-preserving module utilizes the pre-trained ViT model to restore the original features of corrupted images, emphasizing the retention of valuable information. The hashing-preserving module retains semantic information from the Vision Transformer (ViT) by using Kullback-Leibler divergence loss and enhancing quantization and similarity loss to minimize quantization error.

3 Proposed Method

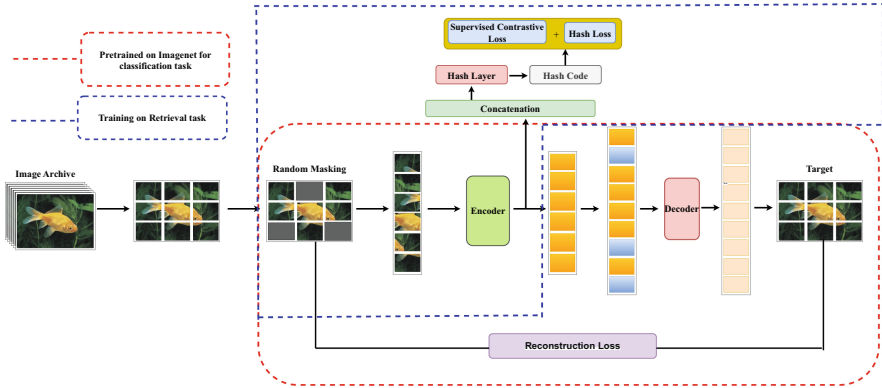


Fig. 1. The framework for image retrieval is named Contrastive Masked Autoencoder hashing (*CMAEH*). The pre-trained MAE_VIT is utilized for image classification in this framework, as denoted by the Red Dotted Line. The model is then trained for the primary retrieval task, as shown by the Blue Dotted Line. We optimize the model by using a combined strategy that includes the Hashing Loss, Supervised Contrastive Loss, and Reconstruction Loss.

Figure 1 illustrates the proposed Contrastive Masked Autoencoder hashing (*CMAEH*) architecture. The architecture includes modules for MAE encoder, patch embedding with random masking, MAE Decoder, pre-training within the classification framework, hashing, and image retrieval.

3.1 Masking

The architecture presented by the authors in [6] is that images are to be divided into regular, separate, and non-overlapping patches. Subsequently, we choose from a range of patches and replace others by the use of masking. In our sampling method, we randomly pick up patches, which we process without replacement, using the uniform distribution. This is called random sampling. Taking advantage of random sampling with a high masking ratio to stop repetition is the way in which our brain works by making the task more challenging, as extrapolating from the nearby visible patches becomes impossible. Similarly, we have implemented two additional masking techniques: **Patch/block masking**: The illustrated patch/block-wise masking method used by [1] has the intention that all the big blocks are removed. The patch masking with 50% block masking gives poor results as compared with random masking with a 50% masking ratio. This task is more complex than random sampling because a bigger training loss is seen. Similarly, the reconstruction is obscure. In the case of **attention grid**

masking, the approach implies that the patches are not randomly selected for masking, but, instead, they are systematically selected at equal intervals. In Fig. 2, we can see some examples of random masking of images. Images may have repetitive patterns and smooth gradients in neighboring pixels or patches. By using random masking on patches within images, our model considers the global context and relationships between different regions of the image. Our model uses the ViT as a backbone network that learns long-range dependencies between different image parts. The decoder part is trained to reconstruct the masked areas based on the remaining visible patches.

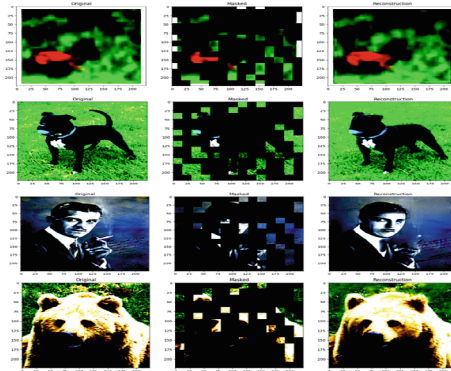


Fig. 2. Reconstruction of the images with masking ratio 0.75 where, 1st, 2nd, 3rd and 4th rows correspond to images taken from CIFAR10, IMAGENET, NUS-WIDE and MS-COCO respectively.

3.2 MAE Encoder

We utilize a ViT [6] encoder, but it is only applied to visible, unmasked patches. In our model, patches are embedded through linear projection coupled with positional embeddings, after which Transformer blocks are used successively to work on the set. The encoder handles $(1 - [\text{mask-ratio}])$ of the whole set at a time. Masked patches have been eliminated, with no usage of mask tokens. In this way, we are capable of efficiently training very big encoders with very limited computation resources and memory. The whole set of the compressed data is appropriately decoded by a lightweight decoder, which will be discussed in the next section.

Pretraining This paper utilizes the pretrained MAE-ViT model [9]. The pretraining is conducted using the ultimate result of the MAE encoder on the ImageNet dataset within a classification framework. The MAE-ViT extracts the important features from the initial vector of the transformer encoder output.

3.3 Hash Layer

In this layer, we fine-tune the pre-trained MAE-VIT model in the image retrieval task. A hash block is added above the MAE encoder output so that the model will learn to hash an image. Then, there is a dropout step from encoder MAE having a dropout factor of 0.5 which is applied to the output MAE encoder. A linear projection is utilized to transform the feature into a 1024-dimensional vector, followed by forwarding it via the ReLU activation function. A linear projection is utilized to generate the final hash features, ensuring they have the same quantity of values as the hash bit length.

3.4 MAE Decoder

As we can see in Fig. 1, the MAE decoder will receive a complete set of tokens, including visible patches that were encoded and the mask tokens. The mask token is a standard learned vector that is applied to detect the missed spot that needs to be predicted. All the tokens in this specific assembly are given positional embeddings. Mask tokens would lack spatial information if this is not provided. The decoder contains an additional sequence of Transformer blocks. The MAE decoder is utilized for image reconstruction during pre-training, and in this paper, we have used the encoder feature for generating hash code for image retrieval tasks. The decoder architecture can be developed independently of the encoder design. We evaluate smaller decoders and have less depth compared to the encoder. In our model, the decoder part is not directly involved in retrieval tasks, but it plays an important role in encouraging the encoder part to learn a latent representation that captures the essential information from the visible patches. This includes not only the basic visual features but also the relationships between different parts of the image that are very useful for retrieval tasks.

3.5 Loss

In this section, we present the reconstruction loss along with joint loss optimization (for retrieval) used in our framework.

Reconstruction Loss. As in [9], we use the normalized image as target reconstruction. We essentially apply MSE (denoted as L_{rec}) loss for masked patches between input image X_{inp} and target image X_{tar} , as shown below,

$$L_{\text{rec}} = \frac{1}{N_{\text{mask}}} \sum (X_{\text{tar}} - X_{\text{inp}})^2 \quad (1)$$

where N_{mask} denotes the total number of patches masked in an image.

Contrastive Loss. Contrastive loss is an objective in a representation learning technique of the SimCLR [13] that aims at maximizing the agreement between differently augmented images representing the same data example in the latent space. Instead of augmented images, we leverage images with the same label in our framework. Like [5], we also focus on maximizing similarity between positive samples, i.e., an image and another image with the same label. We employ cosine similarity metric $siml(x, y)$, as defined below:

$$siml(x, y) = \frac{x^T y}{\|x\|_2 \|y\|_2} \quad (2)$$

where we compute the dot product between any two vectors x and y after they have been normalized. The loss function for a positive pair of examples (i, j) is defined as follows:

$$Loss_c = \sum_{k \in P(i)} -\log \frac{\exp(siml(Z_i, Z_j)/\tau)}{\sum_{k \in P(i)} \exp(siml(Z_i, Z_k)/\tau)} \quad (3)$$

where $P(i)$ denotes the set of indices of images having the same label as i^{th} image, and τ is the temperature parameter.

The final loss is calculated for all positive pairs, including both (i, j) and (j, i) , within a mini-batch.

HashLoss. This paper incorporates three hash loss functions: HashNet loss, DSH loss, and GreedyHash loss. Deep Supervised Hashing (DSH) [14] employs CNN to convert network outputs into binary hash codes through quantization, utilizing a regularizer to ensure the generation of binary outputs from real-valued network outputs. HashNet [3] utilizes weighted cross-entropy loss to learn sparse data while preserving similarity. Gradients are directed through identity mapping to prevent vanishing gradients, as encountered in the GreedyHash method [16]. In HashNet loss [3], Cao et al. proposed a deep learning approach to learn binary hash code. In this paper, authors have used a continuation method to gradually adjust the scale of the learned hash codes, facilitating the optimization process. The loss function is used to find the similarity between hash code obtained by using pairwise similarity between samples and employing a weighted logistic regression loss function. In DSH [14], Liu et al. proposed a deep supervised hashing method to generate the binary code for image retrieval. The loss function is used to find the minimum distance between the hash code of a pair of images, which is done by using pairwise similarity loss and Quantization loss. In [16], Su et al. proposed a novel approach called greedy hashing to learn binary hash codes efficiently. The loss function used here is to find fast and accurate optimization of hash codes within a convolutional neural network (CNN) framework. There are two components used in this loss. First, quantization loss ensures that the learned continuous codes are close to their binary hash code equivalents. Second, similarity preservation loss ensures that the pairwise similarities between samples are preserved in the binary hash codes. The main function of the loss is to

minimize the difference between the inner products of the continuous code and their binary code. In the proposed model CMAEH, the best results are achieved with HashNet loss.

3.6 Joint Optimization

Then, we use a weighted combination of Hash Loss [3] and Supervised Contrastive Loss [13] which is computed as,

$$Joint_{Loss} \equiv \alpha \times Hash_{Loss} + (1 - \alpha) \times SupCon_{Loss} \quad (4)$$

where:

α is the hyper-parameter

$Hash_{Loss}$ is the loss in the hashing module and

$SupCon_{Loss}$ is Supervised Contrastive Learning module.

We can regulate the trade-off between the two loss components and customize the joint optimization loss for the image retrieval task on the CIFAR10 dataset by adapting the values of α .

4 Results

4.1 Experimental Setting

Datasets. In this paper, we have used four datasets, namely CIFAR-10, IMAGENET, MS-COCO, and NUS-WIDE, for evaluating our proposed model. The CIFAR-10 dataset is a collection of 60,000 images consisting of 10 different classes, and each class consists of 6,000 images. Based on the standard experimental process, as done in [2, 23] we conduct experiments on the CIFAR10 dataset. The 5,000 images are randomly picked from the different class sets, which is 500 images for each category. A random sample of 100 images from each category is used in a query set containing 1000 images, as the rest of the images serve as the database. ImageNet dataset is a subpart of the Large Scale Visual Recognition Challenge (ILSVRC 2015). In similar lines to [7, 22], we follow the same retrieval protocol on the ImageNet dataset. We pick 100 random classes. All images from these classes that are in the validation set are part of the query set. All training set images of these classes are considered as the database. There 13,000 images are sampled from the database for the purpose of using them as the training set. In the case of the NUS-WIDE dataset, according to the retrieval protocol in [7], we take the 21 most frequent concepts as image annotations, and this extends into 195K images. The query set has 100 images per concept and is obtained through random sampling. The remaining images are used to build a database of those such that they can be retrieved. Moreover, our training set consists of 500 images from the database picked randomly from each concept. MS-COCO

dataset is a collection of images with a set of 80 categories. We utilize the existing protocol used in [2,22], which contains query, training, and retrieval sets having 5000, 10000, and 117218 images.

Table 1. mAP of our model of different masking ratios and different datasets using different masking strategies.

(a) mAP of our model of different masking ratios and different datasets on random masking method.

Masked Autoencoder Hashing (Random masking)	Masked Ratio	mAP (CIFAR 10)	mAP (ImageNet)	mAP (MSCOCO)	mAP (NUS-WIDE)
	0.00	97.70	81.01	82.80	87.61
0.25	96.50	76.40	79.22	83.40	
0.50	95.90	75.20	77.31	78.31	
0.75	94.40	72.40	73.56	75.40	

(b) mAP of our model of different masking ratios and different datasets on Patch masking method

Masked Autoencoder Hashing (Patch masking)	Masked Ratio	mAP (CIFAR 10)	mAP (ImageNet)	mAP (MSCOCO)	mAP (NUS-WIDE)
	0.00	97.14	80.41	78.67	85.13
0.25	95.82	74.33	74.08	80.14	
0.50	94.63	72.42	73.18	76.51	
0.75	93.11	70.67	68.16	72.96	

(c) mAP of our model of different masking ratios and different datasets on Attention grid masking method

Masked Autoencoder Hashing (Attention masking)	Masked Ratio	mAP (CIFAR 10)	mAP (ImageNet)	mAP (MSCOCO)	mAP (NUS-WIDE)
	0.00	97.45	80.91	81.47	87.13
0.25	96.20	75.98	78.81	83.14	
0.50	95.63	75.10	75.33	78.10	
0.75	93.11	71.53	73.10	75.01	

Evaluation Metrics. In this paper, we adopt the Mean Average Precision (mAP) as the evaluation criterion. The mAP is utilized to compare the correctness of the entire binary codes in terms of the Hamming distances. It is measured by the mean of average precision (AP) of all queries. We have adopted the prevalent evaluation approach [7,22] of measuring the accuracy of our image retrieval methods and computing the Mean Average Precision (mAP). We utilize MAP@1000 for CIFAR10 and ImageNet and MAP@5000 for NUS-WIDE and MS-COCO, respectively. We measure the quality of retrieval using five standard measures: Mean Average Precision (mAP), Precision-Recall curves (PR), Recall curve with respect to the number of retrieved samples (R@N), Precision curve with respect to the number of retrieved samples (P@N) and Top 5 retrieved images.

Implementation Details. In this paper, we used MAE-VIT(large) hashing in which image sizes are resized with $n = 224$. Patch size is 16, the hidden dimension is 1024, the number of encoder and decoder blocks is 24, 8. The number of attention heads is 16. We used 16,32,64 bits length to generate the hash code. We run the model for 150 epochs and test the model after every 30 epochs; the batch size is 32 for all experiments, and the default masking ratio is 25%. We use Adam as an optimizer with a learning rate of $1e^{-5}$. We also try

Table 2. mAP Using different types of loss function on CIFAR-10 dataset using our model

Loss	Model name	Mask ratio	Bits	Dataset	mAP
Reconstruct loss	MAE using ViT	0.25	64	CIFAR10	95.08
HashNet Loss + Reconstruct loss	MAE using ViT	0.25	64	CIFAR10	95.50
HashNet Loss + Contrastive loss + Reconstruct loss	MAE using ViT	0.25	64	CIFAR10	96.10
Deep Supervised Loss + Contrastive loss + Reconstruct loss	MAE using ViT	0.25	64	CIFAR10	96.00
GreedyLoss + Contrastive loss + Reconstruct loss	MAE using ViT	0.25	64	CIFAR10	95.80

Table 3. The comparison of our model with respect to other methods on different datasets with different hash bit code

Methods	Backbone	CIFAR-10			NUS-WIDE			MSCOCO		
		16 bits	32 bits	64 bits	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
ITQ	VGG	0.305	0.325	0.349	0.627	0.645	0.664	0.598	0.624	0.648
DeepBit	VGG	0.194	0.249	0.277	0.392	0.403	0.429	0.407	0.419	0.430
BinGAN	-	0.476	0.512	0.520	0.654	0.709	0.713	0.651	0.673	0.696
GreedyHash	VGG	0.448	0.473	0.501	0.633	0.691	0.731	0.582	0.668	0.710
HashGAN	-	0.447	0.463	0.481	-	-	-	-	-	-
DVB	VGG	0.403	0.422	0.446	0.604	0.632	0.665	0.570	0.629	0.623
TBH	VGG	0.532	0.573	0.578	0.717	0.725	0.735	0.706	0.735	0.722
CIBHash	VGG	0.590	0.622	0.641	0.790	0.807	0.815	0.737	0.760	0.775
ITQ	ViT	0.870	0.901	0.910	0.724	0.756	0.779	0.715	0.805	0.844
GreedyHash	ViT	0.879	0.901	0.915	0.629	0.690	0.752	0.647	0.756	0.836
CIBHash	ViT	0.903	0.925	0.938	0.779	0.810	0.826	0.809	0.846	0.867
IPHash	ViT	0.942	0.951	0.958	0.797	0.816	0.826	0.826	0.875	0.894
Joint Loss(ours)	MAE(ViT)	0.951	0.961	0.968	0.768	0.823	0.834	0.781	0.793	0.816

different masking ratios $m = [0.00, 0.25, 0.50, 0.75]$. The best result is obtained at $m = 0.25$, as shown in the experiments. We have implemented the model in RTX A6000 GPU in pytorch framework; the time complexity of three methods (Random masking, Patch masking, and Grid attention masking) is provided as in Table 4. The average retrieval run time is 0.2 s for 1000 images.

Table 4. Retrieval time of different variants of our proposed architecture.

Masked Autoencoder hashing	Dataset	Training Run Time	Retrieval Running time	Retrieved image time
Random Masking	CIFAR10	1.45 min per epoch	0.2 s for 1000 images	<0.1 s
Patch masking		1.45 min per epoch	0.2 s for 1000 images	<0.1 s
Grid Attention Masking		1.45 min per epoch	0.2 s for 1000 images	<0.1 s

4.2 Experimental Results

In this section, we present the details of experimental results analysis using our model on different datasets and also using different masking ratios. An ablation study is also given to check the effect of the joint loss on our model.

Quantitative Results. From Table 1a, 1b, 1c, we can see the experimental analysis of our model using different masking techniques and different masking ratios. Table 2 shows the quantitative results of our model using different loss function parameters on the CIFAR10 dataset. Table 3 shows the detailed comparative analysis of our model with other state-of-the-art models. We have used the MAE-VIT as the backbone network. While comparing with the latest state-of-the-art methods, our model has shown an average 1.1% increase on the CIFAR10 dataset for all hash code lengths and an average 0.8% increase on NUS-WIDE dataset for 32 and 64 bits hash code length, while on 16 bits hash code length, IPHash has shown great result. In the case of MSCOCO, IPhash has shown great results. Additional results are given in supplementary material.

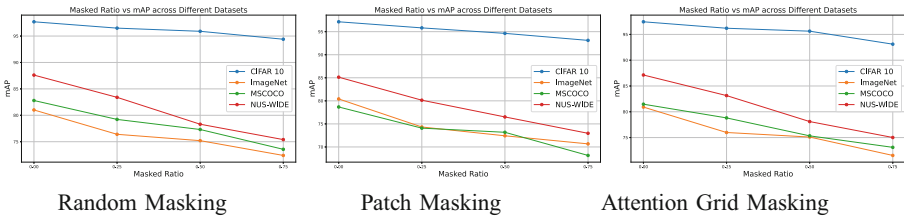


Fig. 3. Masking Ratio vs mAP on the datasets

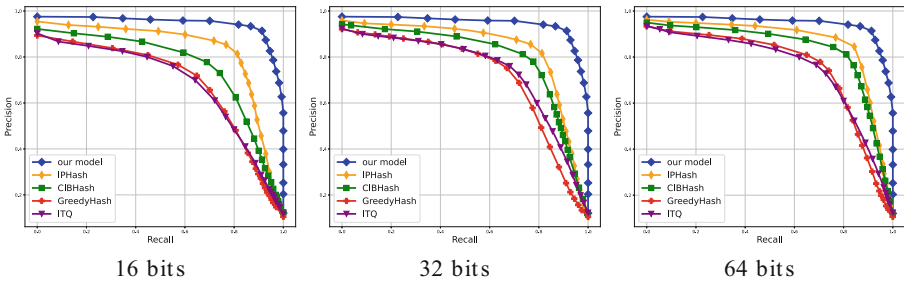


Fig. 4. PR curve on CIFAR-10 dataset

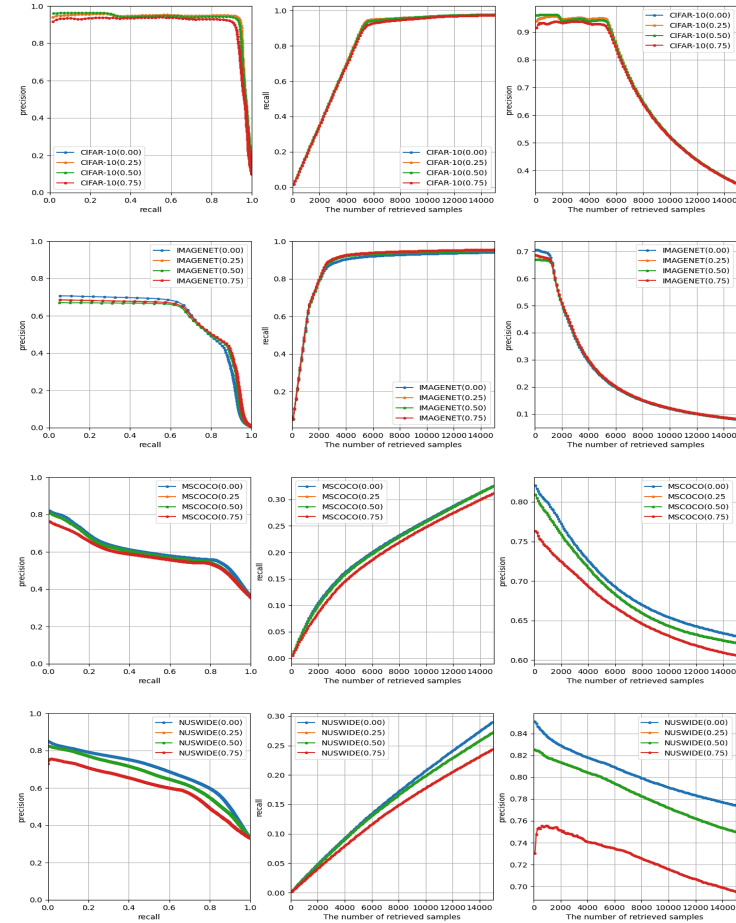
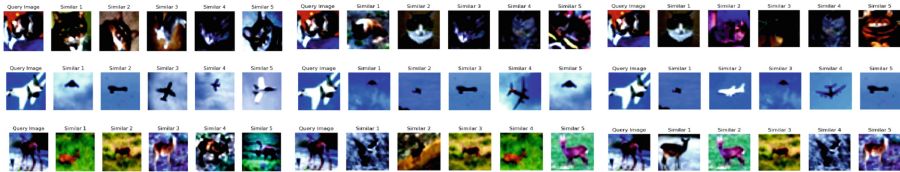


Fig. 5. Pr curve of different masking ratios [0.00, 0.25, 0.50, 0.75] using our model on CIFAR10, ImageNet, MSCOCO and NUSWIDE Dataset, respectively, using random masking



(a) 0.25 Masking ratio (b) 0.50 Masking ratio (c) 0.75 Masking ratio

Fig. 6. Top 5 query image retrieval using different masking ratio [0.25] using our model on CIFAR10 Dataset, respectively, using random masking where 1st rows indicates cosine similarity retrieval, 2nd rows indicates knn search retrieval and 3rd rows indicates distance retrieval method

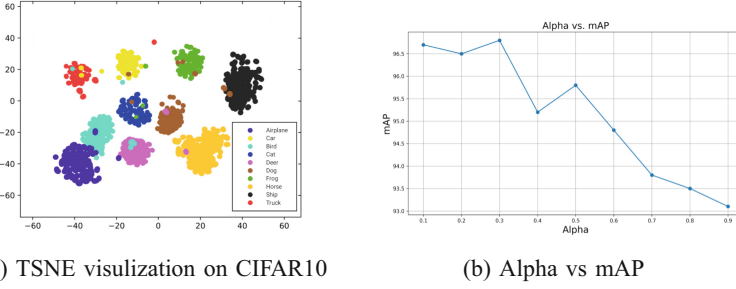


Fig. 7. a)TSNE visualization on CIFAR 10 and b) Alpha vs mAP graph

Qualitative Results. Figure 3 shows the masking ratio vs mAP graphs over various datasets. Figure 4 demonstrates Precision-Recall (P-R) curves with our model on CIFAR-10 with the various methods. We surpass IPHash, CIBHash, GreedyHash, and ITQ in retrieval performance. The proposed technique can save crucial, meaningful information even when it is compressed a lot. Figure 5 shows the roc curve of our model, using different masking ratios on different datasets. Figure 7a presents the T-SNE visualization results of our model on CIFAR-10. In 64-bit hash samples, distinct embeddings (labels) are clearly distinguished, proving our model’s great information-preserving ability using joint loss. Our CMAEH method is capable of generating hash codes that are more discriminative than those generated by existing hashing techniques. Figure 6 indicates the relevant top 5 qualitative retrieval performance of the proposed model.

4.3 Ablation Study

Alpha Parameter: Our model implemented a loss function that integrated a Hash loss with a supervised contrastive loss. The parameter alpha does a very crucial task of maintaining the balance between the two components of this joint loss. The parameter alpha allows us to control the model’s alignment through the change of the relative importance between Hash Loss and contrastive supervised loss during the training process. To conduct a complete analysis of the effect of alpha, we performed a trial with different alpha values. We investigated how the proposed model’s performance varied for various alpha settings, i.e., higher values gave more relevance to the hash loss and lower values gave more weight to supervised contrastive loss. We evaluated both fixed alpha options with adaptive techniques to pick the most efficient approaches for our specific tasks. As shown in Fig. 7b, the mAP value for CIFAR10 is higher with alpha = 0.3 than other alpha values.

5 Conclusion

This paper introduced a novel hashing method called contrastive masked autoencoder hashing, which is based on the MAE-ViT large-scale vision pre-training

model. When learning the information-preserving feature extractor, we utilized a random mask to eliminate spatial redundancy and preserve semantic information in the source image. We used the hashing-preserving module to generate hash codes. Comprehensive experiments are carried out on four benchmark datasets. The results demonstrated that our method outperforms other baseline methods.

References

1. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: bert pre-training of image transformers. arXiv preprint [arXiv:2106.08254](https://arxiv.org/abs/2106.08254) (2021)
2. Cao, Y., Long, M., Liu, B., Wang, J.: Deep cauchy hashing for hamming space retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1229–1237 (2018)
3. Cao, Z., Long, M., Wang, J., Yu, P.S.: Hashnet: deep learning to hash by continuation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5608–5617 (2017)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607. PMLR (2020)
5. Chen, T., Sun, Y., Shi, Y., Hong, L.: On sampling strategies for neural network-based collaborative filtering. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 767–776 (2017)
6. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
7. Fan, L., Ng, K.W., Ju, C., Zhang, T., Chan, C.S.: Deep polarized network for supervised learning of accurate binary hashing codes. In: IJCAI, pp. 825–831 (2020)
8. Gong, Q., Wang, L., Lai, H., Pan, Y., Yin, J.: Vit2hash: unsupervised information-preserving hashing. arXiv preprint [arXiv:2201.05541](https://arxiv.org/abs/2201.05541) (2022)
9. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16000–16009 (2022)
10. 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)
11. Hinton, G.E., Zemel, R.: Autoencoders, minimum description length and helmholtz free energy. Adv. Neural Inf. Process. Syst. **6** (1993)
12. Huang, Z., et al.: Contrastive masked autoencoders are stronger vision learners. IEEE Trans. Pattern Anal. Mach. Intell. (2023)
13. Khosla, P., et al.: Supervised contrastive learning. Adv. Neural Inf. Process. Syst. **33**, 18661–18673 (2020)
14. Liu, H., Wang, R., Shan, S., Chen, X.: Deep supervised hashing for fast image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2064–2072 (2016)
15. Mishra, S., et al.: A simple, efficient and scalable contrastive masked autoencoder for learning visual representations. arXiv preprint [arXiv:2210.16870](https://arxiv.org/abs/2210.16870) (2022)
16. Su, S., Zhang, C., Han, K., Tian, Y.: Greedy hash: towards fast optimization for accurate hash coding in CNN. Adv. Neural Inf. Process. Syst. **31** (2018)
17. Vaswani, A., et al.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)

18. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1096–1103 (2008)
19. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A., Bottou, L.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **11**(12) (2010)
20. Wang, L., Pan, Y., Liu, C., Lai, H., Yin, J., Liu, Y.: Deep hashing with minimal-distance-separated hash centers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 23455–23464 (2023)
21. Wang, Y., Wang, J., Chen, B., Zeng, Z., Xia, S.T.: Contrastive masked autoencoders for self-supervised video hashing. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 2733–2741 (2023)
22. Yuan, L., et al.: Central similarity quantization for efficient image and video retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3083–3092 (2020)
23. Zhu, H., Long, M., Wang, J., Cao, Y.: Deep hashing network for efficient similarity retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)

Author Index

A

Åström, Kalle 289
Adak, Chandranath 196
Agarwal, Akshay 55
Ai, Xinbo 45
Ali, Amin Ahsan 72
Amin, M. Ashraful 72
Aysa, Alimjan 167

B

Bhattacharya, Saumik 135
Biswas, Sanket 135
Blumenstein, Michael 118, 196

C

Chao, Zi-Chun 103
Chaudhuri, Bidyut B. 196
Chen, Jiawei 1
Chen, Jun-Yi 103
Chen, Zhounan 182
Cho, Sangyeon 29
Cho, Yeong-Jun 421
Cu, Vinh Loc 151
Cui, Shuang 406

D

Dang, Yonghao 260
Das, Alloy 135
Deshpande, Gauri 304
Dhar, Ankita 273

E

Eli, Elham 167

F

Fahim, Md. 72
Fuad, Md. Tahmid Hasan 72

G

Gornale, Shivanand S. 118
Gupta, Ashutosh 389

H

Halder, Arnab 118
Hossain, Meheraj 72
Hsieh, Wei-Chun 103
Hsu, Gee-Sern 103
Hu, Lei 209
Hu, Zhenghui 332
Huang, Shuangping 182, 209
Huang, Wenhui 151
Huang, Yubo 87

I

Ionescu, Radu Tudor 16

J

Jeon, Jangyeong 29
Jerripothula, Koteswar Rao 437
Jiang, Liyu 151
Jin, Shaohui 406
Joe, Hae-Won 421

K

Kim, Hye-Geun 421
Kim, Junyeong 29
Kong, Cong 1
Kumar, Mehul 437

L

Lai, Xin 87
Li, Ange 45
Li, Liang 332
Li, Peishan 345
Liang, Lingyu 182
Lin, Junxiang 182
Liu, Guangsheng 45
Liu, Hao 406
Liu, Shiguang 243
Liu, Yun 359
Lladós, Josep 135
Luo, Wenbin 45

M

Ma, Guangyi 345
 Ma, Minuk 29
 Mamat, Hornisa 167
 Marciano, Matteo 273
 Mohan, Karthik 226
 Moon, Yong-Hyuk 421
 Mu, Da 260
 Mukherjee, Himadri 273
 Mukherjee, Prerana 437

N

Na, You-Kyoung 421
 Nechita, Maria Ilinca 16

O

Oskarsson, Magnus 289

P

Pal, Umapada 118, 135
 Palaiahnakote, Shivakumara 118
 Pandey, Suraj Kumar 226
 Patil, Hemant A. 316
 Paul, Subhajit 389
 Peng, Wenjie 182
 Priya, Kumari 196
 Pusuluri, Aditya 316

Q

Qi, Yuankai 332

R

Rahman, A. K. M. Mahbubur 72
 Rahman, Rahat Rizvi 72
 Rogoz, Ana-Cristina 16
 Roy, Kaushik 273
 Ruan, Wenna 374

S

Schuller, Björn W. 304
 Sharma, Aditya 437
 Shi, Hanyi 374
 Shi, Shaoliang 151
 Sultana, Faria 72

T

Tan, Shunquan 1
 Tang, Jin 260
 Tang, Tong 359
 Tegler, Erik 289

U

Ubul, Kurban 167
 Uzair, Mohammed 55

W

Wang, Dong 167
 Wang, Hongkui 332
 Wang, Huimin 406
 Wang, Junfei 345
 Wang, Ningzhi 374
 Wang, Qinghai 243
 Wen, Yimin 151

X

Xu, Mingliang 406
 Xu, Xuebin 167
 Xu, Zhengli 151

Y

Yan, Chenggang 332
 Yang, Jian 332
 Yap, Moi Hoon 103
 Yin, Jianqin 260
 Yin, Zhaoxia 1
 Ying, Jun 359
 Yue, Haobo 260

Z

Zeng, Lingbin 374
 Zhang, Jiehua 332
 Zhang, Wenhao 406
 Zhang, Xinpeng 1
 Zhang, Yonghong 345
 Zhang, Zhedong 332
 Zhang, Zhicheng 260
 Zhu, Anran 87
 Zhu, Zhiyuan 359