

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

LNCS 15331

# Pattern Recognition

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

**31** Part XXXI

ICPR  
2024 INDIA



 Springer

MOREMEDIA 

# Lecture Notes in Computer Science

15331

## 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 XXXI

*Editors*

Apostolos Antonacopoulos   
University of Salford  
Salford, UK

Rama Chellappa   
Johns Hopkins University  
Baltimore, MD, USA

Saumik Bhattacharya   
IIT Kharagpur  
Kharagpur, India

Subhasis Chaudhuri   
Indian Institute of Technology Bombay  
Mumbai, India

Cheng-Lin Liu   
Chinese Academy of Sciences  
Beijing, China

Umapada Pal   
Indian Statistical Institute Kolkata  
Kolkata, India

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Computer Science

ISBN 978-3-031-78118-6

ISBN 978-3-031-78119-3 (eBook)

<https://doi.org/10.1007/978-3-031-78119-3>

© 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. At ICPR 2024 the papers were categorized into six tracks: Artificial Intelligence, Machine Learning for Pattern Analysis; Computer Vision and Robotic Perception; Image, Video, Speech, and Signal Analysis; Biometrics and Human-Machine Interaction; Document and Media Analysis; and Biomedical Image Analysis and Informatics.

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

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

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

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

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

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

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

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

October 2024

Umapada Pal  
Josef Kittler  
Anil Jain

# Organization

## General Chairs

Umapada Pal  
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  
 Anwesha 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 Coquenat  
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 Otroshi Shahreza  
Hebeizi Li  
Heng Zhang  
Hengli Wang  
Hengyue Liu  
Hertog Nugroho  
Hieyong Jeong  
Himadri Mukherjee  
Hoai Ngo  
Hoda Mohaghegh  
Hong Liu  
Hong Man  
Hongcheng Wang  
Hongjian Zhan  
Hongxi Wei  
Hongyu Hu  
Hoseong Kim  
Hossein Ebrahimnezhad  
Hossein Malekmohamadi  
Hrishav Bakul Barua  
Hsueh-Yi Sean Lin  
Hua Wei  
Huafeng Li  
Huali Xu  
Huaming Chen  
Huan Wang  
Huang Chen  
Huanran Chen  
Hua-Wen Chang  
Huawen Liu  
Huayi Zhan  
Hugo Jair Escalante  
Hui Chen  
Hui Li  
Huichen Yang  
Huiqiang Jiang  
Huiyuan Yang  
Huizi Yu  
Hung T. Nguyen  
Hyeongyu Kim  
Hyeonjeong Park  
Hyeonjun Lee  
Hymalai Bello  
Hyung-Gun Chi  
Hyunsoo Kim  
I-Chen Lin  
Ik Hyun Lee  
Ilan Shimshoni  
Imad Eddine Toubal

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

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

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

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

ODTr: Transformer Integrating OCR Auxiliary Map and Image Depth Information for Document Image Unwarping .....	1
<i>Xiangyu Xie, Yuxuan Zhou, and Liangcai Gao</i>	
Oracle Character Recognition Based on Attention Enhancement and Multi-level Feature Fusion .....	13
<i>Zhiwang Han, Nurbiya Yadikar, Xuebin Xu, Alimjan Aysa, and Kurban Ubul</i>	
DocHFormer: Document Image Dewarping via Harmonized Modeling of Hierarchical Priors .....	29
<i>Xinyue Zhou, Guanting Li, Nanfeng Jiang, Da-Han Wang, Xu-Yao Zhang, and ShunZhi Zhu</i>	
Document Image Shadow Removal via Frequency Information-Oriented Network .....	45
<i>Fan Yang, Xinyue Zhou, Nanfeng Jiang, Da-Han Wang, Xu-Yao Zhang, Guanting Li, Wang Man, and Yun Wu</i>	
Improving Online Handwriting Recognition with Transfer Learning Using Out-of-Domain and Different-Dimensional Sources .....	61
<i>Jiseok Lee, Masaki Akiba, and Brian Kenji Iwana</i>	
ROISER: Towards Real World Semantic Entity Recognition from Visually-Rich Documents .....	76
<i>Zening Lin, Jiapeng Wang, Wenhui Liao, Weicong Dai, Longfei Xiong, and Lianwen Jin</i>	
Perception-Enhanced Generative Transformer for Key Information Extraction from Documents .....	91
<i>Runbo Zhao, Jun Jie Ou Yang, Chen Gao, Xugong Qin, Gangyan Zeng, Xiaoxu Hu, and Peng Zhang</i>	
MuLAD: Multimodal Aggression Detection from Social Media Memes Exploiting Visual and Textual Features .....	107
<i>Md. Maruf Hasan, Shawly Ahsan, Mohammed Moshiul Hoque, and M. Ali Akber Dewan</i>	

<b>Fig<sup>4</sup>: A Voting-Based Paradigm for Enhancing Retrieval Augmented Generation</b> .....	124
<i>Wenbo Guan, Xiaoqian Li, Jiyu Lu, and Jun Zhou</i>	
<b>Improving Chinese Emotion Classification Based on Bilingual Feature Fusion</b> .....	139
<i>Haocheng Lan, Jie Ou, Zhaokun Wang, and Wenhong Tian</i>	
<b>SNOBERT: A Benchmark for Clinical Notes Entity Linking in the SNOMED CT Clinical Terminology</b> .....	154
<i>Mikhail Kulyabin, Gleb Sokolov, Aleksandr Galaida, Andreas Maier, and Tomas Arias-Vergara</i>	
<b>Enhancing Automated Short Answer Grading with Prompt-Driven Augmentation and Prompt Adaptive Oversampling</b> .....	164
<i>P. P. Afeefa, Raju Hazari, and Pranesh Das</i>	
<b>SANS: Spatial-Aware Neural Solver for Plane Geometry Problem</b> .....	183
<i>Zi-Hao Lin, Shun-Xin Xiao, Zi-Rong Chen, Jian-Min Li, Da-Han Wang, and Xu-Yao Zhang</i>	
<b>A Multi-modal Framework to Counter Hate Speeches</b> .....	197
<i>Kirtilekha Bhesra and Akshay Agarwal</i>	
<b>TBIA-DBNet: A Two-Branch Image-Adaptive DBNet for Scene Text Detection in Real-World Foggy Scenes</b> .....	208
<i>Zhaoxi Liu, Gang Zhou, Runlin He, Mengnan Zhang, Zhenhong Jia, and Jing Ma</i>	
<b>Breaking Boundaries: Enhancing Script Identification Using a Learnable MULLER Resizer</b> .....	222
<i>Souhaila Djaffal, Yasmina Benmabrouk, Chawki Djeddi, and Moises Diaz</i>	
<b>Arbitrary-Shaped Scene Text Recognition with Deformable Ensemble Attention</b> .....	237
<i>Shuo Xu, Zeming Zhuang, Mingjun Li, and Feng Su</i>	
<b>Primary Key Free Watermarking for Numerical Tabular Datasets in Machine Learning</b> .....	254
<i>Xin Che, Mohammad Akbari, Shaoxin Li, David Yue, Yong Zhang, and Lingyang Chu</i>	
<b>Offline Handwritten Signature Verification Using a Stream-Based Approach</b> .....	271
<i>Kecia Gomes de Moura, Rafael Menelau O. Cruz, and Robert Sabourin</i>	

OCR4HSV: A Multi-task Learning Approach for Handwritten Signature Verification .....	287
<i>Chao-Qun Lin, Da-Han Wang, Yan-Fei Su, De-Wu Ge, and Xu-Yao Zhang</i>	
Learning Explicit Radical Representations for Zero-Shot Chinese Character Recognition .....	303
<i>Song-Liang Pan, Da-Han Wang, Nanfeng Jiang, Xu-Yao Zhang, and Shunzhi Zhu</i>	
Deep Learning for Arabic Word Classification: Leveraging Transfer Learning and Grad-CAM for Morphological Analysis .....	318
<i>Mohamed Hjaiej, Imen Ben Cheikh, and Heithem Abbas</i>	
A Cost Minimization Approach to Fix the Vocabulary Size in a Tokenizer for an End-to-End ASR System .....	331
<i>Sunil Kumar Kopparapu and Ashish Panda</i>	
ASD-Diffusion: Anomalous Sound Detection with Diffusion Models .....	343
<i>Fengrun Zhang, Xiang Xie, and Kai Guo</i>	
FCHiFi-GAN: Aggrandizing Fast Convergence with Batchwise Normalization .....	356
<i>Ravindrakumar M. Purohit, Arushi Srivastava, and Hemant A. Patil</i>	
Adaptive Enhanced Reversible Flow Model for Remote Sensing Image Super Resolution .....	373
<i>Peishan Li, Yonghong Zhang, Junfei Wang, Guangyi Ma, and Ziwei Yuan</i>	
Saliency-Based Neural Representation for Videos .....	389
<i>Qian Cao, Dongdong Zhang, and Xiaolei Zhang</i>	
HNRC: Lightweight Image Compression with Hybrid Neural Representation .....	404
<i>Xinyuan Cheng, Dongdong Zhang, and Xiaolei Zhang</i>	
<b>Author Index</b> .....	419



# ODTr: Transformer Integrating OCR Auxiliary Map and Image Depth Information for Document Image Unwarping

Xiangyu Xie, Yuxuan Zhou, and Liangcai Gao<sup>(✉)</sup>

Wangxuan Institute of Computer Technology, State Key Laboratory of Multimedia  
Information Processing, Peking University, Beijing, China  
{xxy, sherco, gaoliangcai}@pku.edu.cn

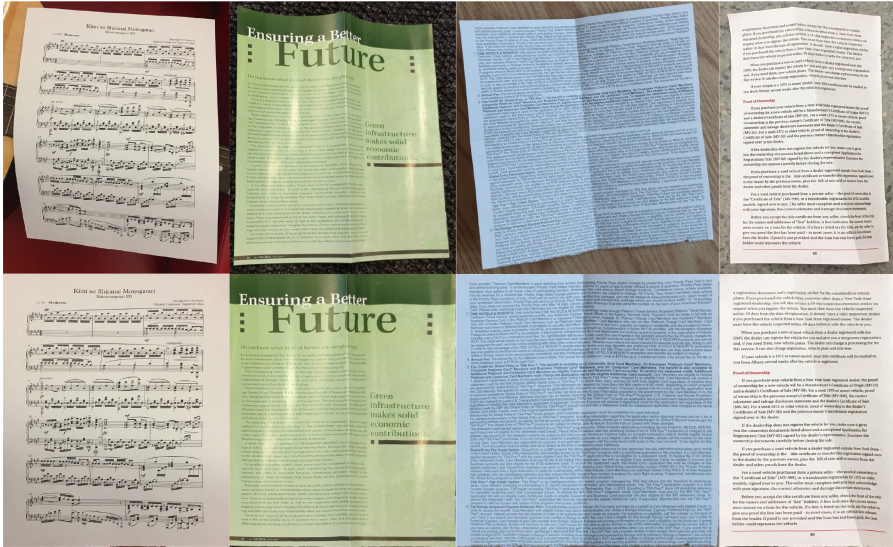
**Abstract.** In this study, we introduce a novel application of transformer for document image unwarping, leveraging depth information and Optical Character Recognition (OCR) results. Our proposed model integrates two key modules: a depth module and an OCR module, into the transformer framework tailored for document image processing. The depth module predicts the relative depth of each pixel in the document image, thereby providing crucial spatial context for unwarping. Concurrently, the OCR module identifies regions suitable for OCR, acting as preferences for the unwarping process. This hybrid approach aims to mitigate text distortion inherent in document image unwarping, consequently enhancing OCR accuracy, although it may reduce our model's capability in structural image unwarping. Experimental results showcase our model's effectiveness, achieving a Character Error Rate (CER) of 24.81%, marking a significant 6.2% absolute enhancement compared to the baseline method, DocTr.

**Keywords:** Transformer · Depth Information · OCR · Document Image Unwarping

## 1 Introduction

Nowadays, the ubiquitous use of portable devices like smartphones for capturing and sharing document images is prevalent. However, paper documents often suffer from geometric distortions, aggravated by the inherent challenges of perspective distortion during casual captures. These distortions hinder the effective exchange of document images in both personal and professional spheres. Furthermore, document image unwarping serves as a vital preprocessing step for Optical Character Recognition (OCR), enhancing its accuracy and efficiency. Consequently, rectifying geometric distortions in document images is imperative for improving OCR performance, underscoring the necessity to prioritize and invest in document image unwarping tasks.

In early stages of tackling document image unwarping challenges, methods primarily relied on hardware-based solutions. These approaches involved deploying stereoscopic vision systems [16, 25] for document modeling, acquiring flat document images using reference points, or pre-modeling specific document types [7, 24, 28] followed by shape transformations to obtain flat images corresponding to their categories. However, these methods were largely reliant on specialized hardware platforms to acquire spatial information from documents, severely limiting their practical applicability in everyday life and work settings.



**Fig. 1.** Illustration of document image unwarping results obtained using our model. The top row exhibits the distorted document images, while the bottom row showcases the corresponding unwarping outcomes.

The advent of Convolutional Neural Networks (CNNs) [10] revolutionized document image unwarping tasks, circumventing the constraints of hardware platforms. CNNs are trained to establish mappings from input images to rectified images. The U-Net [21] architecture, initially deployed for this purpose, demonstrated promising outcomes. Subsequently, Generative Adversarial Networks (GANs) [8] were tailored for document image unwarping, alongside methods such as block-wise unwarping [12], all of which yielded significant successes. Notably, the DocTr [6] model integrated transformer [26] into document image unwarping tasks, yielding substantial enhancements in OCR accuracy.

When examining the outcomes of document image unwarping through the mentioned deep learning methodologies, there is a notable challenge: irregularities at the document image borders. These models often interpret the borders as linear, causing text distortion along the edges. Moreover, incomplete rectification

of document images by the models exacerbates text deformation. These mistakes detrimentally impact the OCR precision, counteracting the objectives of document image unwarping. The top row of the right column in Fig. 3 indicates some text distortions generated during the image unwarping process.

To mitigate this challenge, we propose a strategy of delineating OCR-optimized regions from the original document images, thereby establishing a reference for the model. This method ensures that our model preserves text legibility throughout the document image unwarping process. Furthermore, recognizing the intrinsic influence of text legibility on geometric unwarping outcomes, we incorporate depth information extracted from the images as an additional unwarping reference. This integration aims to encourage the model to achieve maximal flattening of document images. Our contributions can be summarized as follows:

- We introduce a depth information module that forecasts the relative depth details of input document images. This module provides spatial references for subsequent document image unwarping tasks, aiding the model in attaining superior unwarping outcomes.
- We introduce an OCR module to optimize document image unwarping. Initially, OCR operations are executed on distorted document images to isolate easily recognizable regions, generating an OCR mask. Subsequently, image features are extracted from these regions and integrated into the transformer decoder as references for document image unwarping. This approach enables the model to prioritize preserving the legibility of text throughout the unwarping procedure.
- We enhance the encoder-decoder [2] architecture of DocTr [6] by incorporating a depth information module and an OCR module, enabling the model to address document image unwarping and text recognizability concurrently. Experimental findings reveal that our model attains a 24.8% Character Error Rate (CER) in the unwarping results, showcasing a remarkable 6.2% absolute enhancement over the state-of-the-art (SOTA) DocTr [6] model.

## 2 Related Work

### 2.1 Rectification Based on Three-Dimensional Reconstruction

Early techniques for document image unwarping predominantly relied on stereoscopic vision systems to reconstruct documents in three dimensions. Subsequently, corresponding deformations were applied to the images to produce the final flat images. Notably, Fu et al. [7] proposed a pre-modeling approach for flattening book images, where they predefined books as cylindrical-like models. By utilizing textual line information from acquired images, they corresponded distorted regions of the images with flattened regions, automatically adjusting model parameters, thus achieving the task of flattening book images. Adrian et al. [28] introduced a document image unwarping approach leveraging general stereoscopic vision principles. Their method involved modeling the shape of book

images and deriving flat images of books based on reference points. Similarly, Tsoi et al. [24] proposed a technique that employed images captured from multiple viewpoints to generate a comprehensive three-dimensional model, effectively addressing the document image unwarping challenge.

In addition to the pre-modeling methodologies mentioned earlier, numerous researchers have explored hardware-based document image flattening techniques. For example, Gao et al. [16] employed two structured light beams to illuminate document pages and delineate spatial curves. These curves were subsequently flattened onto a plane by solving a system of ordinary differential equations, yielding flat images of documents. Similarly, Ulges et al. [25] utilized stereoscopic vision systems to reconstruct the shape of paper documents afflicted with curved surfaces or folds.

## 2.2 Rectification Based on Deep Learning

**Classical Models for Document Image Unwarping.** In the realm of document image unwarping, convolutional neural network models are used to acquire pixel-level mappings from distorted document images to rectified images. Initially, Ma et al. [15] devised a stacked U-Net network model, pioneering an end-to-end approach for document image unwarping with promising outcomes. Building upon this foundation, Tanmoy et al. [1] augmented this architecture with an edge detection module, resulting in enhanced performance. Subsequently, Ramanna et al. [20] introduced a more streamlined model leveraging Generative Adversarial Networks [8] for document image unwarping tasks, achieving greater efficiency. Furthermore, Li et al. [12] proposed a Patch-based model architecture, facilitating document image flattening by segmenting them into blocks before integration into complete images. Liu et al. [13] designed a pyramid-style encoder-decoder architecture that predicts the unwarping results of document images at multiple resolutions from low to high. They employed three gating modules to introduce structural information such as text lines and table rows to assist in predicting document unwarping images, achieving excellent results.

**Transformer for Document Image Unwarping.** With the enormous success of transformer in the natural language domain [4, 18, 19], it has gradually been introduced into fields such as computer vision [5], speech processing [17], and recommendation systems [23], achieving remarkable results as well. Feng et al. [6] revolutionized document image unwarping by integrating the transformer architecture into their pioneering work, DocTr. They devised a classic encoder-decoder framework, where document images underwent background separation, feature extraction, and subsequent mapping to the final backward map via the Transformer decoder. This approach effectively addressed the document image unwarping challenge. Their method showcased remarkable results in metrics such as MS-SSIM (Multi-Scale Structural Similarity) [27] and LD (Local Distortion) [29], while also exhibiting great performance in OCR accuracy.

Our model is based on the framework of DocTr [6], bolstered by the integration of image depth and OCR auxiliary map, resulting in a substantial enhancement of OCR accuracy for document image unwarping tasks. Furthermore, we conducted ablation experiments to showcase the effectiveness of the modules we devised. Figure 1 displays examples of document image unwarping results achieved through our proposed model.

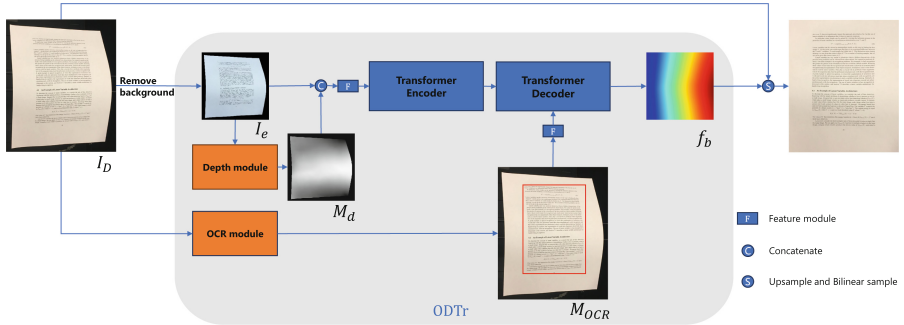


Fig. 2. Architecture of Our Model for Document Image Unwarping.

### 3 Approach

In this section, we present an improved transformer model tailored for document image unwarping, which integrates both image depth information and an OCR auxiliary map. As depicted in the Fig. 2, the model architecture comprises a background segmentation module, a depth information prediction module, an OCR module, and a transformer encoder-decoder architecture.

For the task of document image unwarping, the model takes a distorted document image as input, aiming to predict pixel-wise mappings for unwarping. Specifically, the input image  $I_D \in R^{H \times W \times 3}$  is processed by the model. Initially, the model downsamples the input image to obtain a low-resolution version  $I_d \in R^{H_0 \times W_0 \times 3}$ , where  $H_0 = W_0 = 288$ . Subsequently,  $I_d$  is forwarded to a background segmentation module to eliminate its background, yielding the resulting image  $I_e$ . The background segmentation module leverages the preprocessing module from DocTr [6] to effectively remove the background of the document image. It’s based on U-Net [21] and trained with a binary cross-entropy loss. Because of its effectiveness and universality, we bring it into our ODTr model.

Following this, the depth information prediction module generates the depth information  $M_d \in R^{H_0 \times W_0}$  for  $I_e$ . Subsequently,  $I_e$  and  $M_d$  are concatenated to form  $I'_e \in R^{H_0 \times W_0 \times 4}$ , which serves as input to the transformer encoder for extracting global-aware representations. Concurrently,  $I_D$  is utilized as input



to the OCR module to produce the OCR mask  $M_{OCR}$ . During inference via the transformer decoder, the OCR mask is incorporated as an additional input, functioning as a reference for document image unwarping, ultimately predicting the backward mapping field  $f_b \in R^{H_0 \times W_0 \times 2}$ . Finally,  $f_b$  is upsampled to size  $H \times W$  and transformed into the final backward mapping field  $f_B \in R^{H \times W \times 2}$ .

### 3.1 OCR Module

As previously mentioned, distortions in the text content of document images may arise from irregularities at the image edges and inadequate unwarping. To address this challenge, we developed the OCR Module. Initially, it conducts character recognition on the image  $I_D$  to produce a document region mask  $M_{OCR} \in R^{H_0 \times W_0}$ , which delineates regions of the image conducive to OCR. Subsequently, these segmented regions of the image are fed into a feature head. The resulting output contributes to the attention mechanism, facilitating pixel-level predictions.

### 3.2 Depth Information Prediction Module

Numerous challenges in document image unwarping tasks arise from the absence of spatial information. Moreover, our model introduces the OCR region mask as a guide for document image unwarping, which may influence the geometric unwarping of the image. To solve this contradiction, we devised the depth information prediction module to incorporate three-dimensional information, offering the model a preliminary image shape reference.

The module is constructed based on the U-Net [21] architecture, which takes the image  $I_e$  as input and predicts its depth information  $M_d$  through this module. It's important to note that this depth information represents relative positions. Specifically, the ground truth  $M_{gt}$  is normalized from the z-axis position of the image to the range [0, 1]. The normalization operation is defined as follows:

$$M_{gt} = \frac{Z_{gt} - Z_{min}}{Z_{max} - Z_{min}} \quad (1)$$

where  $Z_{gt}$  represents the value of the image's z-axis position, and  $Z_{min}$  and  $Z_{max}$  denote the minimum and maximum values of the image's z-axis position, respectively. The training loss of the module is defined as the L1 distance between the predicted depth map  $M_d$  and the ground truth  $M_{gt}$ :

$$\mathcal{L}_{depth} = \|M_{gt} - M_d\|_1 \quad (2)$$

### 3.3 Transformer Encoder and Decoder

The transformer encoder and decoder adopt the basic architecture from DocTr [6], comprising  $K$  encoder layers and  $K$  decoder layers. It takes the background-removed image  $I_e$  and the predicted depth information  $M_d$  as inputs. These

inputs pass through the feature module to obtain features  $f_{g_1} \in R^{\frac{H}{8} \times \frac{W}{8} \times c_{g_1}}$ , which are then fed into the transformer encoder to extract global-aware representations  $F_K$ . Simultaneously, the document image, processed through the OCR module to generate OCR detection regions, undergoes the feature module to obtain  $f_{g_2} \in R^{\frac{H}{8} \times \frac{W}{8} \times c_{g_2}}$ . These two sets of features,  $F_K$  and  $f_{g_2}$ , collectively serve as the attention value for the transformer decoder to generate pixel-level predictions  $f_b$ . The backward mapping  $f_b$  is generated as follows:

$$f_{g_1} = F([I_e, M_d]) \quad (3)$$

$$f_{s_1} = Flatten(f_{g_1}) \in R^{N_g \times c_{g_1}} \quad (4)$$

$$F_K = EN(f_{s_1}, E'_p) \quad (5)$$

$$f_{g_2} = F(OCR(I_D)) \quad (6)$$

$$f_b = UP(DE([F_K, Flatten(f_{g_2})], E_p, E_d)) \quad (7)$$

where  $N_g = \frac{H}{8} \times \frac{W}{8}$ ,  $E'_p$  and  $E_p$  represent position embeddings,  $E_d$  is a learnable embedding,  $F(\cdot)$ ,  $EN(\cdot)$ ,  $UP(\cdot)$ ,  $OCR(\cdot)$  and  $DE(\cdot)$  denote the feature module, transformer encoder, upsample layers, OCR module and transformer decoder, respectively. The training loss of the model is defined as the L1 distance between the predicted backward mapping  $f_b$  and the ground truth  $f_{gt}$  as follows:

$$\mathcal{L}_{bm} = \|f_{gt} - f_b\|_1 \quad (8)$$

## 4 Experiments

### 4.1 Baseline Model

We choose DocUNet [15], DRIC [12], DewarpNet [3], and DocTr [6] as baseline models. These models represent classic methods or the best transformer models for document image unwarping. Within these methods, document image unwarping is commonly divided into two tasks: geometric unwarping and illumination correction. Illumination correction typically follows geometric unwarping and can be directly reused. Therefore, for comparison purposes, we focus solely on evaluating the performance of these methods in geometric unwarping.

### 4.2 Datasets and Metrics

The model proposed in this paper is trained on the Doc3D dataset [3]. Furthermore, to facilitate comparison with DocTr [6], the model undergoes performance evaluation on the DocUNet benchmark dataset [15]. To quantify their performance, we employ Multi-Scale Structural Similarity (MS-SSIM) [27] and Local Distortion (LD) [29] metrics to assess the geometric similarity between the model’s results and the ground truth. Additionally, Edit Distance (ED) [11] and Character Error Rate (CER) are metrics commonly used to evaluate the accuracy of character recognition in the generated results.

**Datasets.** The Doc3D dataset [3] stands out as the largest document image unwarping dataset, encompassing 100,000 images of distorted document images synthesized from both real document data and rendering software. It includes various accompanying data such as three-dimensional coordinates, albedo maps, normals maps, depth maps, UV maps, and backward mapping maps.

On the other hand, the DocUNet Benchmark [15] comprises 130 real-world document images, which serve as a widely-used benchmark to evaluate the document image unwarping capabilities of different models.

**Metrics.** MS-SSIM measures the structural similarity between the model’s output and the ground truth, while LD quantifies the local differences between them. These metrics evaluate the model’s image unwarping capabilities from the perspective of image structure. The MS-SSIM score ranges from 0 to 1, with 1 indicating perfect structural similarity. Conversely, a lower LD score signifies better model performance.

On the other hand, ED and CER compare the OCR results of the model’s output with those of the ground truth. ED measures the number of edits (insertions, deletions, substitutions) required to transform the recognized text into the ground truth text. CER represents the percentage of characters in the recognized text that differ from the ground truth text, normalized by the total number of characters in the ground truth text. These metrics provide insights into the accuracy of the model’s image unwarping process in terms of character recognition.

### 4.3 Implementation Details

During the model training process, we train the depth module and the transformer encoder-decoder model separately. For the depth module, we employ the Adam optimizer [9] with a fixed learning rate of  $1 \times 10^{-4}$ . The image size is set to  $288 \times 288$ , and the batch size is set to 16. For the transformer encoder-decoder model, we also set the image size to  $288 \times 288$  and the batch size to 16. Additionally, we utilize the AdamW optimizer [14] with a maximum learning rate set to  $1 \times 10^{-4}$ , adjusting it based on the One-Cycle policy [22].

The generation of the OCR auxiliary map and the calculation of ED and CER are performed using Tesseract (*v3.02.02*) as the OCR engine.

### 4.4 Experimental Results

We compare our model with DocTr [6] and several typical models on the DocUNet benchmark [15], and the results are presented in Table 1. As mentioned earlier, we assess both image structure similarity and OCR accuracy. It’s important to note that, when evaluating OCR accuracy, we focus solely on 30 images in the DocUNet benchmark [15] where the textual content predominates, following the approach of DocTr [6]. The data in the table indicates that our

**Table 1.** Comparisons with the results reported by the original papers trained on Doc3D [3] and evaluated on the DocUNet benchmark [15].

Method	LD ↓	MS-SSIM ↑	ED ↓	CER ↓
Distorted Image	–	–	2051.4	0.68
DocUNet [15]	14.08	–	–	–
DRIC [12]	18.19	–	1840.9	0.61
DewarpNet [3]	8.98	0.4735	1121.1	0.38
DocTr [6]	<b>8.38</b>	<b>0.4970</b>	935.2	0.31
ODTr	10.43	0.4552	<b>625.2</b>	<b>0.25</b>

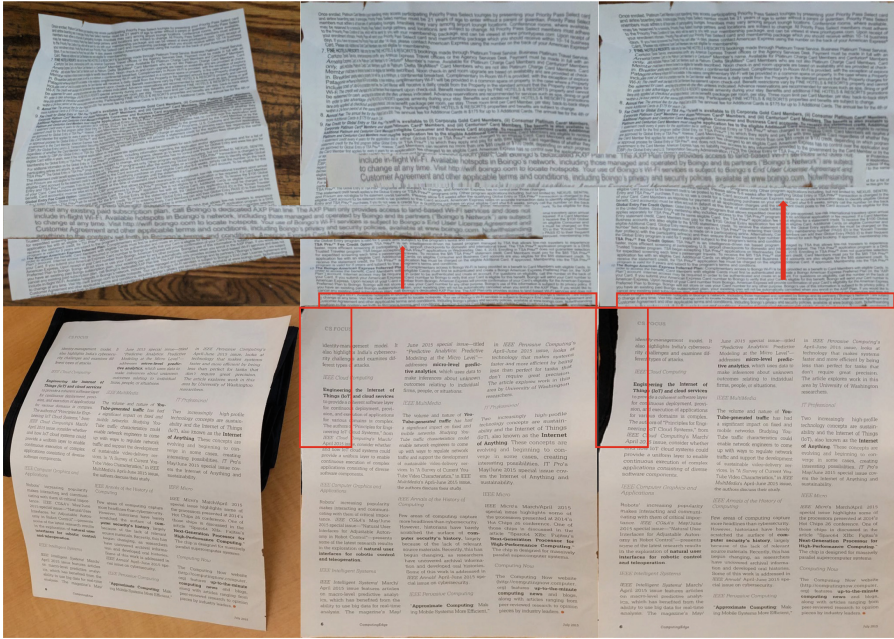
model surpasses the state-of-the-art model in terms of both ED and CER metrics, achieving an absolute improvement of 6% in CER over the state-of-the-art result by DocTr [6]. However, our model’s performance on the structural similarity evaluation metric is slightly lower than the state-of-the-art result.

This outcome is attributed to the incorporation of image regions conducive to character recognition from the original image as references during the document image unwarping process. Figure 3 depicts a comparison between the results of our model and DocTr [6]. From the content within the red boxes, we can observe that during document image unwarping with DocTr [6], text near the edges of the image sometimes experiences distortion. In contrast, our model demonstrates better mitigation of such issues. Moreover, our model prefers to keep text content taking up most of the document image’s space, consistent with the document’s feature. However, these regions easily recognized may contain minor distortions. Consequently, while preserving the recognizability of text regions, this process may introduce some degree of interference to image unwarping.

## 4.5 Ablation Study

In this model, we have designed two modules: the depth information module and the OCR module. These modules aim to extract spatial context and text information from the input image, respectively. To validate the effectiveness of these modules, we trained three models: the basic transformer with the depth module, the transformer with the OCR module, and the complete model trained on the Doc3D dataset. We then evaluated the capabilities of these models on the DocUNet benchmark [15].

As depicted in Table 2, incorporating OCR-friendly regions contributes to enhancing the readability of document images, whereas the inclusion of depth information aids in improving the capability of geometric unwarping. However, it’s noteworthy that the readability of documents may occasionally conflict with geometric unwarping, as described previously.



**Fig. 3.** Comparison examples between our model and DocTr [6]. The left column shows the input document images, the middle column displays the results from our model, and the right column presents the results from DocTr [6].

**Table 2.** Ablation experiments conducted on our model ODTr.

Method	Depth	OCR	LD ↓	MS ↑	ED ↓	CER ↓
Depth only	✓		9.89	0.4620	785.7	0.286
OCR only		✓	11.99	0.4263	453.4	0.198
Full	✓	✓	10.43	0.4552	625.2	0.248

## 5 Conclusion and Future Work

In our study, we propose ODTr, a novel model for document image unwarping. Leveraging depth information and OCR-friendly regions from the original image, our model demonstrates enhanced performance in character recognition.

Our findings underscore the significance of incorporating depth information and text features from input images to enhance document unwarping outcomes. Moving forward, we aim to employ a more rigorous methodology to identify flat regions in original images and train the depth module on a larger dataset to further enhance visual and OCR performance.

**Acknowledgement.** This work is supported by the projects of National Natural Science Foundation of China (No. 62376012) and Beijing Science and Technology Program (Z231100007423011), which is also a research achievement of Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology).

## References

1. Bandyopadhyay, H., Dasgupta, T., Das, N., Nasipuri, M.: A gated and bifurcated stacked u-net module for document image dewarping. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 10548–10554. IEEE (2021)
2. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint [arXiv:1406.1078](https://arxiv.org/abs/1406.1078) (2014)
3. Das, S., Ma, K., Shu, Z., Samaras, D., Shilkrot, R.: Dewarpnet: single-image document unwarping with stacked 3D and 2D regression networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 131–140 (2019)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
5. 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)
6. Feng, H., Wang, Y., Zhou, W., Deng, J., Li, H.: Doctr: document image transformer for geometric unwarping and illumination correction. arXiv preprint [arXiv:2110.12942](https://arxiv.org/abs/2110.12942) (2021)
7. Fu, B., Wu, M., Li, R., Li, W., Xu, Z., Yang, C.: A model-based book dewarping method using text line detection. In: Proceedings of the 2nd International Workshop on Camera Based Document Analysis and Recognition, Curitiba, Brazil, pp. 63–70 (2007)
8. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
11. Levenshtein, V.I., et al.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics Doklady. vol. 10, pp. 707–710. Soviet Union (1966)
12. Li, X., Zhang, B., Liao, J., Sander, P.V.: Document rectification and illumination correction using a patch-based CNN. ACM Trans. Graphics (TOG) **38**(6), 1–11 (2019)
13. Liu, X., Meng, G., Fan, B., Xiang, S., Pan, C.: Geometric rectification of document images using adversarial gated unwarping network. Pattern Recogn. **108**, 107576 (2020)
14. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101) (2017)
15. Ma, K., Shu, Z., Bai, X., Wang, J., Samaras, D.: DocuNet: document image unwarping via a stacked U-Net. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4709 (2018)
16. Meng, G., Wang, Y., Qu, S., Xiang, S., Pan, C.: Active flattening of curved document images via two structured beams. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3890–3897 (2014)

17. Pham, N.Q., Nguyen, T.S., Niehues, J., Müller, M., Stüker, S., Waibel, A.: Very deep self-attention networks for end-to-end speech recognition. arXiv preprint [arXiv:1904.13377](https://arxiv.org/abs/1904.13377) (2019)
18. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
19. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**(140), 1–67 (2020)
20. Ramanna, V.K.B., Bukhari, S.S., Dengel, A.: Document image dewarping using deep learning. In: *ICPRAM*, pp. 524–531 (2019)
21. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) *MICCAI 2015. LNCS*, vol. 9351, pp. 234–241. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
22. Smith, L.N., Topin, N.: Super-convergence: Very fast training of neural networks using large learning rates. In: *Artificial Intelligence and Machine Learning for Multi-domain Operations Applications*, vol. 11006, pp. 369–386. SPIE (2019)
23. Sun, F., et al.: Bert4rec: sequential recommendation with bidirectional encoder representations from transformer. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1441–1450 (2019)
24. Tsoi, Y.C., Brown, M.S.: Multi-view document rectification using boundary. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2007). <https://doi.org/10.1109/CVPR.2007.383251>
25. Ulges, A., Lampert, C.H., Breuel, T.: Document capture using stereo vision. In: *Proceedings of the 2004 ACM symposium on Document engineering*, pp. 198–200 (2004)
26. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
27. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003. vol. 2, pp. 1398–1402. IEEE (2003)
28. Yamashita, A., Kawarago, A., Kaneko, T., Miura, K.: Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system. In: *Proceedings of the 17th International Conference on Pattern Recognition*, 2004. *ICPR 2004*. vol. 1, pp. 482–485 (2004). doi: <https://doi.org/10.1109/ICPR.2004.1334171>
29. You, S., Matsushita, Y., Sinha, S., Bou, Y., Ikeuchi, K.: Multiview rectification of folded documents. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(2), 505–511 (2017)





# Oracle Character Recognition Based on Attention Enhancement and Multi-level Feature Fusion

Zhiwang Han<sup>1</sup>, Nurbiya Yadikar<sup>1,2</sup>, Xuebin Xu<sup>1</sup>, Alimjan Aysa<sup>1,2</sup>,  
and Kurban Ubul<sup>1,2,3(✉)</sup>

<sup>1</sup> School of Computer Science and Technology, Xinjiang University,  
Urumqi 830046, China  
kurbanu@xju.edu.cn

<sup>2</sup> Xinjiang Multilingual Information Technology Key Laboratory,  
Xinjiang University, Urumqi 830046, China

<sup>3</sup> Joint International Research Laboratory of Silk Road Multilingual Cognitive  
Computing, Xinjiang University, Urumqi 830046, China

**Abstract.** Oracle bone characters represent the earliest inscriptions in China. Recognizing and deciphering these characters is significant. Despite some progress made by recent methods, their recognition accuracy remains limited by two major issues: 1) how to focus on characters features within complex background noise images, and 2) how to effectively fuse shallow detail information with deep semantic information. To address these issues, we propose a novel deep learning model called Character Feature Enhancement Network (CFE-Net). The model consists of two key components: Character Feature Enhancement (CFE) and Adaptive Multi-level Classifier Fusion (AMCF). Specifically, CFE utilizes the Spatial Focus Attention Module (SFAM) to focus on extracting foreground character features and suppressing background noise, thereby significantly enhancing high-level semantic representation capabilities. AMCF, on the other hand, achieves multi-level feature fusion by adaptively fusing the outputs of different classifiers, effectively avoiding information loss or interference that simple fusion strategies might cause. We evaluated the CFE-Net on two rubbing oracle bone characters benchmark datasets, OBC306 and Oracle-MNIST. The experimental results demonstrate that CFE-Net significantly outperforms several existing methods in terms of Top-1 accuracy, establishing it as the new state-of-the-art.

**Keywords:** Oracle Bone Characters · Character Image Recognition · Attention Mechanism · Feature Fusion

## 1 Introduction

The automatic recognition technology for oracle bone characters plays a vital role in the preservation and research of digital cultural heritage. Oracle bone



characters, which are an important symbol of ancient Chinese civilization, are mainly carved on tortoise shells or animal bones. Their contents include historical events, social customs, clan situations, and ritual activities. With the discovery of a large amount of oracle bone characters data and the increasing demand for its digitization, determining how to accurately and efficiently identify and interpret this complex textual information has become a major challenge in the field of oracle bone characters recognition.

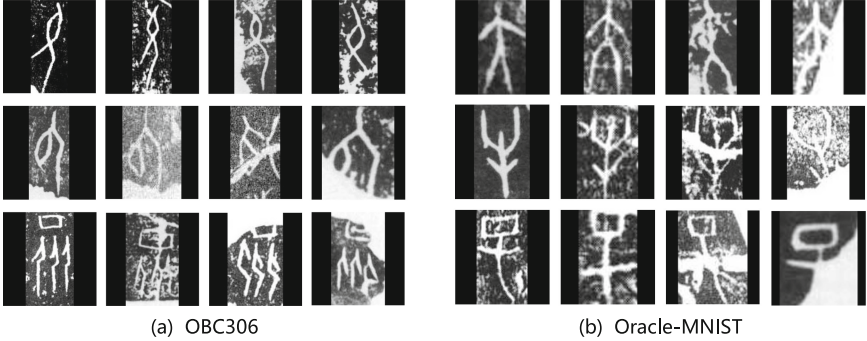
In the field of oracle bone characters recognition, early studies primarily utilized graph theory and topology to extract text features [14, 18, 26]. With technological advancements, deep learning methods began to be applied in this field [20, 25, 34].

Although deep learning methods have achieved significant results in oracle bone characters recognition, the inherent characteristics of rubbing oracle bone characters—complex background noise, irregular character morphology, and character mutilation due to erosion, as shown in Fig. 1—still pose two major challenges: 1) how to effectively extract character features from images with complex background noise; and 2) how to synthesize shallow detail information with deep semantic information of characters. Noted, unless otherwise stated, we simplify oracle bone characters recognition as oracle recognition.

To address the above challenges, we propose a Character Feature Enhancement Network, CFE-Net. Initially, to enhance the model’s ability to capture key character information and effectively suppress the background noise, we designed the Spatial Focus Attention Module (SFAM) and the Character Feature Enhancement (CFE) module. Furthermore, we constructed the Adaptive Multi-level Classifier Fusion (AMCF) module. This module utilizes global average pooling with various kernel sizes to process multi-level feature maps, thus effectively fusing the shallow detail information of characters with deep semantic information. Utilizing an adaptive dynamic weighting strategy, the module optimizes the classification results and more effectively adapts to the diversity of oracle bone characters. To summarize, the main contributions of this study are as follows:

- We propose a new oracle recognition network, CFE-Net, which utilizes the attention mechanism and feature fusion techniques to enhance the ability to capture key information in oracle images and effectively suppress background noise.
- We designed the Spatial Focus Attention Module (SFAM) and the Character Feature Enhancement (CFE) module. These modules enhance the model’s focus on foreground character features and effectively isolate key character information from complex backgrounds.
- We constructed an Adaptive Multi-level Classifier Fusion (AMCF) module. This module optimizes classification results through adaptive weight adjustment, enabling dynamic adjustments based on the hierarchical structure and importance of features, thus enhancing character recognition accuracy.
- We conducted experiments on the challenging rubbing oracle bone characters datasets OBC306 [12] and Oracle-MNIST [29]. The experimental results

demonstrate that the proposed model not only improves baseline performance but also establishes a new state-of-the-art.



**Fig. 1.** Examples of selected samples from the Oracle dataset. (a) OBC306; (b) Oracle-MNIST. In each subgraph, images in the same row belong to the same class

## 2 Related Works

### 2.1 Oracle Character Recognition

Research on oracle recognition has primarily utilized traditional pattern recognition and deep learning methods for both handwritten and rubbing oracle images. Traditional pattern recognition methods rely on graph theory and topology. For example, reference [14] employs a graph isomorphism-based approach, treating oracle characters as undirected graphs and characterizing their structure through nodes (e.g., endpoints and intersections) and edges to achieve matching and recognition. Another study [4] converts oracle characters into topological graphs for matching and recognition. Guo et al. [6] propose hierarchical representations combining Gabor filters and sparse encoder features to improve the efficiency of oracle characters recognition. Despite the initial success of these methods, they are complex and heavily rely on manual feature extraction, which limits their scalability.

In contrast, deep learning methods have greatly improved the performance of oracle characters recognition. Recent research has focused on utilizing Convolutional Neural Networks (CNNs) for automatic feature extraction [2, 30, 36]. Due to the difficulty in obtaining rubbing oracle datasets, early studies were usually limited to self-constructed handwritten or small collections of rubbing oracle characters images. It was not until Huang et al. released the first large-scale rubbing oracle dataset, OBC306 [12], which provided a valuable resource for deep learning research, that more researchers were prompted to turn to rubbing oracle recognition.

The OBC306 [12] dataset suffers from significant sample imbalance, with some categories being extremely rich in samples while others are sparse. To address this challenge, Li et al. [17] combined the mix-up enhancement strategy and triplet loss to increase the sample size of minority classes by combining the information from majority and minority classes, thus improving the accuracy of oracle recognition based on Inception-v4 [28]. Additionally, Li et al. [16] proposed two new data enhancement strategies: Repatch and TailMix. The Repatch strategy enhances the output diversity of the Generative Adversarial Network (GAN) [3] generator by generalizing the samples, while the TailMix strategy increases the number of samples in the tail class by synthesizing data from other classes, effectively improving the model’s recognition ability for minority class samples. Furthermore, Mao et al. [24] enhanced the fusion of shallow and deep features and optimized the classifier structure by improving the ResNeSt [35] network, further enhancing the recognition accuracy of rubbing oracle bone characters.

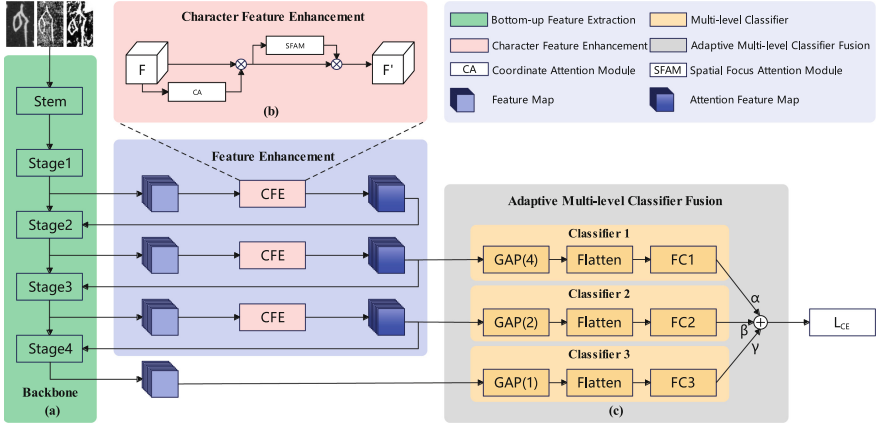
Although research based on ResNeSt [35] and Inception-v4 [28] has achieved significant results in oracle recognition accuracy, these models suffer from complex designs, a large number of parameters, and lengthy training times. To address these issues, this paper proposes achieving efficient and accurate oracle recognition using smaller models such as ResNet-18 [8], aiming to significantly reduce model complexity and training time.

## 2.2 Attention Mechanism

In the field of computer vision, attention mechanisms have become crucial techniques for improving the performance of Neural Networks (NN). These attention mechanisms are effectively integrated into Convolutional Neural Networks (CNNs), enhancing the model’s focus on key features and suppressing irrelevant information. Attention mechanisms can be primarily categorized into channel attention, spatial attention, and a combination of both.

The channel attention mechanism adjusts the feature response by learning the importance of each channel in the feature map, highlighting useful features and suppressing irrelevant information. SE-Net [11] is one of the earliest implementations. It focuses on the global features of channels through a global average pooling layer and employs a simple recalibration strategy to improve feature responses among channels. ECA-Net [31] avoids complex spatial dimensionality transformations and dimensionality reduction operations, using a one-dimensional convolution with adaptive kernel sizes to dynamically capture channel dependencies at different scales. The spatial attention mechanism concentrates on the spatial dimensions of the feature map, optimizing each location by learning the importance of specific areas. GENet [10] aggregates local feature information via the “Gather” operation and modulates the feature response with the “Excite” operation, effectively leveraging spatial attention to enrich the contextual information in the feature map. SPA-Net [5] employs multiple adaptive average pooling within a spatial pyramid structure to model both local

and global contextual semantic information, thereby more fully exploiting spatial semantic information. Considering the importance of channel and spatial dimensions simultaneously, methods like CBAM [33] that combine channel and spatial attention enhance feature representation by sequentially applying these two types of attention. Coordinate attention [9] further innovates by integrating position coding within channel attention, merging spatial position information with channel features to enhance the model’s ability to discern various positions in the image.



**Fig. 2.** Overall architecture. (a) Backbone; (b) Character Feature Enhancement module; (c) Adaptive Multi-level Classifier Fusion module.  $GAP(\hat{u})$  denotes global average pooling with different kernel sizes.  $\alpha$ ,  $\beta$  and  $\gamma$  are three trainable parameters. FC represents fully connected layer.  $L_{CE}$  denotes the cross-entropy loss function

The recognition of complex-morphology oracle bone characters is significantly impacted by background noise and diverse character shapes. We designed a spatial focus attention module that enhances foreground character feature extraction and suppresses background noise.

### 2.3 Multi-level Feature Fusion

Multi-level feature fusion is crucial for visual tasks. Each layer of the backbone network has a distinct receptive field. However, the lack of feature correlation means that information from different levels is not fully utilized, limiting image recognition accuracy. To address this challenge, multi-level feature fusion enhances information utilization efficiency by integrating features from different layers. As a classic example, the Feature Pyramid Network (FPN) [19] enhances multi-level feature utilization by combining low-resolution, high-semantic features with high-resolution, low-semantic features in a structure that relays semantic information from top to bottom, achieving effective feature fusion

across levels. Building on this, PANet [21] further emphasizes effective information exchange between lower and higher layer features by adding an additional bottom-up path to aggregate multi-path, multi-level feature information.

Although FPN [19] and its variants have achieved significant results in image classification and object detection, they exhibit limitations when addressing detail-rich character images against complex backgrounds. The primary challenge with these images lies in the fineness and criticality of details, particularly at character edges and stroke details.

To alleviate this problem, this study proposes a strategy focusing on multi-level feature fusion from the classifier’s perspective. This approach not only minimizes information loss or interference from directly fusing different-level features, but also more effectively utilizes deep semantic information while preserving original detail.

### 3 Proposed Approach

In this section, we present a comprehensive overview of the proposed model. Section 3.1 outlines the overall architecture of the model, Sect. 3.2 designs the character feature enhancement module, and Sect. 3.3 constructs the adaptive multi-level classifier fusion module.

#### 3.1 Overall Architecture

In this study, we propose a Character Feature Enhancement Network (CFE-Net), focusing on attention feature enhancement and multi-level feature fusion to improve oracle recognition accuracy.

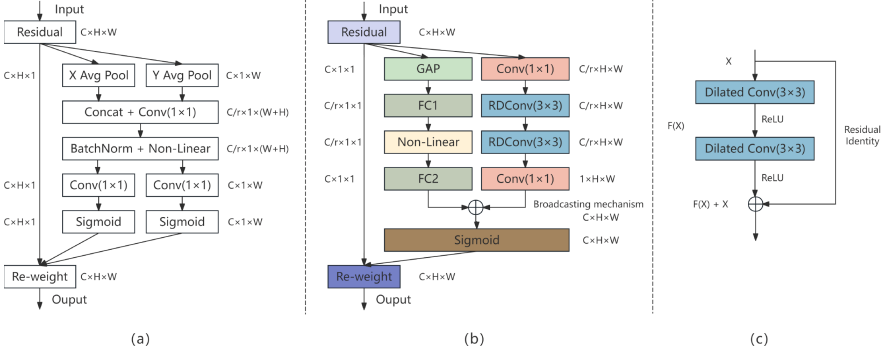
The overall framework, illustrated in Fig. 2, includes the backbone network, the Character Feature Enhancement (CFE) module, and the Adaptive Multi-level Classifier Fusion (AMCF) module. Initially, the backbone network extracts features from the input image. Subsequently, the CFE module emphasizes key features and suppresses irrelevant ones. Finally, the AMCF module fuses features of different levels to optimize recognition results.

#### 3.2 Character Feature Enhancement

Oracle images, which contain complex background noise mixed with characters, significantly impact recognition accuracy. To address this challenge, we designed the Character Feature Enhancement (CFE) module, as shown in Fig. 2(b), aiming to emphasize character features and suppress background noise.

Oracle bone characters are typically elongated, with distinct information along the width and height dimensions. To more effectively focus on information in different spatial dimensions, we introduce Coordinate Attention (CA) [9], illustrated in Fig. 3(a). This module encodes spatial features of the character image along width and height, embedding these features into the channel to enhance the model’s accuracy in locating and focusing on effective features. Specifically,

this module generates two direction-aware feature maps via two one-dimensional global average pooling operations, aggregated horizontally and vertically. These two direction-specific feature maps are then encoded into two attention maps, each capturing long-range dependencies of the input features in their respective directions. Through a multiplication operation, these attention maps enhance important features and suppress unimportant ones in the input feature maps.



**Fig. 3.** From left to right, they are: (a) CA module; (b) SFAM module; (c) Residual Dilated Convolution block

To further enhance the foreground features of character images and suppress background noise, we designed a Spatial Focus Attention Module (SFAM) to intensify spatial feature focus, inspired by CBAM [33]. The SFAM module comprises two branches: channel attention and spatial attention, with its detailed structure illustrated in Fig. 3(b). For a given input feature map  $F \in \mathbb{R}^{C \times H \times W}$ , the module independently computes channel attention  $M_c(F) \in \mathbb{R}^C$  and spatial attention  $M_s(F) \in \mathbb{R}^{H \times W}$ , subsequently fusing these branches via element-wise summation to generate a 3D attention map  $M(F)$ , ranging from (0, 1), using a sigmoid activation function (as shown in Eq. 1). Given the differing dimensions of the two attention maps, they are expanded to  $\mathbb{R}^{C \times H \times W}$  using a broadcasting mechanism prior to fusion. Finally, the resulting 3D attention map is multiplied element-wise with the input feature map  $F$  to produce the enhanced feature map  $F'$  (as shown in Eq. 2).

$$M(F) = \sigma(Mc(F) + Ms(F)) \quad (1)$$

$$F' = F \otimes M(F) \quad (2)$$

where  $\otimes$  denotes element-by-element multiplication,  $\sigma$  represents the sigmoid activation function.

Specifically, the channel attention branch first applies global average pooling to the feature map  $F \in \mathbb{R}^{C \times H \times W}$ , aggregating channel features and generating a channel vector  $F_c \in \mathbb{R}^{C \times 1 \times 1}$ . Subsequently, downscaling and upscaling operations are conducted via two Fully Connected (FC) layers, introducing a

Non-Linear process to estimate inter-channel attention. The hidden layer size is set to  $\mathbb{R}^{C/r \times 1 \times 1}$ , where  $r$ , the reduction ratio, is 16. Within the spatial attention branch, the channel dimension of the feature map  $F$  is initially reduced to  $\mathbb{R}^{C/r \times H \times W}$  using  $1 \times 1$  convolution to decrease computational overhead. Subsequently, two Residual Dilated Convolution (RDConv, illustrated in Fig. 3(c)) blocks (convolution kernel of 3 and dilation rate of 4) are sequentially employed to expand the receptive field and further focus spatial features. Residual dilated convolution expands the convolution kernel’s receptive field by introducing dilations, enhancing the capture of detailed character stroke features while preserving feature map resolution. Residual concatenation facilitates smooth information transfer, preventing information loss and gradient vanishing issues. The number of channels in the feature map is further reduced using  $1 \times 1$  convolution, producing an  $\mathbb{R}^{1 \times H \times W}$  spatial attention output. Finally, the attention maps from both branches are summed and re-weighted with the input feature map post-sigmoid activation to produce the final attention feature map. This strategy, which sums attention maps followed by activation, combines different attention strengths, optimizing the re-weighting process and enhancing overall model performance.

### 3.3 Adaptive Multi-level Classifier Fusion

As the depth of CNNs increases, the resolution of the feature map gradually decreases. Although low-level features are high in resolution and rich in detail, they are less semantic and contain more noise. In contrast, high-level features, rich in semantic information, have lower resolution and capture fewer details. For character images, the details between strokes are crucial for distinguishing different characters. Therefore, effectively fusing low-level detail with high-level semantic information presents a major challenge in oracle recognition.

To address this challenge, we developed an Adaptive Multi-level Classifier Fusion (AMCF) module, illustrated in Fig. 2(c). This module facilitates the fusion of multi-level features at the classifier level. Specifically, we conduct global average pooling (GAP) with various kernel sizes-GAP(4), GAP(2), and GAP(1)-on the output feature maps from stages 2, 3 and 4 of the backbone network to derive three feature maps, each capturing detail and semantic information at distinct levels. These feature maps are transformed into feature vectors along the channel dimension and then input into three distinct classifiers: Classifier1, Classifier2, and Classifier3. Variations in global average pooling dimensions yield feature vectors of differing dimensions. Consequently, we designed a unique classifier structure for each level. Each classifier maps feature vectors of 2048, 1024, and 512 dimensions to corresponding category counts. To achieve optimal feature fusion, we introduce three trainable parameters  $\alpha$ ,  $\beta$  and  $\gamma$ , and employ an adaptive weighting strategy to combine the outputs of different classifiers, namely score1, score2, and score3 (as shown in Eq. 3).

$$Fused\_score = \alpha \times score1 + \beta \times score2 + \gamma \times score3 \quad (3)$$

This multi-level feature fusion strategy, viewed from the perspective of classifier integration, offers several advantages. First, processing features through independent classifiers at various levels prevents information loss and reduces interference from simple fusion, enabling more effective integration of detailed and semantic information. Second, the independent classifier design diminishes the model’s reliance on a single feature level, thereby enhancing noise robustness. Additionally, the adaptive fusion mechanism dynamically adjusts classifier weights to optimize feature information utilization across various input scenarios. The strategy’s flexibility allows the model to handle complex and diverse character images more robustly.

## 4 Experiments

### 4.1 Datasets

To the best of our knowledge, the publicly available rubbing oracle recognition datasets include OBC306 [12] and Oracle-MNIST [29]. We conducted tests on these two rubbing image benchmark datasets to evaluate the effectiveness of our proposed network. Samples from these datasets are shown in Fig. 1, and relevant dataset information is listed in Table 1.

**Table 1.** Oracle recognition dataset OBC306, Oracle-MNIST related information. IR represents imbalance ratio, which denotes the ratio between the largest and smallest class sizes in terms of sample quantities.

Data set	Train set	Test set	Train IR	Test IR	Total number	Class Number
OBC306	232,236	77,286	19,424:1	6,474:1	309,522	277
Oracle-MNIST	27,222	3,000	3,399:2,328	300:300	30,222	10

**OBC306** [12] is currently the largest rubbing oracle bone characters recognition dataset. This dataset comprises 306 classes of oracle characters, totaling 309,551 images. Given the dataset’s long-tailed distribution, to ensure at least one sample from each class in both training and test sets, we removed 29 classes that had only one image, retaining 277 classes totaling 309,522 images. Following the work of [12], we randomly divided the dataset into training and test sets at a 3:1 ratio. Ultimately, the training set includes 232,236 images, and the test set includes 77,286 images, with imbalance ratios of 19,424:1 and 6,474:1, respectively. This alignment is consistent with the ratios reported in the work [15].

**Oracle-MNIST** [29], modeled after the classic MNIST [13] dataset for image classification, was derived from OBC306 by Wang et al. This dataset comprises 10 classes of oracle bone characters totaling 30,222 images, with 27,222 images in the training set and 3,000 images in the test set. Unlike OBC306 [12], this dataset exhibits a relatively balanced distribution of images across each category, with imbalance ratios of 3,399:2,328 in the training set and an equal distribution of 300:300 in the test set.



## 4.2 Evaluation Metrics

In the experiments, this study utilized total accuracy to evaluate the model’s performance, reporting overall accuracy across all categories. This metric is commonly used in the field of image classification and is defined by Eq. 4.

$$A_{\text{Total}} = \frac{1}{H} \sum_{c=1}^C n_c \quad (4)$$

where H denotes the total number of images in the test set, C represents the number of classes in the test set, and  $n_c$  indicates the number of images of class C that were correctly classified by the model. Higher values of accuracy indicate better model performance.

## 4.3 Implementation Details

The proposed method underwent end-to-end training on the PyTorch deep learning framework. In the experiments, input images were uniformly scaled to  $224 \times 224$ , and the batch size was set at 64. Optimization was performed using the AdamW [22] optimizer with a weight decay of  $5e-4$  and an initial learning rate of  $5e-4$ . Initially, training involved a CosineAnnealing [23] decay strategy for 20 epochs, followed by fixing the learning rate at  $1e-6$  for an additional 5 epochs. We employed a cross-entropy loss function to quantify the discrepancies between predictions and actual labels. Additionally, random horizontal flipping and random rotation were implemented to enhance sample diversity. All experiments were conducted on a single GeForce RTX 3090 GPU.

## 4.4 Ablation Study

To validate the effectiveness of the various components of our model, we selected ResNet-18 as the baseline model for ablation studies on the OBC306 dataset. The evaluation criteria included Top-1 and Top-3 accuracy, with detailed results presented in Table 2.

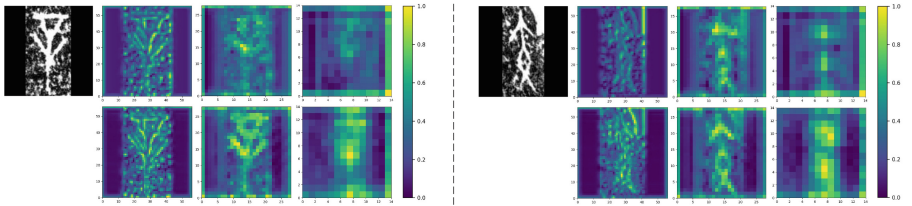
**Table 2.** Ablation experiments on the OBC306 dataset.

Method	Top-1 Accuracy (%)	Top-3 Accuracy (%)
Baseline	91.50	97.01
Baseline+CFE	93.46	97.83
Baseline+AMCF	93.24	97.63
Baseline+CFE+AMCF	94.22	98.07

**Effectiveness of the CFE.** Results in Table 2 demonstrate significant performance improvements attributed to the CFE module. The addition of the CFE

module improves the Top-1 and Top-3 accuracy to 93.46% and 97.83%. This improvement primarily results from effectively combining CA and SFAM within CFE. Specifically, CA captures information from various spatial orientations in the oracle bone character image, while SFAM intensifies spatial feature focus, effectively suppressing background noise and enhancing foreground feature characterization.

Additionally, feature map visualizations for the three stages of CFE implementation are displayed in Fig. 4. With the CFE module integrated, attention to character features is significantly enhanced in the model. Attention is particularly focused on the main parts of the character, underscoring the model’s focus on key features.



**Fig. 4.** Visualization of the feature maps. The first row shows the feature maps generated by the baseline model, while the second row illustrates the feature maps generated after adding the CFE module. Yellow areas indicate higher attentional intensity and blue or black indicate lower attentional intensity (Color figure online).

**Effectiveness of the AMCF.** Similarly, the AMCF module significantly improved model performance. The addition of the AMCF module increased the baseline model’s Top-1 and Top-3 accuracy to 93.24% and 97.63%. This enhancement is attributed to AMCF’s optimization of overall feature representation by effectively integrating the strengths of each classifier and fully utilizing multi-level feature information. Specifically, AMCF trains by feeding varying levels of output feature maps to separate classifiers and fusing the outputs in an adaptively weighted manner. This strategic approach enables each classifier to focus on features received at a specific level, enhancing information accuracy and improving image recognition performance.

#### 4.5 Comparison with Previous Methods

To further validate the state-of-the-art status of the proposed method, we compared it with existing work in the field of rubbing oracle recognition. Experimental results comparing the proposed method with others on OBC306 and Oracle-MNIST are presented in Tables 3 and 4, respectively. The symbol \* denotes the benchmark results for the different models from our own implementations.

As demonstrated by the results in Tables 3 and 4, our proposed CFE-Net exhibits excellent performance on the challenging datasets OBC306 and Oracle-MNIST. On the OBC306 dataset, CFE-Net achieved a Top-1 accuracy of 94.22%,

**Table 3.** Experimental results on the OBC306 dataset.

Method	Backbone	Top-1 Accuracy (%)	Top-3 Accuracy (%)
Simonyan et al. [27]*	VGG16	90.95	96.38
He et al. [8]*	ResNet-18	91.50	97.01
Szegedy et al. [28]*	Inception-v4	92.68	97.49
Guo et al. [7]	Inception-v3	87.73	94.85
Liu et al. [1]	ResNet-18	91.53	-
Li et al. [17]	Inception-v4	91.74	-
Wang et al. [32]	Inception-v4	92.02	-
Mao et al. [24]	ResNeSt	93.53	-
Li et al. [16]	Inception-v4	93.86	-
<b>Ours</b>	ResNet-18	<b>94.22</b>	<b>98.07</b>

**Table 4.** Experimental results on the Oracle-MNIST dataset.

Method	Backbone	Top-1 Accuracy (%)	Top-3 Accuracy (%)
Wang et al. [29]	CNN	93.80	-
Simonyan et al. [27]*	VGG16	96.63	99.67
He et al. [8]*	ResNet-18	96.57	99.70
Szegedy et al. [28]*	Inception-v4	97.27	<b>99.73</b>
Zhang et al. [35]*	ResNeSt-50	97.60	99.70
<b>Ours</b>	ResNet-18	<b>98.50</b>	99.70

significantly outperforming other listed models. Compared to the ResNet-18 baseline model’s 91.50%, CFE-Net showed a 2.72% improvement in Top-1 accuracy and achieved 98.07% in Top-3 accuracy. Even compared to more complex architectures like Inception-v4 and ResNeSt-50, our model remains competitive, surpassing the best current method by Li et al. [16] (93.86%) by 0.36%. On the Oracle-MNIST dataset, CFE-Net also excelled, achieving a Top-1 accuracy of 98.50%, and 4.7% higher than the best current method by Wang et al. [29] (93.80%). These results demonstrate that CFE-Net not only excels in Top-1 accuracy but also maintains superior performance in Top-3 accuracy, effectively handling the task of oracle recognition.

#### 4.6 Comparing the Advantages of Our Model

In this subsection, we provide a detailed comparison of the Top five models-VGG16, ResNet-18, ResNeSt-50, Inception-v4, and our proposed CFE-Net-focusing on recognition accuracy on the test set of OBC306, with results presented in Table 5. Among them, the accuracy rate for ResNet-18, Inception-v4, and ResNeSt-50 reflects the findings from [1], [16] and [24], while the accuracy rates for the remaining models are derived from our own implementations.

**Table 5.** Comparison of different models in terms of Parameters, FLOPs, Weight file size, and Top-1 accuracy.

Model	Params (M)	FLOPs (G)	Weight Size (MB)	Top-1 Accuracy (%)
VGG16 [27]	138.36	15.47	516.50	90.95
ResNet-18 [8]	11.69	1.82	43.25	91.53
ResNeSt-50 [35]	27.48	5.43	99.59	93.53
Inception-v4 [28]	42.68	6.16	164.98	93.86
<b>CFE-Net (Ours)</b>	12.19	2.19	46.57	<b>94.22</b>

Through detailed comparative analysis, it can be clearly seen that CFE-Net offers significant advantages in several aspects. First, CFE-Net has only 12.19M parameters, nearly 91% fewer than VGG16, which has the largest parameter count at 138.36M. Additionally, CFE-Net features 2.19G FLOPs and a 46.57MB weight file, achieving over 70% reduction in parameter quantity, approximately 64% reduction in FLOPs, and nearly 72% reduction in weight file size compared to the high-performing Inception-v4 model. Most importantly, with a Top-1 accuracy of 94.22%, CFE-Net outperforms all current models in terms of performance. In summary, CFE-Net not only offers the best recognition performance but also maintains low resource consumption.

## 5 Conclusion

This study proposes a novel deep learning model, the Character Feature Enhancement Network (CFE-Net), specifically designed to address the challenges of automatic oracle recognition. CFE-Net enhances the recognition accuracy of rubbing oracle bone characters through two innovative modules: Character Feature Enhancement (CFE) and Adaptive Multi-level Classifier Fusion (AMCF). The CFE module significantly enhances the representation of semantic information through the Spatial Focus Attention Module (SFAM), which emphasizes foreground character features against complex backgrounds. Meanwhile, the AMCF module optimizes feature integration across different levels, reducing information loss and suppressing noise interference. Evaluation results on two rubbing oracle recognition datasets, OBC306 and Oracle-MNIST, demonstrate CFE-Net’s effectiveness and superiority in oracle recognition tasks.

**Acknowledgements.** This work was supported by the National Natural Science Foundation of China (NO.62266044, 62061045). It was also supported by the “Tianshan Talents” Leading Talents Program for Scientific and Technological Innovation in Xinjiang Uygur Autonomous Region (2023TSYCLJ0025), and the Open Project of Key Laboratory of Oracle Bone Inscription Information Processing, Ministry of Education (OIP2021E004).

## References

1. Dazheng, L.: Random polygon cover for oracle bone character recognition. In: Proceedings of the 2021 5th International Conference on Computer Science and Artificial Intelligence, pp. 138–142 (2021)
2. Fujikawa, Y., Li, H., Yue, X., Aravinda, C., Prabhu, G.A., Meng, L.: Recognition of oracle bone inscriptions by using two deep learning models. *Int. J. Digit. Hum.* **5**(2), 65–79 (2023)
3. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems **27** (2014)
4. Gu, S.: Oracle bone character recognition method based on topological registration. *Comput. Digit. Eng.* **44**(10), 2001–2006 (2016)
5. Guo, J., et al.: SPANet: spatial pyramid attention network for enhanced image recognition. In: 2020 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2020)
6. Guo, Z., Xu, H., Lu, F., Wang, Q., Zhou, X., Shi, Y.: Improving irregular text recognition by integrating Gabor convolutional network. In: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), pp. 286–293. IEEE (2019)
7. Guo, Z., et al.: An improved neural network model based on inception-v3 for oracle bone inscription character recognition. *Sci. Program.* **2022**(1), 7490363 (2022)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
9. Hou, Q., Zhou, D., Feng, J.: Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13713–13722 (2021)
10. Hu, J., Shen, L., Albanie, S., Sun, G., Vedaldi, A.: Gather-excite: exploiting feature context in convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
12. Huang, S., Wang, H., Liu, Y., Shi, X., Jin, L.: OBC306: a large-scale oracle bone character recognition dataset. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 681–688. IEEE (2019)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
14. Li, F., Zhou, X.: Graph theoretical methods for automatic recognition of oracle bones. *J. Electron. Inf.* **22**(S1), 41–47 (1996)
15. Li, J., Dong, B., Wang, Q.-F., Ding, L., Zhang, R., Huang, K.: Decoupled learning for long-tailed oracle character recognition. In: Fink, G.A., Jain, R., Kise, K., Zanibbi, R. (eds.) Document Analysis and Recognition - ICDAR 2023: 17th International Conference, San José, CA, USA, August 21–26, 2023, Proceedings, Part IV, pp. 165–181. Springer Nature Switzerland, Cham (2023). [https://doi.org/10.1007/978-3-031-41685-9\\_11](https://doi.org/10.1007/978-3-031-41685-9_11)
16. Li, J., Wang, Q.F., Huang, K., Yang, X., Zhang, R., Goulermas, J.Y.: Towards better long-tailed oracle character recognition with adversarial data augmentation. *Pattern Recogn.* **140**, 109534 (2023)

17. Li, J., Wang, Q.-F., Zhang, R., Huang, K.: Mix-up augmentation for oracle character recognition with imbalanced data distribution. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12821, pp. 237–251. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-86549-8\\_16](https://doi.org/10.1007/978-3-030-86549-8_16)
18. Li, Q., Yang, Y., Wang, A.: Recognition of inscriptions on bones or tortoise shells based on graph isomorphism. *Jisuanji Gongcheng yu Yingyong (Computer Engineering and Applications)* **47**(8), 112–114 (2011)
19. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
20. Liu, M., Liu, G., Liu, Y., Jiao, Q.: Oracle bone inscriptions recognition based on deep convolutional neural network. *J. Image Graph.* **8**(4), 114–119 (2020)
21. 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)
22. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101) (2017)
23. Loshchilov, I., Hutter, F.: SGDR: stochastic gradient descent with warm restarts. In: ICLR (International Conference on Learning Representations) (2017)
24. Mao, Y., Bi, X.: Rubbing oracle bone character recognition based on improved ResNeSt network. *J. Intell. Syst.* **18**(3), 450–458 (2023)
25. Meng, L., Kamitoku, N., Yamazaki, K.: Recognition of oracle bone inscriptions using deep learning based on data augmentation. In: 2018 Metrology for Archaeology and Cultural Heritage (MetroArchaeo), pp. 33–38. IEEE (2018)
26. Qu, H., Liu, J., Wu, J.: Oracle bone recognition based on topological features. *Comput. Sci. Appl.* **9**, 1111 (2019)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
28. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-ResNet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
29. Wang, M., Deng, W.: A dataset of oracle characters for benchmarking machine learning algorithms. *Sci. Data* **11**(1), 87 (2024)
30. Wang, M., Deng, W., Liu, C.L.: Unsupervised structure-texture separation network for oracle character recognition. *IEEE Trans. Image Process.* **31**, 3137–3150 (2022)
31. Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: ECA-Net: efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11534–11542 (2020)
32. Wang, W., Zhang, T., Zhao, Y., Jin, X., Mouchere, H., Yu, X.: Improving oracle bone characters recognition via a CycleGAN-based data augmentation method. In: International Conference on Neural Information Processing, pp. 88–100. Springer (2022). [https://doi.org/10.1007/978-981-99-1645-0\\_8](https://doi.org/10.1007/978-981-99-1645-0_8)
33. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
34. Yang, Z., Wang, Q., He, X., Liu, Y., Yang, F., Yin, Z., Yao, C.: Accurate oracle classification based on deep convolutional neural network. In: 2018 IEEE 18th International Conference on Communication Technology (ICCT), pp. 1188–1191. IEEE (2018)

35. Zhang, H., et al.: ResNeSt: split-attention networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2736–2746 (2022)
36. Zhao, X., Liu, S., Wang, Y., Fu, Y.: FFD Augmentor: towards few-shot oracle character recognition from scratch. In: Proceedings of the Asian Conference on Computer Vision, pp. 1622–1639 (2022)



# DocHFormer: Document Image Dewarping via Harmonized Modeling of Hierarchical Priors

Xinyue Zhou<sup>1,2</sup>, Guanting Li<sup>1,2</sup>, Nanfeng Jiang<sup>1,2(✉)</sup>, Da-Han Wang<sup>1,2</sup>,  
Xu-Yao Zhang<sup>3</sup>, and ShunZhi Zhu<sup>1,2</sup>

<sup>1</sup> School of Computer and Information Engineering, Xiamen University of  
Technology, Xiamen 361024, China

2322071057@stu.xmut.edu.cn, qianxule007@gmail.com,  
{2023000066,wangdh,szzhu}@xmut.edu.cn

<sup>2</sup> Fujian Key Laboratory of Pattern Recognition and Image Understanding,  
Xiamen 361024, China

<sup>3</sup> State Key Laboratory of Multimodal Artificial Intelligence Systems Institute of  
Automation of Chinese Academy of Sciences, Beijing, China

xyz@nlpr.ia.ac.cn

**Abstract.** Document Image Dewarping (DID) task aims to address the issue of geometry distortion and improve image quality. In this paper, we propose a simple but effective method, named DocHFormer, that can take hierarchical priors features of images, including document image mask and coordinate positions, as additional information to realize accurate representation. To better exploit these fused information for dewarping, we take them into a harmonized space random shuffle operation, which can stochastically rearrange the pixels across spatial space and further use inverse operation to recover the original order. This way can adapt to allocate each feature pixel with equal probability and thus make full use of multi-type features. Furthermore, we introduce this mechanism into local self-attention to use linear complexity to input resolution and also design a new feed-forward network with structural modeling to boost representation. With the help of the above components, our proposed DocHFormer can achieve competitive performance with lower complexity and also outperform the existing state-of-the-art on several popular datasets.

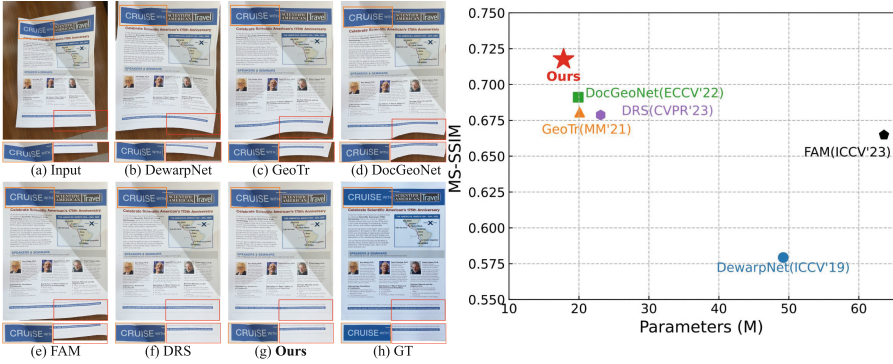
**Keywords:** Document Image Dewarping · Harmonized Space

## 1 Introduction

With the advance of smartphones and thanks to their convenience, people usually use ubiquitous smartphones, equipped with high-quality cameras, to capture photos of documents for archiving and retrieving information. Unlike the controllable operating environment of the scanner, such documents in these photos often suffer from unwanted geometric distortion under the various physical



deformation, *e.g.*, folded, curved and crumpled. These distortions will lead to poor document readability and the deteriorated performance of OCR systems, such as text edit and character recognition. Therefore, it is of great value to design an effective Document Image Dewarping (DID) method to eliminate geometric distortions for real-world applications.



**Fig. 1.** Left: visual results of our proposed and several state-of-the-art methods. Right: the performance-parameters results.

DID tasks have been investigated extensively during the past decades. In traditional methods, several researchers dewarp images by reconstructing 3D document shapes [1–3] or assume parametric models to exploit specific priors on the 2D document images [4, 5]. However, these methods require high-cost hardware equipment or complex optimization process of model parameters, which limit the potential values of applications and are not sufficiently robust for real-world scenes. Recently, with the developments of deep learning technologies, Convolutional Neural Network (CNN) and Transformer have been introduced to DID task and obtained promising performance [6–12]. In CNN-based categories, the works of [6, 7] build the UNet-shape structures to regress a pixel-wise displacement field to dewarp the distorted images. However, it is difficult for CNN to effectively capture the long-range relationship to model the deformation of images with relatively low complexity, as shown in Fig. 1. As a result, the dewarped images still exist curved textline regions. These phenomena also can be observed in Fig. 1. To solve these issues, some researchers [9, 11, 12] introduce the Transformer-based Encoder-Decoder structures to model geometric regions and dewarp the distorted images with mask prior and foreground information. However, these methods do not make full use of another key prior, such as coordinates, to ensure the image integrity with position information. Moreover, common distortions, such as sheet deformation, illumination details and color-cast images, have non-uniform and diverse distributions, which are not addressed well by existing approaches. This is because they ignore inherent weight-sharing property and mainly adopt fixed parameters to learn distortion representation. Therefore, from the results of Fig. 1, the deformation issues with amplified halo noise

can not be effectively addressed. Moreover, the large model complexities of most methods need to be considered carefully. Based on these analyses, we can conclude the main challenges: *How to effectively use different priors to boost model representation and exploit them to improve model generalization ability under the low complexity condition?*

To tackle the aforementioned issue, we propose a simple but effective framework (DocHFormer) that is capable of jointly using explicit modeling and harmonized representation mechanisms to achieve high-quality dewarping. To be specific, first, the distorted images refer to local position deformation and global integrity disruption, which can be represented by image coordinates and masks, respectively. Hence, different existing methods, our proposed DocHFormer suggests to fuse mask and coordinate these two priors with original input to help extract global and local image properties. Second, we adopt a two-step strategy that divides the feature representation into two parts, namely feature transformation and feature reconstruction. Among them, feature transformation adopts shuffle operations to transform the previous fused features into a harmonized space. This way can adapt to allocate each feature pixel with equal probability. Then, the reconstruction uses the inverse operation to reconstruct the image semantics by using the relative position of pixels. With their cooperation, the image content information can be preserved well without key content loss and modeled under low complexity. Finally, we introduce this shuffle mechanism into Local-window Self-Attention (LSA) to boost the representation of complex distorted images. In addition, we also design a structural information-based Feed Forward Network (FFN) to model structure-based interactions in dewarping process. In summary, the main contributions can be included as follows:

- We consider the main properties of distorted images and fuse various priors as auxiliary information into the modeling pipeline to boost representation ability. Specifically, we provide a novel two-step way that can adaptively harmonize the fused features while avoiding learning ambiguity.
- We design the novel random shuffle and inverse shuffle operations that can help the model effectively realize feature harmonization without content loss. Moreover, we introduce this mechanism into local window self-attention to build the basic unit of our proposed framework.
- Our proposed method has strong robustness and high generalization ability on complex distorted images. Extensive and comprehensive experiments conducted on popular datasets demonstrate the superiority of our proposed method over the state-of-the-arts.

## 2 Related Work

### 2.1 Traditional Methods

Traditional DID methods are mainly addressed by 3D reconstruction techniques. Typically, these methods estimate the 3D meshes of the distorted documents and then attempt to restore them to their flat states. These methods achieve

high-quality 3D reconstruction using differentiable rendering with various 3D representations. However, the majority of these approaches require the use of additional hardware [5, 13, 14] or rely on images captured from multiple viewpoints [2, 15, 16], which can be impractical for individual users. Other techniques propose a parametric model for the document’s surface and refine this model by identifying distinct features such as edges [3] and texture patterns [17]. Nevertheless, these simplified models often result in suboptimal performance, and the refinement process entails a significant computational expense.

## 2.2 Deep Learning-Based Methods

In recent years, many works used deep learning to rectify distorted document images. Ma *et al.* [6] firstly stacked the UNet structure to predict the deformation field for each pixel in the warped document images. Li *et al.* [18] introduced a two-step process to rectify and stitch distorted images. Xie *et al.* [8] integrated a smoothing constraint into the learning algorithm to refine the pixel displacement field. Amir *et al.* [19] focused on learning the orientation of text within documents. Das *et al.* [7] proposed a novel method to model the three-dimensional structure of documents using a U-shape structure. Feng *et al.* [12] used a transformer-based network to improve the feature representation in dewarping. Das *et al.* [7] proposed to predict local deformation fields and then integrated them with global context to achieve a superior unwarping result. Li *et al.* [11] combined foreground with text line information to guide the model to focus on the global and local features of the distorted paper. Zhang *et al.* [20] explored polar coordinates to represent document contour. Yu *et al.* [21] utilized an attention-enhanced control point module to better capture local deformations. Kumari *et al.* [22] introduced an innovation network for rectifying unconstrained document images from a single input image based on transfer learning. Verhoeven *et al.* [23] proposed a novel method for grid-based single-image document unwarping. Feng *et al.* [24] proposed a groundbreaking unified framework for document image rectification, without any restrictions on the input distorted images. While these methods can effectively achieve commendable performance, they need to train complicated models and can not generalize well on various images under unrestricted conditions.

## 3 Methodology

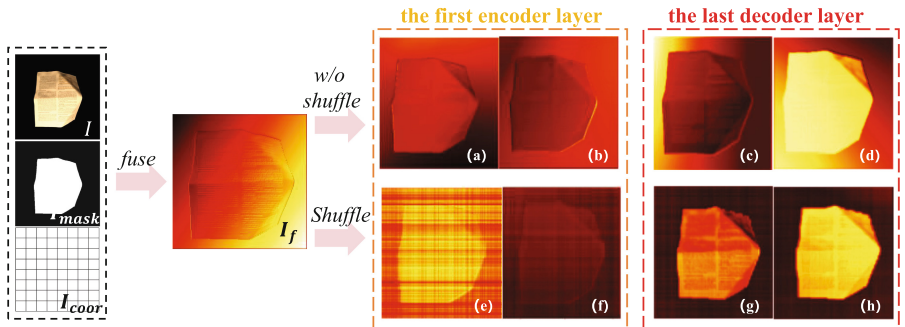
### 3.1 Motivation

Our proposed DocHFormer has strong robustness and can generalize well on complex unrestricted distorted images. This goal is achieved by the cooperation of multi-priors guidance, harmonization and structure modeling. The motivation is described as follows: first, we need to analyze the main properties of distorted images. In general, the noises of unwarped images contain different spatial patterns at global, regional, and local distributions. As we can observe from

some samples in Fig. 1, shadows or illumination issues usually exist in almost all deformed surfaces of documents. While boundary and curve distortions mainly refer to geometric deformation, they should be considered in a regional range, *e.g.*, the regions of coordinate positions. Besides, some tiny details, such as the blueness of textline, are easy to ignore in local areas. In contrast to existing deep models that solely design end-to-end dewarping networks, our framework extracts different spatial features, including mask  $I_{mask}$  and coordinate  $I_{coord}$ , and fuses  $fuse(\cdot)$  them with  $I$  to guide model produce dewarped image  $I_d$ . This process can be formed as follows:

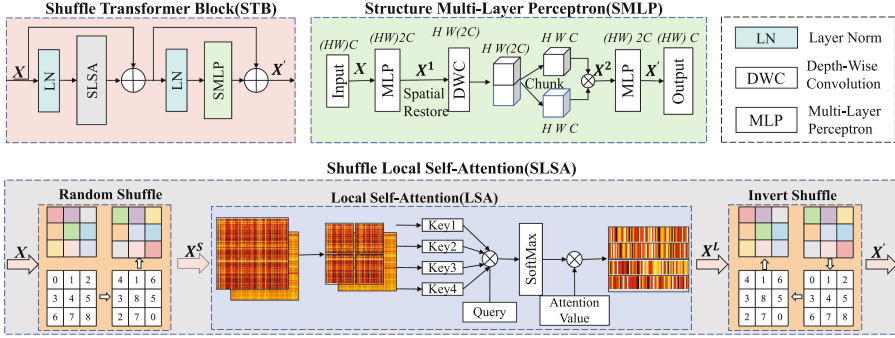
$$I_f = Net(fuse(I, I_{mask}, I_{coord}), \phi), \quad (1)$$

where  $Net(\cdot)$  denotes the designed network, *e.g.*, CNN-based or Transformer-based architecture, that learns image representation with suitable parameters  $\phi$ . However, this way easily produces learning ambiguity issues due to the weight-sharing property. With this, we give the second motivation: the desired model needs to harmonize the fused features and further conduct explicit modeling in training processing.



**Fig. 2.** Visualization of comparison results from the first encoder layer and last decoder layer. The (a) - (d) illustrate the features from the based-transformer without random shuffle, while the (e) - (h) correspond to the features processed by random shuffle. As we can observe the proposed DocHFormer can generate more rich features with details textures.

Second, it is difficult for dewarping process to adaptively focus each pixel since different spatial patterns refer to unique positions. Our motivation is to design a harmonized space that can exchange information between channel and space without disrupting spatial distributions. Figure 2 illustrates the basic modeling pipeline. We attempt to make each pixel of input feature  $X$  can obtain an equal probability through a two-step way, which consists of random shuffle  $S(\cdot)$  and invert shuffle  $I_S(\cdot)$  operations.  $S(\cdot)$  stochastically permutes the elements of input while  $I_S(\cdot)$ . To better avoid the bias from the stochasticity of random shuffle, we make full use of equivariant reordering of Local Self-Attention (LSA).



**Fig. 3.** The Shuffle Transformer Block (STB), which mainly contains Structure Multi-Layer Perception (SMLP) and Shuffle Local Self-Attention (SLSA).

The overall architecture can be seen in Fig. 3. The operation  $LSA(\cdot)$  is embedded between  $S(\cdot)$  and  $I_S(\cdot)$ . This process can be named Shuffle Local Self-Attention (SLSA), which can eliminate the position-based information constraint to the model with the weight-sharing property. SLSA can be step-by-step expressed as follows:

$$\begin{aligned}
 X^S &= S(X), \\
 X^L &= LSA(X^S), \\
 X' &= I_S(X^L),
 \end{aligned} \tag{2}$$

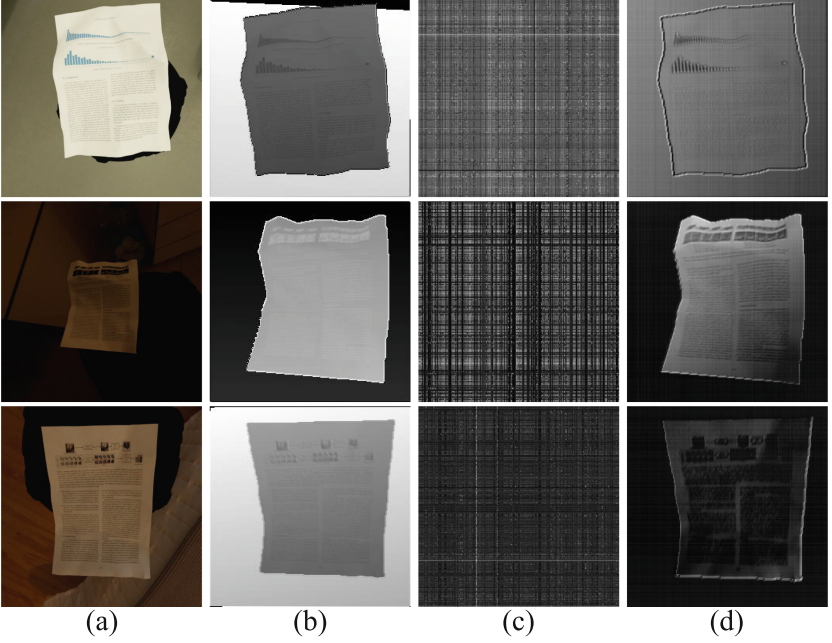
where  $X'$  is the output feature of SLSA. Besides, Fig. 4 shows different distributions of distorted images can be harmonized into a unique space. This verifies the reasonable of our proposed strategy.

Third, we also need to consider the structure information of the distorted images. This is because self-attention can not exploit structure-based information to modeling. Thus, as shown In Fig. 3, we stack Depth-Wise Convolution (DWC)  $DWC(\cdot)$  between two Multi-Layer Perceptrons (MLPs) to Feed Forward Network (FFN), which follows the proposed SLSA. This process is named Structure Multi-Layer Perception (SMLP), which can be expressed as:

$$\begin{aligned}
 X^1 &= MLP(X), \\
 X^2 &= DWC(X^1) \times DWC(X^1), \\
 X' &= MLP(X^2).
 \end{aligned} \tag{3}$$

### 3.2 Network Structure

The overall architecture of the proposed DocHFormer, shown in Fig. 5, is based on a hierarchical encoder-decoder framework. Given a distorted image  $I$ , we perform overlapped image patch embedding with a  $3 \times 3$  convolutional layer. In the network backbone, we stack four STBs to progressively learn features for



**Fig. 4.** Visualization results of different distributions of distorted images: (a) input image, (b) fused output, (c) shuffled image and (d) inverted image.

multi-scale representation. Each level covers its own specific spatial resolution and channel dimension. Additionally, we also add skip-connections to bridge across continuous intermediate features for stable training. In the end, we use a  $3 \times 3$  convolutional layer to output the dewarped backward map  $I_{bm}$ . After that,  $I_{bm}$  and  $I_d$  through Bilinear Sampling (BS)  $BS(\cdot)$  can get high-resolution dewarped images  $I_d$ . This process is obtained by the following process:

$$\begin{aligned}
 I_{bm} &= Net(I), \\
 I_d &= BS(I, I_{bm}), \\
 I_d^{gt} &= BS(I, I_{bm}^{gt}), \\
 loss &= L1(I_{bm}, I_{bm}^{gt}) + L2(I_d, I_d^{gt}) + SSIM(I_d, I_d^{gt}),
 \end{aligned} \tag{4}$$

where  $I_{bm}^{gt}$  is the ground-truth of backward map. We utilize BS operation through the combinations  $I_{bm}^{gt}$  with  $I$  to generate the ground-truth of  $I$ , named  $I_d^{gt}$ .  $L1(\cdot)$ ,  $L2(\cdot)$  and  $SSIM(\cdot)$  denote the Mean Absolute Error, Mean Squared Error and Structural Similarity Index, respectively.

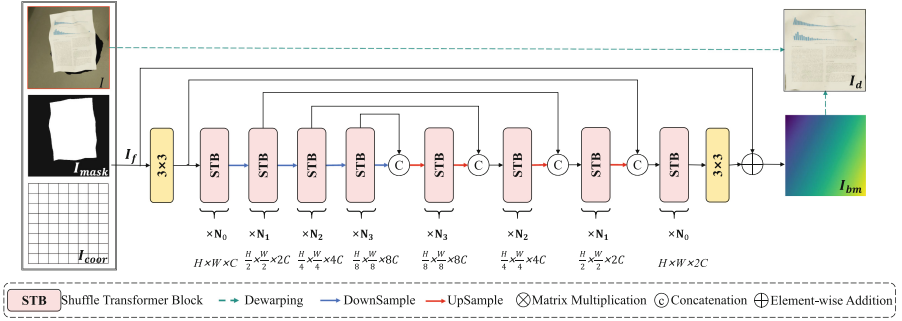


Fig. 5. The overall architecture of the DocHFormer.

## 4 Experiments and Discussion

### 4.1 Datasets and Implementation Details

Our proposed method and comparisons are trained on Doc3D dataset [7], which is a synthetic dataset comprising 100K samples created by real-world document images and rendering software. For each distorted document image, there are corresponding 3D world coordinate maps, albedo maps, normals maps, depth maps, UV maps, and backward mapping maps. As for testing, we adopt the popular DocUNet Benchmark dataset [6] and DIR300 dataset [9]. DocUNet Benchmark dataset contains 130 distorted images in natural scenes captured by mobile devices, while DIR300 includes 300 images of real distorted documents involving more complex backgrounds, distorted degrees, and various lighting conditions. In this paper, we respectively choose 50 and 90 images from the DocUNet Benchmark dataset and DIR300 datasets, a total of 140 distorted images, to create their corresponding testing sets.

In this paper, we implement our model on 5 NVIDIA 3090 GPUs for 60 steps with a global batch size of 8. We set the weight decay of Adam to 0.0005 and use the cosine learning rate scheduler with 0.0002 as the maximum learning rate. Moreover, the patch size of the input image is set to  $256 \times 256$  for training. We measure the quality of dewarped images through popular metrics including Multi-Scale Structural Similarity (MS-SSIM) [25], Local Distortion (LD) [2], and Aligned Distortion (AD) [26]. As for OCR accuracy, we use Character Error Rate (CER) [27] and Edit Distance (ED) [28] to measure the performance of our method. All comparisons are retrained on the above configurations and evaluated on the same platform.

### 4.2 Comparisons with State-of-the-Arts

In this paper, we compare our proposed method with five state-of-the-art algorithms, including DewarpNet (ICCV’19) [7], GeoTr (ACM MM’21) [12], DocGeoNet (ECCV’22) [9], FAM (ICCV’23) [11] and DRS (CVPR’24) [29]. For a



**Table 1.** Quantitative comparisons in DIR300 dataset. “↑” indicates the higher the better, while “↓” means the opposite.

Methods	Venue/Year	MS-SSIM↑	LD↓	ED↓	CER↓	AD↓	Para.
DewarpNet	ICCV’19	0.5793	13.1849	619.256	0.3633	0.2619	49.2M
GeoTr	ACM MM’21	0.6807	6.3877	446.689	0.3325	0.1954	20.1M
DocGeoNet	ECCV’22	0.6911	6.3220	450.422	0.3604	0.1972	19.9M
FAM	ICCV’23	0.6648	7.8056	476.611	0.2713	0.1843	63.6M
DRS	CVPR’24	0.6787	6.5791	470.289	0.3191	0.1855	23.1M
<b>Ours</b>	-	<b>0.7176</b>	<b>5.7301</b>	<b>419.911</b>	<b>0.2867</b>	<b>0.1665</b>	<b>17.8M</b>

fair comparison, we use the available open-source codes provided by the original authors for training and testing.

**Quantitative Results.** As shown in Table 1, our DocHFormer achieves an LD of 5.7301, an ED of 419.911 and an AD of 0.1665. These results significantly outperform previous state-of-the-art methods DocGeoNet [9] and FAM [11]. In addition, from the results of the DIR300 dataset, our proposed DocHFormer achieves a higher MS-SSIM score than all previous SOTA methods, for example with a gain of 0.02 compared with DocGeoNet [9], which demonstrates that our method is capable of improving the integrity of the dewarped image faithfully. Besides, model complexity with 17.8M parameters also shows the promising efficiency of the proposed method.

**Qualitative Results.** The qualitative comparisons are conducted on the DIR300 and DocUNet Benchmark datasets. To compare the local rectified detail, we also show the comparisons of cropped local rectified text. As shown in Fig. 6 and Fig. 7, the proposed DocHFormer shows superior rectification quality. Specifically, the proposed DocHFormer can effectively handle incomplete boundaries and preserve global region details. Moreover, the textlines and curves can be dewarped straighter than other comparisons. Besides, we can see that our method shows less blur and shadows after dewarping (Table 2).

Another goal of DID tasks is to improve subsequent text-related intelligent tasks such as document enhancement and optical character recognition (OCR). To demonstrate the effectiveness, we adopt Tesseract (v5.0.1) [30] as the OCR engine to recognize the text in the images. The visual results of our proposed DocHFormer and several popular DID methods are shown in Fig. 8. As we can see our proposed DocHFormer can OCR engine extract more key textlines from the given scenes and improve performance. This experiment also illustrates the proposed DocHFormer can accelerate the OCR applications in real-world scenarios.



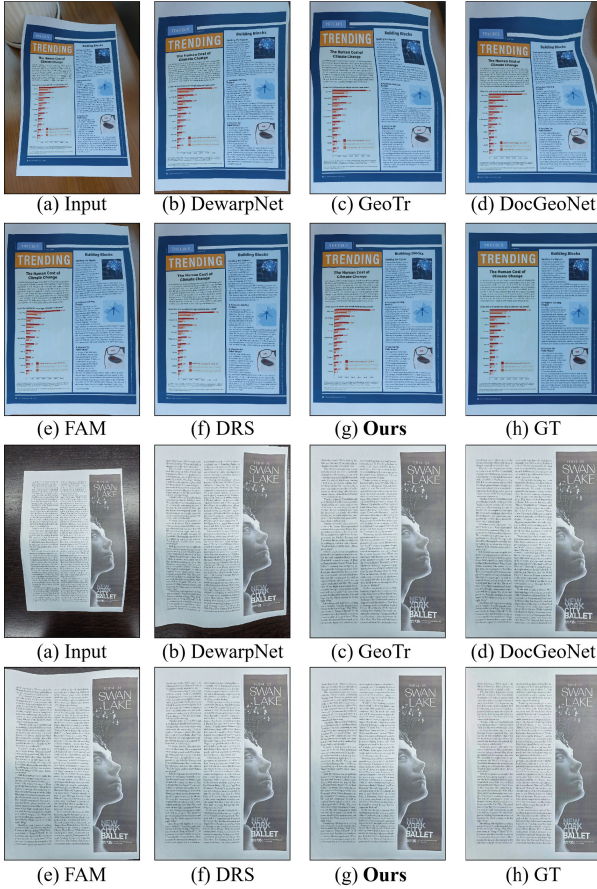
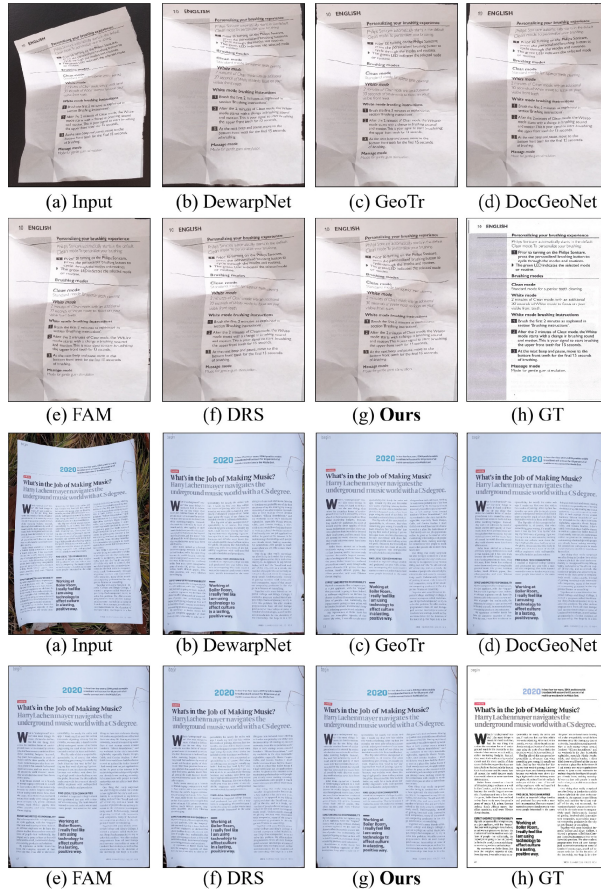


Fig. 6. Qualitative comparisons with previous methods on the DIR300 dataset.

### 4.3 Ablation Study

To verify the effectiveness of our proposed methods and further analyze the key components, we conduct a series of ablation studies on different configurations (Table 3).

**Effects of Multi-priors.** In this paper, we introduce mask and coordinate priors to boost network representation. To demonstrate their contributions to dewarping, we conduct experiments with the following configurations: (1) We remove the mask prior, named **DocHFormer w/o mask**; (2) We remove the coordinate prior, named **DocHFormer w/o coordinate**; (3) We only take the



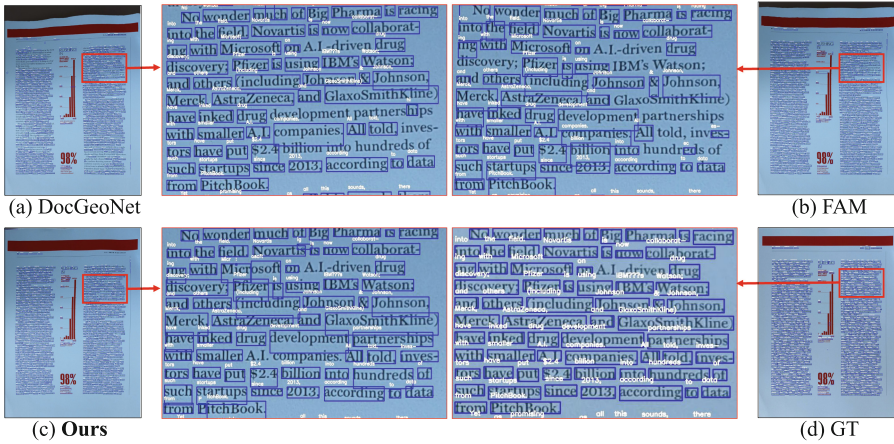
**Fig. 7.** Qualitative comparisons with previous methods on the DocUNet dataset.

distorted image as input, named **DocHFormer w/o any priors**; (4) Our completed version, named **DocHFormer**. Compared with the results of different configurations, LD/ED both increase by a large margin with applying fused priors in four connection schemes. Besides, the results of Fig. 9 achieve the best performance, which validates our current design of different priors.

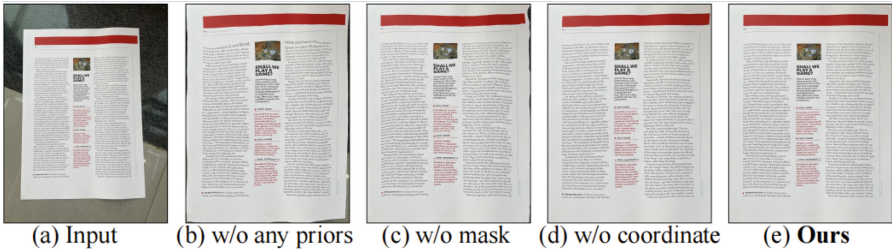
**Effects of the Shuffle Mechanisms and Structure Guidance.** In this paper, we design random shuffle and inverse shuffle operation pairs to form the Shuffle Transformer Block (STB) that can achieve adaptive representation for various features. Besides, we design a SMLP to model structural information. To explore their benefits, we conduct experiments with the following

**Table 2.** Quantitative comparisons in DocUNet Benchmark dataset. “↑” indicates the higher the better, while “↓” means the opposite.

Methods	Venue/Year	MS-SSIM↑	LD↓	ED↓	CER↓	AD↓
DewarpNet	ICCV’19	0.5305	9.6795	732.28	0.4274	0.2950
GeoTr	ACM MM’21	<b>0.5797</b>	9.5183	557.62	0.3548	0.2664
DocGeoNet	ECCV’22	0.5727	<b>9.1102</b>	540.42	0.3590	<b>0.2632</b>
FAM	ICCV’23	<b>0.5801</b>	<b>9.2919</b>	<b>511.34</b>	<b>0.3493</b>	<b>0.2547</b>
DRS	CVPR’24	0.5256	10.5465	587.32	0.381	0.3130
<b>Ours</b>	-	0.5606	9.4366	<b>487.44</b>	<b>0.3422</b>	0.2736



**Fig. 8.** Compared with the recognized text results using Tesseract as OCR engine.

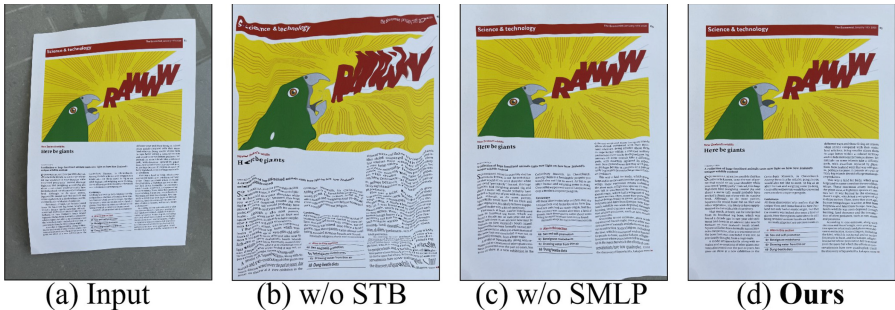


**Fig. 9.** Qualitative comparison of DocHFormer with different priors on DIR300 dataset.

configurations: (1) We replace the STB with original Transformer Block (TB), named **DocHFormer w/o STB**; (2) We remove the SMLP of our proposed STB, named **DocHFormer w/o SMLP**; (3) Our completed version, named **DocHFormer**. From the results of Table 4, without the combinations of harmonization and structure modeling, model performance can be impaired severely.

**Table 3.** Effects of the different priors of DocHFormer.

Configurations	MS-SSIM $\uparrow$	LD $\downarrow$	ED $\downarrow$	CER $\downarrow$	AD $\downarrow$
DocHFormer(w/o any priors)	0.6291	8.3921	834.556	0.3758	0.3122
DocHFormer(w/o mask)	0.6582	7.1826	494.322	0.2997	0.1954
DocHFormer(w/o coordinate)	0.6957	5.8879	433.044	0.3758	0.1748
<b>DocHFormer(Ours)</b>	<b>0.7176</b>	<b>5.7301</b>	<b>419.911</b>	<b>0.2867</b>	<b>0.1665</b>

**Fig. 10.** Qualitative comparison with different configurations of DocHFormer on DIR300 dataset.**Table 4.** Effects of the different modules of DocHFormer.

Configurations	MS-SSIM $\uparrow$	LD $\downarrow$	ED $\downarrow$	CER $\downarrow$	AD $\downarrow$
DocHFormer w/o STB	0.5225	21.1517	1297.022	0.5727	0.5906
DocHFormer w/o SMLP	0.6241	8.3855	721.478	0.3770	0.2638
<b>DocHFormer (Ours)</b>	<b>0.7176</b>	<b>5.7301</b>	<b>419.911</b>	<b>0.2867</b>	<b>0.1665</b>

Similarly, Fig. 10 reveals that the image details and structural integrity can be preserved well.

**Effects of Different Loss Functions.** In this paper, we adopt L1-based, L2-based and SSIM losses to form our final objection loss. To demonstrate the effectiveness of this multi-term loss on model training, we conduct experiments with the following configurations: (1) We remove L1-based loss, named **DocHFormer w/o L1 loss**; (2) We remove L2-based loss, named **DocHFormer w/o L2 loss**; (3) We remove L2-based loss, named **DocHFormer w/o SSIM loss**; (4) Our completed version, named **DocHFormer**. The quantitative and qualitative results of Table 5 and Fig. 11 all verify the reasonable and effectiveness of multi-term loss in model training.





**Fig. 11.** Qualitative comparison of DocHFormer with different loss configurations on DIR300 dataset.

**Table 5.** Effects of different loss on the DIR300 dataset.

Configurations	MS-SSIM $\uparrow$	LD $\downarrow$	ED $\downarrow$	CER $\downarrow$	AD $\downarrow$
DocHFormer w/o L1 loss	0.6798	6.5739	636.189	0.3225	0.2232
DocHFormer w/o L2 loss	0.6636	6.3733	499.533	0.2924	0.2069
DocHFormer w/o SSIM loss	0.6703	6.6898	501.533	0.4060	0.1836
<b>DocHFormer (Ours)</b>	<b>0.7176</b>	<b>5.7301</b>	<b>419.911</b>	<b>0.2867</b>	<b>0.1665</b>

## 5 Conclusion

In this work, we consider the properties of Document Image Dewarping (DID) tasks and present a novel Transformer-based network, called DocHFormer, that can take hierarchical priors features of images with harmonized space to achieve accurate representation. To be specific, we adopt the novel random shuffle and inverse shuffle operations to make each pixel adaptively obtain equal probability, avoiding the learning ambiguity of weight sharing. After that, we embed this mechanism into Local window Self-Attention (LSA) to form the basic feature extractor of our DocHFormer. Besides, we suggest exploiting structure-based information to boost structural information modeling. With these cooperations, our proposed DocHFormer can achieve significant dewarping results and benefit downstream OCR tasks.

**Acknowledgement.** This work is supported by Fujian Provincial Young and Middle-aged Teachers’ Educational Research Project (JZ230050), Unveiling and Leading Projects of Xiamen (No. 3502Z20241011), Open Project of the State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS2024101), Natural Science Foundation of Xiamen (3502Z202373058), and Fujian Key Technological Innovation and Industrialization Projects (2023XQ023).

## References

1. Meng, G., Wang, Y., Qu, S., Xiang, S., Pan, C.: Active flattening of curved document images via two structured beams. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3890–3897 (2014)
2. You, S., Matsushita, Y., Sinha, S., Bou, Y., Ikeuchi, K.: Multiview rectification of folded documents. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(2), 505–511 (2017)
3. He, Y., Pan, P., Xie, S., Sun, J., Naoi, S.: A book dewarping system by boundary-based 3D surface reconstruction. In: 2013 12Th International Conference on Document Analysis and Recognition, pp. 403–407. IEEE (2013)
4. Liu, C., Zhang, Y., Wang, B., Ding, X.: Restoring camera-captured distorted document images. *Int. J. Doc. Anal. Recogn. (IJDAR)* **18**, 111–124 (2015)
5. Zhang, L., Zhang, Y., Tan, C.: An improved physically-based method for geometric restoration of distorted document images. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(4), 728–734 (2008)
6. Ma, K., Shu, Z., Bai, X., Wang, J., Samaras, D.: DocUNet: document image unwarping via a stacked U-Net. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4709 (2018)
7. Das, S., Ma, K., Shu, Z., Samaras, D., Shilkrot, R.: DewarpNet: single-image document unwarping with stacked 3D and 2D regression networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 131–140 (2019)
8. Xie, G.-W., Yin, F., Zhang, X.-Y., Liu, C.-L.: Dewarping document image by displacement flow estimation with fully convolutional network. In: Document Analysis Systems: 14th IAPR International Workshop, pp. 131–144 (2020)
9. Feng, H., Zhou, W., Deng, J., Wang, Y., Li, H.: Geometric representation learning for document image rectification. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*, pp. 475–492. Springer Nature Switzerland, Cham (2022). [https://doi.org/10.1007/978-3-031-19836-6\\_27](https://doi.org/10.1007/978-3-031-19836-6_27)
10. Jiang, X., Long, R., Xue, N., Yang, Z., Yao, C., Xia, G.-S.: Revisiting document image dewarping by grid regularization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4543–4552 (2022)
11. Li, H., Wu, X., Chen, Q., Xiang, Q.: Foreground and text-lines aware document image rectification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 19574–19583 (2023)
12. Feng, H., Wang, Y., Zhou, W., Deng, J., Li, H.: DocTr: document image transformer for geometric unwarping and illumination correction. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 273–281 (2021)
13. Tsoi, Y.-C., Brown, M.S.: Multi-view document rectification using boundary. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007)
14. Meng, G., Pan, C., Xiang, S., Duan, J., Zheng, N.: Metric rectification of curved document images. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(4), 707–722 (2011)
15. Brown, M.S., Tsoi, Y.-C.: Geometric and shading correction for images of printed materials using boundary. *IEEE Trans. Image Process.* **15**(6), 1544–1554 (2006)
16. Koo, H.I., Kim, J., Cho, N.I.: Composition of a dewarped and enhanced document image from two view images. *IEEE Trans. Image Process.* **18**(7), 1551–1562 (2009)
17. Liang, J., DeMenthon, D., Doermann, D.: Geometric rectification of camera-captured document images. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(4), 591–605 (2008)

18. Li, X., Zhang, B., Liao, J., Sander, P.V.: Document rectification and illumination correction using a patch-based CNN. *ACM Trans. Graph. (TOG)* **38**(6), 1–11 (2019)
19. Markovitz, A., Lavi, I., Perel, O., Mazor, S., Litman, R.: Can you read me now? Content aware rectification using angle supervision. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII*, pp. 208–223. Springer International Publishing, Cham (2020). [https://doi.org/10.1007/978-3-030-58610-2\\_13](https://doi.org/10.1007/978-3-030-58610-2_13)
20. Zhang, W., Wang, Q., Huang, K.: Polar-Doc: one-stage document dewarping with multi-scope constraints under polar representation. *arXiv preprint arXiv:2312.07925* (2023)
21. Yu, F., et al.: DocReal: robust document dewarping of real-life images via attention-enhanced control point prediction. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 665–674 (2024)
22. Kumari, P., Das, S.: Am i readable? Transfer learning based document image rectification. *Int. J. Doc. Anal. Recogn. (IJ DAR)*, 1–14 (2024)
23. Verhoeven, F., Magne, T., Sorkine-Hornung, O.: UVDoc: neural grid-based document unwarping. In: *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–11 (2023)
24. Feng, H., Liu, S., Deng, J., Zhou, W., Li, H.: Deep unrestricted document image rectification. *IEEE Trans. Multimedia* (2023)
25. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
26. Ma, K., Das, S., Shu, Z., Samaras, D.: Learning from documents in the wild to improve document unwarping. In: *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–9 (2022)
27. Morris, A.C., Maier, V., Green, P.D.: From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition. In: *Interspeech*, pp. 2765–2768 (2004)
28. Levenshtein, V.I., et al.: Binary codes capable of correcting deletions, insertions, and reversals, in: *Soviet physics doklady*, vol. 10, Soviet Union, pp. 707–710 (1966)
29. Zhang, J., Peng, D., Liu, C., Zhang, P., Jin, L.: DocRes: a generalist model toward unifying document image restoration tasks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15654–15664 (2024)
30. Smith, R.: An overview of the tesseract OCR engine. In: *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, pp. 629–633. IEEE (2007)



# Document Image Shadow Removal via Frequency Information-Oriented Network

Fan Yang<sup>1,2</sup>, Xinyue Zhou<sup>1,2</sup>, Nanfeng Jiang<sup>1,2(✉)</sup>, Da-Han Wang<sup>1,2</sup>,  
Xu-Yao Zhang<sup>3</sup>, Guantin Li<sup>1,2</sup>, Wang Man<sup>1,2</sup>, and Yun Wu<sup>1,2</sup>

<sup>1</sup> School of Computer and Information Engineering, Xiamen University  
of Technology, Xiamen 361024, China

2322071044@s.xmut.edu.cn, 2322071057@stu.xmut.edu.cn,  
{2023000066,wangdh,manwang,ywu}@xmut.edu.cn, qianxule007@gmail.com

<sup>2</sup> Fujian Key Laboratory of Pattern Recognition and Image Understanding,  
Xiamen 361024, China

<sup>3</sup> State Key Laboratory of Multimodal Artificial Intelligence Systems,  
Institute of Automation of Chinese Academy of Sciences, Beijing, China  
xyz@nlpr.ia.ac.cn

**Abstract.** Removing shadows from document images can significantly improve the Quality of Experience (QoE) and boost the performance of the downstream document analysis and recognition tasks. However, existing methods still have limited generalization ability on complex document images and are prone to disrupt the image details. To address this issue, we consider the different shadow types that impact the image content on different frequency sub-bands. This motivates us to exploit frequency-domain information and further design a Frequency Information-oriented Dshadow Network (FID-Net). The proposed FID-Net mainly uses two elaborated modules, named Frequency Feature Extractor (FFE) and a Frequency Feature Refinement (FFR). FFE can generate low/high-frequency features through adaptively decomposing spectra of the shadow image. After that, FFR further refines both frequency features with mutual information operations. With the proposed key designs, extensive experimental results on the commonly used benchmarks demonstrate that the proposed method can learn discriminative shadows and achieve favorable performance against state-of-the-art approaches.

**Keywords:** Document Image Dshadowing · Frequency Domain Representation

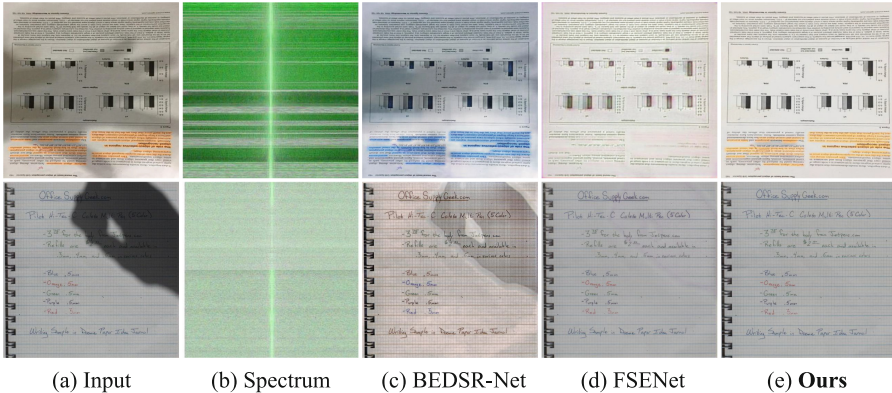
## 1 Introduction

The document images captured under complex conditions, including uncontrollable illumination conditions, camera angles and occlusion *et al.*, exist ubiquitous

F. Yang and X. Zhou—The equal contributions to this work.



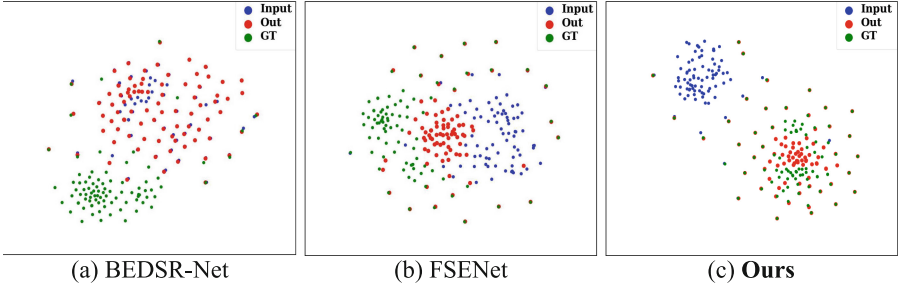
shadows that impair the visual quality of images but impose severe limitations on various subsequent Optical Character Recognition (OCR) [1–5] tasks. Thus, developing an effective method to remove shadows of documents becomes an attractive topic in the current research community. Conventional methods [6–9] mainly rely on carefully designed hand-crafted image priors, *e.g.*, illumination, gradient, and region consistency, to form a function for shadow removal. However, these methods lack the flexibility to generalize to various cases.



**Fig. 1.** The generated results of different types of shadows between our method and recent SOTA methods.

Recently, with the developments of advanced deep Convolutional Neural Network (CNN) models, the shadow removal performance has achieved remarkable progress with a data-driven nature. Nevertheless, as shown in Fig. 1, the produced results of current popular methods have some detail loss and halo artifacts, since they have a limited receptive field to effectively capture long-range image dependencies. To solve this problem, Transformer-based structures have been introduced and present larger receptive fields experimentally outperform the previous methods. However, no matter CNN-based or Transformer-based approaches, most of them mainly design single-stacked architectures to learn mappings between shadow and shadow-free images. These ways are prone to ignore the importance of explicit modeling in feature representations when handling complex distributed shadows, such as local, global and non-homogeneous. Moreover, existing methods only operate deshadow in the image spatial domain, and do not consider frequency domain information. As a result of these issues, we operate on the assumption that different type of shadows impact image content on different frequency sub-bands. Therefore, we subtract shadow images from the ground-truth images and further obtain Fourier spectra of residual images. As illustrated in Fig. 1-(b), we can observe that locally distributed shadows are contaminated with high-frequency content while global and nonhomogeneous shadows are dominated by low-frequency degraded contents, thus indicating the

need to consider the different properties among these shadows in modeling. Based on these facts, we can conclude the main issue: *How to effectively combine frequency with spatial information and explicitly model them for various shadow removals in document images?*



**Fig. 2.** TSNE plots of the degradation embeddings used in our proposed method and the state-of-the-arts.

To address the mentioned-above issue, we propose a Frequency Information-oriented Deshadow Network (FID-Net) to address the above issues. To be specific, we design a Frequency Feature Extractor (FFE) and a Frequency Feature Refinement (FFR) these modules and incorporate them into a U-shaped Transformer backbone. They are designed to identify and refine the relevant frequency components based on the shadow patterns present in the input image. On one hand, FFE explicitly extracts specific frequency elements from the image intermediate features, guided by an adaptive decomposition of the input spectral characteristics that reflect the underlying degradation. On the other hand, FFR further refines these elements by facilitating the exchange of complementary information across different frequency features. With the cooperation of these two modules, the proposed FID-Net not only can learn discriminative representation more effectively than others, as shown in Fig. 2, but also can effectively remove complex shadows with a dynamically adjusted learning strategy.

- We consider the properties of shadows in documents and propose a simple but effective framework that incorporates both spatial and frequency domain information to conduct shadow removal in document images.
- We design two elaborated modules, namely Frequency Feature Extractor (FFE) and Frequency Feature Refinement (FFR). FFE aims to dynamically decouple various features of input images, while FFR further refines these elements by facilitating the exchange of complementary information.
- Extensive experiments on popular datasets demonstrate that our proposed method outperforms state-of-the-art approaches with fewer parameters. Moreover, the downstream application performance of (OCR) can be improved significantly after deshadowing by our proposed method.

## 2 Related Work

### 2.1 Natural Image Shadow Removal

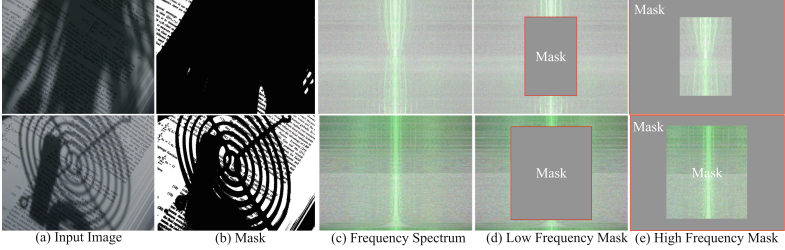
Traditional natural image shadow removal methods mainly rely on various physical properties, such as image gradients and illumination information, to build models. For example, Finlayson *et al.* [10,11] exploited the gradient consistency to remove shadows. Guo *et al.* [12] introduced illumination information to guide shadow detection and further remove them. Gong *et al.* [13] combined two rough user inputs and designed an interactive way to remove shadows. Recently, many deep learning-based methods have been proposed and achieved remarkable progress in this field. Qu *et al.* [14] proposed a DeshadowNet that integrates multi-scale context embedding information for shadow removal. Hu *et al.* [15] captured global features for shadow detection and removal through a direction-aware spatial attention module. Chen *et al.* [16] developed a CANet that attempts to transfer the contextual information of shadow-free to shadow regions to operate shadow removal. Recently, some researchers [17,22], [23] used masks to detect shadows and remove them. Although these methods are effective for natural images, they do not generalize well to document image shadow removal due to the different characteristics between natural images and document images, especially in image properties and evaluation metrics.

### 2.2 Document Image Shadow Removal

The earlier document image shadow removal methods tend to use hand-crafted priors or mathematical formulas to build models [6–9,18]. Bako *et al.* [6] adopted an estimated shadow map to detect and remove shadows. Oliveira *et al.* [9] used natural neighborhood interpolation to remove shadows. Jung *et al.* [8] implemented a water-filling technique to rectify the illumination of document images by transforming the input image into a topographic representation. These methods can achieve good performance to some extent, but they can not effectively work on complex scenes. Recently, deep learning-based methods have emerged in this task and provided significant achievements. Lin *et al.* [19] first introduced background and attention information and proposed a BEDSR-Net for document image shadow removal. Li *et al.* [20] proposed a FSENet that combines a frequency-aware way to remove shadows. Zhang *et al.* [21] presented a BGshadowNet that improves appearance and illumination consistency in a coarse-to-fine way. However, these methods have limited performance in various shadow removal, for example, BEDSR-Net does not do a good job of distinguishing text from shadows, and FSENet does not do a good job of overall contrast in the image. Furthermore, artifact distortion and detail loss are obvious in their generated results. In contrast, our proposed FID-Net explores the properties of shadows in the frequency domain to improve robustness in various shadows removal.

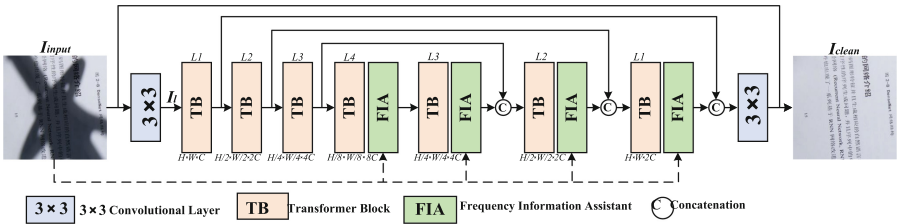
### 3 Methodology

#### 3.1 Motivation



**Fig. 3.** Spatial and frequency domain masks of different shadows.

Existing methods [17,22], [23] utilize masks to model the shadow regions. However, this way fails to accurately capture the distribution due to the fixed threshold, as shown in Fig. 3-(b). Moreover, all these approaches purely operate in the spatial domain and do not consider frequency domain information. Based on this, we attempt to design a model using both spatial and frequency information, thus we transform the input images into frequency versions and find that high/low-frequency components of the spectrum have an obvious difference on different distributed shadows, as shown in Fig. 3-(c). Different spectrum maps accentuate the corresponding informative sub-bands for different frequency distributions. This motivates us to design frequency-based masks to accurately locate shadow regions. This goal is achieved by two elaborated modules, Frequency Feature Extractor and Frequency Feature Refinement (FFR). FFE aims to deal with the amplitude spectrum dynamically and exploit it to guide shadow removal in the frequency domain. Moreover, to refine different frequency features to boost representation, FFR is proposed to facilitate the exchange of complementary information.



**Fig. 4.** The framework of our proposed FID-Net.

### 3.2 Network Structure

Figure 4 shows the overall framework of our proposed FID-Net, which directly learns the mapping between shadow  $I_{input}$  and shadow-free images  $I_{clean}$  via an end-to-end way. First, FID-Net extracts shallow features  $I_l$  by applying a  $3 \times 3$  convolution operation. Second,  $I_l$  are processed through a 4-level encoder-decoder U-shape network. Each level employs several Transformer Blocks (TBs), with the number of blocks gradually increasing from the top level to the bottom level to maintain computational efficiency. Then, the encoder takes  $I_l$  as input, and progressively transforms them into a lower-resolution latent representation. Third, in order to assist the decoding process, we incorporate a Frequency Information Assistant (FIA) module, as shown in Fig. 5, in our FID-Net. FIA is an adapter module that sequentially connects every two levels of the decoder. At each decoder level, we use the FIA to explicitly depart the various shadow contents from the clean image content in the frequency domain, and subsequently assist in refining features in the spatial domain for effective deshadow. We achieve this goal by designing two key modules: Frequency Feature Extractor (FFE) and Frequency Feature Refinement (FFR).

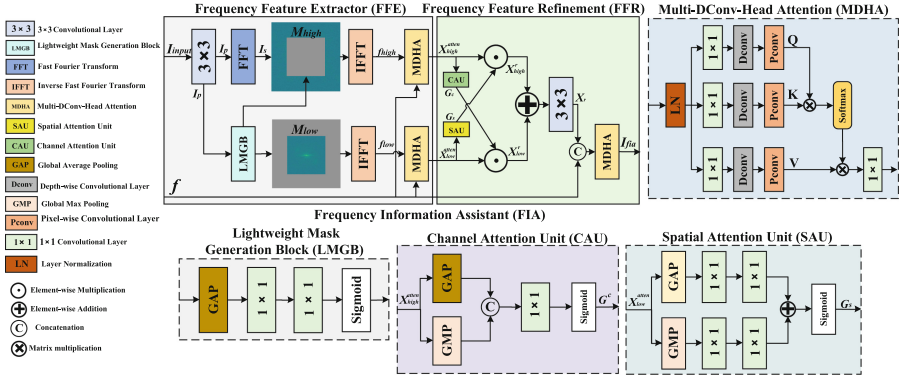


Fig. 5. The framework of our proposed FIA. It consists of FFE and FFR.

**Frequency Feature Extractor.** As illustrated in Fig. 5, FFE takes both the input image  $I_{input}$  and the intermediate features  $F^i$  of each level  $i = 1, 2, 3, 4$ . In particular, FFE adaptively decouples the spectra of  $I_{input}$ , which is used to guide  $f$  to extract the different frequency representations. To be specific, FFE achieves the above process via three steps: (1) Domain Transformation, (2) Mask Representation, and (3) Attention feature extraction.

(1) **Domain Transformation:** We use one  $3 \times 3$  convolution layer to extract initial feature  $I_p$ , and feed it into Fast Fourier Transform (FFT) to obtain spectral domain representation of  $I_{input}$ , named  $I_s$ .

**(2) Mask Representation:** We aim to adaptively depart different frequency contents of  $I_p$  during network learning. This motivates us to design a Lightweight Mask Generation Block (LMGB), as shown in Fig. 5, that uses frequency boundary to divide the frequency sub-band into two parts. Each of them adjustly refers to shadow components according to various distributions. Thus, in LMGB, we project the  $I_p$  into a Global Average Pooling (GAP), a  $1 \times 1$  convolutional layer with GELU activation function to produce two factors ( $\alpha$  and  $\beta$ ) with a range from 0 to 1. In this work, the channel dimension of them is set to 2. Thus, both factors can define the mask size by multiplying with the width  $W$  and height  $H$  of the spectra  $I_s$ . In this work, we set the binary Mask of low frequency  $M_{low}[\frac{H}{2} - \alpha \frac{H}{k} \cdot \frac{H}{2} + \alpha \frac{H}{k}, \frac{W}{2} - \alpha \frac{W}{k} \cdot \frac{W}{2} + \alpha \frac{W}{k}] = 1$ , where  $k$  is set to a small value of 128. Accordingly, the mask for high frequency  $M_{high}$  can be obtained by setting the values within the remaining region as 1. Subsequently, we can obtain the adaptively decoupled features  $f_{low}$  and  $f_{high}$  by applying the learned masks to the spectra via element-wise multiplication and using the Inverse Fast Fourier Transform (IFFT). This process can be expressed as follows:

$$\begin{aligned}
 I_p &= Conv(I_{input}), \\
 I_s &= FFT(I_p), \\
 M_{low}, M_{high} &= LMGB(I_p), \\
 f_{low} &= IFFT(M_{low} * I_{input}), \\
 f_{high} &= IFFT(M_{high} * I_{input}).
 \end{aligned} \tag{1}$$

**(3) Attention Feature Extraction:** As shown in Fig. 5, we use the efficient transformer module, namely Multi-DConv-Head Attention (MDHA), to effectively extract the discriminate contents from  $f$  with the guidance of  $f_{low}$  and  $f_{high}$ . The process can be expressed as:

$$\begin{aligned}
 Q &= H_{dconv}^1(H_{pconv}^1(LN(f_*))), \\
 K &= H_{dconv}^2(H_{pconv}^2(LN(f))), \\
 V &= H_{dconv}^3(H_{pconv}^3(LN(f))), \\
 X_*^{atten} &= V * Softmax(QK^T/\alpha),
 \end{aligned} \tag{2}$$

where  $f_*$  refers to  $f_{low}$  or  $f_{high}$ .  $H_{dconv}(\cdot)$ ,  $H_{pconv}(\cdot)$  and  $LN$  denote depth-wise convolution, pixel convolution, and layer normalization, respectively.  $X_*^{atten}$  refer to the output low frequency  $X_{low}^{atten}$  or high frequency  $X_{high}^{atten}$  features. Both of them will be taken into the Frequency Feature Refinement (FFR) for better representation with cross-interaction.

**Frequency Feature Refinement.** In general, edges and fine texture details exist in high-frequency while global information, such as color and contrast, are presented in low-frequency regions. These two different components should be effectively interacted in the training process for better representation. As shown in Fig. 5, we design an FFR that builds upon Channel Attention Unit (CAU) and Spatial Attention Unit (SAU) to achieve the above goal.

(1) **Channel Attention Unit (CAU)**: This unit computes the channel attention map from high-frequency features that are then used to complement features of the low-frequency branch. As shown in Fig. 5, CAU sequentially leverages two different channel-wise pooling techniques, named Global Average Pooling (GAP) and Global Max Pooling (GMP) to produce two single-channel spatial feature maps. Both them are concatenated and passed through one convolutional layer to produce spatial attention map  $G^s$ , which is used to obtain the refined low-frequency features  $X_l^r$  through element-wise multiplication on  $X_l^a$ . This process can be expressed as:

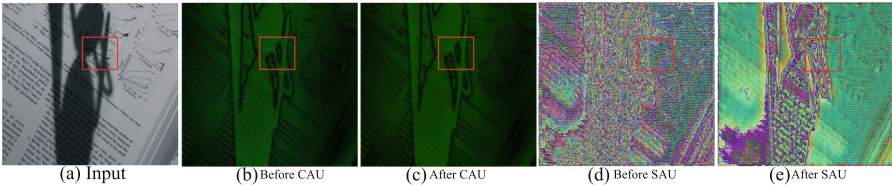
$$\begin{aligned} X_g &= GAP(X_{high}^{atten}), \\ X_m &= GMP(X_{high}^{atten}), \\ G^c &= Sigmoid(Conv(concat(X_g, X_m))), \\ X_{low}^r &= G^c \times X_{low}^{atten}, \end{aligned} \quad (3)$$

where  $X_g$  and  $X_m$  are the intermediate outputs of GAP and GMP operations, respectively.

(2) **Spatial Attention Unit (SAU)**: In contrast to SAU, this unit uses a dual-branch architecture that works on low-frequency features  $X_l^a$  to provide complementary information for high-frequency features. Specifically, the top branch stacks GAP and one Convolutional layer with ReLU. Instead of GAP, the bottom branch adopts GMP at the head. With the combinations of these two branches, we apply the sigmoid function to produce the final channel attention map  $G^c$ , which is used to modulate the refined high-frequency features  $X_{high}^r$  through element-wise multiplication on  $X_{high}^{atten}$ . This process can be expressed as:

$$\begin{aligned} X'_g &= Conv(Conv(GAP(X_{low}^{atten}))), \\ X'_m &= Conv(Conv(GMP(X_{low}^{atten}))), \\ G^s &= Sigmoid(Conv(X'_g + X'_m)), \\ X_{high}^r &= G^s \times X_{high}^{atten}, \end{aligned} \quad (4)$$

where  $X'_g$  and  $X'_m$  are the intermediate outputs of GAP and GMP operations, respectively.



**Fig. 6.** The generated feature maps before and after CAU and SAU process.



Figure 6 respectively shows the features map before and after processing by SAU and CAU. We can observe that the feature details can be refined with fewer noise issues. The ablation study also shows the effectiveness of these two units. In the end, we can obtain the final output via the following process:

$$\begin{aligned} X_r &= Conv(X_{high}^r + X_{low}^r), \\ I_{fia} &= MDHA(Concat(X_r', f)), \end{aligned} \quad (5)$$

where  $X_r$  denotes the refine features, and  $I_{fia}$  denotes the output of FIA module.

### 3.3 Object Function

Our final loss function  $Loss_{final}$  for optimizing the proposed network consists of three components: L1-based loss  $Loss_{l1}$ , SSIM loss  $Loss_{ssim}$  and perceptual loss  $Loss_{per}$ :

$$\begin{aligned} Loss_{l1} &= l1(I_{out}, I_{gt}), \\ Loss_{ssim} &= 1 - SSIM(I_{out}, I_{gt}), \\ Loss_{per} &= \frac{1}{C_l H_l W_l} \|\phi_l(I_{out}) - \phi_l(I_{gt})\|_2^2, \\ Loss_{final} &= Loss_{l1} + Loss_{ssim} + Loss_{per}, \end{aligned} \quad (6)$$

where  $l1(\cdot)$  denotes the Mean Square Error (MSE) denote,  $SSIM(\cdot)$  refers to structural similarity calculation. In  $Loss_{per}(\cdot)$ ,  $C_l$ ,  $H_l$ ,  $W_l$  represents the dimension of the feature map at  $l$ -th convolution layer within the VGG19 network.  $\phi_l(X)$  denote the feature of the  $l$ -th convolutional layer. In this paper, we focus on the last convolutional layer of VGG19 only. The ablation study also shows the different configurations of loss functions.

## 4 Experiments and Discussion

### 4.1 Dataset and Implementation Details

The proposed method is evaluated on two public datasets, RDD dataset [21] and Kligler’s [7]. RDD collects 4916 pairs of shadow and shadow-free images, divided into two groups, 4371 for training and 545 for testing. These images contain paper, menu, and color texts, under different lighting conditions and occluder shadow. As for Kligler’s, due to the lack of training sets, we adopt the same training set of RDD and evaluate on Kligler’s all images, totally 300 images of various intensities and scales shadows.

As for training, we crop patches sized at  $128 \times 128$  pixels for each input and adopt a batch size of 8. The network optimization employs the Adam optimizer ( $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ ) with a learning rate of 0.0002. The whole training epochs are set to 200. The measure metrics include popular Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) and Learned Perceptual Image Patch Similarity (LPIPS). In this paper, all comparisons are retrained on the above configurations and evaluated on the same platform.



## 4.2 Comparisons with State-of-the-Arts

We compare our proposed method with eight state-of-the-art algorithms, including three natural image shadow removal methods (DHAN (AAAI’20) [17], Fu *et al.* (CVPR’21) [22] and ShadowFormer (AAAI’23) [23]) and five document image shadow removal methods (Bako *et al.* (ACCV’16) [6], Jung *et al.* (ACCV’18) [8], BEDSR-Net (CVPR’20) [19], FSENet (ICCV’23) [20] and BGShadowNet (CVPR’23) [21]). For a fair comparison, we use the available open-source codes provided by the original authors for training and testing. To note that, we resize all testing images to the same size ( $512 \times 512$ ) for evaluation.

**Table 1.** Quantitative results on RDD and Kligler’s datasets.

Methods	RDD			Kligler’s		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
DHAN (AAAI’2020)	<b>28.017</b>	0.918	<b>0.058</b>	24.163	<b>0.904</b>	<b>0.094</b>
Fu <i>et al.</i> (CVPR’2021)	24.498	0.821	0.182	21.522	0.832	0.278
ShadowFormer (AAAI’2023)	13.545	0.391	0.654	15.064	0.462	0.667
Bako <i>et al.</i> (ACCV’2016)	20.486	0.892	0.085	<b>24.772</b>	0.896	0.098
Jung <i>et al.</i> (ACCV’2018)	14.278	0.840	0.122	13.723	0.851	0.124
BEDSR-Net (CVPR’2020)	23.969	<b>0.919</b>	0.103	16.055	0.719	0.268
FSENet (ICCV’2023)	13.071	0.385	0.667	15.525	0.404	0.680
BGShadowNet (CVPR’2023)	13.435	0.420	0.677	15.125	0.497	0.671
<b>Ours</b>	<b>32.484</b>	<b>0.9666</b>	<b>0.046</b>	<b>28.842</b>	<b>0.9339</b>	<b>0.077</b>

**Quantitative Evaluation.** Table 1 compares the performance of our proposed FID-Net with the SOTA methods on RDD and Kligler’s, respectively. The evaluation results indicate that the proposed ADR-Net has achieved the best values on PSNR, SSIM and LPIPS. Specially, our FID-Net achieves 32.484dB PSNR and 0.966 SSIM on RDD. It improves the PSNR by almost 4dB over the previous SOTA method DHAN. Furthermore, our method can outperform all methods and especially surpass FSENet almost 85% by PSNR in Kligler’s dataset. In terms of LPIPS on all datasets, our FID-Net also surpasses other methods by a large margin. These results can well validate the effectiveness of our proposed method.

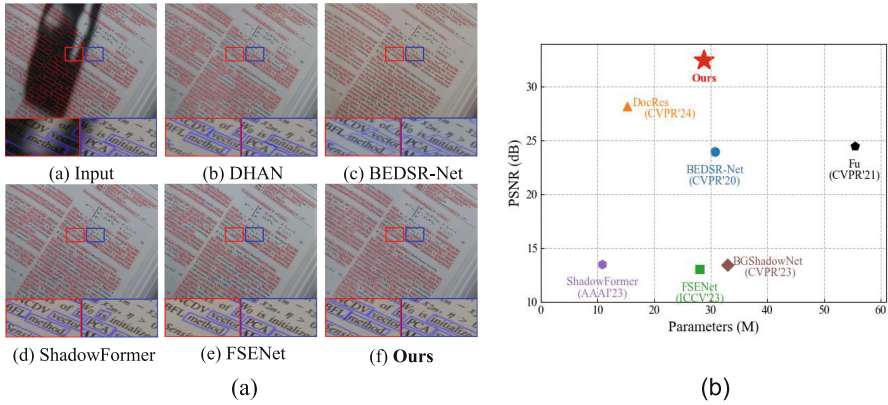
**Qualitative Evaluation.** Figure 7 show visual results on RDD and Kligler’s datasets. We can observe that the images generated by other methods are less natural in the shadows and high-frequency regions. Different from them, our proposed FID-Net generates shadow-free images with better-restored details in the shadows, and are most similar to ground truths, especially in complex distributed shadow regions. To be specific, Fu *et al.* hardly remove the shadows,



Fig. 7. Visual results on RDD dataset and Kligher’s dataset.

BEDSR-Net generate images with graininess and loss of detail, FSENet generates images with color distortion, and the results by BGShadowNet suffer from artifacts and texture loss. Besides, the incomplete and redundant boundaries phenomena that exist in other methods. Compared with them, the shadow-free images generated by our method show fewer artifacts and are much cleaner than others.

**Evaluation on Real-World Applications.** As for OCR accuracy evaluation, we utilize Edit Distance (ED) to evaluate the capacity of text detection on the RDD dataset. The presented results of Table 2 show that our FID-Net performs better on detection accuracy than most comparisons. In terms of visual appearance, in Fig. 8, we provide the character detection results by a popular detection algorithm, namely CharNet [24]. As observed, although parts of shadow removal methods can improve the performance to a large extent, some may even damage the detection performance. As can be seen from Fig. 8-(a), compared with DHAN, FSENet and ShadowFormer, our proposed FID-Net has achieved better performance in text detection, and compared with BEDSR-Net, our method



**Fig. 8.** (a) Visual results on text detection. (b) Comparisons of PSNR and parameters on current popular methods.

is also closer to the original image in the processing of the overall background color. In addition, we conduct runtime and model complexity analysis to explore the potential capacity of the application. Table 2 also reports that our proposed FID-Net is able to outperform most algorithms with lower computational complexity.

**Table 2.** Quantitative results of Edit Distance (ED) on Kligler’s dataset and model complexity.

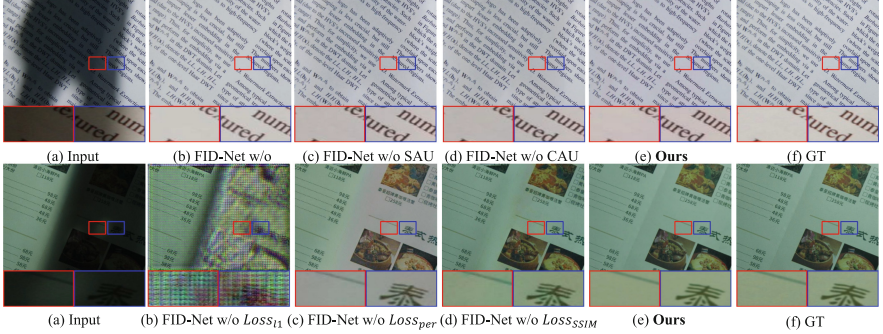
Methods	ED ↓	Parameters (M)
Input	138.31	—
DHAN (AAAI’2020)	121.25	12.85
Fu <i>et al.</i> (CVPR’2021)	174.89	55.42
ShadowFormer (AAAI’2023)	116.08	10.84
Bako <i>et al.</i> (ACCV’2016)	123.37	—
Jung <i>et al.</i> (ACCV’2018)	126.96	—
BEDSR-Net (CVPR’2020)	127.43	30.72
FSENet (ICCV’2023)	117.69	28.03
BGShadowNet (CVPR’2023)	169.19	32.94
<b>Ours</b>	<b>109.63</b>	<b>28.8</b>

### 4.3 Ablation Study

In this section, we conduct ablation studies to test the impacts of various configurations to the overall performance of our proposed method.

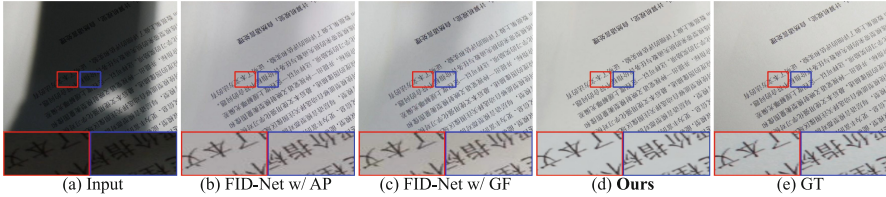
**Table 3.** (a) Quantitative results on different modules. (b) Quantitative results on different loss functions. (c) Quantitative results on different decomposition strategies of LMGB.

Configurations		PSNR↑	SSIM↑	LPIPS↓
(a)	FID-Net w/o None	23.87	0.9516	0.072
	FID-Net w/o Mask	27.12	0.9568	0.058
	FID-Net w/o SAU	24.88	0.9522	0.064
	FID-Net w/o CAU	27.29	0.9616	0.067
(b)	FID-Net w/o $Loss_{l1}$	15.04	0.2603	0.076
	FID-Net w/o $Loss_{per}$	28.00	0.8326	0.058
	FID-Net w/o $Loss_{ssim}$	25.67	0.7768	0.061
(c)	FID-Net w/ AP	27.08	0.9548	0.064
	FID-Net w/ GF	26.46	0.9521	0.063
(d)	<b>Ours</b>	<b>32.484</b>	<b>0.9666</b>	<b>0.046</b>



**Fig. 9.** Visual results on different modules and Loss Functions.

**Effects of the Different Modules.** In this paper, our proposed FID-Net mainly uses the Frequency Feature Extractor (FFE) and Frequency Feature Refinement (FFR) modules to help the model improve feature representation. Therefore, we attempt to explore the effects of these modules via the following configurations: (1) In FID-Net, we remove FFE and FFR, **named FID-Net w/o None**; (2) In FFE, we remove Lightweight Mask Generation Block (LMGB), **named FID-Net w/o Mask**; (3) In FFR, we remove Spatial Attention Unit (SAU), **named FID-Net w/o SAU**; (4) In FFR, we remove Channel Attention Unit (CAU), **named FID-Net w/o CAU**; (5) The overall of our proposed FID-Net. The quantitative and qualitative experiments are reported in Table 3 and Fig. 9, respectively. Especially, LMGB can help model gain the 5.3 dB improvements of PSNR with the decomposition of input spectrum. In addition, SAU and CAU can bring 7.6 dB and 5.2 dB improvements, respec-



**Fig. 10.** Visual results on different decomposition strategies of LMGB.

tively. Furthermore, visual appearance results also demonstrate the effectiveness of these modules.

**Effects of the Different Loss Functions.** In this paper, our proposed FID-Net use three popular losses, including L1-based loss  $Loss_{l1}$ , SSIM loss  $Loss_{ssim}$  and perceptual loss  $Loss_{per}$ , to optimize the model training. To explore the effects of each loss, we conduct the ablation study with the following configurations: (1) We remove L1-based loss, **named FID-Net w/o  $Loss_{l1}$** ; (2) We remove SSIM loss, **named FID-Net w/o  $Loss_{ssim}$** ; (3) We remove perceptual loss, **named FID-Net w/o  $Loss_{per}$** ; (4) The overall of our proposed FID-Net. The quantitative and qualitative results of Table 3 and Fig. 9 respectively show the benefits of each loss, especially, SSIM and perceptual losses can contribute to structural image details and image fidelity.

**Effects of the Decomposition Strategy of Lightweight Mask Generation Block.** In this paper, we design an LMGB to adaptively generate masks to decompose the frequency into low-frequency and high-frequency parts. To verify the effectiveness of this strategy, we conduct the ablation study with the following configurations: (1) We refer to the method [25] and use Average Pooling (AP) to obtain the low-frequency feature region. After that, we subtract it from the input feature to obtain the high-frequency part. We named this process as **FID-Net w/ AP**; (2) We adopt the Gaussian Filter (GF) with size  $5 \times 5$  to obtain the low-frequency part and use the same way to identify the high-frequency part. We named this process as **FID-Net w/ GF**. (3) The overall of our proposed FID-Net. From the reports of Fig. 10 and Table 3, our proposed adaptive decomposition strategy can help produce mostly clear and visually appealing results, which also outperform other strategies across all metrics.

## 5 Conclusion

In this paper, we design a simple but effective Frequency Information-oriented Deshadow Network (FID-Net) that can handle complex shadows in document images and improve visual quality. In particular, our model design is motivated by the observation that different shadows affect distinct frequency bands. Thus,



in the basic unit of FID-Net, we propose two novel components: Frequency Feature Extractor (FFE) and Frequency Feature Refinement (FFR). Each of them has its own contributions to deshadowing. FFE aims to learn specific frequency elements guided by an adaptive decomposition of the input spectral characteristics. While FFR further refines the features through information exchange operations. With the cooperation of these two modules, our proposed FID-Net can achieve high-quality deshadowing with detail preservation and outperform state-of-the-art on several popular datasets.

**Acknowledgement.** This work is supported by Fujian Provincial Young and Middle-aged Teachers Educational Research Project (JZ230050), Unveiling and Leading Projects of Xiamen (No. 3502Z20241011), Open Project of the State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS2024101), Natural Science Foundation of Xiamen (3502Z202373058), and Fujian Key Technological Innovation and Industrialization Projects (2023XQ023).

## References

1. Long, S., He, X., Yao, C.: Scene text detection and recognition: the deep learning era. *Int. J. Comput. Vis.* **129**(1), 161–184 (2021)
2. Wang, Y., Xie, H., Zha, Z.-J., Xing, M., Fu, Z., Zhang, Y.: ContourNet: taking a further step toward accurate arbitrary-shaped scene text detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11753–11762 (2020)
3. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C.D., Finn, C.: DetectGPT: zero-shot machine-generated text detection using probability curvature. In: *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 24950–24962 (2023)
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 (AAAI)*, vol. 34, pp. 11474–11481 (2020)
5. Li, M., et al.: TrOCR: transformer-based optical character recognition with pre-trained models. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 37, pp. 13094–13102 (2023)
6. Bako, S., Darabi, S., Shechtman, E., Wang, J., Sunkavalli, K., Sen, P.: Removing shadows from images of documents. In: Lai, S.-H., Lepetit, V., Nishino, K., Sato, Y. (eds.) *Computer Vision – ACCV 2016: 13th Asian Conference on Computer Vision*, Taipei, Taiwan, November 20–24, 2016, Revised Selected Papers, Part III, pp. 173–183. Springer International Publishing, Cham (2017). [https://doi.org/10.1007/978-3-319-54187-7\\_12](https://doi.org/10.1007/978-3-319-54187-7_12)
7. Kligler, N., Katz, S., Tal, A.: Document enhancement using visibility detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2374–2382 (2018)
8. Jung, S., Hasan, M.A., Kim, C.: Water-Filling: an efficient algorithm for digitized document shadow removal. In: Jawahar, C.V., Li, H., Mori, G., Schindler, K. (eds.) *Computer Vision – ACCV 2018: 14th Asian Conference on Computer Vision*, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part I, pp. 398–414. Springer International Publishing, Cham (2019). [https://doi.org/10.1007/978-3-030-20887-5\\_25](https://doi.org/10.1007/978-3-030-20887-5_25)

9. Oliveira, D.M., Lins, R.D., de França Pereira e Silva, G.: Shading removal of illustrated documents. In: Kamel, M., Campilho, A. (eds.) *Image Analysis and Recognition*, pp. 308–317. Springer, Berlin, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-39094-4\\_35](https://doi.org/10.1007/978-3-642-39094-4_35)
10. Finlayson, G.D., Drew, M.S., Lu, C.: Entropy minimization for shadow removal. *Int. J. Comput. Vis.* **85**(1), 35–57 (2009)
11. Finlayson, G.D., Hordley, S.D., Lu, C., Drew, M.S.: On the removal of shadows from images. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(1), 59–68 (2005)
12. Guo, R., Dai, Q., Hoiem, D.: Paired regions for shadow detection and removal. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(12), 2956–2967 (2012)
13. Gong, H., Cosker, D.: Interactive removal and ground truth for difficult shadow scenes. *JOSA A* **33**(9), 1798–1811 (2016)
14. Qu, L., Tian, J., He, S., Tang, Y., Lau, R.W.: DeshadowNet: a multi-context embedding deep network for shadow removal. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4067–4075 (2017)
15. Hu, X., Fu, C.-W., Zhu, L., Qin, J., Heng, P.-A.: Direction-aware spatial context features for shadow detection and removal. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**(11), 2795–2808 (2019)
16. Chen, Z., Long, C., Zhang, L., Xiao, C.: CANet: a context-aware network for shadow removal. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 4743–4752 (2021)
17. Cun, X., Pun, C.-M., Shi, C.: Towards ghost-free shadow removal via dual hierarchical aggregation network and shadow matting GAN. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 34, pp. 10680–10687 (2020)
18. Brown, M.S., Tsoi, Y.-C.: Geometric and shading correction for images of printed materials using boundary. *IEEE Trans. Image Process.* **15**(6), 1544–1554 (2006)
19. Lin, Y.-H., Chen, W.-C., Chuang, Y.-Y.: BEDSR-Net: a deep shadow removal network from a single document image. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12905–12914 (2020)
20. Li, Z., Chen, X., Pun, C.-M., Cun, X.: High-resolution document shadow removal via a large-scale real-world dataset and a frequency-aware shadow erasing net. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 12415–12424. IEEE (2023)
21. Zhang, L., He, Y., Zhang, Q., Liu, Z., Zhang, X., Xiao, C.: Document image shadow removal guided by color-aware background. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1818–1827 (2023)
22. Fu, L., et al.: Auto-exposure fusion for single-image shadow removal. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10571–10580 (2021)
23. Guo, L., Huang, S., Liu, D., Cheng, H., Wen, B.: ShadowFormer: global context helps image shadow removal. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* (2023)
24. Xing, L., Tian, Z., Huang, W., Scott, M.R.: Convolutional character networks. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 9126–9136 (2019)
25. Cui, Y., Ren, W., Cao, X., Knoll, A.: Focal network for image restoration. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13001–13011 (2023)



# Improving Online Handwriting Recognition with Transfer Learning Using Out-of-Domain and Different-Dimensional Sources

Jiseok Lee<sup>(✉)</sup>, Masaki Akiba, and Brian Kenji Iwana<sup>ID</sup>

Graduate School of Information Science and Electrical Engineering,  
Kyushu University, Fukuoka, Japan  
{jiseok.lee,masaki.akiba}@human.ait.kyushu-u.ac.jp,  
iwana@ait.kyushu-u.ac.jp

**Abstract.** Online handwriting recognition is a widely used technique in our daily lives. Furthermore, deep learning has become one of the most popular and influential methods for online handwriting recognition. However, artificial neural networks typically require massive datasets. Transfer learning is a standard method to overcome the problem of lack of data. Usually, transfer learning works by initiating a network with trained weights and fine-tuning with a smaller dataset. Still, obtaining large amounts of online handwriting can be difficult for pre-training networks. Therefore, we propose pre-training with data sources with dimensions different from handwriting. Namely, we propose using univariate or multivariate data as a source dataset for two-dimensional target data by embedding out-of-domain time series of different dimensions into two-dimensional space. We evaluated the proposed method with four handwritten character datasets: a numerical digit dataset, an uppercase alphabet dataset, a lowercase alphabet dataset, and a Chinese character dataset. Through the evaluation, we demonstrate that transfer learning from datasets with a different dimensionality as online handwriting is possible.

**Keywords:** Online Handwriting Recognition · Transfer Learning · Time Series Recognition

## 1 Introduction

Nowadays, machine learning with deep neural networks [16] is one of the most popular approaches for handwritten character recognition [1], including both offline and online handwriting recognition. While offline handwriting recognition deals with static-rendered images of handwritten text, online handwriting recognition focuses on directly using the data input from the capturing device

---

This work was partially supported by MEXT-Japan (Grant No. 23K16949).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025  
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15331, pp. 61–75, 2025.  
[https://doi.org/10.1007/978-3-031-78119-3\\_5](https://doi.org/10.1007/978-3-031-78119-3_5)



as a sequence of coordinates and features. However, this introduces additional issues, such as temporal distortions, rate, strokes, etc.

Thus, online handwriting recognition systems employ temporal neural networks, such as Recurrent Neural Networks (RNN) [27] and temporal Convolutional Neural Networks (CNN) [15]. For example, there are many neural networks used for various online handwriting recognition fields, such as online handwritten character classification [17, 23, 25, 30], online signature verification [34, 38], online handwritten mathematical expression recognition [42], etc.

However, deep neural networks often require a lot of annotated data for practical training. Getting large amounts of annotated data can be difficult for online handwriting, especially for languages with complex alphabets. Furthermore, training can demand much time for the training loss to converge.

One solution to training networks with insufficient data is to use transfer learning. Transfer learning leverages knowledge from pre-trained models and adapts it to a target task with limited data. Transfer learning can effectively improve the performance of neural networks by fine-tuning the pre-trained model’s parameters on the specific dataset. While the pre-trained model converges the training loss faster, using transfer learning can reduce the need for extensive annotated data.

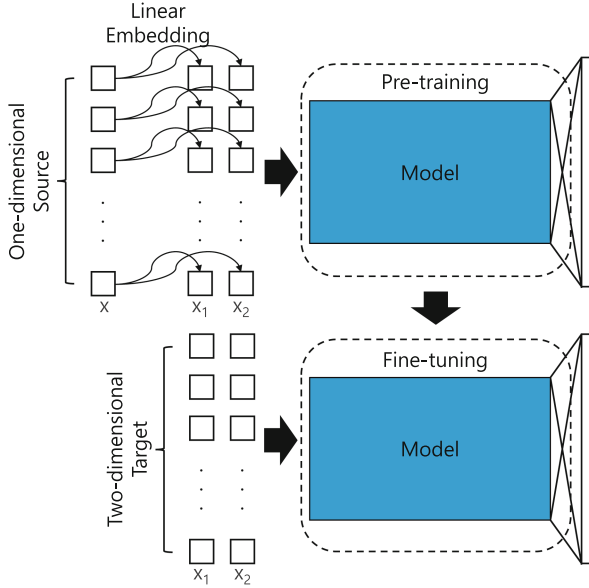
One issue with using transfer learning is that to transfer the parameters of one network to another, the input sizes should be the same size and dimensionality. Thus, to use pre-trained models for online handwriting, the network needs to be pre-trained using similar data. This is less of an issue in the image domain because images are typically one, three, or four-dimensional. However, for online handwriting, finding similar dimensional data can be challenging due to the wide variety of time series characteristics. Thus, in order to utilize various datasets as source data for pre-training, we propose a method of pre-training with *different-dimensional* data, i.e., time series data that is a different dimension than the target and fine-tuning with the online handwriting.

Specifically, we propose using univariate and multivariate time series data, which there is an abundance of, to train temporal CNNs using two-dimensional target data. Figure 1 shows the proposed model. In order to utilize different-dimensional data, we first project the elements of the input time series into two-dimensional space using a learned linear embedding. By doing so, the dimensionality of the input time series can be changed to match the target data. The source data is then used to train the model, and the common part’s parameters can be transferred as usual.

Notably, this allows us to use out-of-domain data, which many exist, to pre-train networks for online handwriting recognition. This is possible because some abstract features in the out-of-domain data may exist that are similar to features that are important for online handwriting.

The contribution of this paper is as follows:

- We propose a new method of pre-training neural networks with a different dimensionality source dataset as the target dataset. This is done using an embedding layer that is added during the pre-training process.



**Fig. 1.** The proposed source embedding for pre-training models with different dimensionality time series. For the pre-training stage, the one-dimensional dataset is embedded into a two-dimensional space, and the weight learned from pre-training from different dimensional data becomes the initial weights of fine-tuning of a given two-dimensional online handwriting dataset.

- We demonstrate that, through our proposed method, transfer learning between different dimensional data is possible.
- We evaluated the proposed method with four online handwritten character datasets: Unipen 1a, Unipen 1b, Unipen 1c, and the Chinese Academy of Sciences Institute of Automation (CASIA) dataset.

## 2 Related Work

In computer vision, there are many examples of pre-training a model with a different task to target task. Taskonomy [40] is a well-known work demonstrating inter-task transferability among multiple computer vision tasks with a graphical structure. For example, they demonstrated that the 2D key points task can be improved with pre-training with unsupervised learning tasks such as denoising, colorization, and in-painting [40]. Also, there are many examples of transfer learning from non-related tasks improving performance in the medical image domain. For example, pre-training ImageNet [7] improved convergence and accuracy in liver lesion segmentation and classification tasks [13], and pre-training JFT [31] and ImageNet [7] showed improvement in Mammography, CheXpert, and Dermatology tasks [21].

## 2.1 Online Handwriting Recognition

There have been many works of online handwriting recognition since hand-held computers such as PDAs became widespread [24]. Online handwriting recognition has the benefit of utilizing its dynamic information as a time series, and it has been shown to sometimes have better performance than offline handwriting recognition [24]. Machine learning has recently been widely adopted for online handwriting recognition. For example, temporal CNNs have shown outstanding performance in Chinese character recognition [39], and also recurrent neural networks (RNNs) have shown to be effective in online handwriting recognition [10, 22].

## 2.2 Transfer Learning on Online Handwriting Recognition

Unlike offline handwriting recognition, there have only been a few works to adopt transfer learning in online handwriting recognition. In one example, transfer learning increased the performance of online Turkish handwriting recognition from 49% to 85% [33]. Chakraborty et al. [3] used transfer learning for handwritten Bangla and Devanagari. Mehralian et al. [20] propose pre-training networks using a self-supervised method which includes stroke masking.

# 3 Transfer Learning Using Different-Dimensional Sources

## 3.1 Transfer Learning

Transfer learning is a powerful approach in machine learning that has been widely used, especially in recognition and classification. The primary use of transfer learning is to pre-train models to make up for the lack of annotated data on specific tasks [26]. However, it is also used as a standard practice to initialize neural networks, even when large amounts of annotated data exist. This is because pre-training models can increase the speed of training convergence and overall accuracy [5, 9, 35].

Transfer learning typically involves two main steps: pre-training and fine-tuning. A model is first trained on a large *source* dataset during pre-training. The source dataset can be from a similar task as the *target* dataset, such as in domain adaptation [36], or an unrelated task. The second step, fine-tuning, uses the weights trained by the pre-training as an initialization for the target task. The fine-tuning process updates the weights of the pre-trained model using the new task.

## 3.2 Linear Embedding

One common limitation of transfer learning is that the target and source datasets should be the same size and dimensionality. In the image domain, dimensionality is not usually an issue because most images have common dimensionalities,

e.g., 1, 3, or 4. For example, it is common practice to use the weights of well-known CNNs pre-trained on ImageNet [7].

Unlike images, time series datasets have various dimensionalities depending on the application and the capture device. Online handwriting, in particular, can have two or more dimensions. Accordingly, there are few applicable time series datasets outside of online handwriting datasets that can be used for pre-training. Following this, we propose to perform transfer learning using an out-of-domain source of a different dimensionality to train online handwriting. We propose training with data from a different dimensional source than online handwriting to do this.

Since the inputs are different dimensions, we propose using a learned linear embedding between the input and the model, as shown in Fig. 1. A tiny dense layer with shared weights between time steps changes the input into two dimensions. Namely, an embedding matrix  $W$  is multiplied by each one-dimensional element of the input time series. A two-dimensional time series  $\mathbf{x}'$  is created by:

$$\mathbf{x}' = Wx_1, \dots, Wx_t, \dots, Wx_T, \quad (1)$$

where  $\mathbf{x} = x_1, \dots, x_t, \dots, x_T$  is the original time series and embedding matrix  $W$  is of size  $(D, I)$ . Each element  $x_t$  can be univariate or multivariate, and  $D$  is the dimensionality of the target, and  $I$  is the dimensionality of the source. By adding the embedding layer, it is possible to train a temporal neural network using an input of any dimensionality with a model that would typically require a different dimensionality.

### 3.3 Pre-training Using Different-Dimensional Sources

In order to exploit transfer learning from out-of-domain data sources, we propose a method to pre-train the model with data that has a different dimension from the target handwriting. The idea is that domains with more data can be used to pre-train the model for specific targets, such as handwriting. Accordingly, the model will learn to extract features from the source domain, hoping that some are common to the target domain. This is similar to using Imagenet [7] to pre-train image neural networks for specialized out-of-domain tasks.

Namely, using the proposed embedding layer, the weights of the model can be fixed for data that would be the same size as what would be required for transfer learning to the target dataset. Specifically, as shown in Fig. 1, the network is trained with one-dimensional data using the model with the embedding layer. Notably, the embedding is only used for the pre-training step, not fine-tuning the online handwritten characters. Next, the rest of the model's parameters, except the output layer, are used in the same way as pre-training for fine-tuning. As in a typical transfer-learning fashion, the output layer is excluded because the number of classes may differ between the source and target tasks.

## 4 Experimental Result

### 4.1 Datasets

We use eight out-of-domain time series datasets for the experiments, four univariate and, four multivariate, and four online handwritten character datasets.

**Univariate Source Datasets.** We show the effect of pre-training for the source datasets with one of four one-dimensional out-of-domain time series datasets. The datasets were selected due to their classification tasks, similar in length to the online handwritten characters, and having large training sets. The four datasets consist of the following.

- *Crop* [32]. Crop consists of spectral measurements of a pixel from aerial photography. The objective is to classify the pixel by the type of land. There are 7,200 patterns in the training set, 46 time steps, and 24 classes.
- *FordA* [6]. This dataset is from the 2018 University of California Riverside (UCR) Time Series Archive [6]. The task of the dataset is to judge whether there is a symptom in an automotive subsystem. The training set has 3,601 patterns of 500 time steps long and two classes.
- *InsectSound* [4]. InsectSound is gained from the UCR computational entomology group. The task of this dataset is to classify flying insects from their sound. This dataset has 25,000 patterns of 600 time steps long and ten classes.
- *NonInvasiveFetalECGThorax1* [28]. NonInvasiveFetalECGThorax1 is from the UCR Time Series Archive. This time series dataset is comprised of fetal electrocardiographic (FECG) signals of the throat. The training set has 1,800 patterns of 750 time steps long and 42 classes.

**Multivariate Source Datasets.** In addition to the one-dimensional datasets, we examined four sources with varying numbers of dimensions.

- *FaceDetection* [8]. The task of this dataset is to classify whether the subject is watching a human face picture or a scrambled image from recorded Magnetoencephalography (MEG) signals. This dataset has 5,890 patterns of 62 time steps with 144 dimensions and two classes in the training set.
- *InsectWingbeat* [4]. This dataset is taken from the UCR computational entomology group. The task is to classify flying insects from the power spectrum of the sound of insects passing through a sensor. Each dimension of the data is a frequency band of the spectrogram. This dataset has 25,000 patterns in the training set, 10 classes, and 200 dimensions.
- *SpokenArabicDigits* [2]. The patterns of this dataset are derived from sounds spoken by native Arabic speakers. The training set has 6,599 patterns of 93 time steps with 13 dimensions and 10 classes corresponding to the 13 Mel Frequency Cepstral Coefficients.

- *WalkingSittingStanding* [41]. The task of this dataset is to classify signals recorded by the wearable sensors in six activities: walking, walking upstairs, walking downstairs, sitting, standing, and lying. The training set of this dataset contains 7,352 patterns of 206 time steps with three dimensions and six classes.

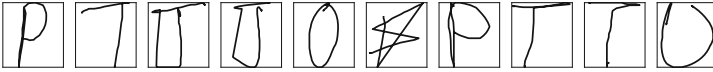
**Online Handwriting Datasets.** To evaluate our proposed method, we used four online handwriting datasets: three from the UNIPEN online handwriting database (Unipen) [11] datasets and one from the CASIA Online and Offline Chinese Handwriting Databases (CASIA) [18] dataset.

- *Unipen 1a* Unipen 1a is a dataset of the Arabic numeral digits. This dataset is comprised of 12,950 time series out of 10 classes from “0” to “9”. Figure 2 shows ten sample data of Unipen 1a.



**Fig. 2.** Sample data of Unipen 1a.

- *Unipen 1b* Unipen 1b is a dataset of the English uppercase alphabet characters. This dataset is comprised of 12,298 time series with 26 classes from “A” to “Z”. Figure 3 shows ten sample data of Unipen 1b.



**Fig. 3.** Sample data of Unipen 1b.

- *Unipen 1c* Unipen 1c is a dataset of the English lowercase alphabet characters. This dataset is comprised of 12,298 data out of 26 classes from “a” to “z”. Figure 4 shows ten sample data of Unipen 1c.
- *CASIA* CASIA is a dataset that contains online and offline Chinese character handwriting data. We used the online handwriting characters and not the offline characters for the research. The online dataset has 1,570,051 online handwritten Chinese characters with 3,740 labels. Figure 5 shows ten random sample data of the CASIA.



Fig. 4. Sample data of Unipen 1c.



Fig. 5. Sample data of CASIA.

## 4.2 Dataset Regularization

We rescaled timesteps of online handwritten characters and one-dimensional time series data into 256 time steps through Gaussian Smoothing. Also, the datasets are regularized so that the training dataset’s minimum and maximum values are -1 and 1.

Regularization is performed so that the values of each of the datasets are of the same magnitude and to remove any bias based on the range.

## 4.3 Model and Settings

For the experiments, we used a Visual Geometry Group Network (VGG) [29] modified with 1D convolutions and 1D max pooling to use with time series. The VGG is made of four convolutional blocks. Each block has three convolutional layers with 64, 128, 256, and 512 filters, respectively. Max pooling is used at the end of each block. Finally, there are three fully connected layers: two layers have 1,024 nodes, and the output layer.

As illustrated in Fig. 1, only the convolutional layers are transferred for the transfer learning. Also, in addition to VGG architecture, we use a small embedding layer between the convolutional layer and the input layer for the pre-training network only during training to implement our proposed method.

To train the network, we use an Adaptive Moment Estimation (Adam) optimizer [14] with an initial learning rate of 0.0001. We trained 10,000 iterations for pre-training and 20,000 and 200,000 iterations for fine-tuning Unipen datasets and CASIA, respectively. CASIA is trained longer due to having a much more extensive training set than the other datasets. The training is conducted with batch size 32 and cross-entropy loss.

## 4.4 Comparative Evaluation

To evaluate the proposed method, we compare the following evaluations.

- **w/o Transfer Learning.** This evaluation is trained without transfer learning. The weights are initialized using the uniform distribution proposed by He et al. [12].

- **Transfer Learning.** We perform typical transfer learning between the online handwriting datasets. For example, we show the results of fine-tuning Unipen 1a on a model pre-trained with Unipen 1b, 1c, and CASIA.
- **TL w/ Retrained First Layer.** One simple way to use transfer learning between datasets of different dimensions is to pre-train a network and replace the conflicting layers with appropriate-sized layers. This way, we created an evaluation that replaces the first convolutional layer with the number of channels to match the input dimensions. The rest of the layers use transfer learning like usual.
- **TL w/ Proposed.** We evaluate the proposed method of using an embedding layer to resize the input to the dimensionality of the online handwriting.

Experiments are done to demonstrate that not only is pre-training with out-of-domain different-dimensional data useful, but it also matches and sometimes surpasses the accuracy of in-domain same-dimensional data. However, it should be noted that pre-training with the cross versions of Unipen might be considered data leakage due to most of the writers being the same across the datasets.

**Evaluation Results.** The experimental results of the Unipen and the CASIA are shown in Table 1. For the Unipen 1a dataset, transfer learning with resized first layer showed 99.08% accuracy, pre-trained with SpokenArabicDigits, had a 98.85% accuracy compared to 98.23% accuracy without transfer learning. Furthermore, this improvement is higher than the highest performance improvement with the two-dimensional online handwriting dataset (Unipen 1b). The Unipen 1b and 1c, the proposed method performed better than any of the comparison online handwriting datasets. Specifically, for Unipen 1b, the best dataset to transfer from was the one-dimensional dataset, FordA and InsectSound, and the 200-dimensional dataset, InsectWingbeat. Also, for Unipen 1c, Crop had the highest results from transfer learning. For the CAISA, transfer learning showed significant effectiveness due to the difficulty of the task compared to the others. Specifically, our proposed method with the one-dimensional dataset, Crop, showed the best recognition accuracy of all. Transfer learning with a resized first layer also showed a lower performance than our proposed method overall. Also, as mentioned previously, using transfer learning between Unipen subsets might not be a fair comparison due to having the same writers, although they have different characters.

One interesting observation is that using transfer learning from CASIA, i.e., Chinese characters, to Unipen tends to hurt the overall accuracy of the network. When pre-training from CASIA, Unipen 1a and 1c had worse results when compared to initializing by random. CASIA is the largest dataset of the evaluated datasets, and thus, it might generally be a first choice for pre-training in a naive application using online handwriting.

Again, from the result of the CASIA, interestingly, these datasets are completely different tasks and are unrelated to online handwriting. Transfer learning from the Unipen datasets performed worse despite being similar to CASIA.



**Table 1.** Experimental result of the proposed method with datasets of the Unipen.

Source	Dimensions	Accuracy (%)			
		Unipen 1a	Unipen 1b	Unipen 1c	CASIA
w/o Transfer Learning		98.23	96.76	96.68	59.11
Transfer Learning					
Unipen 1a	2	–	97.89	96.60	75.01
Unipen 1b	2	98.69	–	96.19	81.19
Unipen 1c	2	98.31	96.84	–	77.02
CASIA	2	97.85	97.17	96.03	–
TL w/ Retrained First Layer					
Crop	1	97.92	96.19	96.03	81.05
FordA	1	98.38	97.17	96.36	69.87
InsectSound	1	98.92	97.49	95.71	57.22
NonInvasiveFetalECGThorax1	1	98.38	97.57	96.28	78.96
FaceDetection	144	98.62	97.41	96.76	76.62
InsectWingbeat	200	98.38	97.73	96.44	76.42
SpokenArabicDigits	13	98.76	97.73	96.52	69.98
WalkingSittingStanding	6	<b>99.08</b>	97.17	96.68	74.20
TL w/ Proposed					
Crop	1	98.31	96.76	<b>97.09</b>	<b>84.16</b>
FordA	1	98.00	<b>98.06</b>	96.84	61.49
InsectSound	1	98.54	<b>98.06</b>	96.52	72.48
NonInvasiveFetalECGThorax1	1	98.69	97.49	96.19	84.05
FaceDetection	144	98.69	97.49	96.52	72.45
InsectWingbeat	200	98.54	<b>98.06</b>	96.36	79.74
SpokenArabicDigits	13	98.85	97.09	96.92	72.33
WalkingSittingStanding	6	98.31	97.33	96.11	82.40

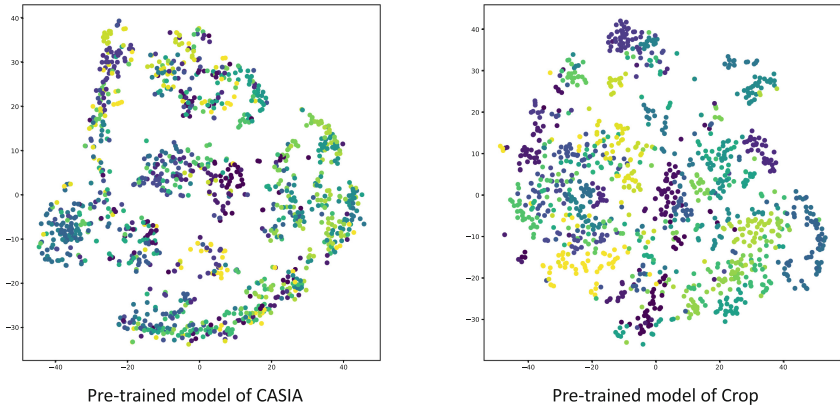
## 5 Discussion

### 5.1 Discriminability of the Pre-trained Networks

We examine the features learned from the convolutions to examine the reason for the improved performance. Through T-distributed Stochastic Neighbor Embedding (t-SNE) [19], the flattened output of the last convolutional block is visualized, i.e., dimensionality reduction using t-SNE has used the  $16 \times 512$  feature vector from the last convolutional layer. We examine before and after fine-tuning and compare pre-training with online handwriting and the proposed out-of-domain one-dimensional time series.

Figure 6 shows how the features of test data of Unipen 1c are distributed. The subfigure on the left shows the distribution of Unipen 1c with the model pre-

trained with the CASIA, and the one on the right is with Crop. It should be noted that this is after pre-training but before fine-tuning. The distribution of features with the model pre-trained with CASIA shows that patterns overlap significantly between the classes. On the other hand, the distribution of the with the model pre-trained model with Crop showed much less overlap. This demonstrates how CASIA is not necessarily a good source of pre-training for online handwriting, whereas out-of-domain and different-dimensional data sources might be.



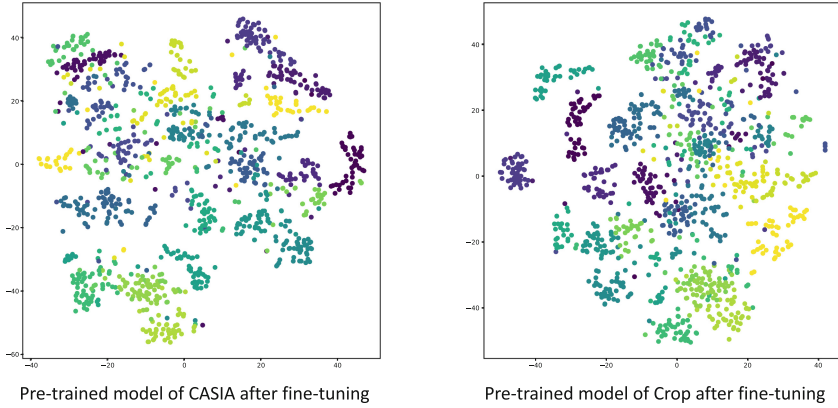
**Fig. 6.** The features of Unipen 1c provided the model pre-trained with CASIA (left) and Crop (right). The colors represent the different classes.

We also visualize the features after fine-tuning. In Fig. 7, the features of Unipen 1c in a model pre-trained by CASIA and Crop and fine-tuned using Unipen 1c are visualized using t-SNE. After fine-tuning, the two distributions become similarly clustered. However, the model with our proposed method still seems to have a better separation between the classes than the model pre-trained with CASIA.

By observing the discriminability of the distributions of the features, we can intuitively infer why the proposed method using Crop ended up having a 1.06% increase in accuracy for Unipen 1c over using CASIA. Because the pre-trained model has a better initialization, it can be fine-tuned towards a more robust classifier.

## 5.2 Comparison Between Datasets

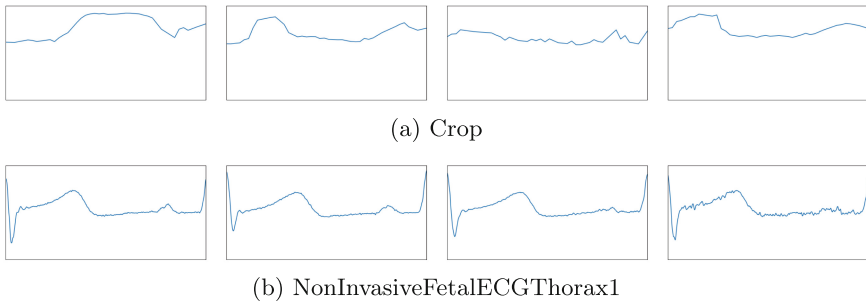
In general, with a small-scale target dataset, pre-training with a closely related source dataset is considered to allow for more efficient training while reducing the risk of overfitting. However, there is also a risk that transfer learning with the wrong source can ruin performance. Therefore, for online handwritten characters, transferring knowledge from different-looking languages, i.e., Latin characters



**Fig. 7.** The features of Unipen 1c provided the model pre-trained with CASIA (left) and Crop (right) and after fine-tuning. The colors represent the different classes.

of Unipen versus Chinese characters of CASIA, might be victim to *negative transfer* [37].

However, despite the different-dimensional time series coming from different domains such as online handwriting, the results show that the features learned in the convolutions helped positively initialize the network. Figure 8 shows examples of time series from the Crop and NonInvasiveFetalECGThorax1. While pre-training with CASIA had a negative effect on the Unipen datasets, the proposed method worked especially well with these two datasets. From the figure, these two datasets have fairly simple time series with only subtle differences between classes.



**Fig. 8.** Examples of time series patterns from (a) Crop and (b) NonInvasiveFetalECGThorax1. The examples are univariate time series and the x-axis represents the time dimension. Each one is from a different class.

## 6 Conclusion

In this research, we propose the source embedding model to solve the problem of a small number of available source datasets for transfer learning of online handwriting datasets. Our proposed method makes it possible to train with a dataset with different dimensions to target tasks in the target task's dimension for transfer learning of isolated online handwriting character recognition.

We evaluated our proposed method with four isolated online handwriting character datasets: Unipen 1a, Unipen 1b, Unipen 1c, and CASIA. Our proposed method showed its potential for all handwriting datasets by improving the classification performance. All the cases in the experiment with Unipen did not show good results; however, our proposed method showed a significant improvement for the CASIA dataset compared to when it trained without transfer learning.

In the future, we want to examine our method in various domains, not only having a two-dimensional target but also every possible combination of dimensions. Also, we will investigate how to predict the transfer learning performance of our proposed method before the training stage.

## References

1. Al-Taei, M.M., Neji, S.B.H., Frikha, M.: Handwritten recognition: a survey. In: IEEE IPAS, pp. 199–205. IEEE (2020)
2. Bedda, M., Hammami, N.: Spoken Arabic digit. UCI Machine Learning Repository (2010). <https://doi.org/10.24432/C52C9Q>
3. Chakraborty, R., Saha, S., Bhattacharyya, A., Sen, S., Sarkar, R., Roy, K.: Recognition of online handwritten Bangla and Devanagari basic characters: a transfer learning approach. In: Singh, S.K., Roy, P., Raman, B., Nagabhushan, P. (eds.) Computer Vision and Image Processing: 5th International Conference, CVIP 2020, Prayagraj, India, December 4-6, 2020, Revised Selected Papers, Part II, pp. 530–541. Springer Singapore, Singapore (2021). [https://doi.org/10.1007/978-981-16-1092-9\\_45](https://doi.org/10.1007/978-981-16-1092-9_45)
4. Chen, Y., Why, A., Batista, G., Mafra-Neto, A., Keogh, E.: Flying insect classification with inexpensive sensors. *J. Insect Behav.* **27**, 657–677 (2014)
5. Chui, K.T., Arya, V., Band, S.S., Alhalabi, M., Liu, R.W., Chi, H.R.: Facilitating innovation and knowledge transfer between homogeneous and heterogeneous datasets: generic incremental transfer learning approach and multidisciplinary studies. *J. Innov. Knowl.* **8**(2), 100313 (2023)
6. Dau, H.A., et al.: Hexagon-ML: the UCR time series classification archive (2018). [https://www.cs.ucr.edu/~eamonn/time\\_series\\_data\\_2018/](https://www.cs.ucr.edu/~eamonn/time_series_data_2018/)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR, pp. 248–255 (2009)
8. Emanuele, Mosi, P.A.: DecMeg2014 - decoding the human brain (2014). <https://kaggle.com/competitions/decoding-the-human-brain>
9. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Transfer learning for time series classification. In: IEEE ICBD (2018). <https://doi.org/10.1109/bigdata.2018.8621990>
10. Graves, A., Liwicki, M., Bunke, H., Schmidhuber, J., Fernández, S.: Unconstrained on-line handwriting recognition with recurrent neural networks. *NeurIPS* **20** (2007)

11. Guyon, I., Schomaker, L., Plamondon, R., Liberman, M., Janet, S.: Unipen project of on-line data exchange and recognizer benchmarks. In: ICPR. ICPR-94, vol. 2, pp. 29–33 (1994). <https://doi.org/10.1109/icpr.1994.576870>
12. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: ICCV, pp. 1026–1034 (2015)
13. Heker, M., Greenspan, H.: Joint liver lesion segmentation and classification via transfer learning. arXiv preprint [arXiv:2004.12352](https://arxiv.org/abs/2004.12352) (2020)
14. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
15. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998). <https://doi.org/10.1109/5.726791>
16. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015). <https://doi.org/10.1038/nature14539>
17. Li, Y., et al.: Fast and robust online handwritten Chinese character recognition with deep spatial & contextual information fusion network. IEEE Trans. Multimedia (2022)
18. Liu, C.L., Yin, F., Wang, D.H., Wang, Q.F.: CASIA online and offline Chinese handwriting databases. In: 2011 International Conference on Document Analysis and Recognition. IEEE (2011). <https://doi.org/10.1109/icdar.2011.17>
19. Van der Maaten, L., Hinton, G.: Visualizing data using T-SNE. J. Mach. Learn. Res. **9**(11) (2008)
20. Mehralian, P., BabaAli, B., Mohammadi, A.G.: Self-supervised representation learning for online handwriting text classification. arXiv preprint [arXiv:2310.06645](https://arxiv.org/abs/2310.06645) (2023)
21. Mustafa, B., et al.: Supervised transfer learning at scale for medical imaging. arXiv preprint [arXiv:2101.05913](https://arxiv.org/abs/2101.05913) (2021)
22. Nguyen, H.T., Nguyen, C.T., Bao, P.T., Nakagawa, M.: A database of unconstrained Vietnamese online handwriting and recognition experiments by recurrent neural networks. Pattern Recogn. **78**, 291–306 (2018)
23. Ott, F., et al.: Benchmarking online sequence-to-sequence and character-based handwriting recognition from IMU-enhanced pens. Int. J. Doc. Anal. Recogn. **25**(4), 385–414 (2022)
24. Plamondon, R., Srihari, S.N.: Online and off-line handwriting recognition: a comprehensive survey. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 63–84 (2000)
25. Popli, R., Kansal, I., Garg, A., Goyal, N., Garg, K.: Classification and recognition of online hand-written alphabets using machine learning methods. IOP Conf. Ser. Mater. Sci. Eng. **1022**(1), 012111 (2021). <https://doi.org/10.1088/1757-899x/1022/1/012111>
26. Ribani, R., Marengoni, M.: A survey of transfer learning for convolutional neural networks. In: SIBGRAPI-T, pp. 47–57 (2019)
27. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986). <https://doi.org/10.1038/323533a0>
28. Silva, I., et al.: Noninvasive fetal ECG: the PhysioNet/Computing in cardiology challenge 2013. In: Computing in Cardiology, pp. 149–152 (2013)
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
30. Singh, H., Sharma, R.K., Singh, V., Kumar, M.: Recognition of online handwritten Gurmukhi characters using recurrent neural network classifier. Soft. Comput. **25**, 6329–6338 (2021)

31. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 843–852 (2017)
32. Tan, C.W., Webb, G.I., Petitjean, F.: Indexing and classifying gigabytes of time series under time warping. In: SIAM ICDM, pp. 282–290 (2017)
33. TAŞDEMİR, E.F.B.: Recognition of online Turkish handwriting using transfer learning. *Gazi Univ. J. Sci. Part C: Design Technol.* **11**(3), 719–726 (2023)
34. Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Ortega-Garcia, J.: DeepSign: deep on-line signature verification. *IEEE Trans. Biometrics, Behav. Identity Sci.* **3**(2), 229–239 (2021)
35. Vásquez-Correa, J.C., et al.: Transfer learning helps to improve the accuracy to classify patients with different speech disorders in different languages. *Pattern Recogn. Lett.* **150**, 272–279 (2021)
36. Wang, M., Deng, W.: Deep visual domain adaptation: a survey. *Neurocomputing* **312**, 135–153 (2018)
37. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. *J. Big Data* **3**(1), 1–40 (2016)
38. 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: ICDAR, pp. 1103–1110 (2019). <https://doi.org/10.1109/ICDAR.2019.00179>
39. Yang, W., Jin, L., Xie, Z., Feng, Z.: Improved deep convolutional neural network for online handwritten Chinese character recognition using domain-specific knowledge. In: ICDAR, pp. 551–555. IEEE (2015)
40. Zamir, A.R., Sax, A., Shen, W., Guibas, L.J., Malik, J., Savarese, S.: Taskonomy: disentangling task transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3712–3722 (2018)
41. Zhang, X., Zhao, Z., Tsiligkaridis, T., Zitnik, M.: Self-supervised contrastive pre-training for time series via time-frequency consistency. *Adv. Neural. Inf. Process. Syst.* **35**, 3988–4003 (2022)
42. Zhelezniakov, D., Zaytsev, V., Radyvonenko, O.: Online handwritten mathematical expression recognition and applications: a survey. *IEEE Access* **9**, 38352–38373 (2021)



# ROISER: Towards Real World Semantic Entity Recognition from Visually-Rich Documents

Zening Lin<sup>1</sup>, Jiapeng Wang<sup>1</sup>, Wenhui Liao<sup>1</sup>, Weicong Dai<sup>2</sup>, Longfei Xiong<sup>2</sup>,  
and Lianwen Jin<sup>1</sup>(✉)

<sup>1</sup> South China University of Technology, Guangzhou, China  
{eeznlin,eejpwang,eelwh}@mail.scut.edu.cn, eelwj@scut.edu.cn

<sup>2</sup> Kingsoft Office, Zhuhai, China  
{daiweicong,xionglongfei}@wps.cn

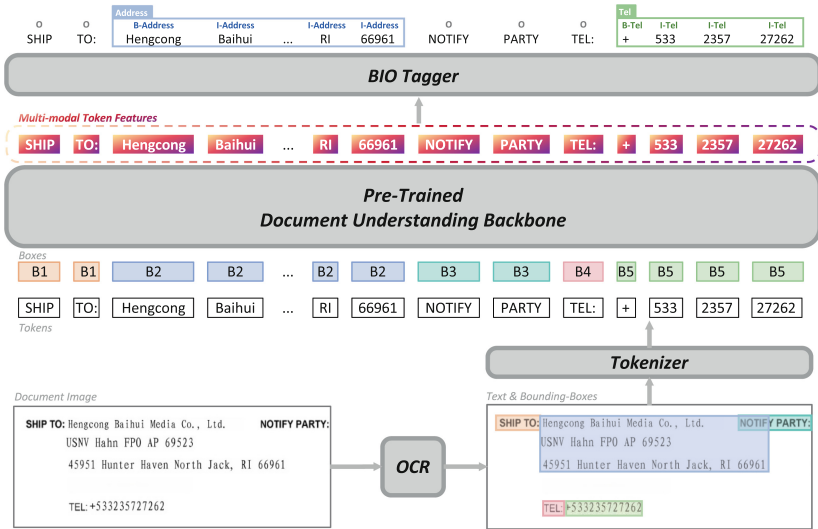
**Abstract.** Visual semantic entity recognition (visual SER) aims to extract contents that fall in key fields from the given visually-rich document image, and it has been widely applied across diverse scenarios. Most existing visual SER methods employ the BIO tagging schema to extract key entities, necessitating well-organized OCR results at the entity level as prior information. However, meeting this prerequisite is challenging in real-world applications. General OCR engines typically provide disordered line-level results, where entities with multiple text lines are split into several segments. Moreover, some adjacent entities may fall into the same detection box, posing challenges for accurate span detection and text aggregation. To address this issue, this paper introduces a novel framework, **ROISER** (**R**eal **w**orld **v**isual **S**emantic **E**ntity **R**ecognition), integrating entity line span detection, line aggregation, and line classification to achieve visual SER with real-world OCR input. Experiment results demonstrate that our model outperforms existing approaches on various benchmarks, showcasing its effectiveness and compatibility for practical applications.

**Keywords:** Visual Information Extraction · Document Understanding · Computer Vision

## 1 Introduction

Visually-rich documents like forms and receipts are widely seen in real-world scenarios. As digitization continues to advance, there is an increasing need to identify key information from document images. Visual semantic entity recognition is a vital step in this progress. It involves extracting texts that belong into predefined categories, such as retrieving the departure time from a train ticket [8] or analyzing purchased item details from receipts [13,22]. In recent years, various visual SER methods [10,12,17,27,29,30] have been proposed. Most of these methods follow a BIO-tagging [24] pipeline (Fig. 1): texts and boxes in the

document are first obtained through an OCR engine, then split into token-level information through a tokenizer. Subsequently, a pre-trained document understanding backbone is employed to generate multi-modal token features. Following this, a classifier (BIO tagger) assigns each token a tag, indicating whether it represents the beginning (B), inside (I), or outside (O) element of a desired entity. The contents of each key field can be parsed by aggregating the corresponding B/I tokens. These visual SER approaches have demonstrated impressive performance in tasks such as form entity identification [7, 32] and receipt information extraction [13, 22].



**Fig. 1.** The BIO-tagging pipeline employed by previous visual SER methods. Entity-level OCR results are processed by the Pre-Trained Document Understanding Backbone and generate multi-modal features for each token. A BIO Tagger is then applied to predict the span and category of the key entities.

Despite the advancements in visual SER, challenges persist when it comes to real-world applications. The aforementioned methods employ fine-annotated OCR labels as the model input, where bounding boxes are provided at the entity level, and texts within an entity are pre-aggregated and sorted in human reading order (Fig. 2 left). However, in real-world applications, document contents are obtained through general OCR engines, where bounding boxes are assigned at the line level (Fig. 2 right). For entities that span across multiple text lines, additional steps for text sorting and grouping are required, which can be problematic for complex layout documents. As depicted in Fig. 2 right, if we simply sort the line boxes from left-top to right-bottom based on their coordinates, the multi-line entity *Hengcong baihui ... RI 66961* will be disrupted by *NOTIFY PARTY*;, causing failure in the BIO-tagging scheme. Although we may employ



document layout analysis algorithms [6, 9] to produce more reasonable grouping results, these methods often involve adjustments of multiple hyper-parameters, making them cumbersome and lacking in robustness. Furthermore, inaccurate OCR detection results can lead to the contents of two entities being merged into the same bounding box, posing difficulties in separating them. As shown in Fig. 2 right, entity *TEL:* and *+533235727262* fall within the same box. These factors present challenges for applying existing visual SER methods in real-world applications.



**Fig. 2.** Illustration of span detection and text aggregation issue. Text bounding boxes are marked in different colours for differentiation. Left: entity-level OCR results used by previous methods. Right: outputs of real-world OCR engines, only line-level bounding boxes are provided, and contents of two entities may be merged.

To address the aforementioned issues, in this paper, we propose a novel framework **ROISER** (**R**eal **w**orld **v**isual **S**emantic **E**ntity **R**ecognition) to handle visual SER with imperfect OCR inputs. Specifically, given the document image and its corresponding line-level OCR results, multi-modal token features are first obtained through a pre-trained document understanding backbone. A novel downstream head is then employed to perform the following three steps: (1) entity line span detection, which serves to identify, separate, and extract entity lines from the token sequence; (2) line aggregation, which connects entity lines belonging to the same field; and (3) line classification, which determines the key category of each entity. By combining the predictions from these three steps, the model effectively extracts organized key information from unstructured OCR input, eliminating the need for rectified input as required in the original BIO-tagging pipeline. Extensive experiments on five public visual SER benchmarks and our in-house dataset can demonstrate the effectiveness of our method.

Our main contribution can be summarized as follows:

- We propose ROISER, a novel pipeline that integrates entity line span detection, line aggregation, and line classification for visual SER. The model effectively addresses the challenges associated with entity extraction and aggregation in practical scenarios.
- ROISER surpasses existing visual SER pipelines on various benchmarks when collaborating with advanced document understanding backbone, demonstrating its versatility and effectiveness.

## 2 Related Work

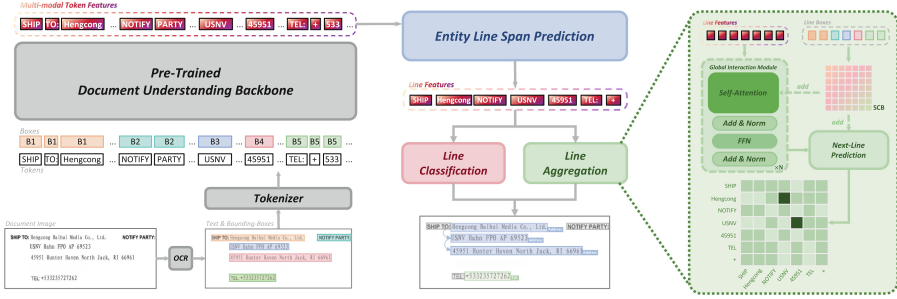
### 2.1 Visual SER with BIO-Tagging Pipeline

With the advancements in deep learning techniques, various transformer-based visual SER approaches have been proposed in recent years. LayoutLM [30] employs a BERT-like [5] architecture that incorporates bounding box embeddings to fuse and extract document multimodal token features. Subsequent works [1, 12, 17, 29, 31] have further integrated visual features during the pre-training phase to improve the understanding of key information in documents. StructuralLM [15] replaces word-level boxes with cell-level ones, strengthening the contextual representation. BROS [10] proposes a novel relative embedding to enhance the layout interaction, achieving satisfactory performance only with text and layout information. LiLT [27] introduces a two-branch architecture, enabling flexible semantic backbone switching and fast adaptation to different language scenarios during the fine-tuning stage. GeoLayoutLM [21] employs various spatial-aware pre-training strategies to further boost the SER performance with stronger positional clues. All of the above methods achieve visual SER through a downstream BIO-tagging head, which necessitates well-organized OCR inputs with properly sorted and aggregated text contents, making them hard to be used in industrial applications where complex layout documents are commonly seen.

### 2.2 Approaches Compatible for General OCR Inputs

XY Cut algorithms [6, 9] are widely applied in layout analysis tasks. These algorithms merge and divide document contents into blocks based on the gaps between line boxes, and can aggregate content within multi-line entities based on block information. However, one limitation of XY Cut algorithms is the need for manual threshold setting, which makes it challenging to apply across documents with different layouts. LayoutReader [28] introduces an encoder-decoder architecture for reading order prediction in a generative manner. However, its generative nature leads to relatively low inference efficiencies, which makes it difficult to satisfy the high throughput demand in most applications. ERNIE-Layout [23] integrates a Layout Parser module to generate a sorted input sequence, and further employs a reading order prediction task in the pre-training stage to enhance the ability to handle inputs that are mistakenly ordered. However, the parser assembles multiple modules based on heuristic rules, and it is not publicly accessible.

Image-to-sequence pipelines like Donut [14] and Dessurt [4] predict the desired content directly from the document image, without the need for text sorting or aggregation operations. EATEN [8] first generates feature maps of each entity, then employs entity-aware decoders to predict the text content. QGN [2] first extracts the prefix token of each key field, then generates the entity content accordingly. These generative approaches require a significant amount of



**Fig. 3.** Model architecture of ROISER. Line-level OCR results are processed by the Pre-Trained Document Understanding Backbone to extract multi-modal features of each token. The downstream head employs Entity Line Span Prediction, Line Classification, and Line Aggregation to obtain the line token span, line category, and neighbouring line relations, then generate the visual SER results.

training data and struggle in documents with complex layouts or low image quality. DocTr [18] generates anchor words for each entity and predicts entity-level bounding boxes using a vision-language decoder, but it necessitates task-specific pre-training, consuming substantial resources. TPP [34] predicts the token path of each entity, thus eliminating the effect of input OCR granularity and order. However, it models the task at the token level, resulting in relatively high memory consumption.

### 3 Methodology

The overall pipeline of ROISER is illustrated in Fig. 3. Our model starts with generating multi-modal representations for each token. Given the OCR results of a document, recognized texts are tokenized into token sequences and further processed by the multi-modal encoder. Any pre-trained BERT-like document understanding model, like LayoutLM [30] or LiLT [27], can serve as the encoder backbone, fusing semantic, layout, and visual (optional) information to predict a sequence of features  $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_N\}$ , where  $\mathbf{f}_i \in \mathbb{R}^{d_e}$  represents the multi-modal embedding of the  $i$ th token,  $d_e$  denotes the backbone hidden size, and  $N$  represents the length of the token sequence. In the downstream head, three steps were applied to organize the unordered input and produce the final SER results, which will be elaborated on in the subsequent sections.

#### 3.1 Entity Line Span Prediction

General OCR engines may produce inaccurate text boxes, resulting in the merging of content from different entities within the same bounding box. To address this issue and identify the boundaries of entity lines from the given unordered

token sequence, we utilize a BEO-tagging scheme, classifying the tokens into three types: B (beginning), E (end), and O (other). Substrings that start with a B token and end with an E token are recognized as entity lines. By employing this approach, we are able to separate the content of individual entities and determine the position of entity lines. The above tagging process is done with a linear classifier:

$$\mathbf{S} = \{s_1, s_2, \dots, s_N\} = \arg \max (\mathbf{F}\mathbf{W}_s + \mathbf{b}_s), \quad (1)$$

where  $\mathbf{W}_s \in \mathbb{R}^{d_e \times 3}$  and  $\mathbf{b}_s \in \mathbb{R}^3$  are the weights and bias of the classifier.  $s_i \in \{0, 1, 2\}$ , category 0, 1, and 2 denote other, beginning, and end tokens respectively.

To simplify the complexity of the task, we adopt the approach of representing each entity line with the embedding of its first token. This allows us to shift the modelling granularity from the token level to the line level, thus reducing the number of elements for subsequent operations:

$$\mathbf{L}' = \{\mathbf{l}'_1, \mathbf{l}'_2, \dots, \mathbf{l}'_K\} = \{\mathbf{f}_i\}_{s_i=1}, \quad (2)$$

where  $\mathbf{l}'_i$  is the entity line feature,  $K$  is the number of entity lines.

To ensure stability and consistency, during the training phase, the selection of line features  $\mathbf{L}'$  is based on the span prediction ground truths  $\hat{\mathbf{S}}$ . While at the inference stage, line features are selected by the predicted  $\mathbf{S}$ .

### 3.2 Line Aggregation

To address the challenges posed by multi-line entities, we employ a line aggregation module to merge neighbouring lines. The module starts with a linear projection layer, which maps the channels of the output features to a smaller size, aiming to further alleviate the memory burden:

$$\mathbf{L} = \{\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_K\} = \mathbf{L}'\mathbf{W}_{proj} + \mathbf{b}_{proj}, \quad (3)$$

where  $\mathbf{W}_{proj} \in \mathbb{R}^{d_e \times d_r}$  and  $\mathbf{b}_{proj} \in \mathbb{R}^{d_r}$  are the projection weights and bias,  $d_r$  is the reduced channel size.

Determining text line adjacency relies on sufficient information interaction between lines. To achieve this, we set up an attention-based Global Interaction Module (GIM) that contains multiple Transformer [26] encoder layers to fuse line features. Let  $\mathbf{L}_i^{(n)} = \{\mathbf{l}_1^{(n)}, \mathbf{l}_2^{(n)}, \dots, \mathbf{l}_K^{(n)}\}$  denotes the input feature of the  $n$ -th GIM layer, the attention score is calculated using the following formula:

$$\alpha_{ij}^{(n)} = \frac{1}{\sqrt{d_r}} (\mathbf{l}_i^{(n)} \mathbf{W}_Q^{(n)}) (\mathbf{l}_j^{(n)} \mathbf{W}_K^{(n)})^T. \quad (4)$$

The relative position holds significant importance in linking-based document understanding tasks. Drawing inspiration from KVPFormer [11], we incorporate a similar spatial compatibility bias (SCB) term to enhance layout awareness:

$$\mathbf{r}_{ij} = (\Delta(B_i, U_{ij}), \Delta(B_i, B_j), \Delta(B_j, U_{ij})), \quad (5)$$

$B_i$  is the line OCR box that token  $i$  falls in, and  $U_{ij}$  is the union box of  $B_i$  and  $B_j$ .  $\Delta(B_i, B_j) = (t_{ij}^{xctr}, t_{ij}^{yctr}, t_{ij}^w, t_{ij}^h, t_{ji}^{xctr}, t_{ji}^{yctr}, t_{ji}^w, t_{ji}^h)$ , each term denotes a relative position information calculated as:

$$\begin{aligned} t_{ij}^{xctr} &= (x_i^{ctr} - x_j^{ctr})/w_i, & t_{ij}^{yctr} &= (y_i^{ctr} - y_j^{ctr})/h_i, \\ t_{ij}^w &= w_i/w_j, & t_{ij}^h &= h_i/h_j, \\ t_{ji}^{xctr} &= (x_j^{ctr} - x_i^{ctr})/w_j, & t_{ji}^{yctr} &= (y_j^{ctr} - y_i^{ctr})/h_j, \\ t_{ji}^w &= w_j/w_i, & t_{ji}^h &= h_j/h_i. \end{aligned} \quad (6)$$

$(x_i^{ctr}, y_i^{ctr})$  is the center point of  $B_i$ , and  $(w_i, h_i)$  are the corresponding box width and height. The bias term is processed by a feed-forward layer for channel size projection, and is subsequently added to the attention score:

$$\alpha_{ij}^{(n)} = \alpha'_{ij} + \mathbf{FFN}_\alpha^{(n)}(\mathbf{r}_{ij}). \quad (7)$$

The output of the attention module is the weighted average of the projected value vectors:

$$\mathbf{h}_i^{(n)} = \sum_j \frac{\exp(\alpha_{ij}^{(n)})}{\sum_k \exp(\alpha_{ik}^{(n)})} (\mathbf{I}_j^{(n)} \mathbf{W}_V^{(n)}). \quad (8)$$

Line representations processed by GIM are concatenated in a pair-wise manner, generating a next-line prediction matrix  $\mathbf{M}$ , in which each term is calculated by:

$$\mathbf{M}_{ij} = [\mathbf{I}_i^{(last)} \oplus \mathbf{I}_j^{(last)}] + \mathbf{FFN}_{pair}(\mathbf{r}_{ij}), \quad (9)$$

Here  $\mathbf{I}^{(last)}$  denotes the output of the last GIM layer,  $\oplus$  denotes vector concatenation. A binary classification step is subsequently applied to  $\mathbf{M}$  to obtain the line aggregation results. If line  $j$  is considered the next line after line  $i$ , element  $\mathbf{M}_{ij}$  is marked as positive, while all other elements remain negative.

### 3.3 Line Classification

To predict which category the entity line falls in, a linear classifier is applied to line features  $\mathbf{L}'$ .

$$\mathbf{C} = \mathbf{L}' \mathbf{W}_{cls} + \mathbf{b}_{cls}, \quad (10)$$

$\mathbf{W}_{cls} \in \mathbb{R}^{d_e \times N_{cls}}$ ,  $\mathbf{b}_{cls} \in \mathbb{R}^{N_{cls}}$  are weights and bias of the classifier.  $N_{cls}$  is the number of entity types in the document, with the background category included.

### 3.4 Optimization and Result Parsing

The three steps above are all supervised by the cross-entropy loss. The overall optimization target is the weighted sum of all losses, denoted as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{sp} + \lambda_2 \mathcal{L}_{la} + \lambda_3 \mathcal{L}_{lc}, \quad (11)$$

Here  $\mathcal{L}_{sp}$ ,  $\mathcal{L}_{la}$ , and  $\mathcal{L}_{lc}$  represent the losses of entity line span prediction, line aggregation, and line classification, respectively.

The final visual SER results can be obtained by parsing the predictions of these three steps. For line aggregation, it has been observed that each text line is connected to at most one other line, and is also linked by at most one line. Consequently, when parsing the next-line prediction matrix, for each entity line, we retain the in/out-linking with the highest confidence. For entity classification, the line category that appears most frequently within the entity lines is assigned as the SER category for that entity.

## 4 Experiments

### 4.1 Datasets and Evaluation Metrics

We conduct experiments on six benchmarks: **RFUND** [19] is the re-annotated version of FUNSD [7] and XFUND [32], which provides line-level OCR results based on the original labels and rectifies various annotation errors. For our experiments, we select RFUND’s English subset RFUND-en, which contains 199 English samples, with 149 samples used for training and 50 for testing. Entities in RFUND are categorized into four types: header, question, answer, and other. **CORD** [22] comprises 1000 shopping receipts in English, of which 800 are for training, 100 for validation, and 100 for testing. Contents of the receipts fall into 30 categories including item name, item count, total price, etc. We employ the word-level OCR results provided in the annotations as the model input. **SIBR** [33] is a bilingual form understanding dataset composed of 600 training samples and 400 testing samples. It contains 600 Chinese invoices, 300 English bills of entry, and 100 bilingual receipts. Entities in the dataset are categorized as header, question, answer, and other. Compared to FUNSD, documents in SIBR have a relatively more complex layout, making it hard to recognize and aggregate entity contents. The dataset is annotated at the line level, with line aggregation (intra-links) labels provided. Its line-level OCR results are used for model input. **FUNSD-r** and **CORD-r** are revised versions of FUNSD and CORD proposed by [34], with different annotation strategies applied compared to RFUND. OCR results of these two datasets are obtained through real-world open-source OCR engines [16], where cases like inaccurate bounding boxes and missing contents exist. Char-level and segment-level annotations are provided, and the latter is used as our model’s input. **INVOICE** is our in-house dataset, which contains 3427 Chinese invoices captured by mobile phone cameras, with 2399 for training and 1028 for testing. Contents to be extracted include payer address, item name, tax rate, total price, etc. The invoices have complex layouts with densely arranged text, and some images may in low quality due to equipment limitations, posing unique challenges. Line-level OCR results are used as the model input.

We evaluate the model’s performance with entity F1 score. A prediction is considered true positive only when the predicted string and category exactly

match the ground truths. In addition, we utilize the multi-line F1 score to evaluate the model’s capability in aggregating fragmented fields, where only entities that span across multiple text lines (boxes) are taken into account.

## 4.2 Implementation Detail

The reduced hidden size  $d_r$  in the line aggregation head is set to  $d_e/2$ . We use a 3-layer Transformer encoder as the GIM layers, where the head size and feed-forward dimension are set to 6 and 384, respectively. The loss weighting parameters  $\lambda_i$  are all set to 1. We utilize OHEM [25] to cope with category imbalance in the next-line prediction matrix during the training phase. Specifically, we select all positive elements and 10 negative elements for back-propagation. We employ AdamW [20] optimizer, with a warmup ratio of 0.1 and scheduled by the linear decay scheduler. The maximum learning rate is set to  $8e-5$  for the document understanding backbone, and  $8e-4$  for the newly added downstream head. Betas and epsilon of the AdamW are set to (0.9,0.99) and  $1e-8$ , with no weight decay applied. We finetuned our model on RFUND-en/CORD/SIBR/FUNSD-r/CORD-r/INVOICE for 300/100/100/200/100/40 epochs, respectively, with a global batch size of 32.

## 4.3 Baseline Settings

We employ two publicly available and advanced document understanding models, LiLT and LayoutLMv3, as our token embedding backbone. InfoXLM [3] is used as the semantic branch of LiLT for multi-lingual compatibility across different datasets. For LayoutLMv3, we initialize the backbone weights with the official *laytoulmv3-base* checkpoint<sup>1</sup> on RFUND, CORD, FUNSD-r, and CORD-r, and the *laytoulmv3-base-chinese* checkpoint<sup>2</sup> on SIBR and INVOICE. Two types of pre-processing strategies were applied to the OCR results before sending them to ROISER: (1) Left-Top-Right-Bottom (LTRB in Table 1) sorting, which arranges the recognized segments based on their bounding box’s centre coordinate in a left-top to right-bottom order. (2) Augmented XY Cut [6] (denoted as XY in Table 1) first divides document contents into multiple blocks and then organizes the segments within blocks in LTRB order. This approach maximizes the alignment of contents within entities to adjacent positions. Specifically, for FUNSD-r and CORD-r, we follow the settings in [34] by directly using the raw OCR order presented in their annotations.

## 4.4 Comparison with Previous Methods

Results on RFUND-en, CORD, SIBR, and INVOICE are shown in Table 1. For RFUND-en, compared to the BIO tagging baseline, our ROISER improves the performance on multi-line entities with both LiLT (36.31→46.11, 49.36→52.60)

<sup>1</sup> <https://huggingface.co/microsoft/layoutlmv3-base>.

<sup>2</sup> <https://huggingface.co/microsoft/layoutlmv3-base-chinese>.

and LayoutLMv3 (40.60→59.07, 58.47→60.64), thus leading to a higher global F1-score, showing its advances in handling the text aggregation issue. For the CORD dataset, ROISER demonstrates better performance compared to the baseline when using word-level OCR results as the model input, showcasing its applicability across different modelling granularities. Figure 4 visualizes an example of line aggregation. Our model correctly extract and merge the desired contents, while the conventional approach fail to aggregate all of the words. It is observed that preprocessing with Augmented XY Cut leads to poor results (32.01 for LiLT and 57.37 for LayoutLMv3) in the BIO tagging pipeline, as the densely arranged contents in CORD make it challenging for XY Cut to correctly split elements based on box gaps, resulting in highly interrupted text order. However, despite these noisy inputs, ROISER successfully groups the content and significantly improves the score (32.01→90.69 for LiLT, 57.37→92.61 for LayoutLMv3), demonstrating its outstanding performance in order correction and text aggregation. Our method also outperforms the BIO-tagging scheme under all settings for SIBR and INVOICE, highlighting its effectiveness in complex layout scenarios.

**Table 1.** Comparison with existing pipelines on RFUND-en, CORD, SIBR, and our in-house INVOICE datasets. Pre. denotes pre-processing strategies. LTRB refers to sorting the OCR boxes from left-top to right-bottom by coordinates, and XY refers to sorting with Augmented XY Cut. ml-F1 denotes F1-scores of multi-line entities. Best scores are marked as **bold**, and second best scores are marked as underline.

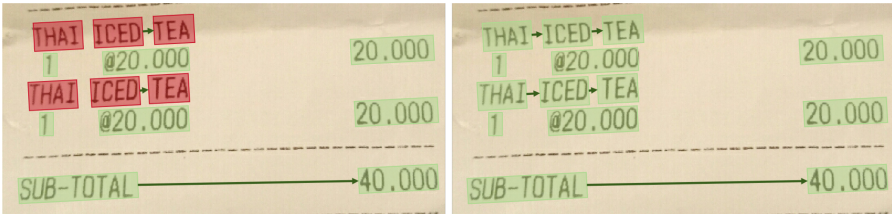
Backbone	Pipeline	Pre.	RFUND-en		CORD		SIBR		INVOICE	
			ml-F1	F1	ml-F1	F1	ml-F1	F1	ml-F1	F1
LiLT[InfoXML] <sub>base</sub> (ACL22)	BIO	LTRB	36.31	78.13	87.62	86.94	22.26	92.46	12.74	93.67
		XY	<u>49.36</u>	78.77	10.04	32.01	47.20	92.92	46.74	93.93
	ROISER (Ours)	LTRB	46.11	<u>82.53</u>	<b>89.91</b>	<b>92.52</b>	<u>60.05</u>	<b>93.70</b>	<b>96.94</b>	<b>99.48</b>
		XY	<b>52.60</b>	<b>83.09</b>	<u>87.88</u>	<u>90.69</u>	<b>62.80</b>	<u>93.49</u>	<u>94.42</u>	<u>98.38</u>
LayoutLMv3 <sub>base</sub> (MM22)	BIO	LTRB	40.60	82.82	90.36	90.98	22.26	92.46	12.85	93.72
		XY	58.47	84.75	43.23	57.37	31.00	92.90	45.75	93.72
	ROISER (Ours)	LTRB	<u>59.07</u>	<u>87.32</u>	<b>92.03</b>	<b>93.50</b>	<b>63.52</b>	<b>93.94</b>	<b>97.76</b>	<b>99.66</b>
		XY	<b>60.64</b>	<b>88.08</b>	<u>91.38</u>	<u>92.61</u>	<u>61.84</u>	<u>93.61</u>	<u>96.09</u>	<u>98.38</u>

The results presented in Table 2 demonstrate that ROISER outperforms both the BIO tagging baseline with various pre-processing strategies and the previous state-of-the-art method, TPP [34], on both FUNSD-r and CORD-r datasets. One common issue in these datasets is the presence of inaccurate bounding boxes that merge different entities. Our model effectively addresses this issue by utilizing the span prediction module. As shown in Fig. 5, our model correctly extracts entities such as *Mail to:*, *Telephone:*, and *FAX:*, even when they are merged within the same box as other contents in the OCR inputs.

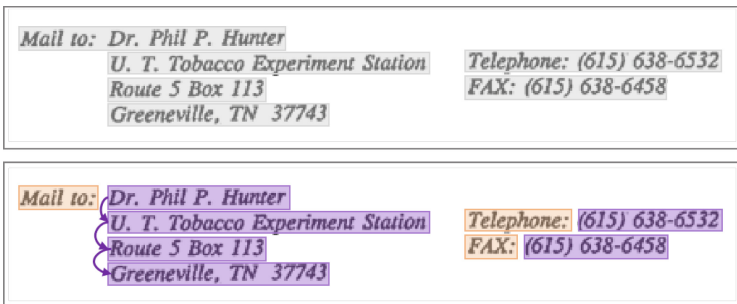


**Table 2.** Performance comparison on FUNSD-r and CORD-r. Pre. refers to pre-processing strategies. None denotes using the raw input order provided in the annotations. LR and TPP mean sorting the input content with LayoutReader [28] and TPP [34], respectively. Best scores are marked as **bold**.

Backbone	Pipeline	Pre.	FUNSD-r	CORD-r
LayoutLMv3 <sub>base</sub>	BIO	None	78.77	82.72
		LR	78.37	70.33
		TPP	79.72	83.24
	TPP (EMNLP23)	None	80.40	91.85
	<b>ROISER (ours)</b>	None	<b>82.50</b>	<b>92.22</b>



**Fig. 4.** Visualization of line aggregation ability on CORD dataset. Green boxes denote the correct prediction, and red are erroneous. Arrays refer to the aggregation prediction. The BIO tagging pipeline (left) fails to merge the word *THAI* with *ICED* and *TEA*, while ROISER (right) successfully handle the case. (Color figure online)



**Fig. 5.** Illustration of span prediction ability on FUNSD-r. Grey boxes in the upper sub-figure represent the input OCR results. The lower sub-figure displays the predictions made by ROISER. Arrows denote the line aggregation prediction, and each entity type is distinguished by a different colour. In this case, adhered entity lines such as *Mail to:*, *Telephone:*, and *FAX:* are correctly split apart. (Color figure online)

#### 4.5 Ablation Studies

Table 3 presents a comprehensive overview of the effectiveness of the global interaction module for the line aggregation task. Results are reported on RFUND-

**Table 3.** Effectiveness of the global interaction module. Encoder refers to the Transformer encoder layers. SCB refers to the spatial compatibility bias, GIM is the global interaction module, and NPM is the next-line prediction matrix  $\mathbf{M}$ .

Setting	Encoder	SCB in GIM	SCB in NPM	ml-F1	F1
1				40.50	79.90
2	✓			0.87	75.95
3	✓	✓		45.11	81.65
4			✓	46.72	81.63
5	✓		✓	41.83	80.00
6	✓	✓	✓	<b>52.60</b>	<b>83.09</b>

**Table 4.** Ablation studies on the number of encoder layers in global interaction module.

# of Layers	Backbone	ml-F1	F1	Backbone	ml-F1	F1
1		47.21	81.14		58.66	86.63
2		48.65	81.96		53.91	86.48
3	LiLT[InfoXLM] <sub>base</sub>	<b>52.60</b>	<b>83.09</b>	LayoutLMv3 <sub>base</sub>	<b>60.64</b>	<b>88.08</b>
4		47.37	82.04		56.64	86.22
5		45.27	80.37		54.35	86.64

en with LiLT[InfoXLM]<sub>base</sub>, and Augmented XY Cut is employed as the pre-processing strategy. When exclusively utilizing the Transformer encoder for line interaction (setting 2), the performance experiences a detrimental impact compared to direct classification (setting 1). Integrating SCB as the attention bias proves advantageous in enhancing performance (setting 3). Incorporating an SCB term into the next-line prediction matrix yields an approximate improvement of 6 points in multi-line F1 and 2 points in global F1 (setting 4), significantly contributing to the overall result. The optimal score is achieved by employing the transformer encoder with SCB as attention bias and including SCB in the next-line prediction matrix (setting 6). It can be concluded that to achieve favourable line global interaction, the Transformer encoder should collaborate with an SCB term. Otherwise, it may harm the results (setting 2 & 5). The relative position assumes a vital role in boosting the line aggregation performance.

We further examine the effect of the number of encoder layers in the global interaction module (Table 4). Results are reported on RFUND-en with Augmented XY Cut sorting. Our findings indicate that setting the layer number to 3 results in optimal performance.

## 5 Conclusion and Future Work

In this paper, we proposed ROISER, a novel pipeline for visual SER tasks that is capable of real-world OCR input. Our approach starts with identifying the entity lines through span prediction and subsequently merges them with a line aggregation module. Entity categories are further determined through line classification. This approach effectively addresses challenges encountered in real-world applications such as multi-line entities and inaccurate OCR detections. Experiments on diverse benchmarks validate the efficacy of ROISER. Future research will focus on further improving the generalization capability of span prediction and line aggregation across different layouts. We hope that our work can draw attention to the struggles faced in the practical application phase, and promote the emergence of more algorithms for real-world scenarios.

**Acknowledgement.** This research is supported in part by National Natural Science Foundation of China (Grant No.: 62441604, 62476093).

## References

1. Appalaraju, S., Jasani, B., Kota, B.U., Xie, Y., Manmatha, R.: DocFormer: end-to-end transformer for document understanding. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 973–983. IEEE (2021)
2. Cao, H., et al.: Query-driven generative network for document information extraction in the wild. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 4261–4271 (2022)
3. Chi, Z., et al.: InfoXLM: an information-theoretic framework for cross-lingual language model pre-training. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 3576–3588 (2021)
4. Davis, B., Morse, B., Price, B., Tensmeyer, C., Wigington, C., Morariu, V.: End-to-end document recognition and understanding with dessurt. In: Karlinsky, L., Michaeli, T., Nishino, K. (eds.) Computer Vision – ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV, pp. 280–296. Springer Nature Switzerland, Cham (2023). [https://doi.org/10.1007/978-3-031-25069-9\\_19](https://doi.org/10.1007/978-3-031-25069-9_19)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019)
6. Gu, Z., et al.: XYLayoutLM: towards layout-aware multimodal networks for visually-rich document understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4583–4592 (2022)
7. Jaume, G., Ekenel, H.K., Thiran, J.P.: FUNSD: a dataset for form understanding in noisy scanned documents. In: ICDAR-OST (2019)
8. Guo, H., Qin, X., Liu, J., Han, J., Liu, J., Ding, E.: EATEN: entity-aware attention for single shot visual text extraction. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 254–259. IEEE (2019)
9. Ha, J., Haralick, R.M., Phillips, I.T.: Recursive X-Y cut using bounding boxes of connected components. In: Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 2, pp. 952–955. IEEE (1995)

10. Hong, T., Kim, D., Ji, M., Hwang, W., Nam, D., Park, S.: BROS: a pre-trained language model focusing on text and layout for better key information extraction from documents. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36(10), pp. 10767–10775 (2022)
11. Hu, K., Wu, Z., Zhong, Z., Lin, W., Sun, L., Huo, Q.: A question-answering approach to key value pair extraction from form-like document images. Proc. AAAI Conf. Artif. Intell. **37**(11), 12899–12906 (2023)
12. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: LayoutLMv3: pre-training for document AI with unified text and image masking. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 4083–4091 (2022)
13. Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., Jawahar, C.: ICDAR2019 competition on scanned receipt OCR and information extraction. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1516–1520. IEEE (2019)
14. Kim, G., et al.: OCR-free document understanding transformer. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII, pp. 498–517. Springer Nature Switzerland, Cham (2022). [https://doi.org/10.1007/978-3-031-19815-1\\_29](https://doi.org/10.1007/978-3-031-19815-1_29)
15. Li, C., et al.: StructuralLM: structural pre-training for form understanding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 6309–6318 (2021)
16. Li, C., et al.: PP-OCrv3: more attempts for the improvement of ultra lightweight OCR system. arXiv preprint [arXiv:2206.03001](https://arxiv.org/abs/2206.03001) (2022)
17. Li, Y., et al.: StrucTexT: structured text understanding with multi-modal transformers. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 1912–1920 (2021)
18. Liao, H., et al.: DocTr: document transformer for structured information extraction in documents. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 19584–19594 (2023)
19. Lin, Z., et al.: PEneo: unifying line extraction, line grouping, and entity linking for end-to-end document pair extraction. arXiv preprint [arXiv:2401.03472](https://arxiv.org/abs/2401.03472) (2024)
20. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proceedings of the 7th International Conference on Learning Representations (ICLR) (2019)
21. Luo, C., Cheng, C., Zheng, Q., Yao, C.: GeoLayoutLM: geometric pre-training for visual information extraction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7092–7101 (2023)
22. Park, S., et al.: CORd: a consolidated receipt dataset for post-OCR parsing. In: Workshop on Document Intelligence at NeurIPS 2019 (2019)
23. Peng, Q., et al.: ERNIE-Layout: Layout knowledge enhanced pre-training for visually-rich document understanding. In: Findings of the Association for Computational Linguistics: EMNLP 2022, pp. 3744–3756 (2022)
24. Ramshaw, L.A., Marcus, M.P.: Text chunking using transformation-based learning. In: Armstrong, S., Church, K., Isabelle, P., Manzi, S., Tzoukermann, E., Yarowsky, D. (eds.) Natural Language Processing Using Very Large Corpora, pp. 157–176. Springer Netherlands, Dordrecht (1999). [https://doi.org/10.1007/978-94-017-2390-9\\_10](https://doi.org/10.1007/978-94-017-2390-9_10)
25. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 761–769 (2016)

26. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
27. Wang, J., Jin, L., Ding, K.: LiLT: a simple yet effective language-independent layout transformer for structured document understanding. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7747–7757 (2022)
28. Wang, Z., Xu, Y., Cui, L., Shang, J., Wei, F.: LayoutReader: pre-training of text and layout for reading order detection. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4735–4744 (2021)
29. Xu, Y., et al.: LayoutLMv2: multi-modal pre-training for visually-rich document understanding. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2579–2591 (2021)
30. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: LayoutLM: pre-training of text and layout for document image understanding. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1192–1200 (2020)
31. Xu, Y., et al.: LayoutXLM: multimodal pre-training for multilingual visually-rich document understanding (2021)
32. Xu, Y., et al.: XFUND: a benchmark dataset for multilingual visually rich form understanding. In: *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3214–3224 (2022)
33. Yang, Z., et al.: Modeling entities as semantic points for visual information extraction in the wild. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15358–15367 (2023)
34. Zhang, C., et al.: Reading order matters: information extraction from visually-rich documents by token path prediction. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13716–13730 (2023)



# Perception-Enhanced Generative Transformer for Key Information Extraction from Documents

Runbo Zhao<sup>1</sup>, Jun Jie Ou Yang<sup>2</sup>, Chen Gao<sup>1</sup>, Xugong Qin<sup>1(✉)</sup>,  
Gangyan Zeng<sup>1</sup>, Xiaoxu Hu<sup>3(✉)</sup>, and Peng Zhang<sup>1,4</sup>

<sup>1</sup> School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

{zhaorunbo, gaochen, qinxugong, gyzeng, zhang\_peng}@njust.edu.cn

<sup>2</sup> Viterbi School of Engineering, University of Southern California, Los Angeles, USA

<sup>3</sup> National Computer Network Response Technical Team/Coordination Center Technology, Beijing, China

hxx@cert.org.cn

<sup>4</sup> Laboratory for Advanced Computing and Intelligence Engineering, Wuxi, China

**Abstract.** Key information extraction (KIE) from scanned documents has attracted significant attention due to practical real-world applications. Despite impressive results achieved by incorporating multimodal information within the generative framework, existing methods fail to understand complex layouts and fuzzy semantics in document images. To settle these issues, we propose a perception-enhanced generative transformer (PEGT), which improves the model through fine-grained multimodal modeling and pre-training tasks tailored for the generative framework. Firstly, we introduce a pre-trained vision-language model to provide transferable knowledge for visual text perception. Then two auxiliary pre-training tasks including absolute position prediction (APP) and semantic relationship reasoning (SRR) are designed for the generative framework. APP learns to predict which grids the texts fall into, improving the model on utilization of the position information. SRR exploits prior information of semantic relationships, injecting the ability for better semantic discrimination into PEGT. Finally, well-designed prompts are leveraged to unleash the potential of PEGT for extracting key information from documents. Extensive experiments on several public datasets show that PEGT effectively generalizes over different types of documents. Especially, PEGT achieves state-of-the-art results in terms of F-measure, i.e., 97.47%, 98.04%, and 84.32% on the SROIE, CORD, and FUNSD datasets, demonstrating the superiority of the proposed method.

**Keywords:** Key Information Extraction · Document Understanding · Generative Model

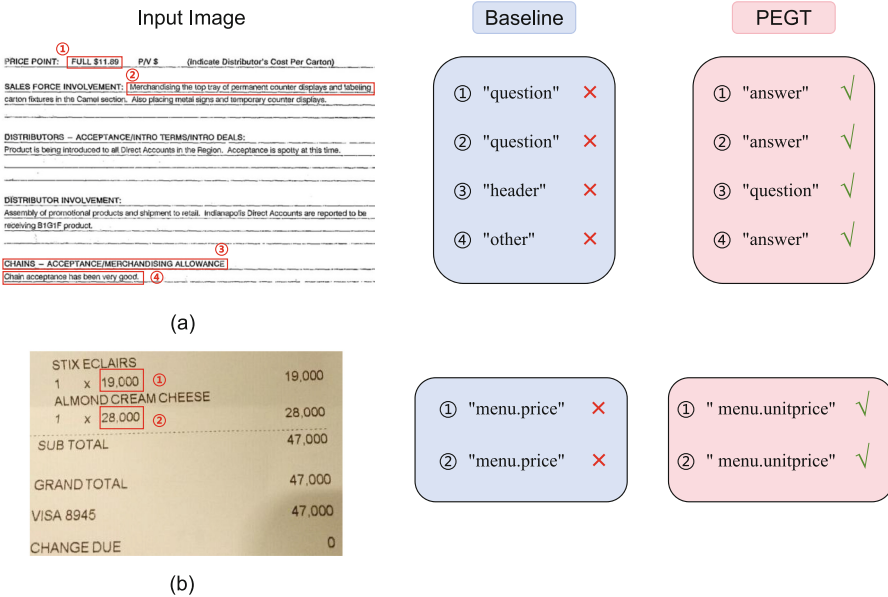
---

X. Qin and X. Hu—Two authors as corresponding authors.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025  
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15331, pp. 91–106, 2025.  
[https://doi.org/10.1007/978-3-031-78119-3\\_7](https://doi.org/10.1007/978-3-031-78119-3_7)

# 1 Introduction

Recently the key information extraction (KIE) from document images, such as scanned receipts [14], forms [15], financial reports [34] and even in the wild scene [18] demonstrate significant value, attracting researchers from the academic and industrial communities. KIE tasks typically involve multiple research areas, including optical character recognition (OCR) [28–32,37] and named entity recognition (NER) [21,35]. Thus utilization of multimodal information, such as text, layout, and visual information is well explored in existing KIE methods. Among these methods, discriminative frameworks [13,41,42] are frequently employed, in which the OCR results generated by common tools are leveraged to provide layout and textual information. However, these models are inevitably affected by the accumulated errors from the OCR process. To mitigate the impact of OCR errors in the input content, a KIE method [3] based on a generative framework [39] has recently been proposed, which shows the potential to correct the OCR errors and generate the desired output with a sequence decoder.



**Fig. 1.** Comparisons of results predicted by the baseline model and PEGT. PEGT exhibits better perception ability of documents, which benefits from pre-training tasks and knowledge from pre-trained vision-language models. (a) When dealing with dense and long text that is difficult to distinguish by the baseline model, PEGT can make more reasonable judgments based on contextual semantic information. (b) Encountering ambiguous or similar text, PEGT makes more accurate, fine-grained predictions based on the position of the text and the layout logic of the document image.

Existing generative large multimodal models, e.g., OFA [39], LLaMA [38], and VisionLLM [40] have demonstrated great success on multimodal understanding, which benefit from the pre-trained large-scale language model as the back-end and the unified decoding paradigm. However, these models fail to comprehend document images with complex layouts and fuzzy semantics. Despite the generation ability inherited from the pre-trained language model, due to the lack of perception ability for the visual modality, when processing document images, the model cannot locate the answer by understanding the layout logic and contextual semantics like humans, which leads to the false predictions as shown in Fig. 1. In our experiments, we try to remove the visual modality, leaving only text and layout features as input, and find that this leads to a minor impact on performance. This finding further indicates that the information of visual modality in existing methods is underutilized.

In addition, pre-training techniques have also been well explored to enable models to learn commonsense knowledge from data and make significant progress in document understanding [13, 41, 42]. BERT [16] proposes masked language modeling (MLM), which learns the bidirectional representation by reconstruction based on random masked texts. In the visual document understanding (VDU) area, existing discriminative methods such as DocFormer [1], SelfDoc [22], and Layoutlmv3 [13] follow this approach and extend it to a multimodal encoder for representation learning to achieve better results. However, the difference in granularity between images and text increases the difficulty of learning a unified multimodal representation. Moreover, considering the importance of the decoder in generative frameworks, existing pre-training tasks that account for encoders alone may lead to sub-optimal performance.

To solve the above problems, we propose PEGT, a perception-enhanced generative transformer that extracts the key information generatively, equipped with better perception ability for VDU. Firstly, inspired by the phenomenon that CLIP responds to visual texts [26], we propose to introduce the pre-trained CLIP [33], which provide commonsense knowledge for the model to enhance the perception of visual text. Besides, two pre-training tasks, including absolute position prediction (APP) and semantic relationship reasoning (SRR) are designed for the generative framework for better perceiving position and semantic information. In particular, the APP learns to predict which grids the texts fall into, improving the model on utilization of the position information. SRR exploits prior information of semantic relationships, injecting the ability for better semantic discrimination into PEGT. PEGT achieves better perception ability with the proposed modules and produces more accurate predictions than the baseline model. As shown in Fig. 1 (a), when dealing with paired long sentences, the baseline model makes it difficult to understand the differences between dense texts, which struggle to make correct decisions and tend to commit easy errors; in contrast, PEGT can better give correct answers considering the semantics between long sentences. For example, the second highlighted text with the ground-truth “answer” that is long and close to a “question” is easily misidentified as part of the “question”. As shown in Fig. 1 (b), the baseline model can not perceive the



semantics and layout structures in documents, e.g., unit prices and total prices often appear in the middle and the far right respectively, making false predictions when distinguishing similar categories; PEGT can better utilize layout information and predict more accurate results. Extensive experiments on three public KIE datasets demonstrate the effectiveness of the proposed method.

The key contributions of this work are summarized as follows:

- We propose PEGT, a perception-enhanced generative transformer for the KIE task that can efficiently utilize visual information that contains pre-trained vision-language knowledge and auto-regressively generates key information from scanned documents based on well-designed prompts.
- Two pre-training tasks, including absolute position prediction (APP) and semantic relationship reasoning (SRR) are designed to enhance the perception ability of the generative model for multimodal information.
- Comprehensive experiments on real-world KIE datasets demonstrate the strong robustness of PEGT against document images containing numerous small texts and complex layouts, which makes it more applicable in practical scenarios.

## 2 Related Works

### 2.1 Multimodal Document KIE

An encoder-based multimodal transformer, accompanied by pre-training techniques, becomes a prevalent method for VDU demonstrates strong feature representation, and achieves SOTA performance in downstream tasks involving KIE tasks. LayoutLM [42] first proposes the pre-training framework to handle multimodal KIE, which incorporates text, layout, and visual features. LayoutLMv2 [41] integrates a spatial-aware self-attention mechanism into the transformer architecture, allowing the model to comprehend the relative positional relationships between texts. LayoutLMv3 [13] learns multimodal representation through self-supervised using three pre-training objectives: word-patch alignment, masked image modeling, and masked language modeling. To simplify the visual branch for VDU, DocFormer [1] designs a multimodal self-attention layer to combine multiple modalities and share learned spatial embeddings across modalities. BROS [2] employs a graph-based classifier to predict entity tags as well as entity relationships. Similar to BROS, LAMBERT [9] designs a layout-aware language model, which does not use the original image but only augments the coordinates of token bounding boxes to the input. StrucText [23] and StrucTextv2 [44] introduce a segment-token aligned encoder to handle entity labeling and entity linking tasks at varying levels of granularity. Unlike the discriminative methods mentioned above, GenKIE [3] proposes a generative framework that provides the potential correct error information in OCR results. ICL-D3IE [10] introduces an in-context learning framework that allows large language models (LLMs) to perform document information extraction using various types of demonstration examples. Despite the strong ability for sequence generation,

existing generative methods can hardly effectively utilize visual information and achieve synergy between different modalities due to the lack of perception ability. Thus the generative frameworks are not fully explored for the KIE tasks.

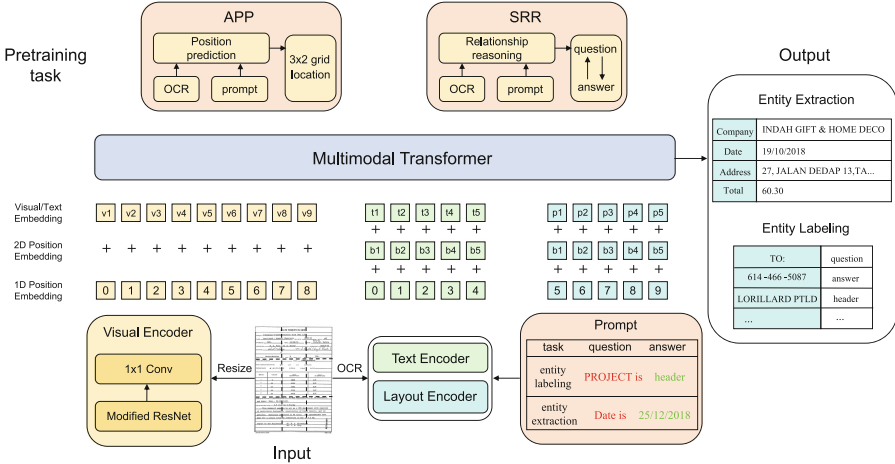
## 2.2 Visual Representation Learning in VDU

Visual representation learning plays an important role in VDU tasks, in which high resolution is required for fine-grained text perception, making the efficiency and performance both matter. Early approaches tend to directly adopt visual encoders pre-trained in the natural image domain, which keeps the pipelines simple and demonstrates good generality. OFA [39] uses ResNet [11] as a visual encoder to unify a diverse set of cross-modal and single-modal tasks in a simple sequence-to-sequence learning framework. Inspired by ViT [6], LayoutLMv3 [13] utilizes the original image patch directly from the document image without complex pre-processing steps. Donut [17] proposes an end-to-end framework, which does not rely on an explicit OCR process to complete various VDU tasks. Subsequent methods [7,8] focus on improving the image encoders, aiming to better perceive the small and dense characters within document images as well as maintain efficiency. Based on DETR [4], DocTr [8] design a CNN backbone with multi-scale visual feature extraction and a vanilla transformer for efficient encoding of visual features, which reduces the feature map resolution to 1/8 of the input image scale for better detection of small entities. DocPedia [7] processes visual input in the frequency domain instead of the pixel space, which can capture more visual and textual information with a limited set of visual tokens. However, how to achieve a good balance between computation efficiency and performance for the design of the visual encoder in VDU remains an open question.

## 2.3 Vision-Language Pre-training

Vision-language pre-training (VLP) exhibits great advantages in processing joint visual-linguistic representations. ViLBERT [25] extends BERT’s architecture into a multimodal, two-flow model by incorporating tasks such as masked multi-modal modeling and multi-modal alignment prediction. LXMERT [36] allows the model to connect visual and linguistic semantics through pre-training tasks, which learn intra-modality and cross-modality relationships. UNITER [5] uses conditional masking in pre-training tasks rather than applying joint random masking to both modalities. Unicoder-VL [20] borrows ideas from the cross-language pre-trained model XLM [19], enabling the model to learn context-aware representations based on visual and linguistic content. These works focus on general vision-language tasks such as image-text retrieval, visual question answering, and visual grounding. For visual text understanding, TAP [43] proposes text-aware pre-training to help the model learn a better aligned representation among text words, visual objects, and scene text. For the KIE tasks, LayoutLMv3 [13] is pre-trained using discrete token reconstruction objectives,

including masked language modeling, masked image modeling for the multi-modal transformer, and word-patch alignment. However, these tasks are mainly designed for encoder pre-training and are not suitable for multimodal generative models which consist of both encoders and decoders.



**Fig. 2.** An overview of PEGT. PEGT is a perception-enhanced multimodal generative model. Given a document image and its OCR results, the encoder’s input is composed of patched visual tokens and textual tokens, which are embedded with the positional features. The decoder generates the output according to a designed prompt. Additionally, PEGT is pre-trained with absolute position prediction (APP) and semantic relationship reasoning (SRR) to better perceive positional and semantic information.

### 3 Methodology

The overall architecture of PEGT is shown in Fig. 2, which consists of a multi-modal encoder-decoder model and two pre-training tasks. The encoder embeds multimodal features from the input and the decoder generates the textual output by following the prompts. To improve the model’s ability to understand layout and semantics, we pre-train it using the perception-enhanced pre-training tasks including APP and SRR, then fine-tune it on downstream KIE tasks. In the subsequent sections, we explain the process of multimodal feature extraction, prompt design, and decoding, and then elaborate on the pre-training method and loss function techniques.

#### 3.1 Multimodal Feature Extraction

**Textual Embedding.** Textual embedding encompasses both word embeddings and position embeddings, which are generated based on the OCR results (text

transcription and box detection). Then we use the BPE (Byte Pair Encoding) tokenizer to split words into subword units. Finally, the <BEG> and <END> tags are used as the start and end identifiers, the <SEP> tag serves to divide the transcripts  $T$  and the prompt  $P$ , and the <PAD> tokens are added at the end to standardize the length of sequences within a batch. The input sequence  $S_{input}$  can be expressed as:

$$S_{input} = \langle \text{BEG} \rangle, \text{BPE}(T), \langle \text{SEP} \rangle, \text{BPE}(P), \langle \text{END} \rangle, \dots, \langle \text{PAD} \rangle. \quad (1)$$

The  $S_{input}$  is then embedded to obtain the word embeddings  $T^{emb}$  and summed with the position embeddings  $P^{emb}$  (including 1D and 2D positional embeddings) as shown in Fig. 2. Following the LayoutLMv3 [13], the 1D positional and the 2D embeddings are based on the index of tokens within the text sequence and the OCRed bounding box information respectively, where the x-axis and y-axis features are integrated to create a two-dimensional spatial layout embedding. All textual tokens in a bounding box use the same layout feature, instead of detailing the bounding box of each token. In addition, all special symbols and the prompted layout feature are set to a blank box  $b_{blank} = (0, 0, \dots, 0)$ .

**Visual Embedding.** Given a document page image  $I$ , it is first resized to a dimension of  $480 \times 480$  pixels, then input into the visual backbone of the model for further processing. Different from direct fine-tuning, we design to adopt a modified CLIP image encoder. Specifically, we froze the weights of the CLIP-ResNet backbone to retain the knowledge from the pre-training process. The image encoder is reformulated by replacing the value-embedding layer and the last linear layer with two respective  $1 \times 1$  convolutional layers as inspired by [45]. It extracts a contextualized feature map from the input image  $I$  maps it from the image token dimension to the model dimension and flattens it into a sequence of patches  $V^{pat}$ . A trainable position information is also added to the visual modality. The visual embedding is formulated as follows:

$$V_i^{emb} = V_i^{pat} \cdot W_v + \text{PE}(i) \in \mathbb{R}^d, \quad (2)$$

where  $W_v$  is the trainable parameters of a linear projection layer,  $\text{PE}(i)$  is the trainable 1D positional embedding.

We concatenate the image embedding  $V^{emb}$  with textual embedding, which contains word embeddings  $T^{emb}$  and position embeddings  $P^{emb}$ . The final document multimodal feature embedding as follows:

$$E = \text{CONCAT}(V^{emb}, T^{emb} + P^{emb}) \in \mathbb{R}^{(T+P) \times d}, \quad (3)$$

then the final output  $E$  of the above three encoders is fed into the following multimodal transformer.

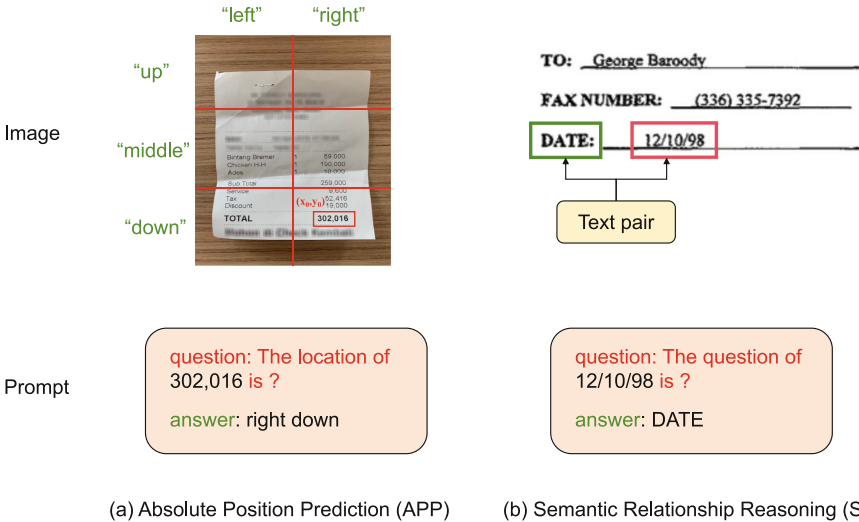
### 3.2 Prompt Design and Decoding

Following the design pattern for the three datasets in GenKIE [3], we provide two kinds of prompts. The first is designed for the entity extraction task, where

the question gives the type of information required and the model generates the corresponding text answer. For example, “Date is ?” will prompt the model to extract date-related content such as “25/12/2018” from the document image. The second is designed for entity labeling tasks, where the question gives an entity in the document image and the model generates the category of the entity, e.g., the category of the “PROJECT SHEET” is “header”. At the same time, to enable the model to better understand and utilize the position information, we add the bounding box information of the queried text to the prompt. The prompt information is appended at the end of the text sequence and fused with other modality features into a multimodal representation, which is fed into the multimodal transformer for answer generation.

The decoder input includes the filled-in prompt, acting as the learning target and ensuring consistency between the training and testing processes. During inference, if the prompt is provided as a template, we apply prefix beam search to restrict the search space, ensuring it begins with the template prefix without requiring the exploration of multiple paths to determine the most probable sequence, which is found to work quite accurately and efficiently on the three datasets.

### 3.3 Perception-Enhanced Pre-training Tasks



**Fig. 3.** Illustration of the proposed pre-training tasks including (a) APP and (b) SRR.

The pre-training tasks aim to enhance the model’s perception of the multimodal inputs. Considering the potential layout logic in document images and

contextual semantics between different texts, we design two kinds of pre-training tasks according to the characteristics of document images. The proposed absolute position prediction (APP) and semantic relationship reasoning (SRR) pre-training tasks are well illustrated in Fig. 3.

**Absolute Position Prediction.** The APP pre-training task is designed to enable the model to understand the logic position information in document images. We find that text fields with similar semantics tend to occur in fixed regions within a document image, e.g., the “title” is usually at the top, and the “total” generally occurs at the bottom right. Therefore, an accurate perception of the position matters a lot. As shown in Fig. 3 (a), according to the layout of the document images, we divide the documents into six grids of  $3 \times 2$ . The learning target of the APP is to predict which grid the text’s top-left coordinate  $(x_0, y_0)$  falls into, which improves the model on understanding the global information relative to the entire documents. To enable the model to distinguish the pre-training task and the fine-tuning task, the APP’s prompt template form is set to “question: The location of  $T$  is ?”. For example, PEGT generates “right down” by filling in the template “question: The location of 302,016 is ?”.

**Semantic Relationship Reasoning.** To learn the semantic relationship between texts, the SRR pre-training task is proposed which forces the model to concentrate on the strongly related texts. Considering that the noise pairs in the documents may degrade the learning process, we propose to select high-quality text pairs. For the FUNSD dataset, the question-answer text pairs are selected to perform the SRR pre-training task. Because they have stronger prior semantic associations than isolated texts labeled as “title” or “other” with fewer amounts and semantic meaning. We select the price types and price numbers in SROIE, menu name, and menu price in CORD similarly. As shown in Fig. 3 (b), the prompt template is designed as “question: The question of  $T$  is ?”, where  $T$  is the corresponding answer text in the document image, e.g., PEGT generates “DATE” by filling in the template “question: The question of 12/10/98 is ?”.

### 3.4 Loss Function

The overall loss function consists of three parts:

$$Loss = Loss_{KIE} + \alpha_1 Loss_{APP} + \alpha_2 Loss_{SRR}, \quad (4)$$

where  $Loss_{KIE}$ ,  $Loss_{APP}$ , and  $Loss_{SRR}$  denote the KIE and the two pre-training tasks,  $\alpha_1$  and  $\alpha_2$  are both set to 1.0. The cross-entropy loss function with label smoothing  $L_{scc}$  is adopted to optimize the three tasks uniformly, which balances the learning of various classes to reduce overfitting and achieves an efficient and flexible loss calculation strategy for diverse task requirements. The formula is expressed as follows:

$$L_{scc} = (1.0 - \epsilon - \epsilon_i) \cdot \left[-\frac{1}{N} \sum_{n=1}^N \log(p_{y_n})\right] + \epsilon_i \cdot \left[-\epsilon \sum_{i=1}^C p_i \log(p_i)\right], \quad (5)$$

where the output probability  $p$  is obtained by feeding  $E$  into the multimodal transformer,  $\epsilon$  is the global smoothing factor,  $\epsilon_i$  is obtained by equally distributing epsilon overall error classes,  $N$  represents the total number of samples,  $y$  represents the real label of the sample,  $p_{y_n}$  represents the probability of the correct category predicted by the model,  $C$  indicates the total number of categories and  $p_i$  represents the probability of the model’s prediction for category  $i$ .

## 4 Experiments

### 4.1 Experiment Settings

In our experiments, we use the modified CLIP-ResNet as the image encoder, while the other network models follow the pre-trained OFA model [39]. We trained the model using two NVIDIA A6000 GPUs. The model dimension is 768. We fine-tuned PEGT on three datasets, uniformly setting the number of epochs to 50, the batch size to 1, the initial learning rate to 5e-5, and the maximum encoder length to 1024. The maximum length for the template prompt is restricted to 128, while the question portion is limited to 32. Taking into account the distinct demands of the entity extraction task and the entity labeling task, we constrain the maximum generated sequence length to 512 for the former and 128 for the latter. We introduce PEGT<sub>base</sub> and PEGT<sub>large</sub>, which are based on the OFA models with 182M and 472M parameters respectively.

### 4.2 Data and Baselines

Our experiments are conducted on three real-world datasets, including SROIE [14], CORD [27], and FUNSD [15]. Table 1 demonstrates the statistics of these datasets. Several baselines are used for comparison, including the SOTA models such as LayoutLMv2 [41], DocFormer [1], GenKIE [3] and StrucText [23]. For GenKIE, the evaluation result is obtained by using the official implementation<sup>1</sup>

**Table 1.** Statistics of different datasets.

Dataset	Type	Labels	Images		
			Train	Val	Test
FUNSD [15]	Form	4	149	0	50
SROIE [14]	Receipt	4	626	0	347
CORD [27]	Receipt	30	800	100	100

<sup>1</sup> <https://github.com/Glasgow-AI4BioMed/GenKIE>.

### 4.3 Comparison with Existing Methods

As shown in Table 2, we evaluate the effectiveness of PEGT for the entity extraction and entity labeling tasks on three public datasets. PEGT can achieve 97.47%, 98.04%, and 84.32% in terms of F-measure on the SROIE, CORD, and FUNSD datasets, respectively. Compared with GenKIE, which is also a generative model, PEGT performs better through the pre-training tasks for learning layout and semantic information. Besides, PEGT achieves the best performance among existing methods on the CORD dataset, which has up to 30 categories, demonstrating its powerful key information extraction capability and robustness in understanding fine-grained semantics. Moreover, PEGT demonstrates comparable performance to other discriminative models with pre-training for encoders, such as LayoutLMv2, DocFormer, and LAMBERT on the three datasets.

**Table 2.** Overall performance of the compared models on the three datasets. The bold font indicates the best performance, the underline indicates the second-best, P is the accuracy rate, R is the recall rate and F is the F1 score. \* indicates results reproduced by using the official implementation.

Model	SROIE			CORD			FUNSD		
	P	R	F	P	R	F	P	R	F
BERT [16]	90.99	90.99	90.99	88.33	91.07	89.68	54.69	67.10	60.26
RoBERTa [24]	91.07	91.07	91.07	-	-	-	66.48	66.48	66.48
UniLMv2 [2]	94.59	94.59	94.59	89.87	91.98	90.92	65.61	72.54	68.90
Bros [12]	94.93	96.03	95.48	95.58	95.14	95.36	81.16	85.02	83.05
LayoutLM [42]	94.38	94.38	94.38	94.37	95.08	94.72	76.77	81.95	79.27
LAMBERT [9]	-	-	96.93	-	-	94.41	-	-	-
LayoutLMv2 [41]	96.25	96.25	96.25	94.53	95.39	94.95	80.29	<u>85.39</u>	82.76
StrucText [23]	95.84	<b>98.52</b>	96.88	-	-	-	<b>85.68</b>	80.97	83.09
DocFormer [1]	-	-	-	96.52	96.14	96.33	80.76	<b>86.09</b>	83.34
GenKIE [3]	<u>97.40</u>	<u>97.40</u>	<u>97.40</u>	95.75	95.75	95.75	83.45	83.45	83.45
GenKIE* [3]	96.84	96.91	96.88	97.90	96.07	96.98	80.00	80.00	80.00
<b>PEGT<sub>base</sub></b>	97.31	97.21	97.26	<b>98.19</b>	<u>97.53</u>	<u>97.86</u>	83.70	83.70	<u>83.70</u>
<b>PEGT<sub>large</sub></b>	<b>97.49</b>	<u>97.45</u>	<b>97.47</b>	<u>98.15</u>	<b>97.92</b>	<b>98.04</b>	<u>84.32</u>	84.32	<b>84.32</b>

### 4.4 Ablation Study

The ablation experiments are performed based on the PEGT<sub>base</sub> model, verifying the effectiveness of the pre-training tasks and the image encoder on the three datasets for the KIE tasks.



**Effectiveness of the Pre-training Tasks.** As shown in Table 3, the proposed APP and SRR pre-training tasks provide a consistent improvement on the three datasets. The APP task increases the score from 97.12% to 97.25% on SROIE, from 97.43% to 97.45% on CORD, and from 80.29% to 81.32% on FUNSD, which shows the effectiveness of learning the position information. The SRR task increases the score from 97.12% to 97.14% on SROIE, from 97.43% to 98.00% on CORD, and from 80.29% to 82.11% on FUNSD, which suggests that learning contextual semantics helps guide the model to understand the document image. Furthermore, the combination of APP and SRR tasks also leads to improvement, i.e., 97.26% on SROIE and 83.70% on FUNSD. PEGT achieves a 3.41% performance gain in terms of F-measure on the challenging FUNSD dataset, and a 0.14% and 0.43% improvement upon a strong baseline on the SROIE and CORD datasets, showing the advantage of the proposed pre-training tasks. We also tried the widely used Masked language modeling (MLM) pre-training method. It can be seen that the mask recovery method is not suitable for our model framework and prompt template.

**Table 3.** Performance comparison of variants with different pre-training tasks on the SROIE, CORD, and FUNSD datasets.

Pre-training task(s)	SROIE			CORD			FUNSD		
	P	R	F	P	R	F	P	R	F
-	97.26	96.99	97.12	97.61	97.26	97.43	80.29	80.29	80.29
APP	97.16	97.33	97.25	97.86	97.04	97.45	81.32	81.32	81.32
SRR	97.22	97.05	97.14	98.15	<b>97.84</b>	<b>98.00</b>	82.11	82.11	82.11
APP + SRR	<b>97.31</b>	<b>97.21</b>	<b>97.26</b>	<b>98.19</b>	97.53	97.86	<b>83.70</b>	<b>83.70</b>	<b>83.70</b>

**Table 4.** Performance comparison of variants with different multimodal embeddings without pre-training tasks. T, V, and L denote the textual, layout, and visual modalities.  $V_{CLIP}$  represents the enhanced visual encoder introduced in this work.

Modality	SROIE			CORD			FUNSD		
	P	R	F	P	R	F	P	R	F
T+L	95.44	94.19	94.81	95.67	92.75	94.19	79.38	79.38	79.38
T+L+V	97.11	<b>97.06</b>	97.08	<b>97.98</b>	96.30	97.13	80.00	80.00	80.00
T+L+ $V_{CLIP}$	<b>97.26</b>	96.99	<b>97.12</b>	97.61	<b>97.26</b>	<b>97.43</b>	<b>80.29</b>	<b>80.29</b>	<b>80.29</b>

**Effectiveness of the Enhanced Visual Encoder.** As shown in Table 4, first, to show the importance of the visual modality information, we try to remove the image encoder from the network. It can be seen that if text modality and layout modality are used alone, the scores have declined in all three datasets. This proves that the visual modality can provide complementary information for the

model to complete the KIE task, which has not been fully explored. Then, we use the modified CLIP image encoder to replace ResNet in the network, which shows improvement in the three datasets. Therefore, introducing commonsense knowledge from the visual modality can enhance the ability to perceive visual text. Specifically, using our image encoder design strategy improves the baseline from 97.08% to 97.12% on the SROIE dataset, from 97.13% to 97.43% on the CORD dataset, and from 79.38% to 80.29% on the FUNSD dataset.

## 5 Conclusion

In this work, we propose a perception-enhanced generative transformer (PEGT) for the KIE task, which improves the model through fine-grained multimodal modeling and pre-training tasks tailored for the generative framework. We introduce a pre-trained vision-language model to provide commonsense knowledge for visual text perception. Then two auxiliary pre-training tasks including APP and SRR are designed for the generative framework to understand position information and semantic relationships better. Extensive experiments on KIE tasks including entity labeling and entity extraction are performed on three public classical datasets, demonstrating the effectiveness of the proposed method.

**Acknowledgements.** This work is supported by the fund of Laboratory for Advanced Computing and Intelligence Engineering.

## References

1. Appalaraju, S., Jasani, B., Kota, B.U., Xie, Y., Manmatha, R.: DocFormer: end-to-end transformer for document understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 993–1003 (2021)
2. Bao, H., et al.: UniLMv2: Pseudo-masked language models for unified language model pre-training. In: International Conference on Machine Learning, pp. 642–652. PMLR (2020)
3. Cao, P., Wang, Y., Zhang, Q., Meng, Z.: GenKIE: robust generative multimodal document key information extraction. In: Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 14702–14713 (2023)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I, pp. 213–229. Springer International Publishing, Cham (2020). [https://doi.org/10.1007/978-3-030-58452-8\\_13](https://doi.org/10.1007/978-3-030-58452-8_13)
5. Chen, Y.-C., et al.: UNITER: UNiversal Image-TExt representation learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX, pp. 104–120. Springer International Publishing, Cham (2020). [https://doi.org/10.1007/978-3-030-58577-8\\_7](https://doi.org/10.1007/978-3-030-58577-8_7)

6. Dosovitskiy, A., et al.: An image is worth  $16 \times 16$  words: transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
7. Feng, H., Liu, Q., Liu, H., Zhou, W., Li, H., Huang, C.: DocPedia: unleashing the power of large multimodal model in the frequency domain for versatile document understanding. arXiv preprint [arXiv:2311.11810](https://arxiv.org/abs/2311.11810) (2023)
8. Feng, H., Wang, Y., Zhou, W., Deng, J., Li, H.: DocTr: document image transformer for geometric unwarping and illumination correction. In: ACM Multimedia, pp. 273–281 (2021)
9. Garncairek, L, et al.: LAMBERT: layout-aware language modeling for information extraction. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12821, pp. 532–547. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-86549-8\\_34](https://doi.org/10.1007/978-3-030-86549-8_34)
10. He, J., et al.: ICL-D3IE: in-context learning with diverse demonstrations updating for document information extraction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 19485–19494 (2023)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
12. Hong, T., Kim, D., Ji, M., Hwang, W., Nam, D., Park, S.: BROS: a pre-trained language model focusing on text and layout for better key information extraction from documents. In: AACL, vol. 36, pp. 10767–10775 (2022)
13. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: LayoutLMv3: pre-training for document AI with unified text and image masking. In: ACM Multimedia, pp. 4083–4091 (2022)
14. Huang, Z., et al.: ICDAR2019 competition on scanned receipt OCR and information extraction. In: ICDAR, pp. 1516–1520. IEEE (2019)
15. Jaume, G., Ekenel, H.K., Thiran, J.P.: FUNSD: a dataset for form understanding in noisy scanned documents. In: ICDARW, vol. 2, pp. 1–6. IEEE (2019)
16. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019)
17. Kim, G., et al.: OCR-free document understanding transformer. In: European Conference on Computer Vision, pp. 498–517. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-19815-1\\_29](https://doi.org/10.1007/978-3-031-19815-1_29)
18. Kuang, J., et al.: Visual information extraction in the wild: practical dataset and end-to-end solution. In: ICDAR, pp. 36–53. Springer, Cham (2023). [https://doi.org/10.1007/978-3-031-41731-3\\_3](https://doi.org/10.1007/978-3-031-41731-3_3)
19. Lample, G., Conneau, A.: Cross-lingual language model pretraining. arXiv preprint [arXiv:1901.07291](https://arxiv.org/abs/1901.07291) (2019)
20. Li, G., Duan, N., Fang, Y., Gong, M., Jiang, D.: Unicoder-VL: a universal encoder for vision and language by cross-modal pre-training. In: AACL, vol. 34, pp. 11336–11344 (2020)
21. Li, J., Sun, A., Han, J., Li, C.: A survey on deep learning for named entity recognition. IEEE Trans. Knowl. Data Eng. **34**(1), 50–70 (2020)
22. Li, P., et al.: SelfDoc: self-supervised document representation learning. In: CVPR, pp. 5652–5660 (2021)
23. Li, Y., et al.: StrucTexT: structured text understanding with multi-modal transformers. In: ACM Multimedia, pp. 1912–1920 (2021)
24. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) (2019)

25. Lu, J., Batra, D., Parikh, D., Lee, S.: ViLBERT: pretraining task-agnostic visual-linguistic representations for vision-and-language tasks. In: *NeurIPS*, vol. 32 (2019)
26. Materzyńska, J., Torralba, A., Bau, D.: Disentangling visual and written concepts in clip. In: *CVPR*, pp. 16410–16419 (2022)
27. Park, S., et al.: CORD: a consolidated receipt dataset for post-OCR parsing. In: *Workshop on Document Intelligence at NeurIPS 2019* (2019)
28. Qiao, Z., Qin, X., Zhou, Y., Yang, F., Wang, W.: Gaussian constrained attention network for scene text recognition. In: *ICPR*, pp. 3328–3335. IEEE (2020)
29. Qin, X., et al.: Towards robust real-time scene text detection: from semantic to instance representation learning. In: *ACM Multimedia*, pp. 2025–2034 (2023)
30. Qin, X., et al.: Mask is all you need: rethinking mask R-CNN for dense and arbitrary-shaped scene text detection. In: *ACM Multimedia*, pp. 414–423 (2021)
31. Qin, X., Zhou, Y., Guo, Y., Wu, D., Wang, W.: Fc<sup>2</sup>rn: a fully convolutional corner refinement network for accurate multi-oriented scene text detection. In: *ICASSP*, pp. 4350–4354. IEEE (2021)
32. Qin, X., Zhou, Y., Yang, D., Wang, W.: Curved text detection in natural scene images with semi-and weakly-supervised learning. In: *ICDAR*, pp. 559–564. IEEE (2019)
33. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: *International Conference on Machine Learning*, pp. 8748–8763. PMLR (2021)
34. Stanisławek, T., et al.: Kleister: key information extraction datasets involving long documents with complex layouts. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) *Document Analysis and Recognition – ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I*, pp. 564–579. Springer International Publishing, Cham (2021). [https://doi.org/10.1007/978-3-030-86549-8\\_36](https://doi.org/10.1007/978-3-030-86549-8_36)
35. Sun, S., Deng, J., Qin, X.: Unearthing historical insights: semantic organization and application of historical newspapers from a fine-grained knowledge element perspective. *ASLIB J. Inf. Manage.* (2023). <https://doi.org/10.1108/AJIM-05-2023-0180>
36. Tan, H., Bansal, M.: LXMERT: learning cross-modality encoder representations from transformers. In: *EMNLP-IJCNLP*, pp. 5100–5111 (2019)
37. Tong, X., Dai, P., Qin, X., Wang, R., Ren, W.: Granularity-aware single-point scene text spotting with sequential recurrence self-attention. *IEEE Trans. Circuits Syst. Video Technol.* (2024)
38. Touvron, H., et al.: LLaMA: open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023)
39. Wang, P., et al.: OFA: unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In: *International Conference on Machine Learning*, pp. 23318–23340. PMLR (2022)
40. Wang, W., et al.: VisionLLM: large language model is also an open-ended decoder for vision-centric tasks. In: *NeurIPS* (2024)
41. Xu, Y., et al.: LayoutLMv2: multi-modal pre-training for visually-rich document understanding. *ACL Assoc. Comput. Linguist.* (2021)
42. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: LayoutLM: pre-training of text and layout for document image understanding. In: *ACM SIGKDD*, pp. 1192–1200 (2020)
43. Yang, Z., et al.: TAP: text-aware pre-training for Text-VQA and text-caption. In: *CVPR*, pp. 8751–8761 (2021)

44. Yu, Y., et al.: StrucTexTv2: masked visual-textual prediction for document image pre-training. In: International Conference on Learning Representations (2023)
45. Zhou, C., Loy, C.C., Dai, B.: Extract free dense labels from CLIP. In: European Conference on Computer Vision, pp. 696–712. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-19815-1\\_40](https://doi.org/10.1007/978-3-031-19815-1_40)



# MuLAD: Multimodal Aggression Detection from Social Media Memes Exploiting Visual and Textual Features

Md. Maruf Hasan<sup>1</sup>, Shawly Ahsan<sup>1</sup>, Mohammed Moshui Hoque<sup>1</sup> (✉),  
and M. Ali Akber Dewan<sup>2</sup>

<sup>1</sup> Chittagong University of Engineering and Technology, Chittagong 4349,  
Bangladesh

{u1604089,u1704057}@student.cuet.ac.bd, moshui.240@cuet.ac.bd

<sup>2</sup> School of Computing and Information Systems, Faculty of Science and Technology,  
Athabasca University, Athabasca, AB T9S 3A3, Canada  
adewan@athabascau.ca

**Abstract.** Aggression detection from memes is challenging due to their region-specific interpretation and multimodal nature. Detecting or classifying aggressive memes is complicated (Bengali) because benchmark datasets and primary language processing software are needed. This paper proposes an innovative meme classification technique that harnesses deep learning (DL) approaches to leverage memes' visual and textual features in Bengali. Various DL frameworks, such as VGG16, VGG19, ResNet50, CNN, BiLSTM, and BiLSTM+CNN, extract visual and textual features from memes. A novel corpus named the Bengali Meme Dataset (AMemD) is also introduced, comprising a substantial amount of multimodal data, including text and image components. Experimental results on AMemD demonstrate the effectiveness of the proposed approach. The CNN combined with VGG16 obtained the highest  $f_1$ -score of 0.738 among all multimodal techniques tested. This pioneering research offers valuable insights into the complex task of aggression detection from memes in Bengali and provides a foundation for future studies in this area. The dataset is available at <https://github.com/Maruf089/Multimodal-Aggression-Detection>.

**Keywords:** Natural language processing · Meme classification · Deep learning · Aggressive memes · Multimodal fusion

## 1 Introduction

With the significant rise in internet usage, social media has emerged as a powerful platform for conveying information, expressing opinions, and conveying emotions on various topics. The proliferation of symbolic, offensive, obscene photos, inappropriate gestures, and provocative textual comments on social media has underscored the need for effective identification and classification of aggressive

content. Memes have become a popular tool for spontaneously transmitting ideas or emotions. Their humorous or sarcastic nature makes them an effective means of spreading information on social media, where posting and sharing memes has recently surged in popularity. Recently, memes have gained popularity to convey information on online media. Typically consisting of visuals with embedded text, they can rapidly spread hatred and offensive content. Memes, being popular forms of internet communication, often contain elements with humorous, satirical, or provocative messages. Their multimodal nature, contextual dependencies, data sparsity, and annotation challenges further complicate the detection process. The propagation of hostile memes and other connected actions through memes, such as trolling and cyberbullying, is rapidly advancing. Detecting aggressive memes often requires context and understanding of the cultural and social implications behind the content. Automated systems can assist in flagging potentially aggressive content, but human judgment and oversight are crucial for accurate and nuanced detection. Meme classification has become increasingly complex due to its implicit meanings, ambiguous, humorous, and sarcastic language use, and the inclusion of eye-catching, comical, and theatrical images.

The detection and classification of aggressive memes are crucial for safeguarding the well-being of users, promoting civil discourse, preventing radicalization, and upholding legal and ethical standards in online spaces. Detecting multimodal aggressive content or social media memes has significantly progressed in high-resource languages. However, detecting aggressive memes in resource-constrained languages needs to be more robust due to the lack of benchmark corpora and language processing tools. Memes embedded with Bengali text have spread exponentially in recent years due to the proliferation of Internet usage. Although few studies concentrated on detecting aggressive content using a single modality (text or image), multimodal aggression content detection is still a work in progress concerning the Bengali language. No substantial tools or techniques have been developed to manage multimodal aggression on social media concerning Bengali. Considering the current constraints of aggressive meme detection in Bengali, this work presents an intelligent system to detect multimodal aggressive memes (image and text) leveraging DL models exploiting visual and textual features. After analyzing the results from each modality, an early fusion approach is utilized to integrate features from both visual and textual modalities for detecting aggressive memes. The key contributions of this work are highlighted as follows:

- Developed AMemD, Aggressive Meme Detection corpus comprising 1718 Bengali memes. This corpus serves as a crucial resource for training and evaluating the multimodal framework for aggressive meme detection.
- Developed MuLAD, an automatic framework to identify multimodal aggressive memes by leveraging various DL architectures (LSTM, Bi-LSTM, CNN, VGG16, VGG19, and ResNet50) exploiting the visual, textual, and multimodal features.

## 2 Related Work

Recent studies have focused on identifying content such as trolling [22], aggression [17], and hate speech [2] from a single modality (e.g., image, text). To detect cyber-trolling in tweets, Sadiq et al. [16] designed and analyzed several methods, including a Multi-Layer Perceptron (MLP) with TF-IDF, and word embeddings and two DNN architectures (CNN+LSTM and CNN+BiLSTM). Their results indicated that the MLP with TF-IDF features surpassed the other techniques, achieving 92% accuracy. Zampieri et al. [24] developed an English offensive language detection dataset and conducted baseline experiments using CNN, BiLSTM, and SVM approaches. The CNN model attained the most elevated macro  $f_1$ -score (0.80) for offensive language detection. Chen et al. [4] suggested a CNN-based technique for identifying verbal aggression in tweets, incorporating sentiment analysis. Additionally, Suryawanshi et al. [21] published a Tamil troll and non-troll memes dataset. They employed pre-trained image categorization algorithms such as ResNet and MobileNet to distinguish between meme categories. Though their approach attained a macro  $f_1$ -score of 0.52, it served inadequately on the troll class, with a lower recall value (0.37).

Multimodal learning has recently acquired popularity owing to its capability to effectively integrate knowledge from various modalities into a unified learning architecture [15]. Yadav et al. [23] surveyed the application of DL approaches for sentiment analysis and followed a growing trend among researchers towards integrating multiple modalities (audio, images, and video), somewhat depending on the text alone. Consequently, researchers increasingly adopt multimodal techniques to detect objectionable content in memes, acknowledging the significant negative impact such content can have on society [14]. Kumari et al. [13] presented a strategy where the visual features are recovered employing pre-trained VGG16, and the text features are dragged using CNN. These features are optimized employing the binary particle swarm optimization technique, attaining a  $f_1$ -score of 0.74. Suryawanshi et al. [20] created a multimodal corpus of 743 offensive and not-offensive memes. To integrate the multimodal features, they employed an early fusion strategy. The integrated technique acquired a  $f_1$ -score of 0.50, while the text-based CNN method surpassed the other models ( $f_1$ -score of 0.54). Hossain et al. [8] introduced an inter-modal attention-based framework for offensive meme detection, employing VGG19 and BERT as feature extractors. Their approach yielded a weighted  $f_1$ -score of 0.635. Sharma et al. [19] curated a dataset designed to identify targets affected by harmful memes and introduced DISARM, a novel multimodal framework. The authors evaluated their models on three test sets: entities encountered during training ( $f_1$ -score of 0.7845), entities not encountered as harmful targets during training ( $f_1$ -score of 0.6498), and entities entirely unseen during training ( $f_1$ -score of 0.641). Gasparini et al. [7] introduced a new dataset to detect multimodal misogynistic content comprising 800 memes. They annotated these memes, categorizing them based on whether they exhibit misogyny, aggression, and irony. Zhu et al. [25] introduced TAME, a novel multimodal framework designed to detect hateful memes in a zero-shot



setting, taking into account the targets of hate speech. Their approach achieved an accuracy of 66.81% in generalized zero-shot learning scenarios.

Sharif et al. [18] used a hierarchical annotation schema to create an aggressive text identification corpus in Bengali. They investigated a variety of ML and DL approaches. The CNN+BiLSTM obtained the best  $f_1$ -scores of 0.87 (coarse) and 0.80 (fine-grained). Kumar et al. [12] introduced ComMA, a multilingual, hierarchical, fine-grained dataset explicitly designed for detecting aggression and bias within comments, memes, and audio content. Their research primarily focused on four languages: Meitei, Bangla, Hindi, and Indian English. They ran a series of experiments with different baseline models. They found that XLM-R produced the highest  $f_1$ -scores for aggression (0.58) and gender bias (0.52), while MuRIL produced the highest  $f_1$ -score for communal bias (0.56). Hossain et al. [10] proposed a new technique for aligning unimodal features before integrating them for multimodal detection of hateful content. They assessed their method using two datasets, MUTE (Bengali) and MultiOFF (English), achieving  $f_1$ -scores of 69.7% and 70.3%, respectively. Dutta et al. [6] presented a multitask learning strategy for classifying emotion in comics. To address the challenge of mislaid modalities, they operated three distinct classifiers, with the conclusive decision being assembled via a unified decision module. Ahsan et al. [1] created a target-aware multimodal aggressive meme dataset in Bengali, which includes 4848 memes categorized into five groups (one non-aggressive category and four aggressive categories). They introduced a novel multimodal framework employing an attention-based fusion of unimodal features, achieving a weighted  $f_1$ -score of 0.742. Hossain et al. [11] created a novel multimodal dataset to detect hateful memes and identify their targets within Bengali meme content. They constructed DORA, a dual co-attention-based multimodal framework tailored for hateful meme detection, achieving  $f_1$ -scores of 0.718 for hateful meme detection and 0.720 for target identification.

Most past studies in Bengali focused on detecting aggressive memes based on a unique modality (e.g., text). However, it is usually critical to comprehend and categorize the contents of a meme regarding multiple modalities. Therefore, exploring visual and textual modalities to detect aggressive memes is essential. This work presents a DL-based framework for detecting multimodal aggressive memes exploiting textual and visual features with a late fusion approach.

### 3 AMemD: A New Aggressive Memes Dataset

At the outset of our research endeavor, we recognized the need for a specialized dataset to detect aggression from Bengali memes. Therefore, we meticulously curated Bengali memes from diverse social media platforms and developed **AMemD**—an innovative multimodal dataset for aggressive meme detection. To maintain uniformity and quality, we adhered to the dataset development protocols outlined by Hossain et al. [9]. Figure 1 illustrates the developmental steps of **AMemD**.

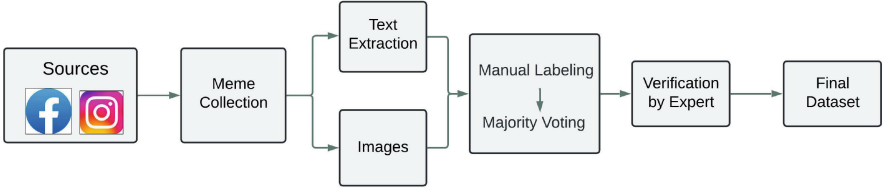


Fig. 1. AMemD development steps

### 3.1 Data Collection and Preprocessing

We gathered data from multiple social media sources, including Facebook and Instagram. Specifically, 1,065 memes out of 1,718 (approximately 62%) were collected from Facebook, while the remaining 653 memes (approximately 38%) were obtained from Instagram. To capture a broad spectrum of content, we utilized a variety of keywords such as “Bengali Memes,” “Bengali Troll Memes,” “Bengali Aggressive Memes,” “Bengali Hateful Memes,” “Bengali Offensive Memes,” “Bengali Funny Memes,” “Bengali Celebrity Memes,” “Bengali Cricket Memes,” “Bengali Political Memes,” “Bengali Sports Memes,” and “Bengali Ironic Memes.” This keyword-based search strategy was employed to ensure inclusivity across different meme categories and minimize dataset bias. During the data collection, we gathered 1,780 memes from publicly accessible meme groups and pages to mitigate copyright concerns. Each meme underwent scrutiny, leading to the removal of memes that met the following criteria: (i) memes that had blurry images or unclear text, (ii) memes that consisted of unimodal content (either image or text), and (iii) memes that were duplicates. As a result of this review, 62 memes were excluded, leaving us with 1,718 memes. Given the absence of standard optical character recognition (OCR) for Bengali, we manually extracted text from the images. Subsequently, we manually reviewed the extracted text to correct typographical errors, including misspellings and grammatical inaccuracies.

### 3.2 Data Annotation

We conducted manual annotation of **AMemD**, categorizing them into two distinct groups: *non-aggressive* and *aggressive*. Annotators were given a clear definition of what constitutes an aggressive meme to maintain consistency and reliability throughout the annotation process.

**Defining Aggressive Memes:** We examined previous studies on detecting aggression [13, 18] and identified aggressive memes as multimedia elements consisting of both an image and accompanying text. These memes can physically intimidate, attack, or intend harm toward an individual, group, or community. Memes also can include factors like political views, religious beliefs, sexual orientation, gender, race, or nationality, or they may include nudity, sexually suggestive material, items promoting violence, or racially charged content. For example,

the meme in Fig. 2b is aggressive as it attacks an internet personality by expressing a wish to assault him physically. On the other hand, the meme in Fig. 2a is non-aggressive since it compares the year 2020 with a bitter vegetable and is lighthearted and humorous.



**Fig. 2.** Instances of aggressive and non-aggressive memes. The criteria used to decide the category: a) contains humorous content by comparing the year 2020 with a bitter vegetable, b) attacks an internet personality, c) makes an offensive remark but does not physically threaten or attack anyone

Aggressive content is distinct from other forms of undesirable content in several ways. While aggressive content is often harmful and offensive, not all harmful, hateful, or offensive content can be classified as aggressive. Offensive content encompasses material that is insulting, derogatory, discriminatory, or otherwise inappropriate [20]. However, it does not necessarily include direct threats or attacks, unlike aggressive content. For instance, Fig. 2c is categorized as offensive because it makes a derogatory comment about an individual’s appearance by likening them to a zebra. Although the content is offensive, it does not contain direct threats or physical harm towards anyone. Strong feelings of animosity characterize hateful content and promote hostility, discrimination, or prejudice against specific individuals or groups based on factors such as race, gender, religion, sexual orientation, or political views [9]. Unlike aggressive content, hateful content often reflects an extreme bias towards certain groups without necessarily including direct threats or physical aggression.

**Process of Annotation:** The initial annotation process was conducted by three undergraduate students, who were instructed to classify memes as aggressive or non-aggressive based on the provided definitions. To ensure accuracy and consistency in their annotations, the students underwent training using a small subset of the data to familiarize themselves with recognizing aggressive content. Annotators were guided to label the images objectively, avoiding bias towards any particular demographic region, culture, sensitive topics, or religious beliefs. Initial labels were determined through a majority voting system, as outlined in Algorithm 1. When disagreements arose, a seasoned expert with over 20 years of

experience in NLP intervened to resolve conflicts and validate the annotations. We assessed the quality of the annotations by calculating the inter-annotator agreement with Cohen’s Kappa coefficient [5]. The pairwise Kappa score between annotators 1 and 2 and annotators 2 and 3 are 0.890 and 0.855, while the score is 0.923 between annotators 2 and 3. The average Kappa value for the AMemD dataset is 0.889, indicating almost perfect agreement based on the Kappa scale [3].

---

**Algorithm 1:** Majority Voting & Initial Label

---

```

1  $T \leftarrow$  Text corpus;
2  $Labels \leftarrow [0,1]$ ;
3  $AL \leftarrow$  Label of annotators;
4  $IL \leftarrow$  Initial Labels;
5 for  $t_i \in T$  do
6   | label_count = [0,0];
7   | for  $a_{ij} \in AL$  do
8   |   | label_count[ $a_{ij}$ ]++;
9   | end
10  |  $IL_i = \text{indexof}[\max(\text{label\_count})]$ ;
11 end

```

---

### 3.3 Dataset Statistics

The dataset comprises 1,718 memes in jpg image format, with a total size of 210 MB. Table 1 displays the distribution of the dataset between the training and the test sets, along with statistics specific to the training set, such as the total data in each class ( $T_t$ ), total word count ( $T_w$ ) and the count of unique words ( $T_{uw}$ ). The training set contains 1425 memes, and the testing set contains 293. The aggressive class has 1,671 words, with 1,141 unique words. Conversely, the non-aggressive class contains 14,778 words, of which 6,112 are unique. The majority of texts consist of fewer than 20 words.

**Table 1.** Dataset distribution

Dataset	Train	Test	$T_t$	$T_w$	$T_{uw}$
Aggressive	140	55	195	1671	1141
Non-Aggressive	1285	238	1523	14778	6112
<b>Total</b>	1425	293	1718	16449	7340

## 4 Methodology

This research presents **MuLAD**, a comprehensive framework for detecting aggression within multimodal memes. **MuLAD** comprises several key components: preprocessing, textual feature extraction, visual feature extraction, and

multimodal fusion. Figure 3 depicts the abstract architecture of the proposed framework.

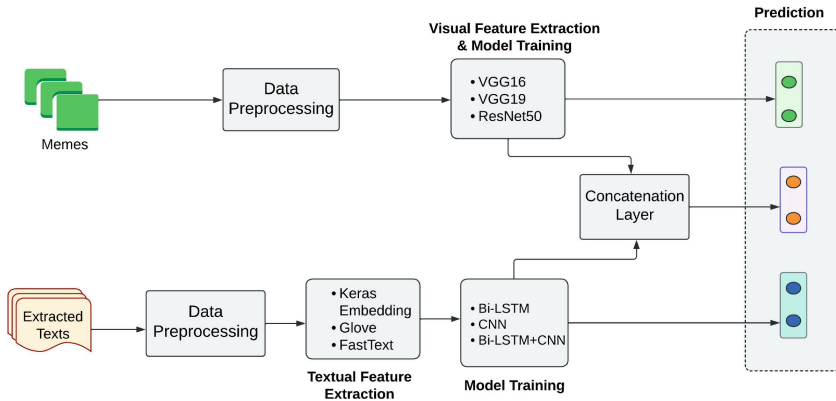


Fig. 3. Abstract framework for multimodal aggressive meme detection (MuLAD)

#### 4.1 Preprocessing

Initially, the raw text data undergoes preprocessing to remove punctuation, hyperlinks, emojis, and special characters. Subsequently, the cleaned text data undergoes tokenization and is converted into a vector of integers via the Keras tokenizer function. To ensure uniformity in the data structure, the tokenized vectors are padded to achieve equal-length sequences. Likewise, the images undergo preprocessing before being utilized in deep learning (DL) models. This process involves resizing the images to  $224 \times 224 \times 3$ , then preprocessing using Keras functions tailored to each pre-trained image model.

#### 4.2 Textual and Visual Feature Extraction

This work exploited various word embedding techniques, such as Keras, FastText, and GloVe, to extract textual features for DL models.

- **Keras:** We instantiated an embedding layer with a vocabulary comprising 8000 words, utilizing a vector space of 64 dimensions and enforcing a maximum sequence length of 130.
- **FastText:** Pre-trained FastText embedding was employed, trained to employ CBOW with position weights. The embedding model was trained with vectors of dimensionality 300, comprising character n-grams (length 5), a window (size of 5), and ten negative samples.

- **Glove:** The GloVe pre-trained model was trained with a dimension of 300, a window size of 5, and a *min\_count* of 5. Pre-trained FastText embedding was employed, trained to employ CBOW with position weights. The embedding model was trained with vectors of dimensionality 300, comprising character n-grams (length 5), a window (size of 5), and ten negative samples

The embedded text was passed to several DL models to extract textual features, including CNN, BiLSTM, and CNN+BiLSTM. The CNN architecture comprises a single convolutional layer with a filter (size of 128) and a kernel (size of 5). The convolution layer is obeyed by a max pooling layer and a dense layer with 32 neurons. Subsequently, a dense layer with a sigmoid activation function was employed for binary classification. Extensive experimentation was conducted with the CNN architecture to refine the framework. Table 2 demonstrates the hyperparameters of the CNN architecture.

**Table 2.** Hyperparameters for the CNN model. The acronyms AF and LR denote the activation function and learning rate.

Hyperparameters	Hyperparameter Space	CNN
Kernel Size	3, 5, 7	5
Pooling Type	‘max’, ‘average’	‘max’
Embedding Dimension	32, 64, 128	64
Batch Size	16, 32, 64, 128	32
AF	‘relu’, ‘sigmoid’	‘relu’
Optimizer	‘adam’, ‘SGD’	‘adam’
LR	0.0001, 0.001, 0.05	0.001

We utilized various pre-trained image models (e.g., VGG16, VGG19, and ResNet50), employing transfer learning techniques to find visual features. Initially, the top layers of these models were discarded. Subsequently, a flattening layer and dropout layers are added to the base model. Additionally, *relu* activation was incorporated into each layer to ensure that negative values were not propagated to subsequent layers. The model is finalized with a dense layer featuring a sigmoid activation function for classification.

### 4.3 Multimodal Fusion

To construct a multimodal framework, we combined the final layers of the textual and visual models through concatenation. Subsequently, a dense layer featuring a sigmoid activation function was added for binary classification. During training, we employed the *‘binary\_crossentropy’* loss function and the *‘adam’* optimizer. Extensive experimentation was conducted, employing three visual models for visual feature extraction and nine textual models for textual feature extraction to identify the optimal framework. Figure 4 depicts the overall architecture of the proposed multimodal framework (MuLAD).

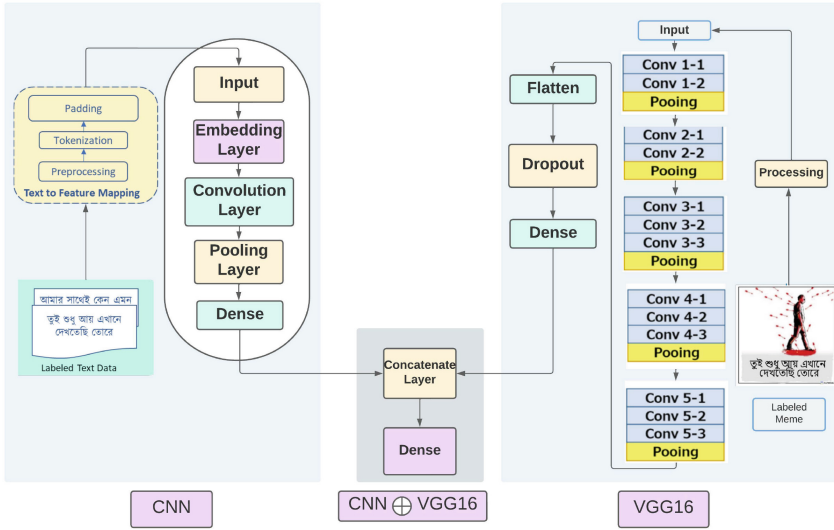


Fig. 4. Proposed framework for multimodal aggressive meme detection (MuLAD)

## 5 Experiments

The experiments were performed on the Google Colab platform, leveraging TensorFlow frameworks to construct DL architectures. Various Keras functions were employed for data preprocessing, model construction, and training. The BNLPT toolkit<sup>1</sup> facilitated textual data preprocessing, while model evaluation utilized the Scikit-Learn library. Data visualization was performed using the Matplotlib library. The weighted  $f_1$ -score is the primary metric for assessing the models’ performance.

### 5.1 Baselines

We investigated three visual and nine textual baseline models to evaluate the proposed multimodal framework’s performance.

**Visual Baselines:** With the transfer learning approach, this work investigated several pre-trained image models (such as VGG16, VGG19, and ResNet50). These models were chosen as baselines because they performed exceptionally well across various image classification tasks. The top two layers of each model were removed first, and then a global average pooling layer was added. The final layer of the model was a dense layer with a sigmoid activation function for binary classification. We trained the models using the ‘binary\_crossentropy’ loss function and ‘adam’ optimizer. We used a LR of  $1e^{-3}$  and a batch size of 32.

<sup>1</sup> <https://pypi.org/project/bnlp-toolkit/>.

In addition, we used the Keras callback technique to store the finest intermediate model while training.

**Textual Baselines:** To establish textual baseline models, we explored DL architectures, including CNN, BiLSTM, and CNN+BiLSTM. Initially, the preprocessed texts underwent embedding using diverse techniques such as Keras, FastText, and GloVe. For Keras embedding, we employed an output dimension of 64 with an input dimension of 7340, representing the vocabulary size. The CNN architecture comprised a convolutional layer with 32 units and a kernel size of 5, followed by a maxpooling layer. To enhance the model’s capability of capturing long-term dependencies within the text, a BiLSTM layer consisting of 100 neurons was integrated into the CNN network to create the BiLSTM+CNN model.

## 6 Results

Table 3 presents the performance measures of the textual baseline techniques on the test dataset, where the metrics A (accuracy), P (weighted precision), R (weighted recall), and WF (weighted  $f_1$ -score) are utilized. Among the models employing Keras embedding, the BiLSTM architecture earned the most elevated WF score of 0.887. Conversely, for the pre-trained GloVe and FastText embeddings, the CNN framework attained the maximum WF score of 0.888 in both cases. Specifically, the CNN model utilizing FastText word embedding demonstrated P and A of 0.907 and 0.901.

**Table 3.** Performance of textual models on the test set

Classifier	Keras				GloVe				FastText			
	A	P	R	WF	A	P	R	WF	A	P	R	WF
BiLSTM	0.901	0.908	0.901	<b>0.887</b>	0.894	0.895	0.894	0.881	0.901	0.912	0.901	0.886
CNN	0.870	0.862	0.870	0.855	0.901	0.907	0.901	<b>0.888</b>	0.900	0.904	0.901	<b>0.888</b>
BiLSTM+CNN	0.881	0.874	0.881	0.867	0.887	0.883	0.887	0.875	0.884	0.879	0.884	0.871

In contrast, VGG16 exhibited the most elevated WF score (0.789) among the visual baseline models, with a P (0.784) and A (0.816). Table 4 demonstrates the outcomes of visual models. These findings suggest that VGG16 outperformed ResNet50 and VGG19 regarding performance metrics. The results indicate that the textual baseline models surpassed the visual baseline models’ performance.

Table 5 shows the performance of the multimodal techniques on the test dataset. The CNN $\oplus$ VGG16 model acquired the most elevated WF score of 0.738 with a P of 0.717 and an A of 0.778.



**Table 4.** Performance of visual models on the test set

Classifier	A	P	R	WF
ResNet50	0.812	0.659	0.812	0.728
VGG16	0.816	0.784	0.816	<b>0.789</b>
VGG19	0.785	0.690	0.785	0.725

**Table 5.** Performance of the multimodal techniques on the test set

Classifier	Keras				GloVe				FastText			
	A	P	R	WF	A	P	R	WF	A	P	R	WF
BiLSTM $\oplus$ ResNet50	0.812	0.659	0.812	0.728	0.812	0.659	0.812	0.728	0.812	0.659	0.812	0.728
CNN $\oplus$ ResNet50	0.764	0.676	0.764	0.713	0.747	0.712	0.747	0.727	0.747	0.712	0.747	0.727
BiLSTM+CNN $\oplus$ ResNet50	0.812	0.659	0.812	0.728	0.812	0.659	0.812	0.728	0.812	0.659	0.812	0.728
BiLSTM $\oplus$ VGG16	0.747	0.669	0.747	0.704	0.375	0.698	0.375	0.411	0.512	0.744	0.512	0.562
<b>CNN<math>\oplus</math>VGG16</b>	0.778	0.717	0.778	<b>0.738</b>	0.450	0.700	0.450	0.502	0.812	0.659	0.812	0.728
BiLSTM+CNN $\oplus$ VGG16	0.812	0.659	0.812	0.728	0.812	0.659	0.812	0.728	0.512	0.744	0.512	0.562
BiLSTM $\oplus$ VGG19	0.812	0.659	0.812	0.728	0.699	0.715	0.699	0.707	0.484	0.719	0.485	0.536
CNN $\oplus$ VGG19	0.785	0.690	0.785	0.724	0.812	0.659	0.812	0.728	0.812	0.659	0.812	0.728
BiLSTM+CNN $\oplus$ VGG19	0.812	0.659	0.812	0.728	0.812	0.659	0.812	0.728	0.812	0.659	0.812	0.728

## 6.1 Class-Wise Performance Analysis

To enhance our understanding of the developed architectural design’s effectiveness, we assessed the model’s performance across individual classes and compared it against the top-performing visual and textual baseline models (refer to Table 6). Notably, the top-performing textual baseline model, employing CNN with FastText embeddings, outperformed the leading visual baseline model (utilizing VGG16 architecture) and the proposed multimodal model. Specifically, it achieved a precision of 0.964, a recall of 0.491, and an  $f_1$ -score of 0.651 for the aggressive class, and a precision of 0.894, a recall of 0.996, and an  $f_1$ -score of 0.942 for the non-aggressive class. Although excelling in precision for the aggressive class, this textual baseline model exhibited inferior recall, resulting in a reduced  $f_1$ -score for this particular class. Conversely, the proposed framework encountered challenges in correctly identifying instances of the aggressive class, evident from its notably low  $f_1$ -score of 0.156. However, it demonstrated commendable performance for the non-aggressive class, obtaining an  $f_1$ -score of 0.872. This discrepancy can be ascribed to the scarcity of training data for the aggressive class, which constrains the model’s ability to effectively learn distinguishing features for this category.

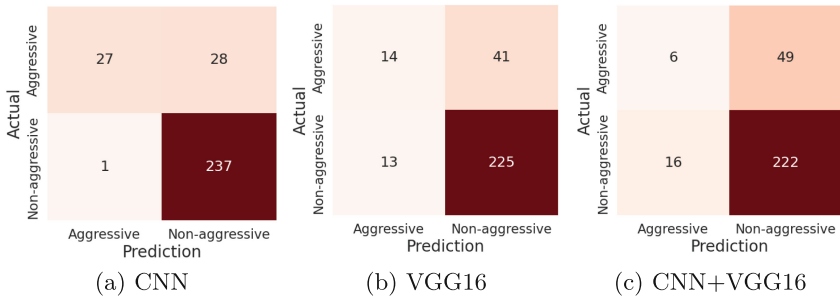
## 6.2 Error Analysis

We performed an exhaustive error analysis encompassing quantitative and qualitative examinations to discern the specific patterns underlying misclassifications within the non-aggressive and aggressive classes.

**Table 6.** Class-wise performance of the best visual, textual, and proposed multimodal framework (MuLAD) on the test set

Class	Classifier	Precision	Recall	$f_1$ -score
Aggressive	VGG16	0.519	0.255	0.341
	CNN (FastText)	<b>0.964</b>	<b>0.491</b>	<b>0.651</b>
	CNN (Keras) $\oplus$ VGG16	0.273	0.109	0.156
Non-aggressive	VGG16	0.846	0.945	0.893
	CNN (FastText)	<b>0.894</b>	<b>0.996</b>	<b>0.942</b>
	CNN (Keras) $\oplus$ VGG16	0.819	0.933	0.872

**Quantitative Analysis:** Quantitative error analysis was conducted on the best baseline models and the proposed architecture using confusion matrices (refer to Fig. 5). Among the 55 aggressive memes evaluated, the CNN $\oplus$ VGG16 model accurately classified merely six images while misidentifying 49 as non-aggressive. This disparity can be attributed to the scarcity of data within the aggressive class and the relatively smaller dataset size. Similarly, the VGG16 model encountered difficulty discerning aggressive memes, correctly identifying only 14 out of the 55. However, it demonstrated proficient performance in classifying non-aggressive memes, accurately categorizing 225 out of 238, with only 13 misclassifications. Conversely, the CNN model improved performance in identifying aggressive memes, correctly categorizing 27 out of the 55 instances. Furthermore, it achieved a notably higher accuracy in classifying non-aggressive memes, accurately identifying 237 out of 238 instances. Notably, all models were biased towards categorizing memes as non-aggressive, evidenced by the higher frequency of misclassifications into this category. This trend likely stems from the inherent overlap in meme content across various classes and the limited availability of data within the aggressive class, posing challenges for multimodal models to predict the actual class accurately.

**Fig. 5.** Confusion matrix for (a) best textual baseline, (b) best visual baseline, and (c) best multimodal model (MuLAD)

**Qualitative Analysis:** For deeper insights, we investigated select inputs to compare their actual class with the predictions made by the top three models, including our proposed method. Figure 6 visually represents the actual and predicted outputs for a subset of sample inputs. Upon analysis, it is evident that the text and image classifiers accurately predict their respective modalities in the first and third samples. Consequently, the multimodal model achieves a correct prediction due to the individual classifiers’ alignment. However, in the second sample, discrepancies arise as the text and image classifiers yield conflicting predictions. Consequently, the multimodal model fails to make an accurate prediction due to the inherent overlapping characteristics present in memes, leading to ambiguity in classification. This challenge could be addressed by expanding the dataset with auxiliary memes exhibiting such overlapping characteristics, thereby enhancing the model’s ability to discern nuanced distinctions.

Meme			
Text on Meme	তুই শুধু আয় এখানে দেখতেছি তোরে	এ যুগের ছেলেমেয়েদের দেখলেই মারতে ইচ্ছে করে	ভাল্লাগে যখন ক্লাসে টিচার পড়া বাদ দিয়ে, নিজের জীবনের গল্প শুরু করে দেন
True Label	Aggressive	Aggressive	Non-aggressive
Text Classifier	Aggressive	Aggressive	Non-aggressive
Image Classifier	Aggressive	Non-Aggressive	Non-aggressive
CNN $\oplus$ VGG16	Aggressive	Non-Aggressive	Non-aggressive

**Fig. 6.** Sample predictions by MuLAD (CNN $\oplus$ VGG16)

## 7 Discussion

Using a multimodal approach, we employed various deep-learning architectures to detect aggression in memes. The proposed model, MuLAD, which combines CNN with Keras embedding and VGG16, obtained the highest f1-score of 0.738 among all multimodal approaches. However, it was outperformed by the best textual baseline model (CNN with FastText embedding). The underperformance of the multimodal model may be attributed to the limitations and imbalance within the dataset. To address these issues in future work, the dataset could be expanded to increase its scope and diversity, incorporating a wider range of

meme styles, domains, and types of aggression. Additionally, improving the quality of annotations by involving more skilled and experienced annotators would help ensure the accuracy and reliability of the data. While this research focused on deep learning architectures, future research could explore more advanced models, such as transformers, attention mechanisms, and large language models (LLMs), to enhance the effectiveness and performance of the multimodal approach.

## 8 Conclusion

This study introduced MuLAD, a multimodal approach for identifying aggressive memes in Bengali. It utilized CNN and VGG16 models and explored various fusion techniques across visual, textual, and visual-textual data to classify memes in a developed dataset. Results showed that the CNN with FastText embedding achieved the most elevated weighted  $f_1$ -score (0.888) among textual approaches, while the VGG16 model scored the highest (0.789) among visual approaches. However, when features were combined from both CNN and VGG16, the performance of the multimodal MuLAD model decreased (0.738  $f_1$ -score). Future research aims to augment the dataset with more multimodal aggressive data to improve the approach. Investigating advanced techniques like transformer-based models (such as BERT, Visual BERT, and ViL-BERT) and large language models (LLMs) could enhance overall performance.

## References



1. Ahsan, S., Hossain, E., Sharif, O., Das, A., Hoque, M.M., Dewan, M.: A multimodal framework to detect target aware aggression in memes. In: Graham, Y., Purver, M. (eds.) Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, St. Julian's, Malta, pp. 2487–2500. ACL (2024)
2. Basile, V., et al.: SemEval-2019 task 5: multilingual detection of hate speech against immigrants and women in Twitter. In: Proceedings International Workshop on Semantic Evaluation, Minneapolis, Minnesota, USA, pp. 54–63. ACL (2019)
3. Berry, K.J., Mielke, P.W., Jr.: A generalization of Cohen's kappa agreement measure to interval measurement and multiple raters. *Educ. Psychol. Measur.* **48**(4), 921–933 (1988)
4. Chen, H., Mckeever, S., Delany, S.J.: Harnessing the power of text mining for the detection of abusive content in social media. In: Angelov, P., Gegov, A., Jayne, C., Shen, Q. (eds.) Advances in Computational Intelligence Systems. AISC, vol. 513, pp. 187–205. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-46562-3\\_12](https://doi.org/10.1007/978-3-319-46562-3_12)
5. Cohen, J.: A coefficient of agreement for nominal scales. *Educ. Psychol. Measur.* **20**(1), 37–46 (1960)
6. Dutta, A., Biswas, S., Das, A.K.: EmoComicNet: a multi-task model for comic emotion recognition. *Pattern Recogn.* **150**, 110261 (2024)

7. Gasparini, F., Rizzi, G., Saibene, A., Fersini, E.: Benchmark dataset of memes with text transcriptions for automatic detection of multi-modal misogynistic content. *Data Brief* **44**, 108526 (2022)
8. Hossain, E., Hoque, M.M., Hossain, M.A.: An inter-modal attention framework for multimodal offense detection. In: Vasant, P., Weber, G.W., Marmolejo-Saucedo, J.A., Munapo, E., Thomas, J.J. (eds.) *ICO 2022. LNNS*, vol. 569, pp. 853–862. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-19958-5\\_81](https://doi.org/10.1007/978-3-031-19958-5_81)
9. Hossain, E., Sharif, O., Hoque, M.M.: Mute: a multimodal dataset for detecting hateful memes. In: *Proceedings of the 2nd conference of the AACL and IJCNLP: Student Research Workshop*, pp. 32–39 (2022)
10. Hossain, E., Sharif, O., Hoque, M.M., Preum, S.M.: Align before attend: aligning visual and textual features for multimodal hateful content detection. *arXiv preprint [arXiv:2402.09738](https://arxiv.org/abs/2402.09738)* (2024)
11. Hossain, E., Sharif, O., Hoque, M.M., Preum, S.M.: Deciphering hate: identifying hateful memes and their targets. *arXiv preprint [arXiv:2403.10829](https://arxiv.org/abs/2403.10829)* (2024)
12. Kumar, R., et al.: A multilingual, multimodal dataset of aggression and bias: the comma dataset. In: *Language Resources and Evaluation*, pp. 1–81 (2023)
13. Kumari, K., Singh, J.P., Dwivedi, Y.K., Rana, N.P.: Multi-modal aggression identification using convolutional neural network and binary particle swarm optimization. *Futur. Gener. Comput. Syst.* **118**, 187–197 (2021)
14. Mishra, P., Yannakoudakis, H., Shutova, E.: Tackling online abuse: a survey of automated abuse detection methods. *CoRR* **abs/1908.06024** (2019). <http://arxiv.org/abs/1908.06024>
15. Morency, L.P., Baltrušaitis, T.: Multimodal machine learning: integrating language, vision and speech. In: *Proceedings 55th ACL: Tutorial Abstracts, Vancouver, Canada*, pp. 3–5. ACL (2017)
16. Sadiq, S., Mehmood, A., Ullah, S., Ahmad, M., Choi, G.S., On, B.W.: Aggression detection through deep neural model on twitter. *Futur. Gener. Comput. Syst.* **114**, 120–129 (2021)
17. Safi Samghabadi, N., Patwa, P., Pykl, S., Mukherjee, P., Das, A., Solorio, T.: Aggression and misogyny detection using BERT: a multi-task approach. In: *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, Marseille, France*, pp. 126–131. ELRA, May 2020
18. Sharif, O., Hoque, M.: Identification and Classification of Textual Aggression in Social Media: Resource Creation and Evaluation, pp. 9–20 (2021)
19. Sharma, S., Akhtar, M.S., Nakov, P., Chakraborty, T.: DISARM: detecting the victims targeted by harmful memes. In: *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 1572–1588 (2022)
20. Suryawanshi, S., Chakravarthi, B.R., Arcan, M., Buitelaar, P.: Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text. In: *Proceedings of the Second Workshop on TRAC*, pp. 32–41. ELRA (2020)
21. Suryawanshi, S., Chakravarthi, B.R., Verma, P., Arcan, M., McCrae, J.P., Buitelaar, P.: A dataset for troll classification of Tamil Memes. In: *Proceedings of the WILDRE5– 5th Workshop on Indian Language Data: Resources and Evaluation, Marseille, France*, pp. 7–13. ELRA, May 2020
22. Mojica de la Vega, L.G., Ng, V.: Modeling trolling in social media conversations. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. ELRA (2018)
23. Yadav, A., Vishwakarma, D.K.: Sentiment analysis using deep learning architectures: a review. *Artif. Intell. Rev.* **53**(6), 4335–4385 (2020)

24. Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: Predicting the type and target of offensive posts in social media. In: Proceedings of the NAACL: Human Language Technologies, Minneapolis, Minnesota, pp. 1415–1420. ACL (2019)
25. Zhu, J., Lee, R.K.W., Chong, W.H.: Multimodal zero-shot hateful meme detection. In: Proceedings of the 14th ACM Web Science Conference 2022, pp. 382–389 (2022)



# $\mathbb{E}^4$ : A Voting-Based Paradigm for Enhancing Retrieval Augmented Generation

Wenbo Guan<sup>1,2,3</sup> , Xiaoqian Li<sup>4</sup>, Jiyu Lu<sup>1,2,3</sup>, and Jun Zhou<sup>3</sup> 

<sup>1</sup> Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100084, China

<sup>2</sup> School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

<sup>3</sup> Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China  
zhoujun@hccl.ioa.ac.cn

<sup>4</sup> School of Computing Science and Engineering, South China University of Technology, Guangzhou 510006, China

**Abstract.** Retrieval Augmented Generation (RAG) has become a common practice to alleviate the hallucination of Large Language Models (LLMs). The retrieval phase of RAG, however, usually solely depends on the original query, which, to some extent, suffers from the problem of *semantic gap* and thus degrades the quality of the retrieved external knowledge. To address this problem and enhance the performance of the traditional RAG, we propose a rEwrite-sElect-votE-rEad paradigm ( $\mathbb{E}^4$ ) that first paraphrases the original query into  $N$  rewritten ones to bridge *the semantic gap* from different perspectives and then determines the most valuable retrieved external knowledge via a voting manner. Besides, in the midst of the above procedures, a certain query-selecting strategy is also required to filter out the extra noise introduced by the query-rewriting process. Following this proposed paradigm, we provide our implementation of  $\mathbb{E}^4$ . Experimental results of our implementation on long context reading comprehension datasets from LongBench demonstrate the effectiveness of our proposed paradigm and provide a profound insight into the whole enhanced RAG process.

**Keywords:** Large language model · Retrieval augmented generation · Voting-based enhancing paradigm

## 1 Introduction

Large Language Models (LLMs), due to their excellent language understanding, logical reasoning and language generation abilities [25], have become the most popular research direction in natural language processing (NLP) domain nowadays. These abilities of LLMs are largely endowed by their large scale parameters [24] where all the *world knowledge* from model pre-training corpus is also stored

(*a.k.a.*, Parameterized knowledge). Since parameterized knowledge is acquired by LLMs via pre-training, which is a high resource and time demanding process, it is intractable to frequently update this kind of knowledge, therefore, leading to the hallucination of LLMs (*e.g.*, Generating outdated content) [8]. Besides, the essence of neural network makes the parameterized knowledge interwoven with each other in the LLMs' parameter space, further deteriorating the model hallucination phenomenon (*e.g.*, Generating fabricated content).

Recently, RAG [6] has become a mainstream solution to address the LLMs' hallucination problem. Different from previous methods that need model training, RAG leaves LLMs' internal parameters intact but additionally introduces knowledge from external source (*e.g.*, Database and Wikipedia). Knowledge of this kind is usually in the form of plain text and referred to as non-parameterized knowledge. The goal of RAG is to find ways that can strike a balance between parameterized and non-parameterized knowledge, making the generating process of LLMs reliable and explainable to the largest extent.

The traditional RAG consists of two steps:

- **Retrieval:** Using the original query to search relevant knowledge from external sources;<sup>1</sup>
- **Read:** Converting the knowledge obtained from the previous step and the original query into a specific prompt which is then fed into LLMs for final content generation.

The original query, however, is not always the optimal choice for knowledge retrieval in the first step of RAG since there exists the problem of *semantic gap* [6] between original query and external knowledge.

To address the aforementioned problem, we propose a voting-based paradigm to enhance the traditional RAG which includes four steps:

- **Rewrite:** Paraphrasing the original query into  $N$  rewritten ones;
- **Select:** Selecting the most valuable rewritten queries and filtering out queries that contain noise with a certain query-selecting strategy;
- **Vote:** Determining the most useful retrieved knowledge via a voting manner according to the selected rewritten queries;
- **Read:** Same as that of the traditional RAG.

To the best of our knowledge, we are the first to conduct a comprehensive research on enhancing RAG from the perspective of query rewriting via a voting manner. The contributions of this paper are as follows:

- A novel voting-based paradigm  $\mathbb{E}^4$  is proposed to enhance the traditional RAG from the perspective of bridging the *semantic gap*.  $\mathbb{E}^4$  is relatively flexible, containing four steps that can be optimized respectively, which provides promising research directions in future work;

---

<sup>1</sup> Methods from information retrieval domain such as sparse and dense retrieval methods are widely used in this step.



- Following this paradigm, we give our implementation of  $\mathbb{E}^4$ , which leverages LLMs to paraphrase the original query and contains various query-selecting strategies;
- Experimental results on long context reading comprehension datasets from LongBench demonstrate the effectiveness of our proposed paradigm and provide a profound insight into the whole enhanced RAG process.

## 2 Related Work

In this section, we introduce the related work from the perspectives of query rewriting, information retrieval and prompt engineering.

### 2.1 Query Rewriting

There are several previous studies that focus on query rewriting. The most typical one is RRR [14]. Not like our method, RRR tries to rewrite the original query via a small model, which is trained by means of reinforcement learning with feedback signals from LLMs. BEQUE [18] is another representative work. It aims at solving the problem of long-tail queries, a phenomenon that is very common in real-life scenarios. BEQUE has significantly improved the recall rate after query rewriting and has been successfully applied to the Taobao. HyDE [4] first adopts LLMs to transform the query into a hypothetical document and then uses this hypothetical document to search knowledge from the external source, making the retrieval process happen at the same semantic level (document *vs.* document).

### 2.2 Information Retrieval

Searching useful knowledge from the external source is the first and most important procedure of RAG and usually depends on information retrieval (IR) [15] methods. IR methods can be roughly divided into two categories, namely, sparse retrieval [3] methods and dense retrieval [26, 27] methods.

Sparse retrieval methods use shallow features, such as term frequency, to calculate lexical similarity between text. BM25 [19] algorithm is the most typical sparse retrieval method. Its inverted index structure makes itself a widely applied IR tool in industry and a strong baseline in academic researches.

Since sparse retrieval methods only focus on the lexical similarity of text, they are not capable of handling situations that entail complicated semantic similarity (*e.g.*, different words with the same meaning). Dense retrieval methods, therefore, are proposed to address this problem. Dense retrieval methods usually first convert the raw text into dense vectors by using a semantic encoder and then calculate the similarity scores between these vectors via some specific vector similarity metrics (*e.g.*, cosine similarity and L2 distance) during retrieval.

Transformer-based architecture [22] is often adopted nowadays to form the backbone of semantic encoders. Renowned pre-trained language model (LM)

BERT [2] is the most popular choice. Other pre-trained LMs, such as RoBERTa [12] and ERNIE [20], are also widely used. After a pre-trained LM is selected, it needs to be further trained to be adapted to the semantic feature extraction task, which is usually achieved by the contrastive learning methods. The goal of contrastive learning is to make the representation vectors with the same semantics close to each other in the latent space, while those with different semantics far away from each other. The representatives of dense retrieval encoders include DPR, Contriever and SimCSE, etc.

### 2.3 Prompt Engineering

Prompt engineering becomes an emerging research field [16, 23] after LLMs have unified most NLP tasks into a sequence-to-sequence paradigm. By leveraging LLMs’ strong instruction-following and in-context learning abilities, prompt engineering can easily adapt LLMs for completing tasks they have never seen before.

LLMs’ prompt usually consists of two parts, task and instruction. Task indicates what LLMs exactly need to do and instruction tells LLMs how to do it. In the realm of RAG, an additional part, namely, task context, is also required which aims at providing supplementary information to help LLMs reduce hallucination and generate more reliable content according to the task.

Our research belongs to the field of prompt engineering and focuses on *how to select the most helpful task context (external knowledge)*.

## 3 Methodology

In this section, we introduce our implementation details of  $\mathbb{E}^4$  in *rewrite*, *select*, *vote* and *read* four steps.

### 3.1 Rewrite

Query rewriting is the key step of bridging the *semantic gap* between original query and external knowledge, which requires the rewritten queries having the same semantics with the original one but asking from a different perspective. Since instruction-aligned LLMs have demonstrated strong instruction-following ability, we leverage LLMs themselves to conduct the query rewriting in our implementation via prompt engineering. The prompt we use is shown in Fig. 1.

In Fig. 1, `<ori.query>` is the placeholder of the original query and `<N>` is the number of the required rewritten queries. In this paper, we use  $Q$  to indicate the original query and  $\hat{Q}_n (n \in [1..N])$  to represent the rewritten queries.

### 3.2 Select

Although there exist constraints on the query-rewriting process which have already been included in the above query-rewriting prompt, LLMs, however, may

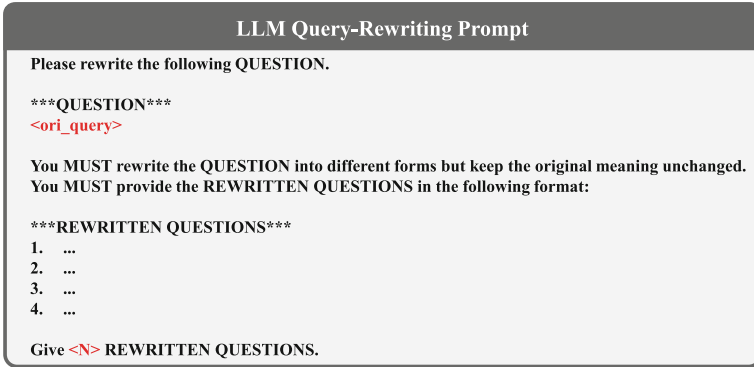


Fig. 1. LLM Query-Rewriting Prompt



Fig. 2. Four Query-Selecting Strategies

not always follow them, generating some queries that distort the meaning of the original one or are even totally irrelevant. These kind of queries will introduce extra noise into the following voting step and, therefore, severely degrade the final RAG performance. To solve this problem, we design three query-selecting strategies from different perspectives, namely, similarity-based strategy (**SBS**), diversity-based strategy (**DBS**) and LLM-based strategy (**LLMB**). These strategies are illustrated in Fig. 2.

**Similarity-Based Strategy.** Similarity-based strategy provides the most intuitive way to conduct query selecting. By calculating the similarity scores between the original query and the rewritten ones, keeping the rewritten queries with the highest  $K$  similarity scores, the most irrelevant rewritten queries are filtered out.

**Diversity-Based Strategy.** Depending solely on similarity may lead to the selected queries lacking diversity which means that these queries are not capable of bridging *the semantic gap* but only repeat the original query’s behavior. To overcome this drawback of the similarity-based strategy, diversity-based strategy

first splits the rewritten queries into  $K$  groups via a certain clustering method and then selects the query that is the most similar to the original query from every group. This strategy is designed to strike a balance between similarity and diversity.

**LLM-Based Strategy.** LLMs, after trained on a massive corpus, acquire an excellent natural language understanding ability, which can help select the most valuable rewritten queries. In this strategy, LLMs are adopted to calculate the conditional probability of the original query given the rewritten one. We believe that the higher the probability of the original query, the more valuable the rewritten one. Queries that with the *top* -  $K$  probabilities will be kept. This process is illustrated as follows:

$$P_{\hat{Q}_n} = P(Q | \hat{Q}_n, I) = \prod_{t=0}^T P(Q_t | Q_{t-1}, \dots, Q_0, \hat{Q}_{nj}, \dots, \hat{Q}_{n0}, I_m, \dots, I_0) \quad (1)$$

where  $Q$  represents the original query,  $\hat{Q}_n$  donates the  $n$ th rewritten query,  $I$  is the guiding instruction,  $Q_t$ ,  $\hat{Q}_{nj}$  and  $I_m$  are the corresponding tokens in  $Q$ ,  $\hat{Q}_n$  and  $I$ . In our research, we investigate the LLM-based strategy with and without the guiding instruction, respectively. The guiding instruction we select in our experiments is shown in Fig. 3. It is worth noting that the guiding instruction used here is not for guiding LLMs to generate content but only act as a prior knowledge when calculating the conditional probability.

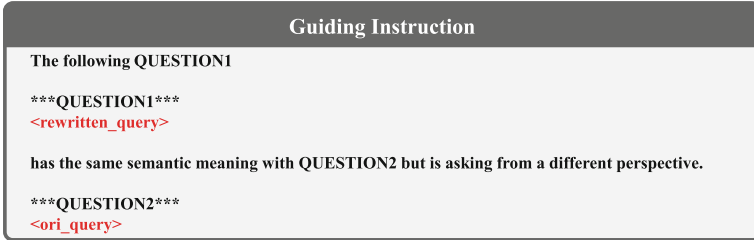


Fig. 3. Guiding Instruction

In Fig. 3, `<rewritten_query>` is the placeholder of the rewritten query and `<ori_query>` is that of the original one.

### 3.3 Vote

After filtering out the useless rewritten queries, the remaining ones are adopted to determine the most helpful retrieved knowledge by voting. In the realm of RAG, the external non-parameterized knowledge is usually organized into the

form of text chunks with certain length in order to be conveniently processed by IR methods. Thus, we regard text chunks as the basic voting units.

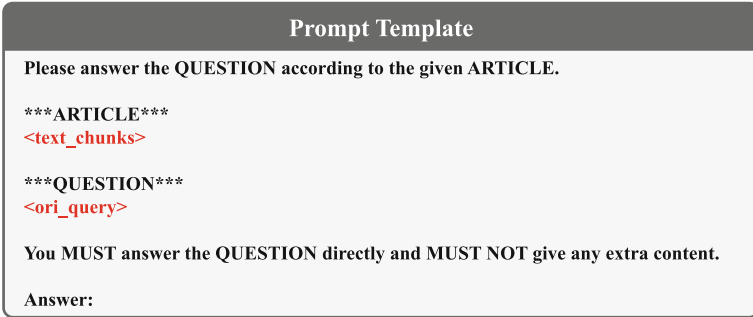
In this paper,  $C$  denotes the external non-parameterized knowledge, where  $C = \{C_0, C_1, \dots, C_L\}$ ,  $C_l$  represents the  $l$ th text chunk,  $L$  is the number of total text chunks.  $S_k$  is the  $k$ th selected rewritten query’s voting score vector on each text chunk candidate  $C_l$  obtained via a certain IR method, where  $S_k = \{S_{k0}, S_{k1}, \dots, S_{kL}\}$ . The voting process is illustrated by Eq. 2.

$$\hat{S} = \sum_{k=1}^K d_k \cdot \text{softmax}(S_k) \quad (2)$$

where  $\hat{S}$  is the final voting results on all text chunk candidates by all selected rewritten queries.  $\text{softmax}(\cdot)$  is the *softmax function*.  $d_k$  is the weighting parameter of each selected rewritten query  $Q_k$  and is calculated via Eq. 3, where  $\hat{d}_k$  is the similarity score between the original query and the  $k$ th selected rewritten query given by a certain IR method.

$$\{d_1, d_2, \dots, d_k\} = \text{softmax}(\hat{d}_1, \hat{d}_2, \dots, \hat{d}_k) \quad (3)$$

### 3.4 Read



**Fig. 4.** Prompt Template

The voting step yields a score vector for all text chunk candidates in  $C$  and the last reading step needs to select the text chunks with the highest  $B$  scores and combines them with the original query according to the template shown in Fig. 4 to form the RAG prompt for LLMs to generate the final answer, where  $\text{<text\_chunks>}$  is the placeholder of the selected  $B$  text chunks and  $\text{<ori\_query>}$  is that of the original query.

In our experiments, we set  $B$  via Eq. 4.

$$B = \lceil \log_p L \rceil \quad (4)$$

where  $\rho$  is called compression factor. It is important to set the correct value of compression factor and details will be discussed in Sect. 5.2.  $\lceil \cdot \rceil$  indicates the *rounding up* operation. Our implementation of  $\mathbb{E}^4$  is summarized in Algorithm 1.

---

**Algorithm 1.** Our implementation of  $\mathbb{E}^4$

---

**Input:** Original query  $Q$ , external non-parameterized knowledge  $C$ , query-selecting strategy pool  $G=\{\text{vanilla-vote}; \text{SBS}; \text{DBS}; \text{LLMBS}; \text{LLMBS-guide}\}$ , pre-defined number of required rewritten queries  $N$ , number of selected rewritten queries  $K$  and compression factor  $\rho$

**Step 1:** Leveraging LLMs to paraphrase the original query  $Q$  into  $N$  rewritten queries  $\hat{Q}_n$

**Step 2:** Choosing one query-selecting strategy  $g$  from strategy pool  $G$

**Step 3:** Using the current strategy  $g$  to select  $K$  queries from  $\hat{Q}_n$

**Step 4:** Using the selected  $K$  rewritten queries and the original query  $Q$  to vote on external non-parameterized knowledge  $C$  via Eq. 2

**Step 5:** Combining text chunks with the highest  $B$  voting scores and the original query  $Q$  to form the final RAG prompt, where  $B$  is calculated via Eq. 4

**Step 6:** Inputting the prompt into LLMs and let LLMs generate the final answer

**Output:** Final RAG answer

---

## 4 Experimental Setup

In this section, we present the details of our experimental setup.

### 4.1 LLM

In our experiments, we choose Llama-2-7B-Chat [21] as the target model. Llama-2-7B-Chat is an instruction-aligned large language model designed and trained by Meta AI. Its training process has three phases including Pre-Training (PT), Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF). Therefore, it is endowed with excellent instruction-following ability, making itself capable of conducting tasks such as query rewriting and reading comprehension.

### 4.2 Task Description and Datasets

We adopt the long context reading comprehension task (LCRC) in our experiments to testify the effectiveness of our proposed  $\mathbb{E}^4$  paradigm and its corresponding implementation.

LCRC task consists of three parts, namely, a long passage, a question based on this passage and an answer list. Since the passages in LCRC datasets are quite long (usually exceed the maximum input lengths of LLMs, *e.g.*, 4096 tokens of

Llama-2-7B-Chat), LCRC becomes a suitable downstream task to measure a RAG framework’s performance.<sup>2</sup>

We select three LCRC datasets, 2WikiMQA, HotpotQA and MultiFieldQA-en, to conduct our experiments. The statistics of these three datasets are listed in Table 1.

**Table 1.** Statistics of datasets

Dataset	Sample Num.	Avg. Passage Length (in tokens)
2WikiMQA	200	8400.03
HotpotQA	300	11210.32
MultiFieldQA-en	150	8056.62

### 4.3 Baselines

Six widely applied IR baselines are adopted in our experiments, including BM25 [19], DPR [10], Contriever [9], SimCSE [5], TAS-B [7] and Nomic [17]. Among them, BM25 is a lexical-similarity-based algorithm that has a profound impact on industry and others are popular semantic-similarity-based methods.

### 4.4 Evaluation Metric

All results in our experiments are reported in the standard reading comprehension evaluation metric *F1-score* which calculates the overlap between characters of the generated answer and the golden answer. *F1-score* integrates the recall and the precision rate together. Equation 5 shows the computing process of it.

$$F1\text{-score} = \frac{2 * \textit{precision} * \textit{recall}}{\textit{precision} + \textit{recall}} \quad (5)$$

### 4.5 Other Experimental Setup

LLMs have been found relatively sensitive to the granularity and order of the external non-parameterized knowledge [1, 13]. In order to fully take these factors into consideration, we set the text chunk’s size to 256 and 512, respectively, and use four sorting strategies to organize the selected text chunks, including a) sorting the text chunks by voting scores in ascending order; b) sorting the text chunks by voting scores in descending order; c) sorting the text chunks according to their positions in original passage; d) interleaving the text chunks at the front and back ends of the sequence alternately by voting scores in ascending order. [11] The combination of two text chunk granularities and four orders finally yields eight experimental settings in total.

<sup>2</sup> The long passage of a data sample in LCRC task can be regarded as the external non-parameterized knowledge and the question as the original query.

To eliminate the impact of randomness on experimental results, greedy decoding strategy is adopted when generating the final answers.

In our main experiment, we set the number of required rewritten queries and that of the selected rewritten queries to 15 and 3, respectively. The clustering method used in **DBS** is *K-Means*. The compression factor  $\rho$  is 2.

## 5 Results and Analyses

In this section, we first show the results of our main experiment and then several case studies as well as an ablation experiment are conducted to provide a profound insight into the whole enhanced RAG process.

### 5.1 Main Experiment

Results of the main experiment are listed in Table 2.<sup>3</sup> We can observe from Table 2 that our proposed voting-based paradigm is able to enhance the traditional RAG with almost all the retrieval baselines.

**Table 2.** Results of the Main Experiment

Strategy ↓ \ Method →		BM25	DPR	Contriever	SimCSE	TAS-B	Nomic
2WikiMQA	Baseline	24.681	27.600	27.094	27.280	29.800	28.908
	+ vanilla-vote	24.423	27.704 ↑	28.725	↑ 27.253	29.695	28.338
	+ SBS	<b>25.096</b>	28.094 ↑	27.845	↑ 27.609	↑ 29.786	28.986 ↑
	+ DBS	-	28.101 ↑	<b>28.926</b>	28.466	↑ 30.159	↑ 28.840
	+ LLMBS	24.109	<b>28.104</b>	26.962	<b>28.758</b>	<b>31.097</b>	29.030 ↑
	+ LLMBS-guide	25.074 ↑	28.058 ↑	27.620	↑ 28.440	↑ 30.782	↑ <b>29.759</b>
HotpotQA	Baseline	<b>33.738</b>	30.937	31.724	33.233	32.262	33.530
	+ vanilla-vote	33.486	31.332 ↑	31.534	32.527	33.014	↑ 34.226 ↑
	+ SBS	32.639	31.426 ↑	<b>32.015</b>	34.534	↑ 32.538	↑ 34.025 ↑
	+ DBS	-	<b>31.649</b>	31.057	33.555	↑ 32.283	↑ 33.712 ↑
	+ LLMBS	33.162	31.094 ↑	31.041	<b>34.994</b>	<b>33.094</b>	<b>35.777</b>
	+ LLMBS-guide	33.546	31.577 ↑	31.904	↑ 34.390	↑ 32.827	↑ 34.809 ↑
MultiField-en	Baseline	42.642	38.644	42.106	41.162	43.120	41.878
	+ vanilla-vote	43.390	↑ 38.737 ↑	42.152	↑ 40.650	42.793	41.284
	+ SBS	42.032	38.629	42.058	41.043	42.802	42.052 ↑
	+ DBS	-	38.794 ↑	42.408	↑ 41.359	↑ 43.271	↑ 42.583 ↑
	+ LLMBS	<b>43.523</b>	<b>39.027</b>	42.447	↑ <b>41.562</b>	<b>43.302</b>	41.681
	+ LLMBS-guide	42.867 ↑	38.766 ↑	<b>42.769</b>	41.202	↑ 43.260	↑ <b>42.755</b>

<sup>3</sup> Here, the best results among eight experimental settings are reported. In each column, the best result is in **bold** and results better than baselines are marked with ↑.

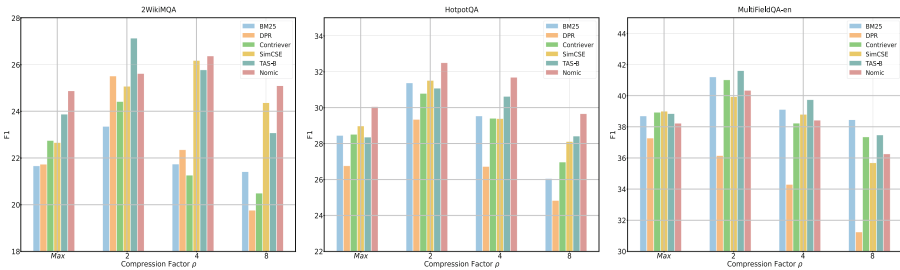


Among all the voting-based methods, the method with **vanilla-vote** is the most unstable one and degrades the performance of the traditional RAG in most cases, since it adopts all the rewritten queries during the voting step, introducing a lot of extra noise. These results demonstrate the necessity of conducting query selecting after query rewriting to dispose of the distracting ones.

Although **SBS** is able to filter out the rewritten queries that contain noise to some extent, the similarity-oriented essence of it limits the final performance of the corresponding method with **SBS**, only having slight improvement compared to the **vanilla-vote** one in most cases.

Methods with **DBS** and **LLMBS**, however, have the most consistent performance in enhancing the traditional RAG, validating our hypotheses above. We argue that **LLMBS-guide**, which naturally integrates with the great potential of LLMs, is a promising research direction in future studies of fully exploiting LLMs’ capacity.

## 5.2 Case Study 1: Compression Factor $\rho$ ’s Role in the RAG Process



**Fig. 5.** Compression factor  $\rho$ ’s impact on the final RAG performance

Compression factor  $\rho$ , according to Eq. 4, determines to what extent the external knowledge (*e.g.*, long passages in the LCRC task) is compressed. External knowledge compression is a necessary procedure in RAG since LLMs usually have a maximum input sequence length due to the hardware and computing resource limitations. Thus, it is impractical to input the whole passage into LLMs during inference. Besides, due to the fact that the longer the input sequence, the more noise it contains, external knowledge compression, via selecting the text chunks with high voting scores only, can act as an additional denoising step to help further improve the final RAG performance. As shown in Fig. 5, we set the compression factor  $\rho$  to *Max*, 2, 4 and 8, respectively, where *Max* represents selecting the text chunks until their total length reaches the length limitation of LLMs (*e.g.*, 4096 tokens of Llama-2-7B-Chat in our experiment). It can be observed from Fig. 5 that setting  $\rho$  to 2 obtains the best performance, further striking a balance between filtering out extra noise and keeping useful information.

### 5.3 Case Study 2: Number of Selected Rewritten Queries $K$ 's Impact on the Final RAG Performance

In our main experiment, we set the number of selected rewritten queries  $K$  to 3. In this section, we are going to study the Number of selected rewritten queries  $K$ 's influence on the final RAG performance. We conduct an additional experiment on 2WikiMQA dataset and set  $K$  to 3, 6, 9 and 12, respectively. Results are in Table 3.<sup>4</sup> Methods with smaller  $K$  values (*e.g.*, 3 and 6) are more consistent in improving the performance of the traditional RAG and are better than those with larger  $K$  values in most situations especially those with **DBS** and **LLMBS**. We argue that the rewritten queries generated by LLMs have redundant information to some extent and, therefore, it is not necessary to take a large proportion of the rewritten queries into account (*e.g.*, 9/15 or 12/15) when voting but the most useful ones only. These results also support our conclusion above from a different perspective that diversity is a more important criterion than similarity when conducting query selecting.

**Table 3.**  $K$ 's Impact on the Final RAG Performance

Method ↓ \ $K$ →		no-vote	3	6	9	12
<b>SBS</b>	BM25	24.681	<b>25.096</b>	24.550	24.721	↑24.636
	DPR	27.600	<b>28.094</b>	27.704	↑27.704	↑27.704
	Contriever	27.094	27.845	↑28.131	↑ <b>28.252</b>	28.116
	SimCSE	27.280	27.609	↑ <b>28.447</b>	28.368	↑27.401
	TAS-B	29.800	29.786	<b>30.490</b>	30.099	↑29.766
	Nomic	28.908	<b>28.986</b>	27.723	28.727	28.132
<b>DBS</b>	BM25	-	-	-	-	-
	DPR	27.600	28.101	↑ <b>28.104</b>	27.649	↑27.704
	Contriever	27.094	28.926	↑ <b>28.948</b>	28.202	↑28.481
	SimCSE	27.280	<b>28.466</b>	26.934	26.671	27.452
	TAS-B	29.800	30.159	↑ <b>30.320</b>	30.311	↑29.989
	Nomic	<b>28.908</b>	28.840	27.716	27.037	27.761
<b>LLMBS</b>	BM25	<b>24.681</b>	24.109	24.401	24.203	24.498
	DPR	27.600	28.104	↑ <b>28.115</b>	27.894	↑27.704
	Contriever	27.094	26.962	27.533	↑ <b>28.415</b>	27.935
	SimCSE	27.280	<b>28.758</b>	26.767	26.939	27.124
	TAS-B	29.800	31.097	↑31.038	↑ <b>31.153</b>	30.151
	Nomic	28.908	<b>29.030</b>	27.865	28.582	27.766
<b>LLMBS-guide</b>	BM25	24.681	<b>25.074</b>	24.146	24.625	25.019
	DPR	27.600	28.058	↑ <b>28.115</b>	28.104	↑27.849
	Contriever	27.094	27.620	↑27.444	↑ <b>28.328</b>	28.222
	SimCSE	27.280	<b>28.440</b>	27.723	↑26.868	25.929
	TAS-B	29.800	30.782	↑ <b>31.640</b>	30.948	↑30.584
	Nomic	28.908	<b>29.759</b>	28.350	27.930	27.559

<sup>4</sup> In each row, the best result is **in bold** and results better than baseline are marked with ↑.

From these analyses, it is reasonable to choose a small  $K$  value in the query-selecting step, which can not only yield a better performance but reduce the computing complexity in the proceeding voting step as well.

## 5.4 Ablation Experiment

**Table 4.**  $d_k$ 's Impact on the Final RAG Performance

Method ↓ \ Strategy →	vanilla-vote	SBS	DBS	LLMBS	LLMBS-guide	
BM25	w/ $d_k$	23.271	<b>23.724</b>	-	22.529	<b>23.137</b>
	w/o $d_k$	<b>23.300</b>	23.169	-	<b>22.992</b>	22.239
DPR	w/ $d_k$	<b>25.442</b>	<b>25.539</b>	<b>25.583</b>	<b>25.367</b>	<b>25.510</b>
	w/o $d_k$	25.276	24.742	24.449	24.585	24.346
Contriever	w/ $d_k$	<b>24.698</b>	<b>25.232</b>	<b>24.911</b>	<b>24.721</b>	<b>24.676</b>
	w/o $d_k$	24.629	25.023	24.725	24.622	24.468
SimCSE	w/ $d_k$	<b>26.087</b>	26.030	<b>25.956</b>	<b>25.854</b>	<b>25.906</b>
	w/o $d_k$	25.555	<b>26.142</b>	25.745	25.667	25.853
TAS-B	w/ $d_k$	26.928	27.002	27.327	<b>27.181</b>	<b>27.585</b>
	w/o $d_k$	<b>27.136</b>	<b>27.293</b>	<b>27.719</b>	27.052	26.930
Nomic	w/ $d_k$	25.913	<b>27.002</b>	<b>26.386</b>	26.586	26.645
	w/o $d_k$	<b>26.153</b>	26.819	26.052	<b>26.605</b>	<b>26.665</b>

In this part, we are going to investigate the impact of the weighting parameter  $d_k$  on the final RAG performance via an ablation study on 2WikiMQA dataset. Results are listed in Table 4.<sup>5</sup> From Table 4 we can observe that adding weighting parameter  $d_k$  to the voting step helps improve the performance in most cases, especially that of methods combined with various query-selecting strategies. We believe that weighting parameter  $d_k$ , according to the original query, provides a more fine-grained signal to further adjust different rewritten queries' degree of importance in the voting process, which is crucial to the final voting results.

## 6 Conclusion

In this paper, we propose  $\mathbb{E}^4$ , a voting-based paradigm for enhancing RAG, which has *rewrite*, *select*, *vote* and *read* four parts. Besides, an implementation of  $\mathbb{E}^4$  is elaborated in this paper, including a LLM-based query-rewriting component and various query-selecting strategies. Experimental results on three long context reading comprehension datasets demonstrate our proposed paradigm's

<sup>5</sup> Mean values of results under eight experimental settings are reported in Table 4.

effectiveness and give us a profound insight into the whole enhanced RAG process.

Since  $\mathbb{E}^4$  contains four independent parts which can be implemented via many other different ways, our implementation, therefore, may not be the optimal one. We believe implementing  $\mathbb{E}^4$  with four parts that can benefit from each other and mutually obtain a better performance a promising research direction in future studies.

**Acknowledgments.** This work is supported by the Youth Innovation Promotion Association of the Chinese Academy of Sciences (E1291902), Jun Zhou (2021025).

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this paper.

## References

1. Chen, T., et al.: Dense x retrieval: what retrieval granularity should we use? ArXiv abs/2312.06648 (2023)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: 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 (2019)
3. Formal, T., Lassance, C., Piwowarski, B., Clinchant, S.: SPLADE v2: sparse lexical and expansion model for information retrieval. ArXiv abs/2109.10086 (2021)
4. Gao, L., Ma, X., Lin, J., Callan, J.: Precise zero-shot dense retrieval without relevance labels. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1762–1777 (2023)
5. Gao, T., Yao, X., Chen, D.: SimCSE: simple contrastive learning of sentence embeddings. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6894–6910 (2021)
6. Gao, Y., et al.: Retrieval-augmented generation for large language models: a survey. ArXiv abs/2312.10997 (2023)
7. Hofstätter, S., Lin, S.C., Yang, J.H., Lin, J., Hanbury, A.: Efficiently teaching an effective dense retriever with balanced topic aware sampling. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 113–122 (2021)
8. Huang, L., et al.: A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions. ArXiv abs/2311.05232 (2023)
9. Izacard, G., et al.: Unsupervised dense information retrieval with contrastive learning. ArXiv abs/2112.09118 (2021)
10. Karpukhin, V., et al.: Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6769–6781 (2020)
11. Liu, N.F., et al.: Lost in the middle: how language models use long contexts. *Trans. Assoc. Comput. Linguist.* **12**, 157–173 (2024)
12. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. ArXiv abs/1907.11692 (2019)

13. Lu, Y., Bartolo, M., Moore, A., Riedel, S., Stenetorp, P.: Fantastically ordered prompts and where to find them: overcoming few-shot prompt order sensitivity. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 8086–8098 (2022)
14. Ma, X., Gong, Y., He, P., Zhao, H., Duan, N.: Query rewriting in retrieval-augmented large language models. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5303–5315 (2023)
15. Mitra, M., Chaudhuri, B.: Information retrieval from documents: a survey. *Inf. Retrieval* **2**, 141–163 (2000)
16. Muresanu, A., Thudi, A., Zhang, M.R., Papernot, N.: Unlearnable algorithms for in-context learning. *ArXiv abs/2402.00751* (2024)
17. Nussbaum, Z., Morris, J.X., Duderstadt, B., Mulyar, A.: Nomic embed: training a reproducible long context text embedder. *ArXiv abs/2402.01613* (2024)
18. Peng, W., et al.: Large language model based long-tail query rewriting in taobao search. In: Companion Proceedings of the ACM on Web Conference 2024, pp. 20–28 (2024)
19. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. *Inf. Retrieval* **3**(4), 333–389 (2009)
20. Sun, Y., et al.: ERNIE: enhanced representation through knowledge integration. *ArXiv abs/1904.09223* (2019)
21. Touvron, H., et al.: LLaMA 2: open foundation and fine-tuned chat models. *ArXiv abs/2307.09288* (2023)
22. Vaswani, A., et al.: Attention is all you need. *Advances in Neural Information Processing Systems*, vol. 30 (2017)
23. Wang, Y., et al.: Self-instruct: Aligning language models with self-generated instructions. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 13484–13508 (2023)
24. Wei, J., et al.: Emergent abilities of large language models. *ArXiv abs/2206.07682* (2022)
25. Wu, T., et al.: A brief overview of ChatGPT: the history, status quo and potential future development. *IEEE/CAA J. Autom. Sinica* **10**(5), 1122–1136 (2023)
26. Zhao, W.X., Liu, J., Ren, R., Wen, J.R.: Dense text retrieval based on pretrained language models: A survey. *ACM Trans. Inf. Syst.* **42**(4), 1–60 (2024)
27. Zhu, Y., et al.: Large language models for information retrieval: a survey. *ArXiv abs/2308.07107* (2023)



# Improving Chinese Emotion Classification Based on Bilingual Feature Fusion

Haocheng Lan, Jie Ou, Zhaokun Wang, and Wenhong Tian<sup>(✉)</sup>

School of Information and Software Engineering, University of Electronic Science and  
Technology of China, Chengdu 610054, Sichuan, China  
haochenglan@alu.uestc.edu.cn tian\_wenhong@uestc.edu.cn

**Abstract.** The growing popularity of Chinese social media platforms such as Sina Weibo has created a large number of user generated text content, which is of great value for understanding public emotions. However, the existence of mixed languages in these texts, especially Chinese and English, and mixed expressions pose a major challenge to current emotion classification methods. To address these issues, we propose a Bilingual Feature Fusion Network (BFFN) that leverages the multilingual capabilities of pre-trained language models to enhance the semantic feature extraction of Chinese text. Additionally, we introduce a Bilingual Cross Attention Mechanism (BCAM) that utilizes emotional features as the primary factor to capture cross-lingual emotional information effectively. Furthermore, we employ a lightweight fine-tuning approach that combines Low-Rank Adaptation (LoRA) and Embedding Fine-tuning (LEF) to reduce the complexity of fine-tuning model weights for downstream tasks. Extensive experiments on various datasets demonstrate the superiority of our proposed method, outperforming state-of-the-art models like ERNIE by 1.43% in accuracy. Our work contributes to the advancement of emotion classification in the context of mixed-language communication culture and provides a practical solution for real-world applications. Our code has been published on the open source community Github (<https://github.com/oujieww/BFFN>).

**Keywords:** Emotion Classification · Bilingual Feature Fusion · Bilingual Cross Attention Mechanism

## 1 Introduction

Emotion classification has become a crucial research area in Natural Language Processing (NLP) due to the increasing popularity of social networking sites. Platforms like Sina Weibo have become repositories of vast amounts of textual data, including user comments and opinions [12, 13]. This wealth of social text

---

Haocheng Lan and Jie Ou contribute equally to this work, and Haocheng Lan completed this work when he was at UESTC.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025  
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15331, pp. 139–153, 2025.  
[https://doi.org/10.1007/978-3-031-78119-3\\_10](https://doi.org/10.1007/978-3-031-78119-3_10)

information holds immense commercial value for businesses and public institutions to understand public emotion [27]. Especially during the COVID-19 outbreak, Sina Weibo played a significant role in keeping Chinese netizens informed [29]. However, many current studies have ignored the changes in current language communication culture, especially the mixing of Chinese and English, and hybrid expressions that are increasingly appearing in our lives. And these mixed words contain strong emotional information as shown in Table 1, the words **shit** and **low** are strong emotion key words.

**Table 1.** These two samples come from the EB dataset. The dataset is in Chinese (ZH), and we use translation software to show the corresponding English(EN) version for readers to understand.

categories	text
angry	ZH: <b>Shit shit shit.</b>
	EN: Shit shit shit, this is impossible. What kind of meeting is this? This is bullshit.
sad	ZH: <b>low.</b>
	EN: If your hair does not follow your heart, your emotion will be very low for the whole day.

Emotion classification has evolved from traditional machine learning techniques like SVM [15], NBC [9], and CRT [1] to deep learning approaches. The introduction of the Transformer network [24] has greatly advanced the field of NLP, with models like BERT [5] and its variants (RoBERTa [14], XLNet [28], Nezha [26], Electra [2], and Ernie [22]) becoming key in emotion classification. However, these methods increase model parameters and training time, and large-scale Chinese datasets are scarce. Recent studies have proposed various improvements, such as combining BERT with BLS [17], employing dynamic encoding and multi-granularity feature fusion (DCCMM) [27], utilizing PCA for feature extraction and fusion (PCA-BERT). Although these technologies have made some progress, they ignore the fact that as time goes by, languages also have mixed expressions. Moreover, these methods directly use fine-tuning technology to align the corresponding models to downstream tasks, without deeply exploring the properties of the model itself, and without fully utilizing the value of pre-training itself, especially the multi-language analysis capabilities. For instance, BERT-based-Chinese [5] is pre-trained in Chinese on BERT-based-uncased [5], while RoBERTA-wwm-ext [4] is an improvement on BERT-based-Chinese, indicating that RoBERTA-wwm-ext possesses English knowledge.

Multimodal emotion classification [8, 11, 18, 21] integrates data from various channels, including text, audio, visual, and physiological signals, to achieve a more comprehensive emotion assessment. However, in the text channel, current approaches still do not account for the growing prevalence of language mixing, such as the use of multiple languages within a single utterance or document, and the multilingual capabilities of the models used are not fully utilized, limiting their effectiveness in analyzing emotion in real-world contexts where language mixing is increasingly common.

To address the aforementioned issues, we propose the **Bilingual Feature Fusion Network (BFFN)** based on a single backbone model, which leverages the English processing capability of the pre-trained language model to enhance the semantic feature extraction of Chinese text, thus improving emotional classification without introducing additional base models. Furthermore, to fully utilize cross-lingual features and thoroughly explore the role of strong emotional information features, we introduce a **Bilingual Cross Attention Mechanism (BCAM)** that employs global salient emotional features as clues to extract and aggregate cross-language word-level emotional features and enhance the model’s performance. Subsequently, we apply Low-Rank Adaptation (LoRA) technology for fine-tuning and combined with Embedding fine-tuning, utilizing Embedding adjustments to further align semantics and reduce the complexity of fine-tuning model weights for downstream tasks. The main contributions of this paper are as follows:

1. Proposed a bilingual feature fusion network (BFFN), which utilizes the English processing ability of the pre-trained language model to improve emotional classification.
2. Proposed a bilingual cross-attention mechanism (BCAM), which uses global salient emotional features as clues to extract and aggregate cross-language word-level emotional features, so as to strengthen the overall emotional features.
3. Introduced a lightweight fine-tuning method that combines LoRA and Embedding Fine-tuning (LEF), reduce the difficulty of fine-tuning model weights for downstream tasks.
4. We conduct extensive experiments on various models and datasets to validate the effectiveness of our proposed method. The results show that our approach outperforms the current state-of-the-art model, ERNIE, by achieving a 1.43% improvement in accuracy.

## 2 Related Work

### 2.1 Transformer in Natural Language Processing

Vaswani et al. proposed Transformers [24], which utilize self-attention mechanisms as the model’s foundation, consisting of an encoder for input sequence encoding and a decoder for target sequence generation. BERT [5], based on the Transformer architecture, employs only the encoder and introduces MLM and NSP tasks to enhance language understanding. RoBERTa-wwm-ext [4] introduces whole word masking based on BERT and is pre-trained on a larger Chinese data set. NEZHA [26] incorporates Functional Relative Positional Encoding, Whole Word Masking, and optimized training techniques. XLNet [28] introduces PLM and a two-stage attention mechanism, with a Chinese version available [3]. ELECTRA [2] refines BERT with RTD and advanced training techniques, and its Chinese version [3] is also offered. ERNIE [22], developed by Baidu, builds on BERT with entity-level enhancements, dynamic masking, and is pre-trained on a larger Chinese corpus.



## 2.2 Emotion Classification

Emotion classification research has shifted focus to deep learning techniques for text classification. Tang et al. were among the first to employ CNNs or LSTMs for sentence encoding [23], while Wang et al. proposed a context-aware bidirectional LSTM model [25]. Kim pioneered the use of CNNs for sentence-level text classification, and Johnson et al. introduced a word-level DPCNN to capture long-distance dependencies [10]. However, RNN-based models suffer from sequence dependency issues and lack parallel computing capabilities, while CNN-based models struggle to capture long-distance features and lose positional information due to pooling layers.

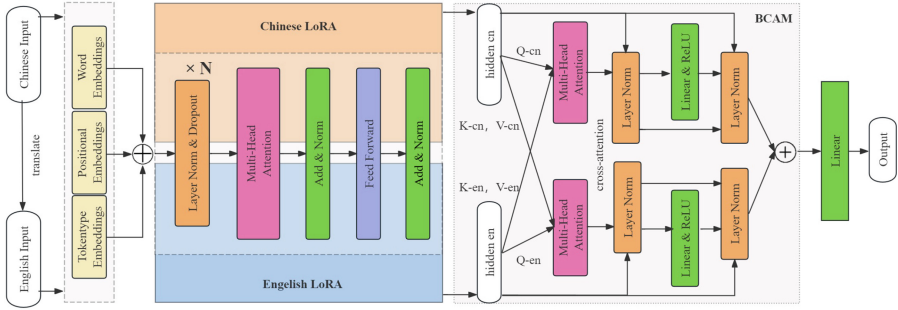
To improve emotion analysis, Peng et al. combined BERT with the Broad Learning System (BLS) [17], which combines BERT and extensive learning system (BLS) for emotion classification, Guo et al. used question-and-answer pairs as inputs for BERT [6], and Yan et al. proposed the DCCMM model [27] using WOBERT Plus and ALBERT for dynamic encoding and multi-granularity feature fusion. Su et al. introduced PCA-BERT [20], which utilizes principal component analysis to extract and fuse effective features from BERT layers. Although these technologies have progressed, they overlook the fact that languages evolve over time, leading to mixed expressions. Moreover, these methods directly apply fine-tuning to align models to downstream tasks without deeply exploring the model’s intrinsic properties and fully harnessing the value of pre-training, particularly multi-language analysis capabilities.

## 3 Method

In this section, we introduce the **Bilingual Feature Fusion Network (BFFN)**, as show in Fig. 1. First, the Chinese text is automatically translated into English. Both the original Chinese text and the translated English text are then fed into two separate LoRA branches, which operate in conjunction with the shared BERT backbone model. The Word Embeddings are aligned and fine-tuned during this process. The hidden representations obtained in the two languages are subsequently passed through the proposed **Bilingual Cross Attention Mechanism (BCAM)** for feature enhancement and fusion, resulting in a final feature vector that is used for emotion classification.

### 3.1 Bilingual Feature Fusion

As world cultures continue to evolve and interact, an increasing number of people are employing mixed language expressions on social media platforms. These cross-linguistic words and phrases often carry relatively strong emotional connotations, making them particularly relevant for emotion classification tasks. Table 1 presents examples that illustrate this phenomenon, where the words “shit” and “low” exhibit clear emotion tendencies and play a crucial role in capturing the overall emotion of the entire sentence. If we rely solely on a pre-trained model based on Chinese language for fine-tuning on the dataset, there



**Fig. 1.** The pipeline of our proposed BFFN.

is a significant risk that these important emotion-bearing elements will be overlooked or misinterpreted.

As illustrated in Fig. 1, we employ the BERT model as a single contributing backbone network and extend it with two LoRA branches for extracting word-level and sentence-level features in Chinese and English, respectively. Subsequently, we fuse the features obtained from these two distinct language sources to generate the final classification features. This approach allows the relevant emotional features extracted from the English branch to enhance the Chinese emotional features, enabling the model to fully capture and utilize the emotional information conveyed by the entire text.

### 3.2 Bilingual Cross Attention Mechanism

The cross-attention mechanism uses the global and local features of sentences and establishes connections between different languages to improve the effect of emotion classification. Directly fusing bilingual feature vectors may not yield the best results. The salient emotional characteristics at the vocabulary level and the general emotion conveyed by the meaning at the sentence level are crucial for accurate classification. A naive fusion approach may inadvertently prioritize unilateral information, potentially introducing side effects that affect the quality of the fused information. For instance, in the second example presented in Table 1, the emotional information expressed in the Chinese portion is relatively weak, lacking the presence of strong emotional words like “as seen in the Chinese part of the first example. Consequently, the emotional features of the final feature vector derived from the Chinese text may be comparatively weak. In contrast, the English portion of the second example contains the strong emotional word “low”, which, in conjunction with the relevant English text “emotion will be very low”, is likely to generate a robust emotional feature in the feature vector corresponding to the English part.

To enhance the effective fusion of bilingual features, we propose a bilingual cross-attention mechanism that takes advantage of the features of each language to assign scores to emotion features at the lexical level. Subsequently, based

on these scores, we acquire the feature vectors corresponding to each language component and then fuse them to obtain the final representation.

The hidden states of the outputs in Chinese and English are transformed into queries ( $Q$ ), keys ( $K$ ), and values ( $V$ ) using linear projections ( $W$ ), as shown in Eqs. (1), (2), and (3). Subsequently, the dot products between the queries ( $Q$ ) and keys ( $K$ ) from different languages are calculated, scaled, and normalized using the softmax function to obtain attention weights. This step enables the model to assess the relevance between the two languages, facilitating effective fusion of feature sequences. Through these attention weights, the values ( $V$ ) of each language are aggregated, thereby enabling cross-linguistic information propagation. The calculations are presented in Eqs. (4) and (5), where  $\Delta H_{\text{cn}}$  and  $\Delta H_{\text{en}}$  represent the information propagated from English to Chinese and from Chinese to English, respectively.

$$Q_{\text{cn}}, Q_{\text{en}} = W_{\text{cn}}^Q H_{\text{cn}}, W_{\text{en}}^Q H_{\text{en}} \quad (1)$$

$$K_{\text{cn}}, K_{\text{en}} = W_{\text{cn}}^K H_{\text{cn}}, W_{\text{en}}^K H_{\text{en}} \quad (2)$$

$$V_{\text{cn}}, V_{\text{en}} = W_{\text{cn}}^V H_{\text{cn}}, W_{\text{en}}^V H_{\text{en}} \quad (3)$$

$$\Delta H_{\text{cn}} = \text{softmax} \left( \frac{Q_{\text{cn}} K_{\text{en}}^T}{\sqrt{d}} \right) V_{\text{en}} \quad (4)$$

$$\Delta H_{\text{en}} = \text{softmax} \left( \frac{Q_{\text{en}} K_{\text{cn}}^T}{\sqrt{d}} \right) V_{\text{cn}} \quad (5)$$

Equations (1), (2), (3), (4), and (5) describe the attention mechanism process using a single head. In this paper, we use 12-heads attention mechanism enhances the representational capacity of this process, with the results from each head being subsequently merged to obtain richer and more nuanced inter-language interaction information. Finally, the information propagated from the other language is then utilized to update the features of the current language, as calculated in Eqs. (6) and (7).

$$\Delta H_{\text{cn-cross}} = \text{LayerNorm}(H_{\text{cn}} + \Delta H_{\text{cn}}) \quad (6)$$

$$\Delta H_{\text{en-cross}} = \text{LayerNorm}(H_{\text{en}} + \Delta H_{\text{en}}) \quad (7)$$

Following the cross-attention layer, a feedforward layer consisting of a single linear layer and a ReLU activation function is introduced to further enhance the model's representational capability, as shown in Eqs. (8) and (9). By incorporating a feedforward layer after the cross-attention layer, leading to a more refined feature representation.

$$H_{\text{cn-cross}} = \text{LayerNorm}(\Delta H_{\text{cn-cross}} + \text{Feedforward}(\Delta H_{\text{cn-cross}})) \quad (8)$$

$$H_{\text{en-cross}} = \text{LayerNorm}(\Delta H_{\text{en-cross}} + \text{Feedforward}(\Delta H_{\text{en-cross}})) \quad (9)$$

$$H_{\text{final}} = \frac{H_{\text{cn-cross}} + H_{\text{en-cross}} + H_{\text{cn}} + H_{\text{en}}}{4} \quad (10)$$

Finally, Eq. (10) is used to fuse and average features, incorporating the residual mechanism. This mechanism allows for the preservation of original features throughout the BCAM calculation process, facilitating the flow of information and gradients throughout the network. By allowing the model to learn residual functions with reference to input characteristics, the residual mechanism improves the overall effectiveness and efficiency of the training process, as demonstrated in [7]. This approach helps to mitigate the vanishing gradient problem and allows for the training of deeper networks, ultimately leading to improved performance in the emotion classification task.

### 3.3 LoRA with Embedding Fine-Tuning

We jointly fine-tune the BFFN by LoRA with Word embedding fine-tuning to construct bilingual processing branches and serve as a lightweight fine-tuning method that reduces memory requirements. LoRA constructs low parameter linear layers for the Chinese processing branch and the English processing branch, respectively. The low parameter linear layers for both languages store the results of BFFN’s Chinese and English fine-tuning, which fully utilizes the model’s bilingual processing capabilities (Chinese and English), reduces memory requirements, and minimizes parameter updates during BFFN fine-tuning. Fine-tuning word embedding can further accelerate the alignment process between embedding representations and the feature space, reducing the difficulty of the LoRA fine-tuning process. Through this adjustment, the model can learn according to task-specific data, optimizing word vectors to better reflect the semantic features related to the current task. This optimization is crucial for improving the model’s performance on specific text classification tasks, particularly when dealing with tasks that are highly sensitive to word meaning, such as emotion classification. By fine-tuning the word embedding, the model can more accurately capture the subtle semantic differences in the text.

### 3.4 Loss

In order to optimize the model parameters and minimize the discrepancy between the predicted and actual probability distributions, we employ the cross-entropy loss function during the training process. The back propagation algorithm is utilized to update the model parameters on the basis of the calculated loss. The mathematical formulation of the cross-entropy loss is given by Eq. (11):

$$\mathcal{L} = - \sum_{x \in \mathcal{D}} \sum_{i=1}^C y_i(x) \log(\hat{y}_i(x)) \quad (11)$$

Where  $\mathcal{L}$  represents the cross-entropy loss,  $\mathcal{D}$  denotes the training dataset, and  $C$  is the total number of emotion categories. For each training sample  $x$ ,  $y_i(x)$  represents the ground truth probability of  $x$  belonging to the  $i$ -th emotion category, while  $\hat{y}_i(x)$  represents the predicted probability of  $x$  belonging to the  $i$ -th

emotion category. By iteratively minimizing the cross-entropy loss during the training process, the model learns to generate predictions that closely match the ground truth probability distribution, thereby improving its performance on the emotion classification task.

## 4 Experiment

### 4.1 Datasets

The SMP2020-EWECT dataset [19], provided by the Social Computing and Information Retrieval Research Center at Harbin Institute of Technology, consists of two parts: pandemic-related and general data. The pandemic dataset (Epidemic dataset) contains 13,606 Weibo posts collected during the COVID-19 pandemic using relevant keywords, while the general dataset (General dataset) comprises 34,768 randomly collected Weibo posts without any specific topic. Both datasets are categorized into six emotional categories: fear, positive, neutral, anger, surprise, and sadness. The data is characterized by its brief length, casual language style, and the presence of noise, non-standard expressions, abbreviations, slang, and spelling errors. In addition, texts often include user information and contextual details, such as @mentions, topic tags, and links, which can influence emotion classification tasks. For the Epidemic dataset, the training set contains 8,606 corpus, the test set contains 3,000 corpus, and the verification set contains 2,000 corpus. As for the General dataset, the training set comprises 27,768 corpus, the test set includes 5,000 corpus, and the verification set consists of 2,000 corpus.

### 4.2 Evaluation Metrics

In this paper, we use Accuracy, Macro-Precision, Macro-Recall, and Macro-F1 as evaluation indicators. Accuracy (Acc) is the ratio of the number of correctly predicted samples to the total number of samples, calculated as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

Where TP, TN, FP, and FN represent True Positive, True Negative, False Positive, and False Negative, respectively. Precision (P) is the ratio of the number of correctly predicted positive samples to the total number of predicted positive samples, while Recall (R) is the ratio of the number of correctly predicted positive samples to the actual total number of positive samples. These metrics are calculated as follows:

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN} \quad (13)$$

The F1-score is the harmonic mean of P and R, which can more comprehensively evaluate the performance of the model:

$$F1 = 2 \times \frac{P \times R}{P + R} \quad (14)$$

In multi-class classification tasks, performance evaluation typically uses micro-averaging and macro-averaging metrics. Macro averaging first calculates TP, TN, FP, and FN for each category, then calculates P, R, and F1 separately, and finally takes the arithmetic mean. The specific calculation process is shown in the following equations:

$$Macro-P = \frac{1}{C} \sum_{i=1}^C P_i, \quad Macro-R = \frac{1}{C} \sum_{i=1}^C R_i, \quad Macro-F1 = \frac{1}{C} \sum_{i=1}^C F1_i \quad (15)$$

The  $P_i$ ,  $R_i$ , and  $F1_i$  are the Precision, Recall, and F1-score for the  $i$ -th category, respectively.

### 4.3 Training Details

In this paper, we set the dimension size of the hidden layer to 768, the batch size to 32, the learning rate to  $2e-5$ , and conducted 30 training epochs. We choose Adam optimizer as the optimizer in the training process. To limit the length of the input sequence, we set the maximum sequence length to 240 references to the settings [6]. In terms of the LoRA parameter setting, we set the  $r$  value to 16,  $alpha$  to 16 for each module of each linear layer in Attention and MLP. We used an Intel(R) Core(TM) i9-10850K CPU and an NVIDIA V100 GPU for all experiments in this paper. Our code is developed based on the Pytorch [16] deep learning framework.

### 4.4 Compare with the States-of-the-Arts

As shown in Table 2, BFFN increased Accuracy by 1.2%, Macro-F1 by 5%, Macro-R by 4.87%, and Macro-P by 1.31% compared to RoBERTa baseline. Compared to the state-of-the-arts ERNIE, BFFN achieved 1.43%, 5.99%, 5.91%, and 4.43% higher scores in Accuracy, Macro-F1, Macro-R, and Macro-P, respectively, on the epidemic dataset. These results effectively demonstrate the effectiveness of the bilingual feature fusion method proposed in this paper. By comparing the results of the RoBERTa and ERNIE, we can also observe that the fine-tuning results of RoBERTa are superior. This can be attributed to the fact that ERNIE is a pure Chinese pre-training model, and its ability to understand English content is inferior to that of RoBERTa. This finding supports the reasonableness of exploring and utilizing the model’s inherent multilingual capabilities to improve emotion classification performance, as presented in this paper.

Our proposed BFFN significantly outperforms the baseline BERT across all evaluation metrics. BFFN achieves 81.03% accuracy, 67.69% Macro-F1, 66.28% Macro-R, and 70.45% Macro-P, surpassing BERT by 2.76%, 9.76%, 8.46%, and 2.56%, respectively. These results demonstrate that the introduction of bilingual feature fusion and cross-lingual attention mechanisms in BFFN effectively captures and utilizes emotion information, thereby enhancing Chinese emotion classification performance.

**Table 2.** The comparison of our proposed BFFN with the state-of-the-arts on the SMP2020 Epidemic dataset.

Model	Acc	Macro-F1	Macro-R	Macro-P
BERT [5]	78.27%	57.93%	57.82%	67.89%
RoBERTa [4]	79.83%	62.69%	61.41%	69.14%
XLNET [3]	79.03%	62.94%	62.30%	64.96%
BERT+CFBLS [17]	76.40%	58.70%	—	—
PCA-BERT [20]	78.97%	63.99%	—	—
Nezha [26]	79.23%	59.63%	60.09%	69.67%
ELECTRA [3]	79.17%	59.72%	58.23%	<b>70.88%</b>
ERNIE [22]	79.60%	61.70%	60.37%	66.02%
<b>BFFN (ours)</b>	<b>81.03%</b>	<b>67.69%</b>	<b>66.28%</b>	70.45%

In Table 3, our proposed BFFN model still achieves the best results. However, unlike the Epidemic dataset, the General dataset has a training set that is more than twice as large, which leads to better performance for the ERNIE. This can be attributed to the fact that the ERNIE itself is relatively advanced and has certain structural advantages. Despite this, the method presented in this paper is still able to surpass the performance of the ERNIE on the General dataset (78.94% vs. 78.6%). This can be mainly attributed to our effective utilization of cross-language features through the Bilingual Cross Attention Mechanism (BCAM). The BCAM demonstrates robustness and adaptability to various datasets, enhancing the model’s generalization ability and enabling it to perform well across different domains.

**Table 3.** The comparison of our proposed BFFN with the state-of-the-art models on the SMP2020 General dataset.

Model	Acc	Macro-F1	Macro-R	Macro-P
BERT [5]	77.78%	74.72%	75.23%	74.41%
RoBERTa [4]	77.82%	75.11%	76.37%	74.18%
XLNET [3]	77.38%	74.23%	75.07%	73.63%
BERT+CFEELS [17]	76.30%	72.40%	—	—
Nezha [26]	76.90%	73.58%	73.84%	73.79%
ELECTRA [3]	77.80%	74.99%	75.08%	74.98%
ERNIE [22]	78.60%	75.89%	<b>76.63%</b>	75.43%
BFFN	<b>78.94%</b>	<b>75.94%</b>	76.23%	<b>75.70%</b>

#### 4.5 Ablation for Feature Fusion Methods

This section will conduct an ablation analysis on the details of the method proposed in this paper, mainly focusing on the basic bilingual feature addition (BFA), the bilingual cross-attention mechanism, and the LoRA with Embedding Fine-tuning (LEF) strategy, and we use the epidemic dataset for the ablation experiment. The backbone models employed in our ablation experiment were BERT and RoBERTa, respectively.

In Table 4, we primarily compared and analyzed different bilingual feature fusion methods. BFA represents the extraction and direct add of bilingual features based on the backbone model, while BCAM represents the Bilingual Cross Attention Mechanism we proposed. This experiment aims to analyze two factors: 1. Whether the bilingual feature fusion itself is effective. 2. Whether the proposed bilingual cross attention mechanism (BCAM) is effective.

**Table 4.** Analyzing the impact of different feature fusion methods.

Method	Acc	Macro-F1	Macro-R	Macro-P
BERT	78.27%	57.93%	57.82%	67.89%
+BFA	78.77%	64.00%	62.18%	67.64%
+BCAM	79.40%	65.41%	63.99%	68.08%
RoBERTa[61]	79.83%	62.69%	61.41%	69.14%
+BFA	79.87%	65.71%	64.42%	68.19%
+BCAM	80.77%	67.00%	66.04%	68.92%

The results presented in Table 4 clearly demonstrate the impact of different feature fusion methods on the performance of emotion classification models. By incorporating bilingual feature addition into the BERT and RoBERTa base models, we observe a notable improvement across all evaluation metrics, including accuracy, Macro-F1, Macro-Recall, and Macro-Precision. This finding underscores the importance of leveraging bilingual features to enhance the model’s ability to understand and classify emotions in text data containing mixed languages.

Moreover, the proposed Bilingual Cross Attention Mechanism (BCAM) further elevates the performance of both BERT and RoBERTa models compared to the BFA method. The BCAM approach achieves the highest scores in all metrics, with an accuracy of 79.40% and 80.77% for BERT and RoBERTa, respectively. This significant improvement can be attributed to the effectiveness of the cross-attention mechanism in capturing and integrating relevant bilingual information, enabling the model to better understand the nuances of emotion expressed in mixed-language text.

These results highlight the superiority of the BCAM method in emotion classification tasks, particularly in scenarios where the text data contains a mix of



languages, such as Chinese and English. By effectively leveraging bilingual features and employing a cross-attention mechanism, BCAM enhances the model’s ability to accurately classify emotions, outperforming both the base models and the BFA approach. This finding emphasizes the importance of considering language diversity and utilizing advanced feature fusion techniques to improve emotion classification performance in real-world applications.

#### 4.6 Ablation for Fine-Tuning Strategies

In this section, we compare and analyze the effectiveness of different training strategies based on the Epidemic dataset. The base model used here is not the original BERT, but rather RoBERTa combined with the BCAM. This approach allows us to analyze not only the effectiveness of different training strategies separately but also the effectiveness of the combination when these strategies are integrated with BCAM.

**Table 5.** The comparison of different fine-tuning strategies.

Methods	Ratio	Accuracy	Macro-F1	Macro-R	Macro-P
LoRA	–	80.77%	67.00%	66.04%	68.92%
LEF	10%	81.00%	67.70%	66.31%	70.42%
	25%	81.00%	67.72%	66.34%	70.42%
	50%	81.03%	67.69%	66.28%	70.45%
	75%	80.96%	66.84%	65.54%	69.31%
	90%	80.93%	67.23%	65.90%	69.85%
	100%	80.96%	67.31%	65.98%	70.00%

Table 5 presents the results of our experiments, focusing on two fine-tuning strategies: Low-Rank Adaptation (LoRA) and LoRA with Embedding Fine-tuning (LEF). The ratio column in the table represents the percentage of word vectors in the embedding matrix that are randomly selected for training. The results demonstrate that the LEF strategy consistently outperforms the LoRA strategy across all evaluation metrics, regardless of the ratio of word vectors selected for training. This indicates that fine-tuning the embedding layer alongside LoRA is beneficial for improving the model’s performance in emotion classification tasks.

The best performance is achieved with a ratio of 50%, the LEF strategy reaches an accuracy of 81.03% and a Macro-F1-score of 67.69%, surpassing the results obtained with higher ratios. This suggests that fine-tuning a subset of word vectors can be more effective than updating the entire embedding matrix. As the ratio of word vectors selected for training increases, we observe a slight decline in performance. This trend is particularly noticeable when the ratio exceeds 75%. One possible explanation for this phenomenon is that updating

a larger portion of the embedding matrix may introduce noise and overfitting, especially when the dataset is relatively small, like the Epidemic dataset used in this study. It is worth noting that even when the entire embedding matrix is fine-tuned (ratio = 100%), the LEF strategy still outperforms the LoRA strategy. This highlights the importance of updating the embedding layer during fine-tuning, as it allows the model to better adapt to the specific characteristics of the emotion classification task.

## 5 Conclusion

This paper introduces a novel approach for emotion classification that addresses the challenges posed by the increasing prevalence of mixed-language content on Chinese social media platforms. The proposed Bilingual Feature Fusion Network (BFFN) harnesses the multilingual capabilities of pre-trained language models to enhance the semantic feature extraction of Chinese text, while the Bilingual Cross Attention Mechanism (BCAM) effectively captures cross-lingual emotional information. Additionally, a lightweight fine-tuning method combining Low-Rank Adaptation (LoRA) and Embedding Fine-tuning (LEF) is employed to reduce the complexity of fine-tuning model weights for downstream tasks. Extensive experiments demonstrate the superiority of the proposed method, outperforming state-of-the-art models like ERNIE by 1.43% in accuracy. This work advances emotion classification in mixed-language contexts, providing a practical solution for applications.

## 6 Future Works

The current work primarily considers the bilingual situation in Chinese and English. As the world's cultures continue to blend, similar problems will exist in other multimedia and multimodal data, including other languages. Therefore, future work based on this paper will consider the following two points:

1. Evaluate the effectiveness of BFFN in mixed language environments outside of the Weibo dataset.
2. Investigate the effectiveness of the proposed BFFN approaches on other languages and mixed-language contexts beyond Chinese and English to further validate their generalizability and robustness.
3. Examine the performance of BFFN in real-time processing scenarios, especially when the data scale is large or biased towards specific domains.
4. Explore the integration of the proposed methods with multimodal emotion classification techniques, incorporating data from various channels such as audio, visual, and physiological signals, to achieve a more comprehensive emotion assessment in real-world scenarios where language mixing is increasingly common.

**Acknowledgements.** This research is supported by the Sichuan International Science and Technology Innovation Cooperation Project, also known as the Hong Kong, Macao, and Taiwan Science and Technology Innovation Cooperation Project, with ID 2024YFHZ0317.

## References

1. Bibi, R., Qamar, U., Ansar, M., Shaheen, A.: Sentiment analysis for Urdu news tweets using decision tree. In: 2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA), pp. 66–70. IEEE (2019)
2. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: ELECTRA: pre-training text encoders as discriminators rather than generators. arXiv preprint [arXiv:2003.10555](https://arxiv.org/abs/2003.10555) (2020)
3. Cui, Y., Che, W., Liu, T., Qin, B., Wang, S., Hu, G.: Revisiting pre-trained models for Chinese natural language processing. arXiv preprint [arXiv:2004.13922](https://arxiv.org/abs/2004.13922) (2020)
4. Cui, Y., et al.: Pre-training with whole word masking for Chinese BERT. arXiv preprint [arXiv:1906.08101](https://arxiv.org/abs/1906.08101) (2019)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
6. Guo, X., Lai, H., Xiang, Y., Yu, Z., Huang, Y.: Emotion classification of COVID-19 Chinese microblogs based on the emotion category description. In: Li, S., et al. (eds.) CCL 2021. LNCS (LNAI), vol. 12869, pp. 61–76. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-84186-7\\_5](https://doi.org/10.1007/978-3-030-84186-7_5)
7. 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)
8. Hu, G., Zhao, Q.: Multi-model fusion framework based on multi-input cross-language emotional speech recognition. *Int. J. Wireless Mobile Comput.* **20**(1), 32–40 (2021)
9. Jamal, N., Xianqiao, C., Aldabbas, H.: Deep learning-based sentimental analysis for large-scale imbalanced twitter data. *Future Internet* **11**(9), 190 (2019)
10. Johnson, R., Zhang, T.: Deep pyramid convolutional neural networks for text categorization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 562–570 (2017)
11. Kim, K., Park, S.: AOBERT: all-modalities-in-one BERT for multimodal sentiment analysis. *Inf. Fusion* **92**, 37–45 (2023)
12. Li, Z., Zhou, L., Yang, X., Jia, H., Li, W., Zhang, J.: User sentiment analysis of Covid-19 via adversarial training based on the BERT-FGM-BiGRU model. *Systems* **11**(3), 129 (2023)
13. Liu, B., et al.: Context-aware social media user sentiment analysis. *Tsinghua Sci. Technol.* **25**(4), 528–541 (2020)
14. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) (2019)
15. Mishra, A., Singh, A., Ranjan, P., Ujlayan, A.: Emotion classification using ensemble of convolutional neural networks and support vector machine. In: 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN), pp. 1006–1010. IEEE (2020)

16. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32** (2019)
17. Peng, S., et al.: Emotion classification of text based on BERT and broad learning system. In: U, L.H., Spaniol, M., Sakurai, Y., Chen, J. (eds.) *APWeb-WAIM 2021, Part I. LNCS*, vol. 12858, pp. 382–396. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-85896-4\\_30](https://doi.org/10.1007/978-3-030-85896-4_30)
18. Qi, Q., Lin, L., Zhang, R., Xue, C.: MEDT: using multimodal encoding-decoding network as in transformer for multimodal sentiment analysis. *IEEE Access* **10**, 28750–28759 (2022)
19. SMP2020: *Smp2020smp2020-ewect* (2020). <https://smp2020ewect.github.io/>
20. Su, M., Cheng, D., Xu, Y., Weng, F.: An improved BERT method for the evolution of network public opinion of major infectious diseases: case study of Covid-19. *Expert Syst. Appl.* **233**, 120938 (2023)
21. Sun, L., Liu, B., Tao, J., Lian, Z.: Multimodal cross-and self-attention network for speech emotion recognition. In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4275–4279. IEEE (2021)
22. Sun, Y., et al.: ERNIE 3.0: large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137* (2021)
23. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1422–1432 (2015)
24. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017)
25. Wang, Y., Feng, S., Wang, D., Zhang, Y., Yu, G.: Context-aware Chinese microblog sentiment classification with bidirectional LSTM. In: Li, F., Shim, K., Zheng, K., Liu, G. (eds.) *APWeb 2016. LNCS*, vol. 9931, pp. 594–606. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-45814-4\\_48](https://doi.org/10.1007/978-3-319-45814-4_48)
26. Wei, J., et al.: NEZHA: neural contextualized representation for Chinese language understanding. *arXiv preprint arXiv:1909.00204* (2019)
27. Yan, S., Wang, J., Song, Z.: Microblog sentiment analysis based on dynamic character-level and word-level features and multi-head self-attention pooling. *Future Internet* **14**(8), 234 (2022)
28. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: XLNet: generalized autoregressive pretraining for language understanding. *Adv. Neural Inf. Process. Syst.* **32** (2019)
29. Zeng, L., Li, R.Y.M., Zeng, H.: Weibo users and academia’s foci on tourism safety: implications from institutional differences and digital divide. *Heliyon* **9**(3), e12306 (2023)



# SNOBERT: A Benchmark for Clinical Notes Entity Linking in the SNOMED CT Clinical Terminology

Mikhail Kulyabin<sup>1</sup>(✉), Gleb Sokolov<sup>2</sup>, Aleksandr Galaida<sup>2</sup>, Andreas Maier<sup>1</sup>,  
and Tomas Arias-Vergara<sup>1</sup>

<sup>1</sup> Pattern Recognition Lab, Department of Computer Science,  
Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany  
mikhail.kulyabin@fau.de

<sup>2</sup> Moscow Institute of Physics and Technology - State University (MIPT), Moscow,  
Russia

**Abstract.** The extraction and analysis of insights from medical data, primarily stored in free-text formats by healthcare workers, presents significant challenges due to its unstructured nature. Medical coding, a crucial process in healthcare, remains minimally automated due to the complexity of medical ontologies and restricted access to medical texts for training Natural Language Processing models. In this paper, we proposed a method, “SNOBERT,” of linking text spans in clinical notes to specific concepts in the SNOMED CT using BERT-based models. The method consists of two stages: candidate selection and candidate matching. The models were trained on one of the largest publicly available datasets of labelled clinical notes. SNOBERT outperforms other classical methods based on deep learning, as confirmed by the results of a challenge in which it was applied.

**Keywords:** NLP · SNOMED · BERT · Entity Linking · NER

## 1 Introduction

Most medical data is stored in free-text documents, usually filled in by healthcare workers. Analyzing this unstructured data can be challenging, as it can be difficult to extract meaningful insights. Medical coding remains an under-automated process despite being widely applicable in healthcare, medical insurance, and medical research, mainly due to the vast amount of codes in medical ontologies and the minimal access to medical texts for training natural language processing systems [8]. In recent years, the problem of Named Entity Recognition (NER) within medical texts has received increasing attention from the research community [17]. By applying standardized terminology, healthcare organizations can convert this free-text data into a structured format that computers can readily analyze, stimulating the development of new medicines, treatment pathways, and better patient outcomes. One of the most comprehensive

and multilingual clinical healthcare terminologies in the world is Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) [3], a systematically organized computer-processable collection of medical terms that provides codes, terms, synonyms, and definitions used in clinical documentation and reporting.

Annotating medical data according to SNOMED CT terminology is a time-consuming and labor-intensive process that often requires the annotator to have prior medical training. Automating such a process using Natural Language Processing (NLP) methods is called an Entity Linking (EL) problem. EL is the task of linking entities within a text to a suitable concept in a reference Knowledge Graph [18]. This work presents a ‘‘SNOBERT’’ method for linking text spans in clinical notes with specific topics in the SNOMED CT clinical terminology that distinguishes itself with a novel two-stage approach leveraging advanced NLP models and a refined preprocessing strategy. In the first stage, we applied the candidate selection with a BERT-based model, whose embeddings are then matched with extracted embeddings from the entire dataset. The method was tested in the ‘‘SNOMED CT Entity Linking Challenge’’ [7].

## 2 Related Works

With the recent advances of deep learning (DL) technologies, NLP applications have received an unprecedented boost in performance [13]. However, DL has rarely been used to solve the entity linking problem in SNOMED CT terminology since the massive size of the corpus needed to train on such a large set of classes [5]. Nevertheless, few papers have been published in the field. Hristov et al. [8] proposed a method that integrates transformer-based models, such as BERT, pre-trained on biomedical data, with support vector classification using the transformer embeddings for fine-tuning and predicting SNOMED CT codes for medical texts. This hybrid approach leverages the strengths of both deep learning and classical machine learning techniques to achieve high accuracy in medical text coding, particularly in morphology and topography coding.

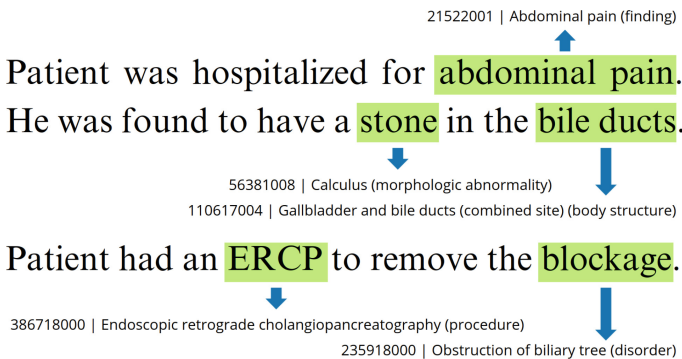
The KGE4SCT method [4] is a technique that utilizes Knowledge Graph Embeddings (KGEs) to automatically post-coordinate SNOMED CT clinical terms. It does this by using a vector space to capture the ontology’s graph-like structure. The method uses vector similarity and analogies to derive post-coordinated expressions for clinical terms that are not explicitly present in SNOMED CT. This facilitates the encoding of clinical information that has been extracted from text. The effectiveness of this method has been validated on a subset of SNOMED CT and a set of manually post-coordinated concepts.

## 3 Data

In this work, we used the MIMIC-IV-Note dataset, which contains 331,794 de-identified hospital discharge summaries from 145,915 patients provided by the Beth Israel Deaconess Medical Center (BIDMC) and Massachusetts Institute of Technology (MIT) [10]. The challenge provided annotated data, comprising up to

300 annotated discharge summaries from the original MIMIC-IV-Note dataset. The full dataset consists of a public training subset (204 notes) and a private test subset (around 70 notes) comprising approximately 75,000 annotations across discharge summaries.

Each entry in the discharge dataset includes a note ID and the anonymized discharge text. Annotations indicate the concept ID, start and end points, and the corresponding note ID. Each note ID consists of a subject ID, note sequence position number, and note type (in this work, we use only discharges). Annotations consist of note IDs, starting and end points of concepts according to SNOMED CT clinical terminology, and their corresponding IDs. Fig. 1 shows an example of an annotated part of a synthetic discharge note according to the SNOMED CT terminology.

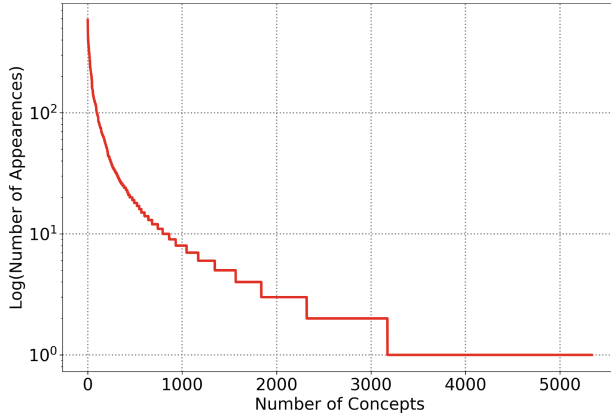


**Fig. 1.** Example of synthetic discharge note annotated according to SMOMED CT terminology, e.g. “blockage” corresponds to “235918000” concept ID.

Medical notes frequently use abbreviations that can be context-dependent and may assume prior knowledge. In addition, the knowledge bases used in medical notes can contain hundreds of thousands of concepts, with many of these concepts occurring infrequently. As a result, there can be a “long tail” effect in the distribution of concepts, Fig. 2. Thus, 2162 concepts out of 5336 appear only once in the annotated data. This effect causes zero-shot learning (ZSL), a problem in DL in which, at test time, a learner observes samples from classes that were not observed during training and needs to predict the class to which they belong.

## 4 Method

This section presents the proposed method for the clinical notes EL. Figure 3 illustrates the scheme of the method that consists of two stages: Candidate Selection and Candidate Matching. In the first stage, we solved the NER classification problem, and in the second stage, for each classified span from the



**Fig. 2.** Distribution of the concepts in the annotated dataset: “long tail” distribution effect.

first stage, we linked the corresponding concept ID in SNOMED terminology. In further sections, we describe each of the stages in detail (Fig. 4).

#### 4.1 Preprocessing

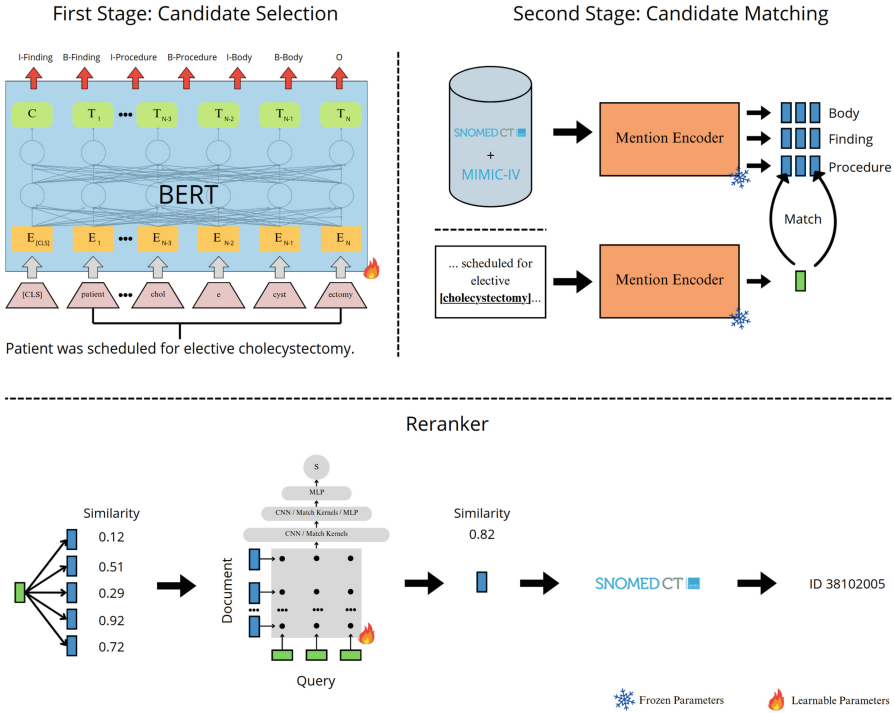
We utilized the NER pipeline [20] to address certain annotation inaccuracies, such as those caused by shifts due to tags. Approximately 10 notes out of 204 underwent corrections, involving adjustments to around 150 annotation IDs. These corrections specifically targeted errors resulting from shifted annotations. Furthermore, most annotated notes are missing some labels in the paragraphs with the following headers: ‘medications on admission:’, ‘\_\_\_ on admission:’, ‘discharge medications:’. We excluded these parts from the training process. All HTML markup elements, such as the line break element (‘br’) or the new line (‘n’), have also been removed from the notes.

While a robust baseline for EL, the dictionary method falls short when it comes to ZSL, highlighting the need for alternative solutions [2]. We generated a static dictionary of the most common concepts from training data and matched them with test data in the post-processing step using a string-matching search. Levenshtein ratio or Stolois distance could be utilized as a matching metric. However, we applied “one-to-one” matching, linking only complete coincidences.

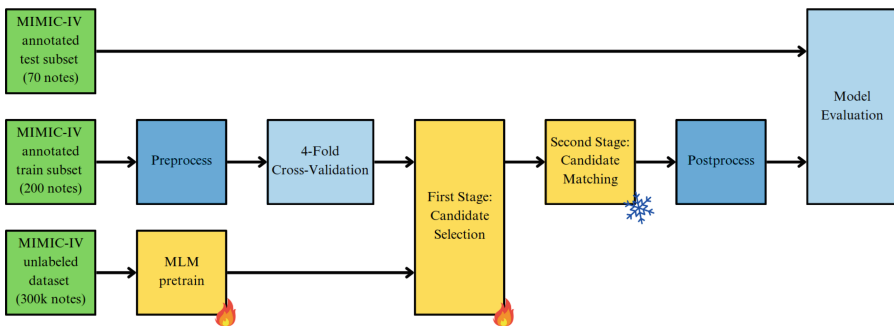
#### 4.2 First Stage: Candidate Selection

NER is the task of identifying rigid designators’ mentions from text belonging to predefined semantic types such as person, location, organization, etc. [14]. NER involves processing raw text through stages, including Sentence Segmentation, where text is divided into sentences, and Word Tokenization, which breaks text into individual words. Subsequent stages include Part of Speech Tagging, assigning grammatical tags based on word roles and context, and Entity Detection,





**Fig. 3.** SNOBERT scheme. The method consists of two stages. In the Candidate Selection stage (I), the BERT model is utilized to classify the text’s tokens into seven classes. In the Candidate Matching stage (II), the Mention Encoder matches the extracted embeddings from the training and testing datasets within these classes. Reranker is used to rerank the top matches and to get the final similarity score.



**Fig. 4.** Training pipeline of the proposed approach. The method uses a two-stage solution: Candidate Selection and Candidate Matching. All the models were trained on the MIMIC-IV dataset. The model from the first stage was trained on the annotated training subset. An optional pretrain step was done on the full unlabelled dataset. Models were evaluated on the test annotated subset.

which identifies and categorizes key elements in the text, highlighting the core function of NER in extracting meaningful information from unstructured data.

We proposed to separate the concepts according to the first-level hierarchy (the first entry in the path to the concept) according to SNOMED terminology. Thus, we can emphasize the “Finding,” “Procedure,” “Body structure,” and “None” classes. During Tokenization, the words got flags in “B-I-O” tagging format: “B” (Beginning) means the first token of the first word within an annotation; “I” (Inside) - is the first token of a subsequent word within an annotation; “O” (Outside) - is as a stand-alone token or single word token. Consequently, we have seven classes: “I-Finding,” “B-Finding,” “I-Procedure,” “B-Procedure,” “I-Body,” “B-Body,” and “O”.

### 4.3 Second Stage: Candidate Matching

To link classified terms from the first stage, we matched their embeddings with embeddings of terms from SNOMED terminology by cosine similarity. For this purpose, the whole database extracted concepts from the “Body structure,” “Findings,” and “Procedure” paths, which is about 200k unique IDs with the Mention Encoder. In this work, we applied the cambridgeltl/SapBERT-from-PubMedBERT-fulltext-mean-token model trained with UMLS 2020AA [15].

### 4.4 Postprocessing

A reranker is used to improve the performance of the model’s initial predictions. The model generates a list of possible predictions, which might not be ranked optimally in order of correctness. A re-ranker evaluates these predictions and adjusts their ranking based on more refined or specific criteria. Using only the top one or top five predictions may cause the correctly predicted vector to be missed. In our method, as a re-ranker, we applied the MedCPT model [9] trained on 18M semantic query-article pairs from PubMed.

### 4.5 Metrics

We evaluated the proposed method on both stages. First stage was evaluated with Macro-F1 score across all labels:

$$Macro - F1 = \frac{1}{N} \sum_i \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i}, \quad (1)$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i}, \quad (2)$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i}, \quad (3)$$

where  $TP = True Positive$ ,  $TN = True Negative$ ,  $FP = False Positive$ ,  $FN = False Negative$ , and  $N$  is a number of classes.

The second stage was evaluated with cosine similarity:

$$\cos(\theta) = \frac{\sum_i^n A_i B_i}{\sqrt{\sum_i^n A_i^2} \sqrt{\sum_i^n B_i^2}}, \quad (4)$$

where A and B are comparing vectors.

The final results were evaluated using a class macro-averaged character intersection over union (mIoU). IoU is a popular metric for measuring localization accuracy and computing localization errors. It calculates the amount of overlap between a prediction and a ground truth:

$$IoU_{class} = \frac{P_{class}^{char} \cap G_{class}^{char}}{P_{class}^{char} \cup G_{class}^{char}}, \quad (5)$$

$$macro\ IoU = \frac{\sum_{classes \in P \cup G} IoU_{class}}{N_{classes \in P \cup G}}, \quad (6)$$

where  $P_{class}^{char}$  is the set of characters in all predicted spans for a given class category,  $G_{class}^{char}$  is the set of characters in all ground truth spans for a given class category, and  $classes \in P \cup G$  are the set of categories present in either the ground truth or the predicted spans.

## 4.6 Training

For the training, we employed the four-fold cross-validation. Each fold consisted of 51 discharge notes. For the first stage, we used a domain-specific pretrained language model for Biomedical Natural Language Processing [19]. We utilized the base version, microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext, which was pretrained on abstracts from PubMed and full-text articles from PubMedCentral and is available in the HuggingFace repository [6]. We ran four experiments for each setup so that three folds were used each time for training and one for validation. We trained each split on 100 epochs using EarlyStoppingCriteria. Therefore, 75 epochs on average were needed. We used ADAM optimizer with  $3e^{-5}$ , batch size of 8, and class weighting. Training took 30 min on 4 GPUs (NVIDIA A100-SXM4-40GB). A more detailed description of the training configuration is shown in our training repository [11].

We found a slight improvement of 0.0005 in IoU when applying the Masked Language Model (MLM) pretraining technique. To accomplish this, we used the larger microsoft/BiomedNLP-BiomedBERT-large-uncased-abstract weights as an initial model instead. This optional pretraining step took 24 h on 4 GPUs.

## 5 Results

Table 1 shows the averaged cross-validation evaluation results for the first (I) stage and final evaluation. MLM pretraining slightly improves the F1 score in the first stage, and therefore, mIoU in the second with the best score of 0.4302

**Table 1.** Evaluation metrics of the proposed method

Model (I)	GPUs	F1 (I)	mIoU	Epoch
BiomedBERT large	1	0.7429	0.4231	75
BiomedBERT base	1	0.7487	0.4199	76
<b>BiomedBERT large</b>	4	<b>0.7514</b>	<b>0.4302</b>	74
BiomedBERT base	4	0.7499	0.4257	72

training on 4 GPUs. MultiGPU outstands single due to the batch size and its synchronization in the Bert model.

Table 2 shows the evaluation results of the second stage before reranking for each class. The results for the top five best candidates are significantly higher than those of the top one, which shows the need to use a reranker in the post-process, as the best candidate after the second stage is not always at the top of the similarity score.

**Table 2.** Evaluation metrics of the second stage.

Category	Similarity@1	Similarity@5
Body structure	0.715	0.811
Findings	0.614	0.703
Procedure	0.478	0.694

Table 3 shows the final test results of the three best methods and the baseline of the competition. The dictionary-based method achieved the highest score. However, this is a time-consuming method that requires manual effort.

**Table 3.** Results on the test dataset.

Author	Method	mIoU
Bilu et al.	Dictionary-based *	0.4202
Ours	SNOBERT	0.4194
Popescu et al.	Faiss + Mistral	0.3777
Baseline	deberta-v3-large	0.1794

\* Time-consuming semi-manual method

## 6 Discussion

Even though the MIMIC-IV-Note dataset we used for training the models has some limitations, it is extensive, well-organized, and properly annotated. Our

method requires a large dataset, which can be challenging and expensive to prepare. Medical notes often contain abbreviations and assumed knowledge, and the knowledge base itself can include hundreds of thousands of medical concepts.

Technology must be able to be used in countries where English isn't the primary language. The MIMIC-IV-Note dataset contains 300 discharge notes that were annotated in English. It's essential to understand that models trained only on this English-labeled data are limited to working with English text. However, parts of our solution, like the pre-trained SapBERT model, are multilingual, as they were trained on texts from different language domains.

## 7 Conclusion

We proposed a “SNOBERT” method of EL of text spans in clinical notes with specific topics in the SNOMED CT clinical terminology, tested in practice in the “SNOMED CT Entity Linking Challenge.” Our method uses two stages; however, an end-to-end approach could improve the linking score [1] but would need more training data from the other side. The limited annotation problem could be solved using synthetic data generated with Large Language Models [12]. SapBERT from the second stage can be changed by the BioLORD model, trained on a cumulative dataset of biomedical concepts' names and descriptions [16]. With BioLORD, we achieved results similar to SapBERT's. However, given that the data used to train this model originates from UMLS, that makes different restrictions.

The proposed method showed confident results, and so did a fair comparison with alternative methods during the “SNOMED CT Entity Linking Challenge” [7], losing only to the dictionary-based method by less than a percent. Since our method used a dictionary in the last step, the two methods can be easily combined, potentially yielding a significant improvement in the score.



## References

1. Ayoola, T., Tyagi, S., Fisher, J., Christodoulopoulos, C., Pierleoni, A.: ReFinED: an efficient zero-shot-capable approach to end-to-end entity linking. In: Loukina, A., Gangadharaiah, R., Min, B. (eds.) Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, pp. 209–220. Association for Computational Linguistics, Hybrid, Seattle (2022). <https://doi.org/10.18653/v1/2022.naacl-industry.24>,
2. Basaldella, M., Liu, F., Shareghi, E., Collier, N.: COMETA: a corpus for medical entity linking in the social media. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3122–3137. Association for Computational Linguistics, Online (2020). <https://doi.org/10.18653/v1/2020.emnlp-main.253>,
3. Benson, T.: Principles of Health Interoperability HL7 and SNOMED. Springer, Heidelberg (2012)

4. Castell-Díaz, J., Miñarro-Giménez, J.A., Martínez-Costa, C.: Supporting snomed ct postcoordination with knowledge graph embeddings. *J. Biomed. Inf.* **139**, 104297 (2023). <https://doi.org/10.1016/j.jbi.2023.104297>
5. Gaudet-Blavignac, C., Foufi, V., Bjelogrić, M., Lovis, C.: Use of the systematized nomenclature of medicine clinical terms (snomed ct) for processing free text in health care: systematic scoping review. *J. Med. Internet Res.* **23**(1), e24594 (2021)
6. Gu, Y., et al.: Domain-specific language model pretraining for biomedical natural language processing (2020)
7. Hardman, W., et al.: Snomed ct entity linking challenge (2024)
8. Hristov, A., et al.: Clinical text classification to snomed ct codes using transformers trained on linked open medical ontologies. In: Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing, pp. 519–526 (2023)
9. Jin, Q., et al.: Medcpt: contrastive pre-trained transformers with large-scale pubmed search logs for zero-shot biomedical information retrieval. *Bioinformatics* **39**(11), btad651 (2023)
10. Johnson, A.E., Bulgarelli, L., Shen, L., Gayles, A., Shammout, A., Horng, S., Pollard, T.J., Hao, S., Moody, B., Gow, B., et al.: MIMIC-IV, a freely accessible electronic health record dataset. *Sci. Data* **10**(1), 1 (2023)
11. Kulyabin, M., et al.: A benchmark for clinical notes entity linking in the snomed ct clinical terminology. <https://github.com/MikhailKulyabin/SNOBERT>
12. Kweon, S., et al.: Publicly shareable clinical large language model built on synthetic clinical notes (2023)
13. Lauriola, I., Lavelli, A., Aioli, F.: An introduction to deep learning in natural language processing: models, techniques, and tools. *Neurocomputing* **470**, 443–456 (2022). <https://doi.org/10.1016/j.neucom.2021.05.103>. <https://www.sciencedirect.com/science/article/pii/S0925231221010997>
14. Li, J., Sun, A., Han, J., Li, C.: A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **34**(1), 50–70 (2022). <https://doi.org/10.1109/TKDE.2020.2981314>
15. Liu, F., Shareghi, E., Meng, Z., Basaldella, M., Collier, N.: Self-alignment pretraining for biomedical entity representations. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4228–4238 (2021)
16. Remy, F., Demuyne, K., Demeester, T.: BioLORD: learning ontological representations from definitions for biomedical concepts and their textual descriptions. In: Findings of the Association for Computational Linguistics: EMNLP 2022, pp. 1454–1465. Association for Computational Linguistics, Abu Dhabi (2022). <https://aclanthology.org/2022.findings-emnlp.104>
17. Reyes-Aguillón, J., et al.: Clinical named entity recognition and linking using bert in combination with Spanish medical embeddings. In: CLEF (Working Notes), pp. 341–349 (2022)
18. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.* **27**(2), 443–460 (2014)
19. Tinn, R., et al.: Fine-tuning large neural language models for biomedical natural language processing (2021). <https://doi.org/10.48550/ARXIV.2112.07869>
20. Tkachenko, M., Malyuk, M., Holmanyuk, A., Liubimov, N.: Label studio: data labeling software (2020–2022). <https://github.com/heartexlabs/label-studio>



# Enhancing Automated Short Answer Grading with Prompt-Driven Augmentation and Prompt Adaptive Oversampling

P. P. Afeefa<sup>(✉)</sup> , Raju Hazari , and Pranesh Das 

Machine Learning Laboratory, Department of Computer Science and Engineering,  
National Institute of Technology Calicut, Kozhikode, India  
{afeefa\_p220097cs, rajuhazari, praneshdas}@nitc.ac.in

**Abstract.** Automated Short Answer Grading (ASAG) comes under automatic answer script evaluation where the answer length is limited from one phrase to one paragraph. The main task in ASAG is generating a good sentence embedding for both the student and the reference answers. The existing works on the embedding creation perform better when using different deep-learning techniques and language models. However, the deep-learning techniques' performance mainly depends on the training set size and quality. Most of the publicly available datasets typically have a limited number of reference and student answer pairs. To automate the dataset expansion, text augmentation techniques can be used. Conventional methods like back-translation, synonym replacement, and random deletion may replace some important technical words with other non-relevant terms, resulting in a loss of contextual meaning. We propose a new augmentation strategy for the ASAG datasets using LLM (Large Language Model) prompting. The effect of the proposed strategy is analysed on sentence transformer fine-tuning. We experimented with four different sizes of augmented training sets to determine the impact of the size of augmented training data on fine-tuning the sentence transformer model. Results indicate that sentence transformer fine-tuned using a 50% prompt-driven augmented dataset generates better embeddings. After having good embeddings, the traditional classifiers can be used to classify the student answers to different scores. We introduce “Prompt Adaptive Oversampling (PAO)” to address the class imbalance issue during grade classification. The effectiveness of the proposed strategy is analysed on two different public datasets: SPRAG, and Mohler-ASAG. The proposed method performs better while training highly imbalanced datasets. [The source code of this work is available here.](#)

**Keywords:** ASAG · Text-augmentation · Class-imbalance · Over-sampling · Siamese network · Sentence-transformer · Pre-trained models · Prompting · LLM

## 1 Introduction

Automatic answer script evaluation is an area of research that seeks to develop algorithms and methods for automatically evaluating student answers in the education system. As the use of online education platforms continues to grow, the need for efficient and accurate methods of assessing student answers has become increasingly important. Traditionally, the evaluation of student answers has been done manually, which can be time-consuming and lead to grading inconsistencies. Automatic answer script evaluation aims to address these issues by developing systems that can evaluate student answers, and provide feedback and grades to students without human intervention. This work focuses on Automatic Short Answer Grading (ASAG), a kind of answer script grading where the answer script length is limited from one phrase to one paragraph. ASAG can be considered as both a regression and a classification problem. In the regression approach, ASAG predicts a continuous score for each short answer. While in the classification approach, it categorizes each short answer into predefined grade categories. Both approaches have their advantages and disadvantages. The regression approach provides more nuanced grading by assigning a specific numerical score, but it requires a continuous scale for grading and may be sensitive to the consistency of human graders. On the other hand, the classification approach simplifies grading by categorizing answers into discrete grade categories, but it may lose some granularity in grading and requires careful definition of grade boundaries. To apply regression, the dataset should contain continuous scores. Most publicly available datasets do not contain continuous scores as grades. But, if the score in the dataset is continuous, it can be converted to categorical by applying some boundary conditions. Hence, in this study, we consider ASAG as a classification problem.

Most previous works in ASAG are considered a pairwise comparison between the student and reference answers [8]. A vector representation that encapsulates the context and meaning of the student and reference answer is necessary for comparing them accurately. There are many types of vector representations for texts. Word2Vec [16] stands out as a leading choice because it consistently represents words regardless of their contextual nuances. Since it is a static representation, it is giving context-free vector representations. So, instead of this static representation, if the vectors are created by considering the surrounding words in a sentence, it would be more related to the context. Because of this realization, bi-directional LSTM [11] was introduced. The drawback of this network is its computation complexity. It predicts the tokens based on past and future but does not encounter them simultaneously. Then, BERT [6] was introduced to address this issue. It will take both the previous and next tokens into account at the same time. Any network can be used to fine-tune these vectors after having a good vector representation of both the student answer and the reference answer. A Siamese network [9] is a type of neural network architecture commonly used for tasks involving similarity or dissimilarity measurement between pairs of inputs. The architecture of a Siamese network can be customized based on the characteristics of the task. The appropriate layers, embeddings, or attention mechanisms



can be chosen to suit the specific requirements of our data. In the context of short answer grading, the Siamese network can be used to assess the similarity between a student's answer and a reference answer. Some works are already done using the Siamese network [4, 5, 7, 13, 19, 20]. A deep hybrid Siamese neural network is introduced in [4]. This network consists of a combination of Siamese Convolutional Neural Network(CNN), Siamese Bi.LSTM layer, interaction enhancement layer, and dot product attention mechanism. The final layer of this network is the classification layer. However, the computation complexity of this method is greater due to the very deep network architecture. In [7], a pre-trained transformer model, T5, in conjunction with a Siamese BI-LSTM architecture is used. A framework called GradeAid is introduced in [5], which combines lexical and semantic features. Lexical features are computed with the TF-IDF method and semantic features are computed with the BERT cross encoder. It uses a regression method for scoring. The limitation of this work is that the regressor's output values can be under zero and over the maximum score. So, these scores need to be capped manually. In [13] a Siamese LSTM network for automatic short-answer scoring is introduced. They use an earth movers distance pooling mechanism to combine the Siamese network output and subsequently add a regression layer for score calculation. Another Siamese BiLSTM for the ASAG purpose is introduced in [19]. Here, the Glove is used for word embedding and after the Siamese BiLSTM layer, a dense layer is added to combine these embeddings and then the probability distribution of the score is predicted. The sentence transformer [21] is a kind of Siamese network introduced to compute the vector representation for a sentence rather than a word. Here, the base model is BERT. So, the sentence transformer could achieve state-of-the-art performance even though it contains a single pair of BERT layers compared to other Siamese networks. So, in this work, the sentence transformer is used to fine-tune the embeddings of the answers. However, the accuracy of every deep-learning technique is greatly influenced by the size and quality of the dataset. The publicly available dataset only consists of thousands of answer pairs. To automate the dataset scaling, text augmentation techniques can be used. A study about the effectiveness of transfer learning and dataset expansion using various augmentation techniques is conducted in [2]. They experimented with augmentation techniques like back-translation, random deletion and synonym replacements. These augmentation techniques do not ensure the preservation of the exact contextual meaning of the sentence. Sometimes these augmentation techniques may lead to the replacement of some important technical words with other non-relevant terms further leading to the loss of contextual meaning. So, one of the objectives of this work is to propose a new dataset augmentation technique by prompting a Large Language Model(LLM) [14]. This technique will preserve the technical terms after the augmentation.

After embedding the student and reference answers, the classifiers are trained to build a score prediction model. The class imbalance problem is one of the major challenges in classification. The class imbalance problem is the situation where the classifiers are trained on the datasets having the number of samples

in some classes significantly higher than the other classes. In such cases, the model may achieve high accuracy, but perform poorly in the case of minority classes. Different types of oversampling techniques can be used to address these issues. Randomly selecting instances from minority classes to duplicate the samples is the simplest method of oversampling technique. Synthetic Minority Over-sampling Technique (SMOTE) [3] is another method that generates synthetic samples of the minority class by interpolating between existing minority class instances. Adaptive Synthetic Sampling Approach for Imbalanced Learning (ADASYN) [10] is an extension of SMOTE that adaptively generates new instances based on the difficulty of learning the minority class samples. The limitation of these oversampling techniques is that if the minority classes consist of outliers, they will be replicated in the oversampled dataset. To overcome this issue, we propose Prompt Adaptive Oversampling (PAO), which combines prompt-driven augmented samples with existing synthetic oversampling techniques.

The main contributions of this work are:

- Proposing a new augmentation strategy that preserves the contextual meaning for automatic short answer grading datasets using the prompting technique.
- Fine-tuning sentence transformer with the original and augmented training sets and comparing the performance.
- Compare the performance of the proposed augmentation strategy with other existing augmentation strategies on sentence transformers with different training set sizes. Hence find out the best model for answer-pair embedding.
- Apply the proposed oversampling strategy on different datasets to solve class-imbalance issues and compare the effect of this oversampling with other strategies on different classifiers.

The rest of the paper is organized in the following way. Section 2 provides a detailed description of the proposed methodology. Section 3 provides the experimentation details and a comparative study. Section 4 concludes the paper.

## 2 Proposed Methodology

The architecture of the proposed methodology is shown in Fig. 1. It comprises of three phases. Phase 1 augment the original dataset using the prompt-driven technique. Phase 2 is the sentence transformer fine-tuning. In phase 3, the training set is balanced using PAO and the classifier is trained using a balanced dataset. A detailed description of this architecture’s component is provided in the following sub-sections.

### 2.1 Dataset Preprocessing

Initially, the dataset is converted to a CSV format to make it convenient for further processing. Here, the first four columns are indicated by the question\_id,

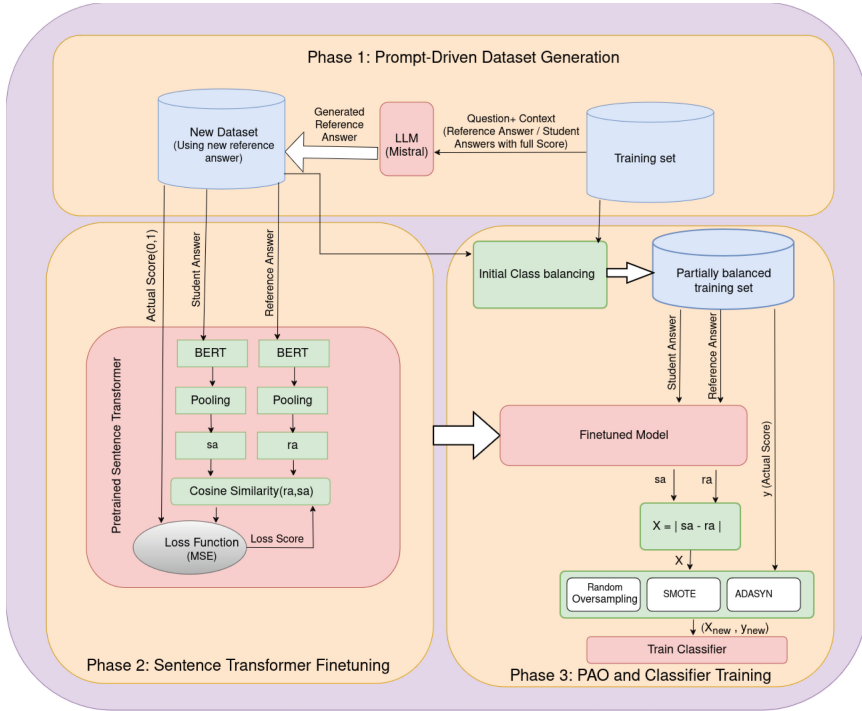


Fig. 1. Architecture of Proposed Methodology

the question, the corresponding reference answer, and the student answer respectively. As a fifth column, we have added the student mark as the average score of two evaluators. During the fine-tuning phase, the range of scores (between 0 and 5) is needed to map to the range of cosine similarity values (0 to 1). It is done by simply dividing the score by 5. In Phase 1, all the five columns are required. For Phase 2 and Phase 3, the columns ‘Reference answer’, ‘Student answer’ and ‘Score’ are required. So, at the beginning of each phase, a data frame which contains the above columns is created. The dataset is initially split into training and test sets. Here, we choose 20% of the total dataset as the test set and the remaining as the training set. The dataset split is done before dataset scaling.

## 2.2 Prompt-Driven Dataset Generation

In Phase 1, a dataset is generated by adding new reference answers to the original dataset entries. The initial expansion is done by adding the student answers with the full score as another set of reference answers. The additional reference answers are then generated by prompting an LLM. Prompting an LLM using a question from a dataset will give more generalized answers in some situations. Providing the context and question, LLM produces a better reference answer. The reference answer or the answers with the full score corresponding to that

**Algorithm 1.** Prompt-Driven Dataset Generation

---

```

1: Input: Original training set, and the new reference answer set
2: Output: Augmented training set
3: initialize new dataframe named generated_dataset
4: train  $\leftarrow$  training set
5: new  $\leftarrow$  new reference answer set
6: m  $\leftarrow$  size(original training set)
7: for i  $\leftarrow$  0 to m - 1 do
8:   result  $\leftarrow$  rows from 'new' where new.id = train.id(Studenti)
9:   for j  $\leftarrow$  0 to size(result)-1 do
10:    new_row  $\leftarrow$  {train.id(i), train.Question(i), result.answer(j),
    train.StudentAnswer(i), train.Score(i)}
11:    add new_row as the last index of current generated_dataset
12:  end for
13: end for
14: Save updated generated_dataset as CSV file

```

---

question can be used as the context. Repeat the above procedure for the entire questions-correct\_answer pairs in the original dataset. Mistral-7B-Instruct-v0.1<sup>1</sup> is used here as the LLM model. It is an instruct fine-tuned version of the Mistral-7B-v0.1 [12] which uses a variety of publicly available conversation datasets. This model is open source and small (only 7B parameters). However, it outperforms other best open-source models. To handle the data efficiently during prompting, a 4-bit quantization technique [15] is used. Algorithm 1 is introduced to generate a new dataset with LLM-generated reference answers.

The 'new reference answer set' in Algorithm 1 is a dataframe containing three columns - *id*, *question*, and *answer*. Before doing the prompt-driven augmentation, an initial augmentation is done by creating new instances where the new reference answers are the student answers with full score. For initial augmentation also the same algorithm is used. During initial data expansion, the column *answer* consists of the student answers with full scores. During the prompt-driven dataset expansion, the column *answer* contains the new set of reference answers generated using LLM prompting. The variable *m* varies in different datasets, indicating the original training set's size.

### 2.3 Fine-Tuning the Sentence Transformer

The sentence transformer can be fine-tuned by setting two objectives - classification and regression. Here, we focus on fine-tuning the model with regression as the primary objective. The sentence transformer is designed as a Siamese network to get meaningful embedding for the answer pairs. The base layer of the sentence transformer comprises a pair of pre-trained BERT models. A pair of pooling layers succeed the BERT. Finally, the output layer connects these identical pairs of layers. This network aims to maximize the cosine similarity between

<sup>1</sup> <https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1>.

similar text inputs and minimize the cosine similarity between dissimilar texts. The student and the reference answer are fed as input text pairs during fine-tuning. The corresponding student score converted to the range of 0 to 1 is used as the actual similarity measure between the student and the reference answer. At each iteration, the cosine similarity between the student and reference answer denoted by  $sa \in \mathbb{R}^n$  and  $ra \in \mathbb{R}^n$  respectively are calculated.

$$\text{cos\_similarity}(sa, ra) = \frac{\sum_{i=1}^n sa_i * ra_i}{\sqrt{\sum_{i=1}^n sa_i^2} \sqrt{\sum_{i=1}^n ra_i^2}} \quad (1)$$

where  $n$  is the dimension of the student and reference answers. The mean square error (MSE) loss is then calculated and fine-tuned the sentence transformer by back-propagating this loss.

$$\text{MSE} = \frac{\sum_{j=1}^N (\text{actual\_score}_j - \text{cos\_similarity}(sa_j, ra_j))^2}{N} \quad (2)$$

where  $N$  is the number of data points in the training set. The sentence transformer used for this work is *all-MiniLM-L6-v2*<sup>2</sup>. This is a simple model that requires less computational resources but performs well like other large models. The result is minimally influenced by the hyperparameters used during fine-tuning. The following hyperparameter setting is used during the fine-tuning to facilitate experiments.

- Learning rate = 2e-05
- Scheduler = warmupconstant
- Number of epochs = 10
- Warmup steps = 500
- Batch size = 64
- Optimizer: Adam
- Evaluation steps = 500

## 2.4 Prompt Adaptive Oversampling (PAO) and Classifier Training

Oversampling is a technique used to address class imbalance in the dataset while training a classifier. When certain classes are underrepresented, the classifier struggles to learn patterns effectively from these classes. One commonly used oversampling technique is randomly duplicating examples from minority classes in the training set to balance the class distribution. However, random oversampling can lead to overfitting especially when applied to highly imbalanced datasets. If some classes have extremely few instances compared to others, and if some outliers are present in those classes, there is a possibility to duplicate such outliers multiple times, further leading to overfitting. Adding more instances without duplicating the same instances can solve this problem. The underrepresented class instances from the new Prompt-driven dataset generated in Phase

<sup>2</sup> <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>.

**Algorithm 2.** Prompt-Adaptive Oversampling (PAO)

---

```

1: Input: Original training set, and the prompt-driven augmented instances
2: Output: Balanced training set ( $X, y$ )
3: train  $\leftarrow$  training set
4: aug  $\leftarrow$  prompt-driven augmented instances
5:  $m \leftarrow$  length(train)
6:  $l \leftarrow$  number of labels
7: count_t  $\leftarrow$  count of each label in ‘train’
8: count_n  $\leftarrow$  count of each label in ‘aug’
9: for  $i \leftarrow 0$  to  $l - 1$  do
10:    $b \leftarrow []$ 
11:   if count_t[ $i$ ] < max(count_t) *  $\alpha$  then
12:      $b.append(\max(\text{count\_t}) * \alpha - \text{count\_t}[i])$ 
13:   else
14:      $b.append(0)$ 
15:   end if
16: end for
17: for  $i \leftarrow 0$  to  $l - 1$  do
18:   if count_n[ $i$ ] <  $b[i]$  then
19:      $frac=1$ 
20:   else
21:      $frac=\frac{b[i]}{\text{count\_n}[i]}$ 
22:   end if
23:   train.concatenate( $frac$  fraction of samples from each question in aug)
24: end for
25: sa  $\leftarrow$  apply fine-tuned SentenceTransformer encoding on train[StudentAnswer]
26: ra  $\leftarrow$  apply fine-tuned SentenceTransformer encoding on train[ReferenceAnswer]
27:  $X \leftarrow |sa - ra|$ 
28:  $y \leftarrow$  train[label]
29:  $(X, y) \leftarrow$  Apply synthetic sampling technique on  $(X, y)$ 
30: return  $(X, y)$ 

```

---

1 are used here to make the classes partial-balanced. After having the partial-balanced dataset, the student and reference answer pairs are fed to the fine-tuned sentence transformer model. Let  $sa$  and  $ra$  be the embeddings generated by this model that correspond to the student and reference answers respectively. The input feature for the classifier training can be found using the following equation:

$$X = |sa - ra| \quad (3)$$

The resulting embedding  $X$  represents the degree of similarity between  $sa$  and  $ra$ . After generating the features  $X$  for each pair of answers, synthetic oversampling techniques such as Random oversampling, SMOTE, or ADASYN can be applied to balance the classes completely. The target feature for the classifier is the discrete score for each student. The algorithm for the proposed PAO is given in Algorithm 2. The parameter  $\alpha$  used here determines the percentage of prompt-driven samples used for oversampling.

### 3 Result and Analysis

#### 3.1 Datasets

In this study, we use two publicly available datasets called the Mohler dataset [18] and the Short Programming Related Answer Grading (SPRAG) dataset [1]. The Mohler dataset is the earliest ASAG dataset. It consists of 80 questions from 12 assignments (prepared by Mohler et al. from North Texas University). It consists of 2442 student responses. The SPRAG dataset consists of Python programming-related student responses and their corresponding scores. It consists of 144 different questions with a total of 4039 student responses. In these two datasets, two annotators evaluated the student responses independently. Here, the average score is considered for the experimentation. The score range is from 0 to 5.

#### 3.2 Evaluation Metrics

The primary contribution of this work lies in the development of prompt-driven data augmentation. This approach to text augmentation for dataset expansion proves particularly valuable when the augmented text maintains semantic similarity while introducing lexical diversity relative to the reference text. To assess the effectiveness of the proposed augmentation strategy, we introduce a composite metric termed the Diversity-Similarity Trade-off Score (DSTS). The equation for DSTS is provided below.

$$DSTS = \frac{1}{2}(SS + LD) \quad (4)$$

Where the Semantic Similarity ( $SS$ ) is quantified in terms of Cosine similarity (1) and the Lexical Diversity ( $LD$ ) can be computed as  $LD = 1 - LO$ . Where the Jaccard similarity [23] is used to find out Lexical Overlap ( $LO$ ).

The results of the fine-tuned sentence transformer models are evaluated using the Pearson correlation coefficient, Spearman's rank correlation, and the root mean squared error (RMSE). The Pearson correlation coefficient is a measure of the strength of the association between the two variables, the equation for which is given by:

$$r = \frac{\sum_{i=1}^n (x_i - \tilde{x})(y_i - \tilde{y})}{\sqrt{\sum_{i=1}^n (x_i - \tilde{x})^2 (y_i - \tilde{y})^2}} \quad (5)$$

Where  $x_i$  and  $y_i$  denote the  $i^{th}$  values for the distribution of variables X and Y, and  $\tilde{x}$  and  $\tilde{y}$  indicate the mean values of X and Y distribution. Here we can consider X to be the distribution of the actual scores and Y to be the distribution of the scores predicted by the model. Another metric used for this experiment is Spearman's rank correlation coefficient between the cosine similarity of the sentence embeddings and the gold labels. The formula for Spearman's rank correlation coefficient is given below:

$$\rho = 1 - \frac{6 \sum d^2}{N(N^2 - 1)} \quad (6)$$

where  $d$  is the difference between the ranks of corresponding data points, and  $N$  is the number of data points. RMSE is the standard deviation of the prediction error. Its formula is given as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}} \quad (7)$$

where,  $N$  = Number of data points

$y_i$  = Actual similarity value

$\hat{y}_i$  = Predicted similarity value

To evaluate the classifier, Accuracy and F1-score are used as metrics. The following formula can be used to compute the accuracy.

$$Accuracy = \frac{\sum_{i=1}^N f(y_i = \hat{y}_i)}{N} \quad (8)$$

where  $y_i$  is the actual score,  $\hat{y}_i$  is the predicted score,  $N$  is the total number of instances in the training set, and  $f(.)$  is the function that returns ‘1’ if  $(y_i = \hat{y}_i)$  and return ‘0’ otherwise. The main objective of a good classifier model is to obtain high precision and recall. F1-score can be used to express these metrics as a single metric. The following formula can be used to calculate the F1-score.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (9)$$

where *Precision* measures the accuracy of positive predictions. It is calculated as the ratio of true positives to the sum of true positives and false positives. *Recall* measures the proportion of actual positives that are correctly predicted. It is calculated as the ratio of true positives to the sum of true positives and false negatives.

### 3.3 Experimental Setup & Results

This subsection analyses the effectiveness of the prompt-driven augmented dataset on the sentence transformer finetuning and the effect of prompt-adaptive oversampling on classification. Both the datasets used for this study are used to fine-tune the sentence transformer separately. Each fine-tuned sentence transformer learns the domain-specific features of the answers. To analyse the effect of the proposed augmentation strategy on sentence transformer fine-tuning, the results are compared against other augmented datasets and the original dataset. In [2], the authors experimented with the effect of the different augmented datasets on binary grading. Some of those augmentation techniques- back translation, random deletion and synonym replacement -are used for the comparative study. An example of original student answer and augmented answers corresponding to the question ‘*Differentiate exploratory and explanatory analysis?*’ are listed in Table 1.



**Table 1.** Example of an original student answer and the augmented student answer.

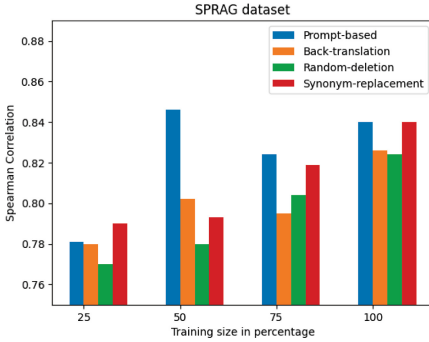
Method	Augmented Answer
Original	Exploratory analysis is done while searching for insights and used to find the answer to many questions. Explanatory analysis is performed to provide the accurate, insightful and visually appealing results to others.
Back-translation	Explorer analysis is performed when searching for insights and used to find many questions. Explanation analysis is to provide accurate, visible and visually attractive results for others.
Random deletion	Exploratory is done while searching for insights and used to find the answer to many questions. Analysis is performed the insightful and visually results to others.
Synonym replacement	explorative analysis is make while searching for insights and used to find the answer to many questions. Explanatory analysis is performed to provide the accurate, insightful and visually appealing results to others.
Prompt-based	Exploratory analysis is the initial step in data analysis where the goal is to understand the data and identify patterns, while explanatory analysis is the final step where the goal is to communicate the findings to others in a clear and concise manner.

The back translation technique [22] uses an existing machine translation algorithm with two steps - Forward translation and Backward translation. Here, we use the Google translation algorithm with the source language as English and the target language as Chinese. So, during forward translation, English-to-Chinese translation occurs and during backward translation, this Chinese text is again translated back to English.

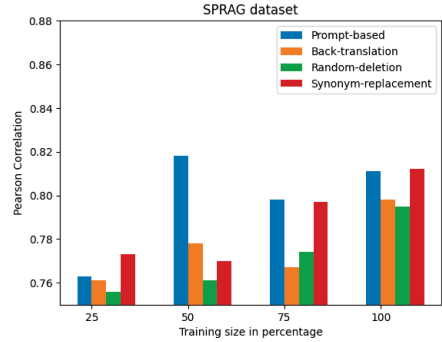
In the random deletion technique [24], randomly deleting  $n$  number of words with a probability  $p$ . Since we augment short text answers, we set  $n$  as 3. So, if the actual answer text consists of a total  $m$  number of tokens, the augmented text contains a maximum of  $m$  tokens and a minimum of  $m - 3$  tokens.

Synonym replacement [24] is an augmentation technique that replaces some random words with their synonyms. The synonyms of each word can be collected from some lexical database. Here we use the help of WordNet [17] for this purpose.

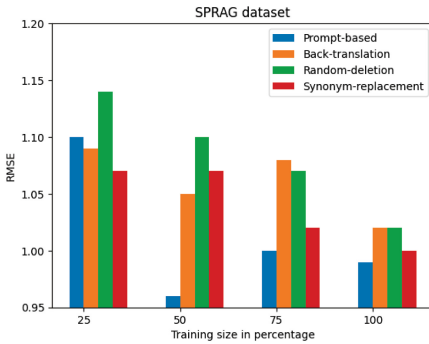
Table 2 analyses the effectiveness of the proposed augmentation strategy on dataset expansion. The metric DSTS in Eq. 4 is used here for the evaluation. The higher value of DSTS is obtained in the case of the proposed augmentation strategy which indicates that the augmented texts using the proposed strategy are more semantically similar as well as more lexically diverse to the reference answer.



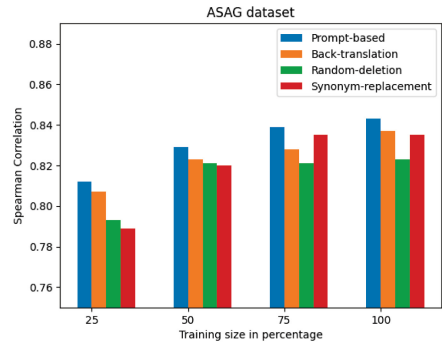
(a) Spearman Correlation on SPRAG



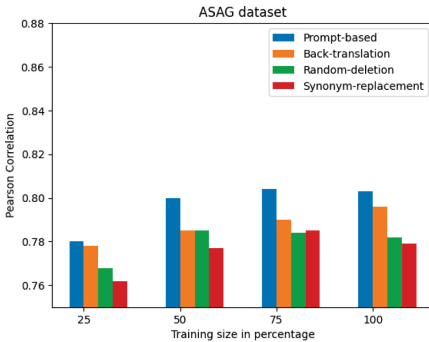
(b) Pearson Correlation on SPRAG



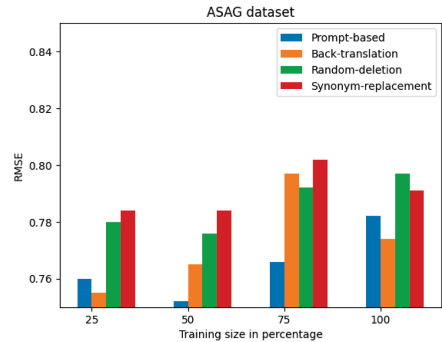
(c) RMSE on SPRAG



(d) Spearman Correlation on ASAG



(e) Pearson Correlation on ASAG



(f) RMSE on ASAG

**Fig. 2.** Comparison of performance of the prompt-based augmentation with other augmentation strategies in different training sizes on sentence transformers fine-tuning.

Since we use the automated augmentation techniques, there is a chance to get good results in low-resource scenarios as well. So every augmented dataset is split into 4 different sizes (25%, 50%, 75%, and 100%) to analyse the result in the low resource scenario.

**Table 2.** Analysis on the effectiveness of prompt-based text augmentation over other techniques.

Text augmentation technique	DSTS on Mohler-ASAG	DSTS on SPRAG
Back-translation	0.785	0.786
Random deletion	0.569	0.542
Synonym replacement	0.627	0.599
<b>Prompt-based</b>	<b>0.887</b>	<b>0.849</b>

The plot of Spearman correlation, Pearson correlation, and RMSE of the different augmentation techniques are shown in Fig. 2. The Fig. 2(a), 2(b), and 2(c) are the results obtained from fine-tuned *all-MiniLM-L6-v2* on SPRAG dataset and the Fig. 2(d), 2(e), and 2(f) are the results obtained from fine-tuned *all-MiniLM-L6-v2* on Mohler-ASAG dataset. The best result is achieved while using 50% of the prompt-based augmented training set in both cases. Table 3 compares the best results of the model trained with the augmented datasets and the performance of the model trained with the original dataset. The best fine-tuned sentence transformer model obtained here is used to embed the student and reference answers during grade classification.

**Table 3.** Comparison between the effect of different augmentation techniques

<i>all-MiniLM-L6-v2 on SPRAG dataset</i>			
Dataset	RMSE	$\rho$	$r$
Original dataset	1.00	0.799	0.793
Back-translation based augmented dataset	1.02	0.826	0.798
Random deletion based augmented dataset	1.02	0.824	0.795
Synonym replacement based augmented dataset	1.00	0.840	0.812
<b>Prompt based augmented dataset</b>	<b>0.96</b>	<b>0.846</b>	<b>0.818</b>
<i>all-MiniLM-L6-v2 on Mohler-ASAG dataset</i>			
Original dataset	0.837	0.659	0.702
Back-translation based augmented dataset	0.765	0.837	0.796
Random deletion based augmented dataset	0.776	0.823	0.785
Synonym replacement based augmented dataset	0.784	0.835	0.779
<b>Prompt based augmented dataset</b>	<b>0.752</b>	<b>0.843</b>	<b>0.804</b>

Both the datasets used for this work consist of continuous scores. Hence, these scores must be converted to discrete score labels before classification. The number of labels we choose for classification can significantly affect the classification model’s performance and interpretability. The higher number of score

**Table 4.** Comparison of performance of classifier with different oversampling techniques on SPRAG dataset

<i>6-way classification</i>						
Oversampling technique	SVM		KNN		XGBoost	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
No oversampling	0.582	0.535	0.556	0.501	0.562	0.509
RO	0.582	0.529	0.566	0.506	0.563	0.509
SMOTE	0.582	0.535	0.566	0.512	0.563	0.515
ADASYN	0.583	0.535	0.563	0.513	0.562	0.509
PAO with RO	0.580	0.528	0.567	0.506	0.564	0.522
<b>PAO with SMOTE</b>	<b>0.585</b>	<b>0.542</b>	<b>0.579</b>	<b>0.529</b>	<b>0.576</b>	<b>0.539</b>
PAO with ADASYN	0.583	0.540	0.567	0.513	0.566	0.520
<i>3-way classification</i>						
<b>No oversampling</b>	<b>0.759</b>	<b>0.732</b>	<b>0.720</b>	<b>0.711</b>	0.710	0.704
<b>RO</b>	0.756	0.729	0.714	0.654	<b>0.743</b>	<b>0.713</b>
SMOTE	0.754	0.727	0.736	0.699	0.735	0.705
ADASYN	0.755	0.729	0.730	0.703	0.741	0.715
PAO with RO	0.755	0.725	0.713	0.649	0.739	0.709
<b>PAO with SMOTE</b>	0.755	0.729	0.734	0.699	<b>0.745</b>	<b>0.713</b>
PAO with ADASYN	0.757	0.731	0.728	0.696	0.736	0.705
<i>2-way classification</i>						
No oversampling	0.848	0.822	0.828	0.819	0.844	0.817
RO	0.849	0.824	0.819	0.806	0.845	0.820
SMOTE	0.853	0.829	0.845	0.823	0.845	0.823
<b>ADASYN</b>	<b>0.854</b>	0.832	0.846	0.827	<b>0.849</b>	<b>0.827</b>
PAO with RO	0.849	0.824	0.817	0.769	0.845	0.820
<b>PAO with SMOTE</b>	<b>0.854</b>	<b>0.832</b>	<b>0.852</b>	<b>0.829</b>	0.845	0.823
<b>PAO with ADASYN</b>	<b>0.854</b>	<b>0.832</b>	0.847	0.827	<b>0.849</b>	<b>0.827</b>

labels provides the highest granularity in distinguishing between different performance levels. However, it requires more data to model each class effectively. Here, the experiments are done as 6-way, 3-way and 2-way grade classification using Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and XGBoost (Extreme Gradient Boosting) classifiers. Here, both SVM and KNN are the traditional machine-learning models and XGBoost is one of the ensemble learning models. KNN is a simple model that performs well even when the dataset size is very small. SVM can effectively learn the data with high dimensions and it is more robust to outliers. XGBoost works by combining multiple decision trees. The deep-learning models are not suitable in the case of low-resource scenarios. So, the above three models are used for further analysis. The following seven combinations of the training data are used here for analysis in the case of both datasets.

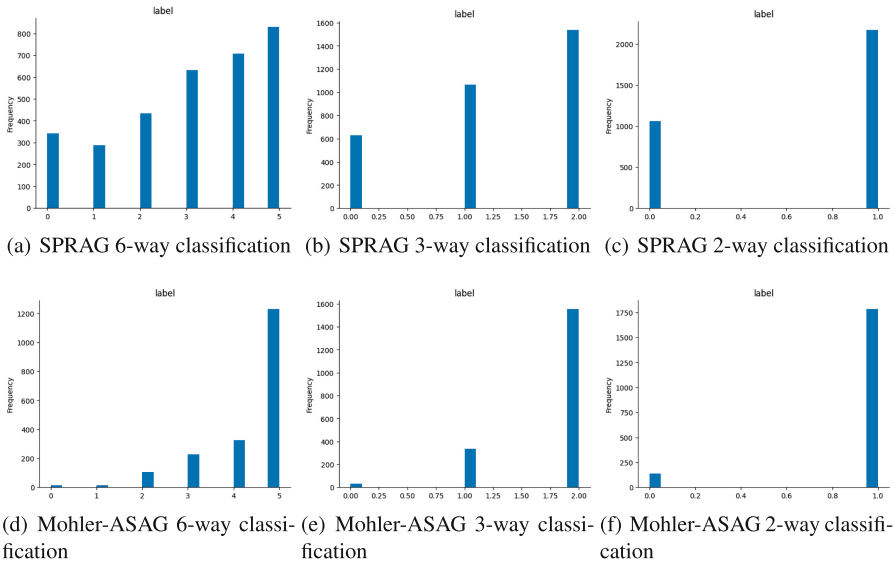
**Table 5.** Comparison of performance of classifier with different oversampling techniques on ASAG dataset

<i>6-way classification</i>						
Oversampling technique	SVM		KNN		XGBoost	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
No oversampling	0.717	0.479	0.729	0.494	0.722	0.493
RO	0.722	0.498	0.733	0.531	0.724	0.488
<b>SMOTE</b>	0.722	0.500	<b>0.740</b>	<b>0.552</b>	0.720	0.532
ADASYN	0.717	0.497	0.729	0.537	0.722	0.533
<b>PAO with RO</b>	0.717	0.496	0.736	0.535	<b>0.726</b>	0.538
<b>PAO with SMOTE</b>	<b>0.724</b>	<b>0.502</b>	<b>0.740</b>	0.545	0.724	<b>0.553</b>
PAO with ADASYN	–	–	–	–	–	–
<i>3-way classification</i>						
No oversampling	0.823	0.620	0.828	0.622	0.830	0.657
RO	0.825	0.654	0.828	0.679	0.823	0.627
<b>SMOTE</b>	0.825	0.654	<b>0.830</b>	<b>0.700</b>	<b>0.832</b>	<b>0.697</b>
ADASYN	0.825	0.654	0.821	0.677	0.821	0.666
<b>PAO with RO</b>	<b>0.830</b>	<b>0.660</b>	0.830	0.651	0.825	0.683
<b>PAO with SMOTE</b>	<b>0.830</b>	<b>0.660</b>	<b>0.830</b>	<b>0.700</b>	<b>0.832</b>	<b>0.697</b>
PAO with ADASYN	–	–	–	–	–	–
<i>2-way classification</i>						
No oversampling	0.936	0.714	0.926	0.647	0.928	0.665
RO	0.933	0.698	0.933	0.728	0.931	0.693
<b>SMOTE</b>	<b>0.938</b>	<b>0.729</b>	0.931	0.723	<b>0.943</b>	<b>0.757</b>
<b>ADASYN</b>	<b>0.938</b>	<b>0.729</b>	0.933	0.736	0.931	0.704
PAO with RO	0.936	0.714	0.933	0.728	0.933	0.698
<b>PAO with SMOTE</b>	<b>0.938</b>	<b>0.729</b>	<b>0.936</b>	<b>0.741</b>	0.940	0.743
<b>PAO with ADASYN</b>	<b>0.938</b>	<b>0.729</b>	0.933	0.736	0.938	0.738

- No oversampling: This is the original training set without doing any oversampling.
- RO: The training set where the random oversampling is applied.
- SMOTE: The training set where the SMOTE is applied.
- ADASYN: The training set where the ADASYN oversampling is applied.
- PAO with RO: The training set where PAO with random oversampling is applied.
- PAO with SMOTE: The training set where PAO with SMOTE is applied.
- PAO with ADASYN: The training set where PAO with ADASYN sampling is applied.

Table 4 and Table 5 show the results obtained while classifying the SPRAG test set grades and the ASAG test set grades respectively. Both these dataset’s scores are in the range of (0,5). Hence while doing 6-way classification, the scores are capped to 0, 1, 2, 3, 4, and 5. In the case of 3-way classification, the scores less

than 1.5 are capped as 0, the scores between 1.5 and 2.5 are converted as 1, and the scores greater than 2.5 are considered as 2. In 2-way classification, the scores less than 2.5 are capped as 0, and others are capped as 1. All the experiments are done using Google Colab T4 GPU with system RAM of 51 GB, GPU RAM of 15 GB, and Disk Space of 201.2 GB. Table 4 shows the performance of the classifiers in SPRAG dataset. The proposed methodology performs better in the 2-way and 6-way classification. 3-way classification is performing well without any oversampling. While analysing the training set distribution of these three classification data (Fig. 3), the majority of classes in the 3-way classification contain the number of instances greater than the average number of class instances. In the case of 6-way classification PAO with SMOTE is performed well in all classifiers. However, in the case of 2-way classification, PAO with SMOTE performed well in SVM and KNN, and PAO with ADASYN performed well in XGBoost.



**Fig. 3.** Class distribution of both SPRAG and Mohler-ASAG datasets

Table 5 demonstrates the experimental results of classifiers on the Mohler-ASAG dataset. In the majority of cases, PAO with SMOTE is performing well. Although the SMOTE is better performing in XGBoost 2-way classification, PAO with SMOTE achieves comparable performance. However, PAO with ADASYN failed in the case of 6-way and 3-way classification. This is because of the working principle of ADASYN. ADASYN failure generally occurs when the minority class samples are highly isolated in the feature space. This algorithm checks the number of majority class neighbours to find out the amount and the

direction of synthetic samples that have to be generated. Hence, ADASYN is not working for every dataset distribution.

**Table 6.** Best  $\alpha$  obtained after tuning

Oversampling technique	SPRAG			Mohler-ASAG		
	SVM	KNN	XGBoost	SVM	KNN	XGBoost
<i>6-way classification</i>						
PAO with RO	0.6	0.6	0.6	0.7	0.7	0.7
PAO with SMOTE	0.6	0.7	0.6	0.1	0.8	0.7
PAO with ADASYN	0.6	0.6	0.6	—	—	—
<i>3-way classification</i>						
PAO with RO	0.5	0.5	0.5	0.6	0.6	0.6
PAO with SMOTE	0.5	0.5	0.5	0.6	0.1	0.6
PAO with ADASYN	0.5	0.5	0.5	—	—	—
<i>2-way classification</i>						
PAO with RO	0.5	0.5	0.5	0.1	0.1	0.1
PAO with SMOTE	0.8	0.5	0.5	0.1	0.1	0.1
PAO with ADASYN	0.5	0.5	0.5	0.1	0.1	0.1

While executing PAO, the hyper-parameter  $\alpha$  used in Algorithm 2 is tuned manually to get better results in all classifiers. The value of  $\alpha$  ranges from 0 to 1. The experiments were conducted by varying the value of  $\alpha$  in 0.1 intervals. The best values of  $\alpha$  obtained in all cases are listed in Table 6. The experimental results of PAO in Table 4 and Table 5 are obtained by setting these tuned  $\alpha$  values.

## 4 Conclusion and Discussion

This work proposes a new prompt-driven dataset augmentation technique and prompt-adaptive oversampling technique to improve the performance of short answer grade classification. The performance of the proposed prompt-driven augmentation is analysed with other traditional text augmentation methods, with the DSTS metric showing superior results, indicating higher semantic similarity and greater lexical diversity compared to other approaches.

By experimenting with various sizes of augmented training sets, we found that the sentence transformer fine-tuned using a 50% prompt-driven augmented dataset generates better embeddings. But it need not be always 50%. This will change according to the dataset properties like the number of instances, dataset distributions, etc.

Additionally, we introduce PAO to improve grade classification accuracy. The performance of the proposed methodology is analysed by training three different

classifiers- SVM, KNN, and XGBoost. In most of the cases, PAO with SMOTE is performing well. In a few cases, other synthetic oversampling techniques like RO, SMOTE, and ADASYN perform better. However, in such cases, PAO with SMOTE still yields comparable performance with RO, SMOTE, and ADASYN. The oversampling techniques are not that much of useful in the cases where the size of the majority classes exceeds the average size of one class. The parameter  $\alpha$  used in the PAO algorithm is tuned manually to achieve optimal results. In the future, an automated method for parameter tuning can be proposed.

The experimental analysis was conducted on two publicly available datasets, Mohler ASAG and SPRAG, both within the computer science domain. To evaluate the generalizability of the proposed method, future research could expand the analysis to include datasets from diverse domains.

## References

1. Bonthu, S., Sree, S.R., Prasad, M.K.: SPRAG: building and benchmarking a short programming related answer grading dataset (2022)
2. Bonthu, S., Sree, S.R., Prasad, M.K.: Improving the performance of automatic short answer grading using transfer learning and augmentation. *Eng. Appl. Artif. Intell.* **123**, 106292 (2023)
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)
4. Chen, Y., Luo, J., Zhu, X., Wu, H., Yuan, S.: A cross-lingual hybrid neural network with interaction enhancement for grading short-answer texts. *IEEE Access* **11**, 37508–37514 (2023)
5. Del Gobbo, E., Guarino, A., Cafarelli, B., Grilli, L.: GradeAid: a framework for automatic short answers grading in educational contextsdesign, implementation and evaluation. *Knowl. Inf. Syst.* **65**, 4295–4334 (2023)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)* (2018)
7. Gomaa, W.H., Nagib, A.E., Saeed, M.M., Algarni, A., Nabil, E.: Empowering short answer grading: Integrating transformer-based embeddings and bi-LSTM network. *Big Data Cognit. Comput.* **7**(3), 122 (2023)
8. Haller, S., Aldea, A., Seifert, C., Strisciuglio, N.: Survey on automated short answer grading with deep learning: from word embeddings to transformers. *arXiv preprint [arXiv:2204.03503](https://arxiv.org/abs/2204.03503)* (2022)
9. He, A., Luo, C., Tian, X., Zeng, W.: A twofold Siamese network for real-time object tracking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4834–4843 (2018)
10. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1322–1328. IEEE (2008)
11. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint [arXiv:1508.01991](https://arxiv.org/abs/1508.01991)* (2015)
12. Jiang, A.Q., et al.: Mistral 7b. *arXiv preprint [arXiv:2310.06825](https://arxiv.org/abs/2310.06825)* (2023)



13. Kumar, S., Chakrabarti, S., Roy, S.: Earth mover's distance pooling over Siamese LSTMs for automatic short answer grading. In: IJCAI, pp. 2046–2052 (2017)
14. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **55**(9), 1–35 (2023)
15. Liu, S.y., Liu, Z., Huang, X., Dong, P., Cheng, K.T.: LLM-FP4: 4-bit floating-point quantized transformers. arXiv preprint [arXiv:2310.16836](https://arxiv.org/abs/2310.16836) (2023)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint [arXiv:1301.3781](https://arxiv.org/abs/1301.3781) (2013)
17. Miller, G.A.: WordNet: a lexical database for English. *Commun. ACM* **38**(11), 39–41 (1995)
18. Mohler, M., Bunescu, R., Mihalcea, R.: Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 752–762 (2011)
19. Prabhudesai, A., Duong, T.N.: Automatic short answer grading using Siamese bidirectional LSTM based regression. In: 2019 IEEE International Conference on Engineering, Technology and Education (TALE), pp. 1–6. IEEE (2019)
20. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. arXiv preprint [arXiv:1908.10084](https://arxiv.org/abs/1908.10084) (2019)
21. 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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics (2019)
22. Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, pp. 86–96. Association for Computational Linguistics, 7–12 August 2016
23. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to data mining (2006)
24. Wei, J., Zou, K.: EDA: easy data augmentation techniques for boosting performance on text classification tasks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6382–6388 (2019)



# SANS: Spatial-Aware Neural Solver for Plane Geometry Problem

Zi-Hao Lin<sup>1,2</sup>, Shun-Xin Xiao<sup>1,2</sup>(✉), Zi-Rong Chen<sup>1,2</sup>, Jian-Min Li<sup>1,2</sup>,  
Da-Han Wang<sup>1,2</sup>, and Xu-Yao Zhang<sup>3</sup>

<sup>1</sup> School of Computer and Information Engineering, Xiamen University of  
Technology, Xiamen 361024, China  
giannis@foxmail.com, xiaoshunxin.tj@gmail.com, {lijm,wangdh}@xmut.edu.cn

<sup>2</sup> Fujian Key Laboratory of Pattern Recognition and Image Understanding,  
Xiamen 361024, China

<sup>3</sup> State Key Laboratory of Multimodal Artificial Intelligence Systems,  
Institute of Automation of Chinese Academy of Sciences, Beijing 100190, China  
xyz@nlpr.ia.ac.cn

**Abstract.** Geometry problem solving (GPS) is an important research direction in artificial intelligence. Previous studies have demonstrated the effectiveness of neural solvers in GPS. However, they are deficiencies in accurately representing spatial relationships of geometric primitives within visually rich geometric diagrams. This paper presents a novel neural solver termed spatial-aware neural solver (SANS) that can perceive spatial relationships between geometric primitives. SANS includes two new modules: multimodal dual-branch spatial awareness pre-trained language module and point-primitive spatial-aware attention module. The pre-training module employs a dual-branch visual-textual point-matching strategy to align visual and textual points, and utilizes semantic structure pre-training to model global relationships. Additionally, the point-primitive spatial awareness attention module enhances the model's ability to perceive spatial relationships between geometric primitives by accounting for the relative positions of points. Experiments show that SANS achieves 81.5 and 74.1 of accuracy on the Geometry3K and PGPS9K datasets.

**Keywords:** Geometry problem solving · Multimodal · Pre-training · Neural solver · Attention

## 1 Introduction

The task of automatic geometry problem solving (GPS) has long been a challenging endeavor in AI. Recently, it has garnered significant attention from both the computer vision and natural language processing communities [17, 22, 29]. This challenge is notable not only due to its academic significance, but also because of its playing a crucial role in educational contexts [3]. A geometric

problem encompasses both a textual description and its corresponding geometric diagram. The textual component outlines the conditions and objectives of the problem, while the diagram offers structural and semantic insights that the textual part cannot provide. The effective integration of geometric images and text is key to enhancing understanding of geometric problems.

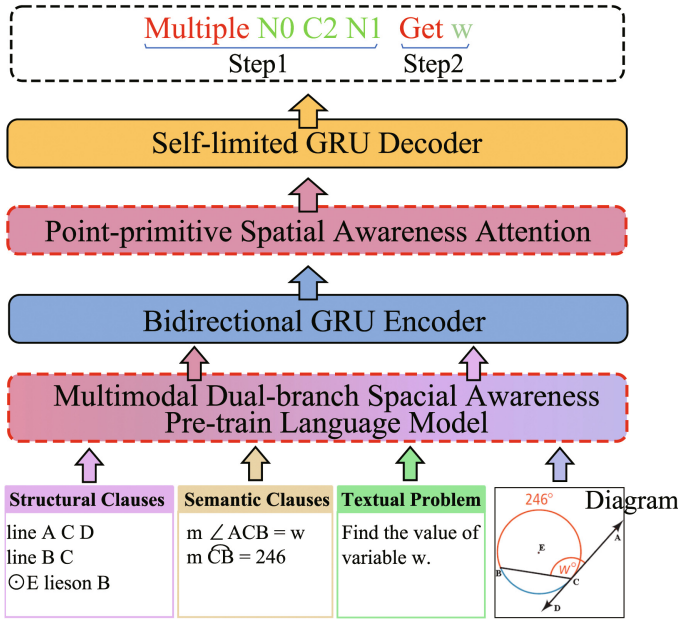


Fig. 1. Overview of spatial-aware neural solver.

Due to the differences between geometric images and natural scene images, simple image feature extraction fails to capture the unique structural information of geometric images. Conversely, the methods adopted by Lu et al. [17] and Zhang et al. [30], which are based on describing images with textual descriptions, yield superior reasoning results. In detail, textual clauses can be categorized into structural clauses, semantic clauses, and textual problems. Among them, structural clauses describe the structural relationships between geometric primitives. For example, “line A C D” describes a structural relationship that points “A”, “C” and “D” lie on the same line in order. Semantic clauses describe the semantic relationship between textual and geometric primitives. For example, “ $m\angle ACB = w$ ” illustrates the semantic relationship between the “ $\angle ACB$ ” and the textual “ $w$ ”. Textual problems describe the solving objectives of geometric problems. Geometric diagrams provide unique spatial relationships among geometric primitives that textual clauses cannot provide. For example, the semantic clause describes “ $\angle ACB$ ”, which angle is “ $w$ ”. However, it does not specify whether “ $\angle ACB$ ” is obtuse or acute, which needs the geometric primitive spatial relationship to be determined.

We propose a neural solver called spatial-aware neural solver (SANS) to solve the problems mentioned above. The overall architecture of the proposed SANS model is shown in Fig. 1. It contains two submodules: multimodal dual-branch spatial awareness pre-trained language module (MDSP) and point-primitive spatial awareness attention (PP-SAA). They improve the model’s ability to capture spatial relationships across diverse geometric diagrams. In the pre-training phase, our model employs a contrastive learning-based dual-branch visual-textual points matching (DB-VTPM) strategy. This approach achieves fine-grained matching between geometric images and textual texts and optimizes feature extraction for images and texts. Combining DB-VTPM with the structural semantic pretraining (SSP) [30], which enables the multimodal pre-training module to comprehend text clauses and gain a preliminary understanding of spatial relations of geometric primitives. During the model training phase, we design a PP-SAA by introducing point-primitive spatial-aware attention that enhances the model’s ability to perceive spatial relationships between geometric primitives.

The contributions of this work are summarized in four aspects: (1) We propose a spatial-aware neural solver for GPS, which can represent and fuse geometric diagrams effectively. (2) We introduce the MDSP module, achieving optimized feature representations for geometric images and geometric text and facilitating cross-modal alignment. (3) We propose PP-SAA to enhance the model’s spatial perception capabilities. (4) Our SANS significantly outperforms existing symbolic solvers and neural solvers on the Geometry3K and PGPS9K datasets.

## 2 Related Work

### 2.1 Geometry Problem Solving

GPS is a long-standing and challenging mathematical reasoning task [26]. In existing research, geometry problem solvers can be classified into symbolic solvers and neural geometric solvers. For existing symbolic solvers [17, 21, 22], the main-stream approach is to parse geometric illustrations and problem texts into formal languages, followed by symbolic reasoning based on manually defined complex geometric axioms. However, this leads to the complex design of symbolic solvers, making them difficult to apply in practical geometry problem solving. The first neural geometric solver, proposed by [6], utilizes multiple auxiliary tasks to address the semantic gap between geometric images and text. In recent years, the neural solver PGPSNet [30] has adopted structural semantic pretraining (SSP), data augmentation, and self-restricted decoding to fuse multimodal information. These methods did not consider the spatial relationships between geometric primitives in geometric diagrams. In contrast, we propose a neural solver that can perceive the spatial relationships of geometric primitives, thereby improving the joint understanding of diagram images and texts.

## 2.2 Multimodal Reasoning

Multimodal reasoning refers to the process of reasoning using data from multiple modalities (e.g., images, text, etc.). In multimodal reasoning, it is essential to effectively utilize information from different modalities to address a variety of complex tasks, such as visual question answering [2, 13] and visually-rich document understanding [11, 27, 28], among others. One of the critical challenges in multimodal reasoning is how to effectively understand and integrate information from different modalities, which involves aspects like learning representations of cross-modal features, aligning and integrating modalities, and applying domain knowledge [8]. GPS can be regarded as a specialized multimodal reasoning problem [30], where the unique characteristics of its dataset make understanding internal features within each modality and the fusion of cross-modal information crucial for GPS.

## 2.3 Multimodal Pre-training

Compared with single-modality pre-training, multimodal pre-training effectively addresses issues such as modality completion and cross-modal alignment and can compensate for the limitations of a single modality. Therefore, multimodal pre-training helps extract common features across modalities [23]. To achieve cross-modal alignment, understanding, and fusion, researchers can design appropriate pre-training objectives for multimodal pre-training, such as contrastive loss [12, 14, 20], image-text matching [13, 25], and others. Although these strategies show excellent performance on natural scene images, the application to GPS tasks directly is hindered by the specificity and small scale of GPS datasets. In existing GPS works, the structural-semantic pre-training strategy proposed in PGPSNet [30] achieves excellent performance by specifically modeling geometric text. However, due to it adopting single modality pre-training, it would overlook the information provided by the visual modality. Therefore, our proposed multimodal dual-branch spatial awareness pre-trained language model (MDSP) effectively improves both single modality expression and multimodal fusion in GPS while endowing the model with the ability to perceive spatial relationships among geometric primitives.

## 3 Method

Before introducing the neural solver model, we first define the GPS task. The GPS task is formalized as providing a geometric problem  $P = [PD, PT]$ , where PD represents geometric diagram images, and PT represents textual clauses. By learning and applying geometric knowledge through the model, a solution program is generated, and numerical results for geometric problems are obtained through program execution on calculators.

### 3.1 Overall Framework

To better fuse features from geometric diagrams and textual clauses, we propose a spatial-aware neural solver (SANS). For data processing, we adopt the geometric image parser [29] to parse geometric diagrams into textual clauses, including structural clauses and semantic clauses. After that, image embedding and text embedding are concatenated as tokens together. Then, these modal tokens are fed into the Multimodal Dual-Branch Spatial Awareness Pretraining (MDSP) and processed by the bidirectional GRU encoder to perform fusion encoding. Subsequently, we take the corresponding part of the text  $H_T = \{h_i^t\}_{i=1}^L$  from context encoding  $H$ . The text encoding  $H_T$  is fed into the point-primitive spatial awareness attention (PP-SAA) module to further boost the spatial relationship of geometric primitive awareness. Finally,  $H$  is input into a self-limited GRU decoder for decoding, and a sequence solution program is generated in an autoregressive manner [30]. The proposed MDSP and PP-SAA modules will be detailed in the following sections.

### 3.2 Image and Text Embedding

**Visual Embedding.** To accelerate model convergence, we put the geometric image  $PD$  into the CNN encoder to quickly extract coarse-grained global visual features of the diagram. Next, the feature map is divided into  $N$  patches via linear projection. The image could be further expressed as  $D = \{d_i\}_{i=1}^N$ , and the visual feature could be expressed  $F_{PD}$ , where  $N$  is a number of diagram patches. To enable the subsequent transformer model to better process the image features, we incorporate 1D positional embeddings into the image features extracted by the backbone, compensating for the CNN’s inability to capture positional information. The visual encoding can be represented as:

$$e_i^{pd} = PatchEmb(d_i) + PosEmb(i), 1 \leq i \leq N \quad (1)$$

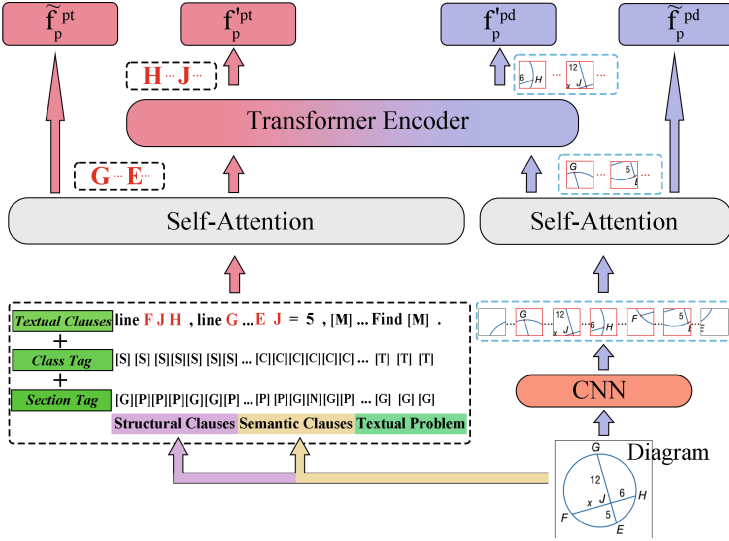
$d_i$  represents the feature extracted from each image patch through CNN, and  $PosEmb(*)$  represents the positional embedding of image patches. For subsequent pre-training tasks, we use a transformer to encode the  $e_i^{pd}$  to  $\tilde{F}_{PD} = \{\tilde{f}_i^{pd}\}_{i=1}^N$ .

**Textual Token Embedding.** To represent geometric text with fine-grained, textual token embeddings  $e_j^{pt}$  fuse not only positional encoding but also the embeddings of class tags and section tags [30]. In detail, textual token embeddings are formulated as:

$$e_j^{pt} = TokenEmb(pt_j) + PosEmb(i) + ClassEmb(pt_j) + SectEmb(pt_j), \quad 1 \leq j \leq L \quad (2)$$

where  $L$  is the maximum length of the problem text. We feed the textual token embeddings into a transformer encoder [24] to obtain the textual feature named

$F_{PT}$  by learning the intrinsic relationships between textual tokens simultaneously. Similar to the processing of visual embeddings,  $F_{PT}$  is linearly mapped to  $\tilde{F}_{PT} = \{\tilde{f}_j^{pt}\}_{j=1}^L$ .



**Fig. 2.** Pipeline of multimodal dual-branch visual-textual points matching pre-training. The red dot text and image patches denote the matched points, and the image patch shows where the points are located.  $f_p^{*}$  represents the output from inter-modality, while  $\tilde{f}_p^*$  represents the output from intra-modality. [M] denotes the mask marker. (Color figure online)

### 3.3 Multimodal Dual-Branch Spatial Awareness Pre-training Language Module

Geometric problems are often solved by humans through reasoning about spatial relationships using visual and textual information. Previous neural geometry solvers [6, 18, 30] have not fully utilized the spatial relationships of geometric primitives, resulting in suboptimal performance. We propose a multimodal dual-branch spatial awareness pre-training language module (MDSP), which is illustrated in Fig. 2, with two pre-training strategies: structural semantic pre-training (SSP) and dual-branch visual-textual point matching (DB-VTPM).

**Structural Semantic Pre-training.** While textual clauses describe the fine-grained structural and semantic information, these clauses lack overall structure and context. To enable the multimodal pre-training module to comprehend text clauses, we adopt the structural semantic pre-training (SSP) [30]. This strategy

aims to recover the masked text in a unified text generation manner. The training loss is denoted as  $L_{SSP}$ . Specifically, we maintain the masking of text tokens  $p_i^t$ , with only 30% being masked, following the approach outlined in [7].

**Dual-Branch Visual-Textual Point Matching (DB-VTPM).** To address the semantic feature differences between geometric diagrams and texts, we propose the DB-VTPM strategy. This strategy matches visual and textual points intra and inter modality. It aims to enhance cross-modal learning by integrating information from different modalities, allowing the model to gain an initial insight into the spatial relationships of geometric primitives in the geometric diagrams described by the text clauses.

Specifically, we obtain the patch containing the visual point by using the coordinates of the point primitives in the geometric image. Then, we combine the visual features  $F_{PD}$  and textual features  $F_{PT}$ . Finally, those feature maps will be input into a transformer encoder [24] for cross-modal learning to obtain the fused textual features  $F'_{PT}$  with integrated visual features and the fused visual features  $F'_{PD}$  with integrated textual features. We utilize contrastive learning loss [9, 14, 20] to force the textual points and the image patches containing those points to be more similar in the same semantic space, achieving the matching between textual points and visual points. The training loss is represented as:

$$L_{intra} = -\frac{\exp(s(\tilde{f}_p^{pt}, \tilde{f}_p^{pd})/\tau)}{\sum_{i=1}^N \exp(s(\tilde{f}_j^{pt}, \tilde{f}_i^{pd})/\tau)} \quad (3)$$

$$L_{inter} = -\frac{\exp(s(f_p'^{pt}, f_p'^{pd})/\tau)}{\sum_{i=1}^N \exp(s(f_j'^{pt}, f_i'^{pd})/\tau)} \quad (4)$$

where  $f_p^{pt}$  represents the features of textual points, and  $f_p^{pd}$  represents the features of the image patches containing those points. The function  $s(*, *) = W_{PT}(f_{pt})^T W_{PD}(f_{pd})$ , where  $W_{PT}$  and  $W_{PD}$  are linear projection layers, and  $\tau$  is a temperature coefficient. The total training loss for this task is represented as  $L_{VTPM} = L_{intra} + L_{inter}$ , where  $L_{intra}$  represents intra-modality and  $L_{inter}$  represents inter-modality.

Overall, our pre-training module adopts a multi-task learning approach, combining SSP with the DB-VTPM strategy. The SSP task effectively models the global context by comprehending the semantics and context of the text. The DB-VTPM strategy is used to calculate semantic similarity among modalities, allowing for better semantic alignment between the image and text modalities in the same semantic space. Moreover, it provides initial learning of the spatial relationships of geometric primitives. The pre-training loss is defined as  $L_{all} = \alpha \times L_{SSP} + \beta \times L_{VTPM}$ , with  $\alpha$  and  $\beta$  serving as hyperparameters.

### 3.4 Point-Primitive Spatial Awareness Attention

While pre-trained models hold valuable prior knowledge that allows for a preliminary understanding of the spatial relationships between geometric primitives,



this capability may diminish during downstream training due to the varied training objectives of GPS. To bolster the model’s perception of these relationships, we design a point-primitive spatial awareness attention (PP-SAA) mechanism. This module will be positioned between the bidirectional GRU encoder and the self-restricted GRU decoder.

To implement PP-SAA, we design a relative distance table to calculate the attention score between the relative distance and point symbols. By analyzing the coordinates of the point primitives depicted in the diagram images, we can calculate their relative distances in 2D space. These distances offer valuable insight into the positions of the geometric primitives in relation to one another. For instance, if the relative distance between point symbol “A” and point symbol “B” on the x-axis is positive, it indicates that point symbol “B” is located to the right of point symbol “A”. Similarly, the y-axis follows the same principle. By utilizing this method, SANS can gain a heightened perception of the spatial relationship between the geometric primitives. In particular, we employ the original attention mechanism to capture the correlation between text  $i$  and text  $j$ :

$$a_{ij} = \frac{Q(h_i^t)K(h_j^t)}{\sqrt{d_{head}}} \quad (5)$$

where  $Q(*)$  and  $K(*)$  are the query matrix and key matrix, respectively. Considering the large range of values for the relative positions, we incorporate the relative positions of point primitives in space as bias terms to prevent adding too many parameters. Similar approach has been demonstrated to be effective in text-only Transformer architectures [4, 27]. The relative distance bias of the point text on the x-axis and y-axis is represented by  $b(x)$  and  $b(y)$ . The attention score is further expressed as

$$\hat{a}_{ij} = a_{ij} + b_{p_i-p_j}(x) + b_{p_i-p_j}(y) \quad (6)$$

where  $p_i$  and  $p_j$  denote the point symbols in the text. Finally, the attention output is represented as:

$$h_i = \sum_j \text{softmax}\left(\frac{\hat{a}_{ij}}{\sum_k \hat{a}_{ik}}\right)V(h_j^t) \quad (7)$$

where  $V(*)$  denotes the value matrix. After the PP-SAA module, the model is made to enhance the understanding of the spatial relationship between the geometric primitives.

## 4 Experiments

### 4.1 Datasets and Implementation Details

**Datasets.** Due to the scarcity of datasets in the GPS field, we utilize two widely used datasets: Geomety3K and PGPS9K to evaluate the effectiveness of

the proposed SANS. These datasets comprise geometric content from American grades 6–12, covering 30 different types of problems. Geometry3K contains 8,433 training samples and 589 test samples, while PGPS9K has 8,022 training samples and 1,000 test samples distributed evenly across different problem types. This paper utilizes chart annotations to structure the textual clauses, textual semantic clauses, and point coordinates. The solution programs are composed of multiple steps, each containing an operator and relevant operands. These operators correspond to geometric theorems, while operands are arranged according to theorem formulas. Due to the smaller size of GPS datasets compared to natural language corpora, the pretraining process is conducted on the PGPS9K dataset.

**Data Augmentation.** To enrich the diversity of geometric problems, we performed data augmentation on text and diagram separately. For the text, we adopted four augmentation strategies from [30]: token replacement, connection rotation, representation transposition, and clause shuffle. The diagrams are flipped randomly, and change the position of the point in the text description accordingly.

**Metrics.** Consistent with PGPSNet [30], we adopt three metrics, namely Completion, Choice, and Top-3, to evaluate the numerical performance of our model. Specifically, in Completion, the neural solver selects the first executable solution program as the completion result. In Choice, the process involves selecting the correct option from four candidates, but one is randomly chosen if the answer is not among them. Regarding the Top-3 metric, it is considered correct if the solution lies within the top three high-confidence solutions. We using the Choice selection as the evaluation metric for the ablation study in Sect. 4.3.

**Implementation Details.** For geometric diagrams, we scale the image to 256 on the longest side and center it on a blank  $256 \times 256$  screen. We utilize ResNet10 [10] as our visual backbone to extract image features, which are then mapped to 64 image patches using linear mapping. Our pre-training module adopts the transformer [24] with 6-layers, 8-heads, 256-inputs, and 1024-hidden dimensions. For the PP-SAA module, we use a 1-layer transformer encoder with the same number of heads and feature dimensions. In terms of hyperparameters, our configuration remains consistent with PGPSNet [30] to ensure the fairness of the experiments. In the training stage, we employ the AdamW optimizer [16] with a weight decay of  $1 \times 10^{-2}$  and a decay rate of 0.5 for step-down scheduling. The training batch size is set to 128, and the learning rate is initialized to  $1e^{-4}$ , with an initial learning rate of  $1e^{-3}$  for the pre-training module.

**Comparing with State-of-the-Arts Methods.** To evaluate the performance of SANS, we conduct a comparative analysis with several state-of-the-art GPS solvers, including symbolic solvers like Inter-GPS [17] and GeoDRL [19], as well as neural solvers like NGS [6], Geoformer [5], SCA-GPS [18], and PGPSNet

**Table 1.** Performance comparison among excellent GPS solvers. The first rows indicate the performance solved by humans. The bolded portion indicates optimal performance.

Model	Geometry3K			PGPS9K		
	Completion	Choice	Top-3	Completion	Choice	Top-3
Human Expert [17]	–	90.9	–	–	–	–
Baseline [17]	–	35.9	–	–	–	–
InterGPS (Predict) [17]	44.6	56.9	–	–	–	–
InterGPS (Diagram GT) [17]	64.2	71.7	–	59.8	68.0	–
InterGPS (All GT) [17]	<b>69.0</b>	75.9	–	–	–	–
GeoDRL (Predict) [19]	–	68.4	–	–	–	–
NGS [6]	35.3	58.8	62.0	34.1	46.1	60.9
Geoformer [5]	36.8	59.3	62.5	35.6	47.3	62.3
SCA-GPS [18]	–	76.7	–	–	–	–
PGPSNet [30]	65.0	77.9	80.7	62.7	70.4	79.5
LLaVA-v1.5 [15]	7.6	11.2	–	6.3	9.1	–
GPT-4V [1]	38.7	41.4	–	30.2	35.7	–
SANS (ours)	68.8	<b>81.5</b>	<b>83.5</b>	<b>66.4</b>	<b>74.1</b>	<b>82.5</b>

“Predict” (the formal language of input images and text is predicted by its parser), “Diagram GT” (the formal language of input charts uses ground truth data), and “All GT” (the formal language of input charts and textual problems are ground truth data)

[30]. The results reported in Table 1 reveal some noteworthy distinctions among these models. GeoDRL optimizes the search strategy of Inter-GPS. Our SANS demonstrate superior performance on the Geometry3K dataset compared with Inter-GPS (All GT), achieving a 5.6% improvement in the Choice metric but slightly lower in the Completion metric.

As to neural solver, NGS is a pioneering neural geometric solver, which leverages auxiliary self-supervised tasks to bolster cross-modal semantic representation, while Geoformer jointly tackled geometric proof and computation problems. Compared to them, our SANS showed significant improvement in all metrics. The character alignment method SCA-GPS, resulted in a 4.8% lower in the Choice metric compared to SANS. PGPSNet improves image-text understanding through semantic modeling and text pre-training, yet our SANS adopts a finer-grained cross-modal fusion strategy and enhances spatial awareness of geometric primitives, resulting in improvements of 3.8%, 3.6%, and 2.8% in the Completion, Choice, and Top-3 metrics, respectively. Moreover, our model outperformed all other state-of-the-art solvers on the PGPS9K dataset.

We also compare our approach with general multimodal large models, and the results show that the performance of GPT-4V [1] and LLaVA-v1.5 [15] is poor in GPS contexts. This underperformance may be attributed to the limited

capability of general multimodal large models to perceive the complex geometric structures and spatial relationships present in geometric images.

## 4.2 Ablation Study

**Effect of Modules.** We conduct ablation experiments on the Geometry3K dataset to examine the effectiveness of the modules proposed in our SANS. As shown in Table 2, we select PGPSNet [30] as our baseline for the ablation experiments. The SS-PLM module of PGPSNet only pre-trains the text modality, while our SANS pre-training all input modalities before training. The results show that our multimodal pre-training module (MDSP) outperforms PGPSNet’s single-modal pre-training module and improves performance by 2.4%. Additionally, our point-symbol spatial awareness attention module further enhances GPS performance by enhancing the differences in relative distances of text points in geometric space during training. Ultimately, our model achieves an accuracy of 81.5%, which is a 3.4% improvement over the baseline.

**Table 2. Impact of the design submodules on Geometry3K.**

Module	Accuracy
Baseline	77.9
+ MDSP	80.3(+2.4)
+ MDSP + PP-SAA	81.5(+3.4)

**Table 3. Effectiveness of pre-training strategies on Geometry3K.**

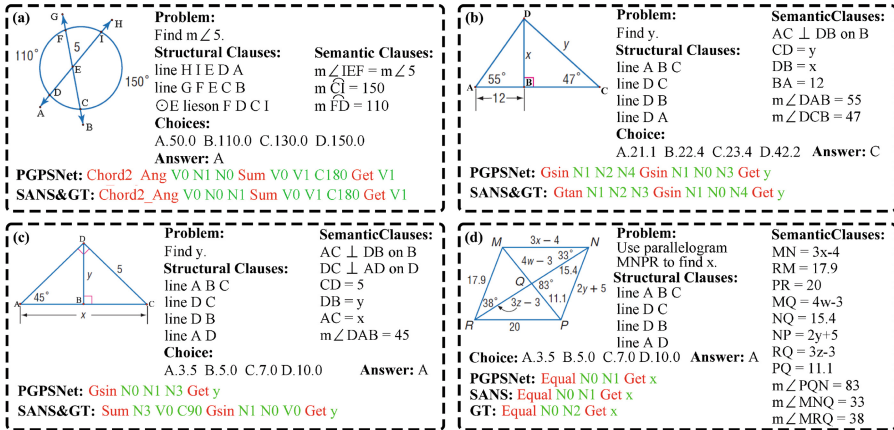
Module	Accuracy
None	60.2
+ SSP	73.2(+13.0)
+ DB-VTPM	77.6(+17.4)
+ SSP + DB-VTPM	81.5(+21.3)

**Pre-training Strategies.** To validate the efficacy of our proposed DB-VTPM strategy, we conduct distinct pre-training experiments utilizing both SSP and DB-VTPM methodologies. Subsequently, we conduct fine-tuning on the Geometry3K dataset to perform ablation studies, and the results are detailed in Table 3. The SSP strategy effectively models global relationships, and the DB-VTPM strategy enhances the model’s cross-modal learning of geometric diagrams and texts during pre-training and preliminary learning of point primitive distribution in geometric diagrams. When comparing the results between the fourth row and

the outcomes from the second and third rows, the simultaneous application of both strategies allows for a deeper understanding of spatial relationships among geometric primitives, which advances the model’s comprehension of geometric diagrams and texts.

### 4.3 Case Analysis

As shown in Fig. 3, we conduct a case analysis. In case (a), this problem requires the model to perceive the spatial relationships between geometric primitives. SANS accurately understood the positions of various geometric primitives in the diagrams, while PGPSNet [30], lacking spatial awareness of geometric primitives, which confuses the orientations of angles. In case (b) - (c), the perception of the relationships between geometric angles is key to solving this problem. SANS brilliantly handled the correspondence between angles in geometric diagrams, whereas PGPSNet confused the relationships between edges and angles, leading to errors in problem-solving. In case (d), the complex spatial relationships between geometric primitives led to incorrect solutions by both SANS and PGPSNet. In conclusion, SANS promotes the development of GPS by enhancing the spatial perception among geometric primitives.



**Fig. 3.** The cases analysis on the Geometry3K dataset. (a), (b), and (c) represent the problems correctly solved by SANS, whereas (d) denotes the instance where SANS provided an incorrect solution.

## 5 Conclusion

In this paper, we proposed a neural solver for GPS. Specifically, the designed MDSP is used to promote the performance of cross-modal fusion through learning the spatial relationships between geometric primitives. Furthermore, during the training phase, the PP-SAA module can force the model to pay more attention to the space relationship between points. Experimental results demonstrate the effectiveness of our proposed MDSP and PP-SAA. Compared to the recent

state-of-the-art models, SANS achieves superior performance on publicly available benchmarks. Moving forward, we aim to design a finer-grained fusion of geometric images and text, further enhancing the model’s understanding of spatial relationships among geometric primitives.

**Acknowledgements.** This work is supported by National Natural Science Foundation of China (61773325, 62222609, 62076236), Unveiling and Leading Projects of Xiamen (No. 3502Z20241011), Open Project of the State Key Laboratory of Multimodal Artificial Intelligence Systems (No. MAIS2024101), Natural Science Foundation of Xiamen (No. 3502Z202373058), and Fujian Key Technological Innovation and Industrialization Projects (No. 2023XQ023), and the Education and Scientific Research Projects for Middle-Aged and Young Teachers of Fujian Province (grant number JAT231102).

## References

1. Achiam, J., et al.: GPT-4 technical report. arXiv preprint [arXiv:2303.08774](https://arxiv.org/abs/2303.08774) (2023)
2. Antol, S., Agrawal, A., et al.: VQA: visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2425–2433 (2015)
3. Bajaj, R., Sharma, V.: Smart education with artificial intelligence based determination of learning styles. *Procedia Comput. Sci.* **132**, 834–842 (2018)
4. Bao, H., et al.: UniLMv2: pseudo-masked language models for unified language model pre-training. In: International Conference on Machine Learning, pp. 642–652. PMLR (2020)
5. Chen, J., et al.: UniGeo: unifying geometry logical reasoning via reformulating mathematical expression. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 3313–3323 (2022)
6. Chen, J., et al.: GeoQA: a geometric question answering benchmark towards multimodal numerical reasoning. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 513–523 (2021)
7. Cho, J., Lei, J., Tan, H., Bansal, M.: Unifying vision-and-language tasks via text generation. In: Proceedings of the 38th International Conference on Machine Learning, vol. 139, pp. 1931–1942 (2021)
8. Goyal, Y., Khot, T., Agrawal, A., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: elevating the role of image understanding in visual question answering. *Int. J. Comput. Vision* **127**(4), 398–414 (2019)
9. Grill, J.B., et al.: Bootstrap your own latent a new approach to self-supervised learning. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, pp. 21271–21284 (2020)
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., Lv, T., Cui, L., Lu, Y., Wei, F.: LayoutLMv3: pre-training for document AI with unified text and image masking. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 4083–4091 (2022)
12. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. In: Proceedings of the 38th International Conference on Machine Learning, pp. 4904–4916 (2021)
13. Kim, W., Son, B., Kim, I.: ViLT: vision-and-language transformer without convolution or region supervision. In: Proceedings of the 38th International Conference on Machine Learning, pp. 5583–5594 (2021)

14. Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., Hoi, S.C.H.: Align before fuse: vision and language representation learning with momentum distillation. In: Proceedings of the Neural Information Processing Systems, pp. 9694–9705 (2021)
15. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. *Adv. Neural Inf. Process. Syst.* **36** (2024)
16. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proceedings of the 36th International Conference on Learning Representations (2019)
17. Lu, P., et al.: Inter-GPS: interpretable geometry problem solving with formal language and symbolic reasoning. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pp. 6774–6786 (2021)
18. Ning, M., Wang, Q.F., Huang, K., Huang, X.: A symbolic characters aware model for solving geometry problems. In: Proceedings of the 31st ACM International Conference on Multimedia, pp. 7767–7775 (2023)
19. Peng, S., Fu, D., Liang, Y., Gao, L., Tang, Z.: GeoDRL: a self-learning framework for geometry problem solving using reinforcement learning in deductive reasoning. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2023, pp. 13468–13480 (2023)
20. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning, pp. 8748–8763 (2021)
21. Sachan, M., Xing, E.: Learning to solve geometry problems from natural language demonstrations in textbooks. In: Proceedings of the 6th Joint Conference on Lexical and Computational Semantics, pp. 251–261 (2017)
22. Seo, M., Hajishirzi, H., Farhadi, A., Etzioni, O., Malcolm, C.: Solving geometry problems: combining text and diagram interpretation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1466–1476 (2015)
23. Wang, X., et al.: Large-scale multi-modal pre-trained models: a comprehensive survey. *Mach. Intell. Res.* **20**(4), 447–482 (2023)
24. Waswani, A., et al.: Attention is all you need. In: Proceedings of the Conference on Neural Information Processing Systems (2017)
25. Wei, X., Zhang, T., Li, Y., Zhang, Y., Wu, F.: Multi-modality cross attention network for image and sentence matching. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10938–10947 (2020)
26. Wenjun, W.: Basic principles of mechanical theorem proving in elementary geometries. *Selected Works Of Wen-Tsun Wu*, p. 195 (2008)
27. Xu, Y., et al.: LayoutLMv2: multi-modal pre-training for visually-rich document understanding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pp. 2579–2591 (2021)
28. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: LayoutLM: pre-training of text and layout for document image understanding. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1192–1200 (2020)
29. Zhang, M.L., Yin, F., Hao, Y.H., Liu, C.L.: Plane geometry diagram parsing. In: Proceedings of the Joint Conference on Artificial Intelligence (2022)
30. Zhang, M.L., Yin, F., Liu, C.L.: A multi-modal neural geometric solver with textual clauses parsed from diagram. In: Proceedings of the 32nd International Joint Conference on Artificial Intelligence, pp. 3374–3382 (2023)



# A Multi-modal Framework to Counter Hate Speeches

Kirtilekha Bhesra<sup>1</sup> and Akshay Agarwal<sup>2</sup>(✉)

<sup>1</sup> C.V Raman Global University, Bhubaneswar, India

<sup>2</sup> Indian Institute of Science Education and Research Bhopal, Bhopal, India  
akagarwal@iiserb.ac.in

**Abstract.** The proliferation of offensive, hateful, and toxic content on social media platforms has reached unprecedented levels. These deleterious expressions not only tarnish the fabric of online interactions but also pose significant threats to individual well-being, potentially precipitating mental health issues such as depression. Manifesting in various modalities including audio and text, this digital toxicity exerts a corrosive influence, leaving enduring impacts on the psyche of individuals. The literature has begun addressing this issue through the lens of natural language processing. However, conventional toxic language detection systems often exhibit biases, particularly in misidentifying text featuring mentions of minority groups as harmful. Furthermore, the overreliance on spurious correlations undermines the efficacy of these systems in detecting implicitly toxic language. Notably, existing benchmark datasets such as ToxiGen predominantly comprise text-based content. Thus, to address this gap, this study presents a pioneering effort in assembling an audio-based hate speech dataset. Subsequently, a multimodal hate speech detection algorithm integrating audio and text inputs is proposed, demonstrating a significant performance enhancement over conventional text-based models.

## 1 Introduction

The phenomenon of hate speech, although perennial, has assumed greater significance in the digital age. Its ramifications extend beyond individual targets to encompass entire societies by starkly contradicting principles of tolerance, inclusion, and human rights. Not only does hate speech subject its victims to discrimination, abuse, and violence, but it also perpetuates social and economic marginalization. Left unchecked, it can sow seeds of conflict, impeding societal peace and development while engendering egregious human rights violations [28], including acts of atrocity. Consequently, addressing and counteracting hate speech becomes imperative. In recent years, researchers have proposed various machine learning and deep learning methods to detect online hate speech, which includes content spreading hostility or inciting violence based on race, religion, gender, or other identity traits [9, 12, 17, 21]. Nonetheless, existing research predominantly revolves around text-based models, with a limited exploration



into speech-based approaches [13, 14]. However, one can safely assume that hate speeches are not limited to text but predominately used through voice as well. One scenario of hate speech in audio form can be seen during several national elections which see a sharp jump in the content of hate speech audio/videos on the social media platforms. This study aims to investigate whether audio data can furnish unique and valuable signals for hate speech detection, drawing upon the success of audio-based person identification methodologies.

Recent trends indicate a concerning uptick in toxic content within media, particularly in movies where hate-driven dialogue seeks to elicit strong reactions. Likewise, certain comedians employ divisive and derogatory humor targeting caste, religion, and gender, perpetuating the normalization of hate speech. This issue transcends entertainment to permeate public platforms such as YouTube, Meta, and Twitter, which witness a substantial influx of audio content daily, some of which contain hateful elements. Despite moderation efforts, the sheer volume of audio content renders it challenging to identify and remove all instances of hate speech, potentially fostering its unchecked proliferation and contributing to toxicity and division online. While some platforms utilize human moderators to identify and remove harmful content, the sheer volume of daily uploads poses a formidable challenge. Even though platforms try to moderate, there's just too much content to check. For instance, Facebook employs approximately 15,000 moderators to review content flagged by both AI and users, yet it still encounters approximately 300,000 content moderation errors daily<sup>1</sup>. Moreover, moderators themselves involve emotional and psychological risks. Compliance with regulatory mandates further complicates matters, as failure to expeditiously remove hateful content may result in fines. While larger platforms deploy machine learning algorithms for detection, smaller platforms may lack the resources to develop datasets and models for hate speech detection in audio. Hence, there arises a necessity to develop efficient and effective hate speech detection models capable of detecting hate speech not only in text but also in audio. We assert that it underscores the need for a comprehensive benchmark encompassing both spoken and written language to fortify society against this pervasive menace.

## 1.1 Current Limitation and Contribution

In recent years, notable strides have been made in text-based hate speech detection [3]. A pivotal factor contributing to the success of hate speech detection technology lies in the availability of publicly accessible hate speech text datasets. However, unlike numerous openly accessible text-based datasets, the absence of large-scale datasets capturing both text-based and audio-based hate speech concurrently is conspicuous. Remarkably, there has been scant exploration into hate speech detection in audio, highlighting a critical research gap in the field. Our objective is to bridge this void by leveraging both text and audio data, employing multimodal fusion to yield distinctive and valuable insights into hate speech

---

<sup>1</sup> <https://www.theverge.com/2020/11/13/21562596/facebook-ai-moderation>.

detection. While text-based hate speech detection has historically shown significant efficacy [26], parallelly, audio modality shows success in several tasks including person identification and sentiment analysis [15, 22]. By amalgamating these modalities, we aim to develop a more comprehensive and precise hate speech detection system. This multimodal approach capitalizes on the strengths of text and audio analyses, presenting a robust framework poised to potentially surpass existing methodologies and contribute to the evolution of dependable solutions in hate speech detection. Section 2 provides a brief overview of the existing hate speech detection algorithms followed by a description of the proposed audio hate speech detection dataset in Sect. 3. Section 4 describes the proposed multimodal hate speech detection algorithms; whereas, its effectiveness is reported in Sect. 5. Section 6 provides the overall summary of the findings reported in this paper along with possible future directions to advance the hate speech detection task.

## 2 Related Works

In recent years, significant progress has been made in text-based hate speech detection due to the availability of public datasets. We will first summarize the existing datasets. Following that, we will analyze multimodal models, which integrate text, videos, and audio for hate speech detection. At present, there are many benchmark datasets based on text content. In the text-based hate speech datasets, most work has focused on explicit or overt hate speech, failing to address a more pervasive form based on coded or indirect language. To fill this gap, Implicit Hate [10] dataset introduces a theoretically justified taxonomy of implicit hate speech and a benchmark corpus with fine-grained labels for each message and its implication, ToxiGen [12] offers a large machine-generated dataset to train models on subtler forms of hate speech. It includes a wide variety of grammatically correct yet subtly offensive language targeting thirteen different minority groups. ETHOS [20] is another hate speech detection dataset for comments found on YouTube and Reddit. It comes in two versions: a simple classification (hateful or not) and a more detailed one that identifies specific types of hate speech. HateXplain [18] covers multiple aspects of hate speech detection by annotating each post from three different perspectives: the basic, commonly used 3-class classification (i.e., hate, offensive or normal), the target community (i.e., the community that has been the victim of hate speech/offensive speech in the post), and the rationales, i.e., the portions of the post on which their labeling decision (as hate, offensive or normal). Automatic hate speech detection using machine learning approaches is a relatively recent field, and hence, only a few works for the detection of hate speech are available especially targeting audio hate speech [2, 4, 27].

Although automatic identification of offensive comments has seen tremendous success from the point of processing natural text into hate speech and cyberbullying [1, 25]. The natural language processing field has witnessed several tools for analyzing large datasets, such as social media content [7, 29]. Further, techniques such as deep learning architectures including convolutional neural network

(CNN), recurrent neural networks (RNN), and long short-term memory (LSTM) and ensemble learning including random forest, bagging, and boosting have significantly contributed to automatic hate speech detection in social media [21]. However, we believe that sentence transformer and speech t5 which are already pre-trained on large internet datasets containing a variety of hate words might be instrumental in learning word embedding highly effective as compared to handcrafted models.

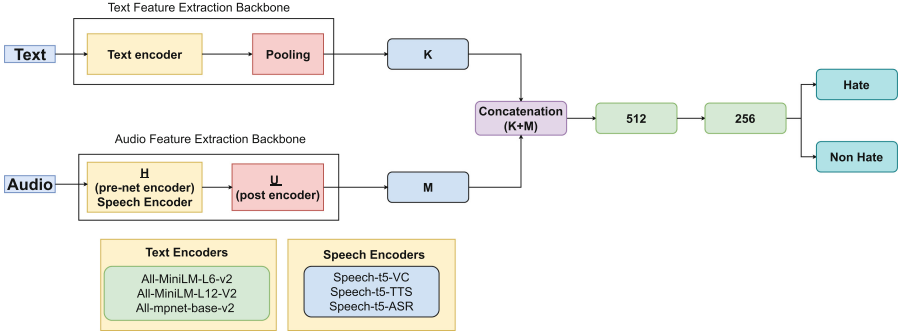
Hate speech detection is not limited to the utilization of any one particular data modality but is seen as the fusion of multiple input modalities. For example, in the context of multimodal hate speech detection, Rana et al. [23] present a hate speech detection video dataset (HSDVD). Das et al. [8] used a multimodal approach (text, audio, video) for hate speech detection but primarily focused on video content. It is important to note that a significant portion of their video collection consists of content where only images and text are associated with music, shifting the focus of their research towards text and video rather than audio detection. To the best of our knowledge, we are the first to experiment with multi-modal hate speech detection, where we leverage text and audio modalities. We are confident that our dataset, along with the benchmark model trained on it, will assist moderators in discerning authentic instances of hate speech while minimizing false alarms.

### 3 Proposed Audio Hate Speech Dataset

To thoroughly investigate the potential of the audio modality, we have generated both hate and non-hate speech samples by transforming text excerpts from the *ToxiGen dataset* [12] into audio representations. 1000 random text samples where 502 samples belong to the non-hate class and 498 samples belonging to the hate class are selected to generate the audio samples. We utilized the *Speech T5* a state-of-the-art speech synthesis model designed for converting text into natural-sounding speech [5]. Since it is a natural sound-like encoder, this cutting-edge technology has applications in various fields, including voice assistants, and interactive media, and hence is an ideal choice for our work. Within the *SpeechT5* framework, [19], various speakers are available, including two distinct male voices namely BDL and RMS, and two female voices namely CLB and KSP. To ensure a comprehensive evaluation, we generated the audio dataset using four distinct voices: two male voices (BDL and RMS) and two female voices (CLB and KSP). The prime reason behind this is since hate speech is concerned with any particular demographic identity, covering a wide spectrum of demographic variables can ensure the universal detection of hate samples. By utilizing 4 different voices, in total, we have generated 4000 audio samples. The proposed dataset is publicly available for research purposes at the following [link](#).

### 4 Methodology

In this paper, we formulate the hate speech detection problem as follows: Let  $\mathcal{D} = \{(a_i, t_i, y_i)\}_{i=1}^N$  be a dataset, where  $a_i$  is the audio embedding,  $t_i$  is the



**Fig. 1.** A schematic diagram of the proposed multi-modal hate speech detection algorithm. Pre-net converts the input speech into the hidden representations used by the transformer (SpeechT5).  $K$  and  $M$  represent the dimensions obtained using the particular text and audio encoder, respectively.

corresponding text embedding.  $y_i \in 0, 1$  represent binary label, where ( $y = 0$ ) corresponds to non-hate speech and ( $y = 1$ ) belong to the hate class. This research aims to learn a binary classifier  $f_\theta(a_i, t_i)$  that can predict the label  $y_i$  for a given audio and text embedding pair. The simplest possible classifier can be learned by minimizing the cross-entropy loss function  $\ell$ :

$$\ell(f_\theta(a_i, t_i), y_i) = -y_i \log(f_\theta(a_i, t_i)) - (1 - y_i) \log(1 - f_\theta(a_i, t_i))$$

Figure 1 shows the schematic diagram of the proposed multimodal hate speech detection architecture. The architecture uses the embeddings of both text and audio modalities and combines them to harness the distinct strengths of each. Since the literature comprises several audio and text embedding techniques; therefore, the effective selection of an accurate embedding method is critical. Further, to comprehensively understand the strengths and weaknesses of embedding methods, multiple audio and text embedding techniques are explored in this research. Once audio and text embeddings are obtained, a binary classification network attached at the end of the proposed architecture is trained for multimodal hate speech detection.

In this work, to effectively encode the text modality we used several sentence transformer encoders [24], namely (i) all-MiniLM-L12-v2 (**enc-1**), (ii) all-MiniLM-L6-v2 (**enc-2**), and (iii) all-mpnet-base-v2 (**enc-3**). It is to be noted here that these models are distilled versions of BERT (bidirectional encoder representations from transformers) and are optimized for creating dense vector representations of sentences while being computationally efficient, with L12 and L6 denoting the number of layers. all-mpnet-base-v2, on the other hand, is a sentence transformer based on Microsoft’s MPNet (masked and permuted network), which captures both the word order and contextual meaning effectively, leading to improved performance on natural language understanding tasks. We assert that these encoders that are pre-trained on large-scale text datasets are effective in extracting semantic information. By utilizing these state-of-the-art text

encoders, we aimed to capture nuanced linguistic characteristics and semantic nuances within our textual data. Each encoder offers unique strengths in capturing different aspects of textual information, thus enriching our analysis and providing a comprehensive understanding of the underlying linguistic dynamics. Our preliminary study also demonstrates the effectiveness of these pre-trained text encoders for hate speech detection [6].

In our work endeavor, similar to text encoders, to extract the discriminative features from the audio samples, we have utilized one of the popular methods, namely, the Microsoft SpeechT5 processor [5,16]. The three distinct voice feature extraction methods used in this research are (i) `speecht5-vc` (**Venc-1**), (ii) `speecht5-tts` (**Venc-2**), and (iii) `speecht5-asr` (**Venc-3**). Since the text and audio modalities are complementary to each other, the fusion of their encoding advances the hate speech detection task. Further, to comprehensively evaluate the effectiveness of individual demographic entities, we trained hate detection models on each voice representing different genders. This holistic approach not only enabled us to unravel the intricacies of spoken language but also laid the groundwork for exploring diverse applications in the realm of speech processing and analysis.

Once the text and audio fused vectors are obtained, they are passed to two-dense classification layers to learn the decision boundary between the hate and genuine samples. These two dense layers contain 512 and 256 neurons respectively. To avoid overfitting on our small-scale dataset, we trained our model for 100 epochs with early stopping. The Adam optimizer with an initial learning rate of  $10^{-3}$  and batch size of 128 is used to train the classification layers. Further, a learning rate scheduler with a factor of 0.1 and patience of 20 is also used for effective parameter tuning. In the proposed research, we have performed the ablation study by combining each text encoder with each audio encoder. In other words, a total of 9 combinations (3 text encoders and 3 audio encoders) have been extensively evaluated to identify the effective combination of hate speech detection. Text encoders namely `enc-1` and `enc-2` provide the feature of dimension 384 and the dimensionality of the feature obtained from the `enc-3` text encoder is 768. Each audio encoder used in this research provides the feature vector of dimension 768.

## 5 Experimental Results and Analysis

The analysis of the multimodal hate speech detection approach reveals insightful findings concerning multiple demographic entities and the effectiveness of the amalgamation of different feature encoders. The results of multimodal hate detection for female and male entities coupled with text encoding are reported in Table 1 and Table 2, respectively. The results of hate speech detection are reported in terms of accuracy (Acc), balanced accuracy (B-Acc), area under the ROC (RA), and F-1 (F1) score. The analysis can be broadly divided into two groups: (i) analysis based on the effectiveness of individual demographic voice and (ii) the effectiveness of encoders. While both genders exhibit strong performance in multimodal concatenation, *female voice generated using CLB display a*

*slightly higher accuracy compared to male voices.* This disparity suggests potential variations in the underlying linguistic and acoustic characteristics between male and female speech patterns, influencing the effectiveness of multimodal fusion techniques. The female voice encoded using CLB yields an accuracy of 84.70% as compared to the 83.10% obtained using the BDL-encoded male voice. Furthermore, a comparative analysis between male and female voices reveals a nuanced trend in performance.

As mentioned earlier, effective generation of voice is equally important similar to encoding of them. It can also be seen from the hate speech detection results, where a drastic performance difference between the female voices generated using CLB and KSP is noticed. A similar observation can be seen on the male voice modality as well, where the use of RMS voice yields 11.10% lower performance than the BDL voice. *Interestingly, in both male and female cases, the amalgamation of text encoder-3 (all-mpnet-base-v2) with speecht5-vc (Venc-1) voice encoder resulted in the highest hate speech detection performance.* These results highlight the efficacy of leveraging diverse encoders to capture and concatenate rich textual and auditory features, enhancing classification accuracy in male voice classification tasks.

**Table 1.** Hate speech detection using the combination of text and female voice (CLB and KSP). The best classification performance across each voice technique is highlighted and the second best is underlined.

Model	CLB Voice				KSP Voice			
	Acc	B-Acc	RA	F1	Acc	B-Acc	RA	F1
enc-1 + Venc-1	78.70	78.58	78.65	78.58	66.50	66.50	65.44	65.50
enc-1 + Venc-2	49.50	50.00	32.82	50.00	50.00	50.00	33.33	50.00
enc-1 + Venc-3	76.50	76.50	76.49	76.50	61.00	61.22	58.92	61.22
enc-2 + Venc-1	77.70	77.57	77.67	77.57	65.00	65.26	64.75	65.26
enc-2 + Venc-2	49.50	50.00	32.82	50.00	51.00	50.00	34.45	50.00
enc-2 + Venc-3	76.10	76.16	76.11	76.16	69.00	68.50	68.47	68.50
enc-3 + Venc-1	<b>84.70</b>	84.67	84.69	84.67	<u>70.00</u>	70.35	69.81	70.35
enc-3 + Venc-2	49.50	50.00	32.82	50.00	47.50	50.00	30.59	50.00
enc-3 + Venc-3	<u>82.80</u>	82.68	82.76	82.68	<b>70.50</b>	71.80	69.24	71.80

Surprisingly, in our experiments, we observed that the “Microsoft Speech-T5-tts” encoder encounters difficulties in accurately processing embeddings of female voices, leading to erratic predictions. Consequently, the model exhibited random predictions, resulting in an accuracy of 49%, which notably stands as the lowest among all predictions. This observation underscores the need for further investigation and potential refinement of the text-to-speech (TTS) encoder to ensure robust performance across diverse voice characteristics.

**Table 2.** Hate speech detection using the combination of text and male (RMS and BDL) voices. The best classification performance across each voice technique is highlighted and the second best is underlined.

Model	RMS Voice				BDL Voice			
	Acc	B-Acc	RA	F1	Acc	B-Acc	RA	F1
enc-1 + Venc-1	63.00	62.98	63.02	62.98	78.30	78.08	78.21	78.08
enc-1 + Venc-2	49.00	50.00	32.22	50.00	75.40	75.12	75.26	75.12
enc-1 + Venc-3	<u>71.00</u>	71.30	70.74	71.30	78.20	77.89	78.07	77.89
enc-2 + Venc-1	60.00	59.17	58.15	59.17	76.50	76.39	76.44	76.39
enc-2 + Venc-2	48.50	50.00	31.68	50.00	74.10	73.98	74.05	73.98
enc-2 + Venc-3	63.50	64.42	61.58	64.42	76.00	76.11	75.95	76.11
enc-3 + Venc-1	64.50	65.42	63.77	65.42	<b>83.10</b>	83.18	83.11	83.18
enc-3 + Venc-2	47.50	50.00	30.59	50.00	<u>82.90</u>	82.78	82.87	82.78
enc-3 + Venc-3	<b>72.00</b>	72.22	72.04	72.22	81.80	81.86	81.79	81.86

**Table 3.** Performance of unimodal text-based hate speech classification.

Model	Accuracy	Precision	Recall	F1
HateBERT	50.3	52.4	10.7	17.8
RoBERTa	79.8	93.8	63.9	76.0

### 5.1 Performance of Unimodal vs Multimodal Hate Classification

For a comprehensive comparison between unimodal and multimodal models, we have now performed experiments with individual modalities used in this research. To effectively encode the text modalities, two state-of-the-art unimodal models, HateBERT and RoBERTa. We believe that these algorithms are developed using the ToxiGen [12] dataset, they must be effective for hate speech detection. The results of the unimodal algorithms for text are reported in Table 3. It is observed that the RoBERTa outperforms the HateBERT model by a significant margin. For instance, the accuracy obtained by HateBERT is 29.5% less than the accuracy obtained by the RoBERTa model. Further, in comparison to the utilization of state-of-the-art (SOTA) text encoding algorithms, we have performed

**Table 4.** Performance of unimodal audio-based hate speech classification.

Gender (Generator)	Accuracy	B-Acc	F1	RA
Male (BDL)	47.88	50.92	43.24	50.92
Male (RMS)	51.21	51.51	50.45	51.51
Female (CLB)	52.28	52.51	48.26	52.28
Female (KSP)	47.27	47.09	42.28	47.09

experiments with the text encoders used in this research. To perform the hate speech classification using individual text encoders, we have used the two-layer feed-forward neural network with batch normalization. The first layer of this network consists of neurons equal to the dimension of the input feature embedding; whereas, the second layer contains 64 neurons. As found in multimodal detection, enc-3 (all-mpnet-base-v2) yields the highest accuracy of 80.3% surpassing the performance of other encoders by at least 4.7%. We have performed the ablation study with deeper neural networks as well; however, through five-fold cross-validation two-layer network is found the best for hate speech detection.

Further, similar to the comparison with state-of-the-art (SOTA) text-based hate speech detection architecture, we utilized the audio spectrogram transformer (AST) [11] to perform hate detection. The results of hate audio detection using 4 different voice variations are reported in Table 4. AST, known for its effectiveness in audio analysis, ensured unbiased evaluations across both modalities. Once the AST feature vector is obtained, it is passed to the two-layer neural network comprising 527 and 128 neurons in the first and second layers, respectively. When the AST model is used on the male voice generated using BDL, it yields an accuracy of 47.88%. However, when the RMS-generated male audios are used for evaluation, the performance of hate speech detection increases by 3.33%. Similarly, the performance of the CLB-generated female voices is 5.01% better than the KSP-generated female voices in detecting hate speeches. However, the proposed multimodal architecture yields significantly higher accuracy of 84.7% than any unimodal model. The superior performance of the multimodal approach can be attributed to its ability to integrate and leverage the strengths of both text and audio modalities. Therefore, we can easily assert that while the generated audio modalities are synthetic, they act as auxiliary information enhancing the hate speech detection performance by capturing nuanced features. Similar to the evaluation of individual text encoders, we have evaluated the effectiveness of individual audio encoders as well. For classification, again we have used the two-layer neural network. Specht5-vc (Venc-1), specht5-tts (Venc-2), and specht5-asr (Venc-3) yield the hate audio classification accuracy of 68.7%, 51.3%, and 69.8%, respectively. Similar to text, compared to deeper neural networks, two-layer neural performed best with each audio encoder.

## 6 Conclusion and Future Directions

This paper represents a significant step forward in the ongoing effort to identify and mitigate hateful content by harnessing signals from both text and audio modalities. In pursuit of advancing research in this crucial direction, we introduce a novel hate audio speech detection dataset<sup>2</sup> by synthetically transforming the text corpus using the voice of multiple demographics. In the realm of hate speech detection, relying solely on text data is not enough since the hate content is not limited to any one input modality. It can be seen from the lower performance of hate speech detection using only the text corpus. Therefore, we have proposed

<sup>2</sup> [https://github.com/kirti1545/HateSpeech\\_Dataset](https://github.com/kirti1545/HateSpeech_Dataset).



a multimodal hate speech detection architecture by combining the audio with text. The proposed algorithms outperform the unimodal hate speech detectors by a significant margin reflecting the potential of audio modality even when it is synthetically generated. It demonstrates that even when an audio modality is missing in hate content it can be used by synthetically generating it to detect hate speech effectively. Since, the dataset developed as part of this research is still not large enough as compared to the text dataset used, in the future, we aim to extend this dataset. Furthermore, addressing bias inherent in toxic language detection systems, particularly concerning false positives associated with mentions of minority groups, stands as a critical challenge that we are committed to addressing in forthcoming studies. We aspire to develop context-aware toxicity analysis methodologies that leverage sentiment analysis and contextual cues to enable more nuanced language identification.

## References

1. Al-Garadi, M.A., et al.: Predicting cyberbullying on social media in the big data era using machine learning algorithms: review of literature and open challenges. *IEEE Access* **7**, 70701–70718 (2019)
2. Al-Hassan, A., Al-Dossari, H.: Detection of hate speech in social networks: a survey on multilingual corpus. In: *International Conference on Computer Science and Information Technology*, vol. 10, pp. 10–5121. ACM (2019)
3. Alkomah, F., Ma, X.: A literature review of textual hate speech detection methods and datasets. *Information* **13**(6), 273 (2022)
4. Alrehili, A.: Automatic hate speech detection on social media: a brief survey. In: *IEEE/ACS International Conference on Computer Systems and Applications*, pp. 1–6 (2019)
5. Ao, J., et al.: SpeechT5: unified-modal encoder-decoder pre-training for spoken language processing. *arXiv preprint [arXiv:2110.07205](https://arxiv.org/abs/2110.07205)* (2021)
6. Bhesra, K., Shukla, S.A., Agarwal, A.: Audio vs. text: identify a powerful modality for effective hate speech detection. In: *The Second Tiny Papers Track at ICLR* (2024)
7. Cheng, J., Danescu-Niculescu-Mizil, C., Leskovec, J.: Antisocial behavior in online discussion communities. In: *AAAI Conference on Web and Social Media*, vol. 9, pp. 61–70 (2015)
8. Das, M., Raj, R., Saha, P., Mathew, B., Gupta, M., Mukherjee, A.: HateMM: a multi-modal dataset for hate video classification. In: *International AAAI Conference on Web and Social Media*, vol. 17, pp. 1014–1023 (2023)
9. Del Vigna<sup>12</sup>, F., Cimino<sup>23</sup>, A., Dell’Orletta, F., Petrocchi, M., Tesconi, M.: Hate me, hate me not: hate speech detection on Facebook. In: *Italian Conference on Cybersecurity*, pp. 86–95 (2017)
10. ElSherief, M., et al.: Latent hatred: a benchmark for understanding implicit hate speech. *arXiv preprint [arXiv:2109.05322](https://arxiv.org/abs/2109.05322)* (2021)
11. Gong, Y., Chung, Y.A., Glass, J.: AST: Audio spectrogram transformer. *arXiv preprint [arXiv:2104.01778](https://arxiv.org/abs/2104.01778)* (2021)
12. Hartvigsen, T., Gabriel, S., Palangi, H., Sap, M., Ray, D., Kamar, E.: ToxiGen: a large-scale machine-generated dataset for adversarial and implicit hate speech detection. In: *Annual Meeting of the Association for Computational Linguistics*, pp. 3309–3326 (2022)

13. Ibañez, M., Sapinit, R., Reyes, L.A., Hussien, M., Imperial, J.M., Rodriguez, R.: Audio-based hate speech classification from online short-form videos. In: IEEE International Conference on Asian Language Processing, pp. 72–77 (2021)
14. Imbwaga, J.L., Chittaragi, N.B., Koolagudi, S.G.: Automatic hate speech detection in audio using machine learning algorithms. *Int. J. Speech Technol.* **27**(2), 447–469 (2024)
15. Lin, Y., Ji, P., Chen, X., He, Z.: Lifelong text-audio sentiment analysis learning. *Neural Netw.* **162**, 162–174 (2023)
16. Liu, Y., et al.: DelightfulTTS: the microsoft speech synthesis system for blizzard challenge 2021. arXiv preprint [arXiv:2110.12612](https://arxiv.org/abs/2110.12612) (2021)
17. MacAvaney, S., Yao, H.R., Yang, E., Russell, K., Goharian, N., Frieder, O.: Hate speech detection: challenges and solutions. *PLoS ONE* **14**(8), e0221152 (2019)
18. Mathew, B., Saha, P., Yimam, S.M., Biemann, C., Goyal, P., Mukherjee, A.: HateXplain: a benchmark dataset for explainable hate speech detection. In: AAAI Conference on Artificial Intelligence, vol. 35, pp. 14867–14875 (2021)
19. Matthijs: speech5-tts-demo. <https://huggingface.co/spaces/Matthijs/speech5-tts-demo/blob/main/app.py>
20. Mollas, I., Chrysopoulou, Z., Karlos, S., Tsoumakas, G.: ETHOS: an online hate speech detection dataset. arXiv preprint [arXiv:2006.08328](https://arxiv.org/abs/2006.08328) (2020)
21. Mullah, N.S., Zainon, W.M.N.W.: Advances in machine learning algorithms for hate speech detection in social media: a review. *IEEE Access* **9**, 88364–88376 (2021)
22. Poria, S., Cambria, E., Howard, N., Huang, G.B., Hussain, A.: Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing* **174**, 50–59 (2016)
23. Rana, A., Jha, S.: Emotion based hate speech detection using multimodal learning. arXiv preprint [arXiv:2202.06218](https://arxiv.org/abs/2202.06218) (2022)
24. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. arXiv preprint [arXiv:1908.10084](https://arxiv.org/abs/1908.10084) (2019)
25. Rodriguez, A., Argueta, C., Chen, Y.L.: Automatic detection of hate speech on Facebook using sentiment and emotion analysis. In: International Conference on Artificial Intelligence in Information and Communication, pp. 169–174 (2019)
26. Saleh, H., Alhothali, A., Moria, K.: Detection of hate speech using BERT and hate speech word embedding with deep model. *Appl. Artif. Intell.* **37**(1), 2166719 (2023)
27. Schmidt, A., Wiegand, M.: A survey on hate speech detection using natural language processing. In: International Workshop on Natural Language Processing for Social Media, pp. 1–10 (2017)
28. Van Spanje, J., De Vreese, C.: The good, the bad and the voter: the impact of hate speech prosecution of a politician on electoral support for his party. *Party Polit.* **21**(1), 115–130 (2015)
29. Weir, G., Owoeye, K., Oberacker, A., Alshahrani, H.: Cloud-based textual analysis as a basis for document classification. In: International Conference on High Performance Computing & Simulation, pp. 672–676 (2018)



# TBIA-DBNet: A Two-Branch Image-Adaptive DBNet for Scene Text Detection in Real-World Foggy Scenes

Zhaoxi Liu, Gang Zhou<sup>(✉)</sup>, Runlin He, Mengnan Zhang, Zhenhong Jia, and Jing Ma

Key Laboratory of Signal Detection and Processing, Department of Computer Science and Technology, Xinjiang University, Urumqi, China  
gangzhou\_xju@126.com

**Abstract.** Though deep learning-based scene text detection methods have achieved promising results on conventional datasets, these methods are unable to maintain optimal performance in adverse weather conditions, such as foggy weather. To alleviate this problem, we propose a Two-Branch Image-Adaptive DBNet (TBIA-DBNet) framework. Specifically, to avoid missing discriminable features from the original image in one branch, we design an Image Enhancement Network (IENet) in another branch. Additionally, we design a Fusion Module based on Coordinate Attention (FMCA) to fully integrate original and enhanced features. Experimental results demonstrate that TBIA-DBNet significantly enhances scene text detection performance in foggy weather. Notably, it improves detection accuracy by nearly 10% in real-world foggy weather conditions compared to existing methods.

**Keywords:** Scene Text Detection · Dehazing · Real World · Two-Branch Network

## 1 Introduction

Scene text detection has consistently been a focal point of research within the realm of computer vision. Nowadays, scene text detection predominantly focuses on representing arbitrary-shaped text and designing post-processing methods to recover text outlines from geometric attributes [1–5]. Nevertheless, there is a notable absence of research examining scene text detection in adverse weather conditions. Degraded images under adverse weather such as fog appear blurred and distorted, which leads to reduced accuracy in text detection. The visualization results of traditional scene text detection algorithms in various weather are presented in Fig. 1, although DBNet [6] achieves excellent test results in clear weather, its performance significantly deteriorates under foggy conditions.

This work was supported by National Natural Science Foundation of China (No. 62166040, No. 62261053) and Tianshan Talent Training Project-Xinjiang Science and Technology Innovation Team Program (2023TSYCTD0012).



**Fig. 1.** A comparative analysis of the results obtained by DBNet [6] for scene text detection. In the figure, the green curve indicates correctly detected text areas, the yellow curve indicates incorrectly detected text areas, and the red curve indicates missing text areas.

While there is a limited number of algorithms addressing scene text detection under adverse weather, numerous papers on object detection under adverse weather are available for reference. The most straightforward method is to conduct object detection via image restoration. AOD-Net [7] was the first CNN-based end-to-end dehaze network, seamlessly embeddable into other deep models. Instead of relying on separate and intermediate parameter estimation steps, it produced clear images directly from foggy images. Furthermore, DSNet [8] employs two subnetworks to concurrently learn visibility enhancement and object detection. Shared feature extraction layers enable DSNet to mitigate the impact of image degradation. IA-YOLO [9] noted that DSNet encountered challenges in adjusting the parameters to achieve a balanced weight distribution between detection and recovery during the training phase. Therefore, IA-YOLO introduced a differentiable image processing module and employed a compact convolutional neural network to predict its parameters. BAD-Net [10] replaced IA-YOLO’s dehazing module with that of AOD-Net. Experimental results indicated that IA-YOLO’s detection loss was difficult to converge and that detection accuracy significantly decreased. BAD-Net observed that IA-YOLO was not a reliable framework for object detection under foggy weather and proposed a two-branch foggy weather object detection framework. These methods have demonstrated notable efficacy in foggy weather conditions.

However, object detection and scene text detection differ significantly. In general, text detection requires higher positioning accuracy (IoU greater than 80%) and scene text itself is more easily confused with the background. This necessitates rethinking suitable methods for text detection under foggy conditions. Although the latest BAD-Net method considers feature fusion between the two branches, scene text detection requires more scale features to increase the accuracy of scene text positioning.

To address the above limitations, this paper designed a Two-Branch Image-Adaptive DBNet (TBIA-DBNet) model. Specifically, the two branches are divided into original image feature and enhanced image feature branches. To improve the ability of feature extraction in adverse weather, the enhanced image feature branches adopts Laplacian pyramid method [11] to decompose the image into a low frequency (LF) component and multiple high frequency (HF) components to fully learn image features. The image has been processed by the IENet, and the weather-specific information is suppressed, thereby facilitating the restoration of latent information. Additionally, we designed FMCA to fuse features from the two branches at multiple scales, employing coordinate attention [12] to capture long-range dependencies while preserving precise positional information. This approach ensures that the features of the two branches at different scales can be maximally complementary. In the realm of scene text detection, even without any bells and whistles, our method can compete or outperform current state of the art methods on foggy weather datasets.

In summary, the primary contributions of this paper are as follows:

- A Two-Branch Image-Adaptive DBNet scene text detection network called TBIA-DBNet is proposed. Unlike typical single-branch networks, TBIA-DBNet utilizes the differences between its two branches to form complementarity, maximizing the extraction of image features.
- We propose an Image Enhancement Network (IENet), which is based on Laplacian pyramid for adaptive image enhancement. Furthermore, we propose FMCA, which efficiently integrates features from different scales in two branches to improve the robustness and complementarity of the network.
- Comprehensive experimental results illustrate that our proposal can achieve superior detection performance in foggy weather conditions.

## 2 Related Work

### 2.1 Traditional Scene Text Detection

Scene text detection is a critical preliminary step in the process of scene text recognition. Its main function is to automatically detect text information in an image or video and convert it into an editable or searchable text form for subsequent processing and application. Currently, scene text detection methods are commonly categorized into two groups: regression-based and segmentation-based approaches.

Regression-based approaches consider text as objects and focus on identifying their locations by directly predicting the bounding boxes that enclose the text instances. For example, EAST [13] and ABCNet [14] achieve efficient pixel-level regression of text objects through direct prediction without using anchor mechanisms or proposal generation. TextBoxes [15], building on the SSD [16] object detection method, employs larger default box aspect ratios and convolutional kernels to detect long text. Additionally, TextBoxes++ [17] applies quadrilateral regression to multi-directional text instances. LOMO [18], building on these

direct regression methods, introduces an iterative refinement module that iteratively refines bounding box proposals for ultra-long text, and then predicts the centerline, text region, and boundary offsets to reconstruct text instances. Although these methods have achieved strong performance in quadrilateral text detection, most texts appear in irregular shapes, which makes it challenging for these methods to effectively handle various irregular texts.

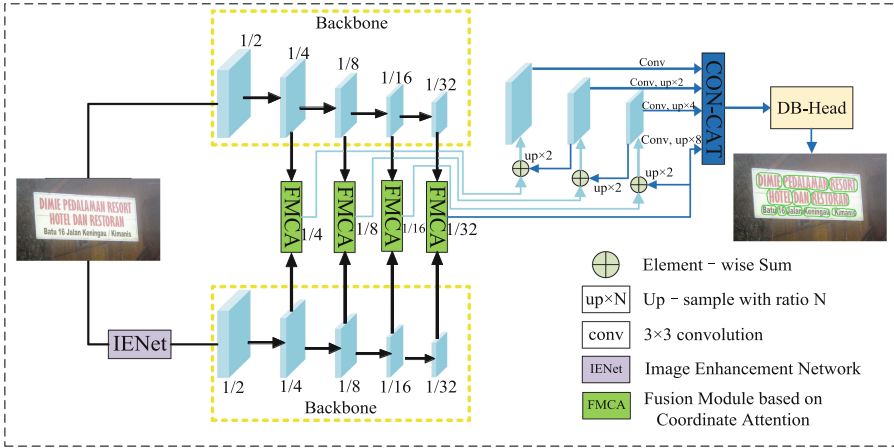
Segmentation-based methods are capable of accurately describing scene text of diverse shapes through the utilisation of pixel-level segmentation masks. For example, PSENet [19] uses different scaling checks to scale text regions step by step to generate complete text boundaries. PAN [20] generates an efficient text detector by utilizing a lightweight feature extraction network, a computationally cheap segmentation head, and several post-processing stages. Inspired by Masks R-CNN [21], SPCNet [22] proposes a supervised pyramid context network based on instance segmentation to detect text of arbitrary shape. I3CL [23] designs a text detection method for interinstance and real exception based on Masks R-CNN. DBNet [6] designs a differentiable binarization module that is capable of predicting text regions directly through segmentation. Nevertheless, these methods are designed based on high-quality images and do not account for the adverse effects of low-quality images under adverse weather conditions on text detection.

## 2.2 Scene Text Detection in Adverse Weather

At present, there is a limited amount of research available on scene text detection specifically under adverse weather conditions [24]. However, there have been some studies conducted on object detection in similar scenarios. The classic approach is to preprocess the image first [25–29], they were designed to eliminate fog and enhance image quality. Nevertheless, enhancements in image quality do not invariably enhance detection accuracy. Others directly implement the end-to-end network structure through the combination of enhancement and detection methods [10,30]. This combined approach has proven to be effective [10]. However, these are all methods based on object detection, and how to effectively apply them in text detection also proves the necessity of our work.

## 3 Proposed Methodology

Figure 2 illustrates the architectural design of our approach. It mainly contains two backbone, an Image Enhancement Network (IENet), four Fusion Module based on Coordinate Attention (FMCA), an FPN [31] layer and an output network. Following [6], we employ ResNet50 [32] as the backbone and DB-Head [6] as the output network. We first simultaneously input the pre-processed image into two distinct branches. Then, we employ FMCA to merge the feature maps acquired from the four stages of ResNet50. Finally, the fused features are transmitted to the FPN layer, where multi-scale features are fused again for the ultimate detection.



**Fig. 2.** The overall framework of the proposed TBIA-DBNet. TBIA-DBNet employs a two-branch architecture, where one branch directly extracts original image features, while the other branch captures features processed by IENet. FMCA is used to fuse four different scale output features from two branches. The fused features are then passed to the FPN layer for final detection.

### 3.1 Image Enhancement Based on Laplacian Pyramid

The Laplace pyramid has demonstrated outstanding performance in image processing [33], and DE-YOLO [30] have confirmed its effectiveness in object detection. Unlike previous work, we have redesigned a network for text detection that processes both high and low-frequency information after Laplacian pyramid decomposition [11]. This includes designing a multi-scale convolution module to expand the receptive field when dealing with low-frequency information. When processing high-frequency information, we introduce the SFT [34] module, which can reconstruct high-resolution images with rich semantic regions by transforming the intermediate features of the network with only one forward pass, as illustrated in Fig. 3.

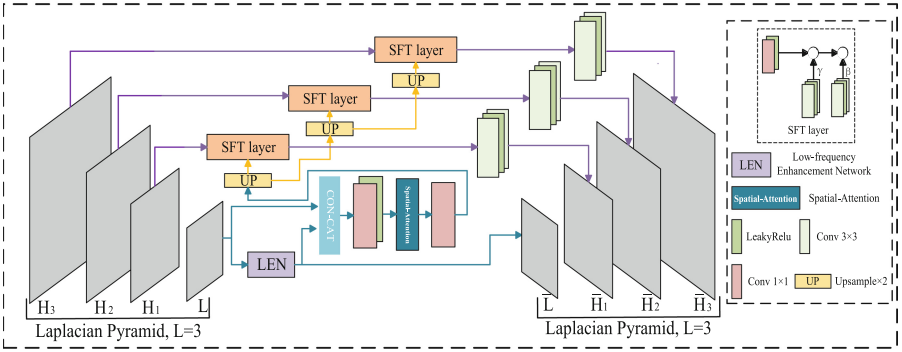
We employed Laplacian pyramid decomposition to dissect an input image with dimensions  $H \times W$  into a low-frequency component and three high-frequency components. The low-frequency component and high-frequency components for the  $i$ th decomposition layer of the Laplacian pyramid ( $1 \leq i < N$ ) are computed based on the following equations:

$$L = G_N(I) \tag{1}$$

$$H_i = G_i(I) - B(\text{upsample}(G_{i+1}(I))) \tag{2}$$

here  $B(\bullet)$  denotes the application of a two-dimensional Gaussian kernel with a size of  $5 \times 5$ .  $N$  denotes the total number of decomposition levels. And





**Fig. 3.** An illustration of the Image Enhancement Network (IENet).



**Fig. 4.** The Laplace pyramid decomposition image detail display.

“upsample” refers to enlarging an image by a factor of 2.  $G_i(I) \in \mathbf{R}^{\frac{h}{2^{i-1}} \times \frac{w}{2^{i-1}} \times 3}$  represents the  $i$ th level of image in Gaussian pyramid [11], which can be expressed as:

$$G_i(I) = \begin{cases} I & i = 1 \\ \text{downsample}(B(G_{i-1}(I))) & 2 \leq i \leq N \end{cases} \quad (3)$$

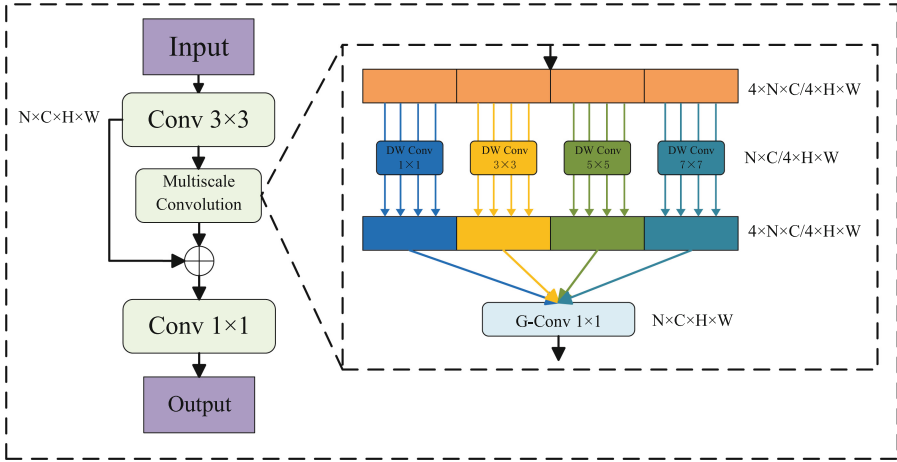
where “downsample” refers to reducing an image by a factor of 2. As evident from equations (1)–(3), the decomposition process is entirely reversible.  $H_1$ ,  $H_2$ ,  $H_3$ , and  $L$  after Laplacian pyramid decomposition are shown in Fig. 4.

Equation (3) reveals that the  $N$ th layer has the lowest resolution, which contains the global information of the image and removes a lot of high-frequency noise. Consequently, inspired by Lin et al. [35], we designed a Low-frequency Enhancement Network (LEN) to learn its underlying features as much as possible, as illustrated in Fig. 5. Our proposed Multiscale Convolution can be summarized as follows:

$$F_{out} = G_{1 \times 1}(\text{Concat}(DW_{k_i \times k_i}(x_i), \dots, DW_{k_n \times k_n}(x_n))) \quad (4)$$

where  $x = [x_1, x_2, x_3, x_4]$  means to split up the input feature  $x$  into multiple heads in the channel dimension and  $k_i \in \{1, 3, 5, 7\}$  denotes the kernel size.  $DW$  denotes depth-wise separable convolutions.  $G_{1 \times 1}$  represents grouped convolution using a  $1 \times 1$  kernel size.





**Fig. 5.** An illustration of the Low-frequency Enhancement Network (LEN).

Equation (2) shows that  $H_i$  consists of residual details of the HF component. From high decomposition to low decomposition, the coarse to fine details of the image are stored separately in  $\{H_i\}$ . We spliced the low frequency component with the feature processed by LEN and then employed spatial attention mechanisms [36] to direct the network’s focus towards text regions. Subsequently, we gradually upsample them to improve the image enhancement.

We generate new HF components by combining the processed LF information with the HF components using an affine transformation [34]. This transformation is primarily used for high-resolution reconstruction, and its formula can be expressed as follows:

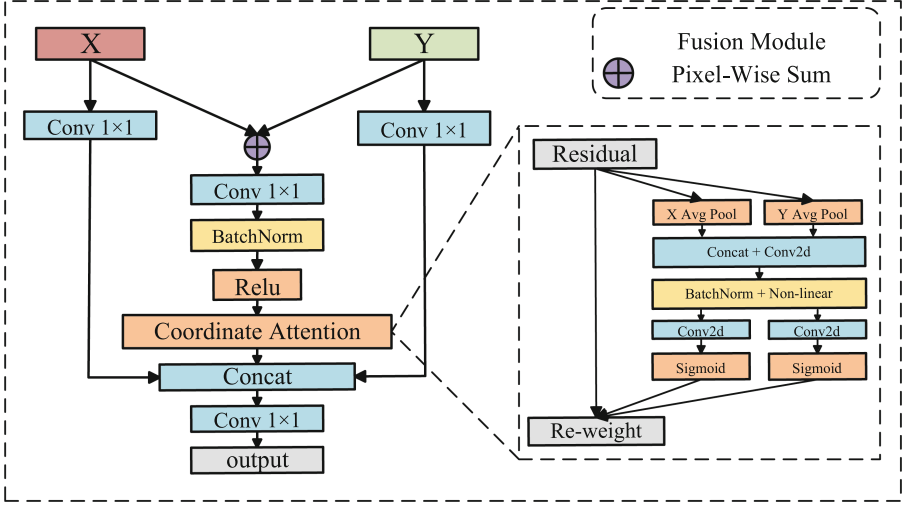
$$\text{SFT}(F_i | \alpha_i, \beta_i) = \alpha_i \odot F_i + \beta_i \tag{5}$$

where  $F_i$  denotes high frequency features.  $\odot$  denotes element-wise multiplication.  $\alpha_i$  and  $\beta_i$  correspond to the scale and shift parameters for the  $i$ th decomposition layer. Through the newly acquired LF and HF component, we gradually merge and enhance the image.

### 3.2 Fusion Module Based on Coordinate Attention

An illustration of the FMCA is shown in Fig. 6. FMCA combines the features from the original branches and the enhanced branches. Then the multi-scale features are fused by FPN. We believe that features at the same scale exhibit stronger correlations and are more effective in fusing fog-invariant characteristics. Further integrating fog-invariant features across different scales can help us achieve precise localization of scene text.

Specifically, we add the features from the two branches point by point to obtain the fusion features. These fused features are subjected to  $1 \times 1$  convolution



**Fig. 6.** An illustration of the Fusion Module based on Coordinate Attention (FMCA).

layers, BN layers, and ReLU activation functions to reduce their channel size. After that, we use coordinate attention [12] to capture long-range dependencies while preserving precise positional information. We adjust the channel size for both the original feature and the enhanced feature, both of which are reduced by a  $1 \times 1$  convolution. Finally, the spliced channels are restored to the original channel size through the  $1 \times 1$  convolution layer.

### 3.3 Loss Function

IA-YOLO [9] demonstrates that combining image recovery loss with detection loss during object detection training can lead to longer training times, difficulty in achieving convergence of the total training loss, and a reduction in detection accuracy. Therefore, following the IA-YOLO approach, we define the loss function of the proposed method solely based on detection loss as follows:

$$L_{total} = L_s + \alpha L_b + \beta L_t \quad (6)$$

where  $L_s$  denotes the probability map loss,  $L_b$  is the binary map loss, and  $L_t$  stands for the threshold map loss. Note that  $L_s$  is the balanced cross-entropy loss,  $L_b$  and  $L_s$  are equal. In this work, we retained the settings from DBNet [6] with  $\alpha$  and  $\beta$  set to 1.0 and 10.

## 4 Experiment

### 4.1 Datasets

We utilize two datasets for scene text detection under foggy weather.

**Table 1.** Comparison of the proposed method to the state-of-the-art on the HTT and REF datasets. Note that all models were trained on HTT datasets, and all results are represented by the metric F-measure.

Method	Venue	HTT	REF
PAN [20]	ICCV'19	74.0	37.4
PSENet [19]	CVPR'19	77.5	25.4
DBNet [6]	AAAI'20	79.6	50.1
FCENet [40]	CVPR'21	73.6	25.9
DB++ [41]	TPAMI'22	79.8	50.0
TCM [42]	CVPR'23	80.5	38.3
TBIA-DBNet (Ours)	-	<b>82.2</b>	<b>60.9</b>

HTT is a synthetic fog dataset that is rendered by the synthetic fog algorithm [9] for Total-Text [37]. The dataset comprises 1,255 training images and 300 test images. All text instances are annotated with word-level polygon markings. Real-English-Fog (REF) consists of 204 real foggy images that we collected under real-world foggy weather conditions.

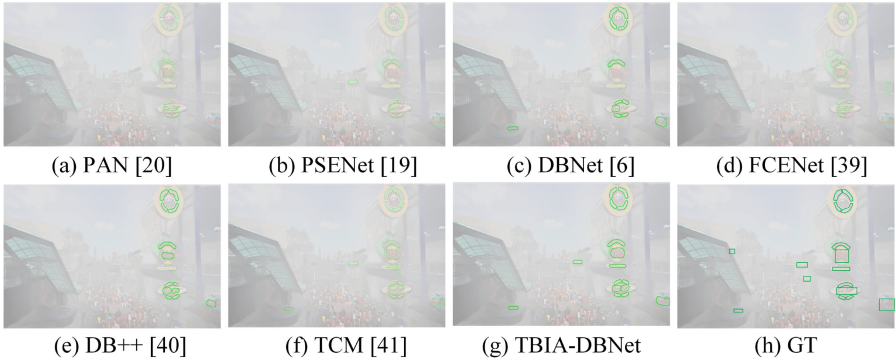
## 4.2 Implementation Details

In this paper, all the models utilise a pretrained ResNet50, which was trained on the SynthText dataset [38] for 100,000 iterations. Subsequently, the models are fine-tuned on particular datasets for 1,200 epochs. All the experiments are performed using SGD optimizer. The training batch size is set to 16. We employ a “poly” policy [39] with the initial learning rate of 0.007, and use a weight decay of 0.0001 and a momentum of 0.9. We use the same data augmentation as DBNet [6]. Our experiments were conducted by the PyTorch-based torchvision detection framework. All models are trained and tested using two NVIDIA 3090 GPUs.

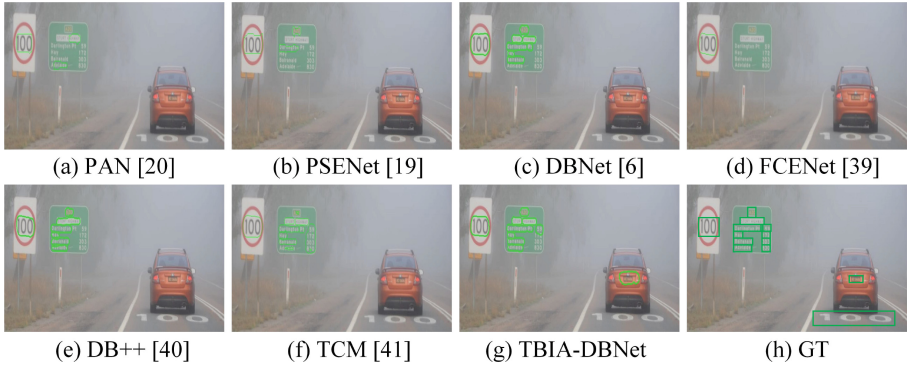
## 4.3 Comparisons with State-of-the-Art Methods

In this paper, we choose DBNet as the baseline model. Our experimental results are depicted in Table 1, compared to the baseline, TBIA-DBNet exhibits a remarkable 2.6% improvement in F-measure on the HTT datasets. TBIA-DBNet’s performance improvement is even more pronounced on real-world foggy weather datasets, with an impressive 10.8% increase. These enhancements substantially outperform state-of-the-art methods, highlighting TBIA-DBNet’s outstanding text detection performance in foggy conditions. These results validate that our approach significantly enhances the scene text detection network’s feature extraction capacity under foggy weather.

As shown in Fig. 7 and Fig. 8, we provide some representative qualitative samples, including synthetic foggy images and real-world foggy scenes images.



**Fig. 7.** Visualization results of TBIA-DBNet in comparison to other methods on the HTT datasets.



**Fig. 8.** Visualization results of TBIA-DBNet in comparison to other methods on the REF datasets.

The results indicate that the TBIA-DBNet demonstrates superior performance in foggy weather conditions.

#### 4.4 Ablation Study

To evaluate the effectiveness of each module in our proposed framework, we performed ablation experiments with different settings and assessed the network’s performance on two separate test datasets.

As show in Table 2, we use TBIA-DBNet to represent our proposed network framework. Compared to baseline, when we combined IENet into DBNet, the F-measure improved by 1.5% and 7.4%, respectively, on the HTT and REF datasets. This shows that IENet effectively reveals more potential features conducive to text detection, and greatly improves the accuracy of text detection in foggy weather. In the second line, we sum distinct image features from separate branches. and F-measure improved by 2% and 9% respectively on HTT and

**Table 2.** Ablation study on various modules of our approach on the HTT and REF datasets.

IENet	Two-Branch	FMCA	HTT	REF
✗	✗	✗	79.6	50.1
✓	✗	✗	81.1	57.5
✓	✓	✗	81.6	59.1
✓	✓	✓	<b>82.2</b>	<b>60.9</b>

**Table 3.** Image Enhancement Network and several advanced dehaze networks were subjected to ablation experiments on the HTT and REF datasets.

Methods	HTT	REF
DBNet [6]	79.6	50.1
AODNet [7] -DBNet	80.2	52.4
DM [10] -DBNet	80.6	54.2
DENet [30] -DBNet	80.8	55.8
IENet-DBNet	81.1	57.5
TBIA-DBNet (Ours)	<b>82.2</b>	<b>60.9</b>

REF datasets. This shows that the two-branch network effectively improves the complementarity of the two branches and avoids the performance degradation of the single-branch network that may be caused by the omission of important potential features in the dehaze network. In the third line, the text detection performance is optimal when we combine the features of the two branches and replace the sum with FMCA. Compared with the sum method, the F-measure of FMCA is improved by 0.6%, and 1.8% on the HTT, and REF datasets.

We compared our method with several popular defogging networks combined with detection networks. As shown in Table 3. Importantly, for quantitative comparison in this paper, we replaced the detection models in other methods with our baseline. The results indicate that under single-branch conditions, IENet-DBNet achieved the highest F-measure. The reason is that, unlike other dehazing networks, IENet directly uses the Laplacian pyramid to decompose foggy images into LF and HF components, rather than relying on atmospheric scattering models [43]. We believe that methods using the atmospheric light scattering model to predict clear images can introduce noise detrimental to detection. The reason is that even clear regions in the image without fog are forcibly restored, which may introduce noise and negatively impact detection. Although DENet [30] considers the influence of the atmospheric light scattering model, its network is implemented based on object detection. Additionally, the last row shows that our proposed TBIA-DBNet achieves the highest F-measure, indicating that the two-branch network structure we propose is more effective than the typical single-branch network structure.

## 5 Conclusion

In this paper, we introduce TBIA-DBNet, a novel two-branch detection frameworks designed for scene text detection in foggy weather. The proposed IENet can effectively remove weather-specific information and reveal more latent information. Furthermore, through the design of FMCA, we seamlessly integrate both original and enhanced features, effectively improving the complementarity and richness of the target features. Our proposed approach is capable of adaptively handling real-world foggy weather conditions. It is a robust framework that bridges low-level image enhancement with high-level vision tasks, allowing for the seamless replacement and extension of each module. A significant number of experiments have demonstrated the effectiveness of our method. In the future, we are interested in expanding our approach to enable text recognition in challenging weather conditions.

## References

1. 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)
2. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9365–9374 (2019)
3. Dai, P., Zhang, S., Zhang, H., Cao, X.: Progressive contour regression for arbitrary-shape scene text detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7393–7402 (2021)
4. Liao, M., Pang, G., Huang, J., Hassner, T., Bai, X.: Mask TextSpotter v3: segmentation proposal network for robust scene text spotting. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12356, pp. 706–722. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58621-8\\_41](https://doi.org/10.1007/978-3-030-58621-8_41)
5. Sheng, T., Chen, J., Lian, Z.: CentripetalText: an efficient text instance representation for scene text detection. In: Advances in Neural Information Processing Systems, vol. 34, pp. 335–346 (2021)
6. 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)
7. Li, B., Peng, X., Wang, Z., Xu, J., Feng, D.: AOD-Net: all-in-one dehazing network. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4770–4778 (2017)
8. Huang, S.C., Le, T.H., Jaw, D.W.: DSNet: joint semantic learning for object detection in inclement weather conditions. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(8), 2623–2633 (2020)
9. Liu, W., Ren, G., Yu, R., Guo, S., Zhu, J., Zhang, L.: Image-adaptive yolo for object detection in adverse weather conditions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 1792–1800 (2022)
10. Li, C., et al.: Detection-friendly dehazing: object detection in real-world hazy scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* (2023)





11. Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image code. In: *Readings in Computer Vision*, pp. 671–679. Elsevier (1987)
12. Hou, Q., Zhou, D., Feng, J.: Coordinate attention for efficient mobile network design. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13713–13722 (2021)
13. 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)
14. 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)
15. 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)
16. Liu, W., et al.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
17. Liao, M., Shi, B., Bai, X.: TextBoxes++: a single-shot oriented scene text detector. *IEEE Trans. Image Process.* **27**(8), 3676–3690 (2018)
18. Zhang, C., et al.: Look more than once: an accurate detector for text of arbitrary shapes. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10552–10561 (2019)
19. 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)
20. 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)
21. 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)
22. Xie, E., Zang, Y., Shao, S., Yu, G., Yao, C., Li, G.: Scene text detection with supervised pyramid context network. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9038–9045 (2019)
23. Du, B., Ye, J., Zhang, J., Liu, J., Tao, D.: I3CL: intra-and inter-instance collaborative learning for arbitrary-shaped scene text detection. *Int. J. Comput. Vision* **130**(8), 1961–1977 (2022)
24. Tian, J., Zhou, G., Liu, Y., Deng, E., Jia, Z.: FTDNet: joint semantic learning for scene text detection in adverse weather conditions. In: Fink, G.A., Jain, R., Kise, K., Zanibbi, R. (eds.) *ICDAR 2023*. LNCS, vol. 14191, pp. 137–154. Springer, Cham (2023). [https://doi.org/10.1007/978-3-031-41734-4\\_9](https://doi.org/10.1007/978-3-031-41734-4_9)
25. Guo, C., et al.: Zero-reference deep curve estimation for low-light image enhancement. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1780–1789 (2020)
26. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(12), 2341–2353 (2010)
27. Liu, X., Ma, Y., Shi, Z., Chen, J.: GridDehazeNet: attention-based multi-scale network for image dehazing. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7314–7323 (2019)
28. Dong, H., et al.: Multi-scale boosted dehazing network with dense feature fusion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2157–2167 (2020)

29. Qin, X., Wang, Z., Bai, Y., Xie, X., Jia, H.: FFA-Net: feature fusion attention network for single image dehazing. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11908–11915 (2020)
30. Qin, Q., Chang, K., Huang, M., Li, G.: DENet: detection-driven enhancement network for object detection under adverse weather conditions. In: Wang, L., Gall, J., Chin, T.J., Sato, I., Chellappa, R. (eds.) ACCV 2022. LNCS, vol. 13843, pp. 2813–2829. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-26313-2\\_30](https://doi.org/10.1007/978-3-031-26313-2_30)
31. 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)
32. 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)
33. Liang, J., Zeng, H., Zhang, L.: High-resolution photorealistic image translation in real-time: a Laplacian pyramid translation network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9392–9400 (2021)
34. Wang, X., Yu, K., Dong, C., Loy, C.C.: Recovering realistic texture in image super-resolution by deep spatial feature transform. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 606–615 (2018)
35. Lin, W., Wu, Z., Chen, J., Huang, J., Jin, L.: Scale-aware modulation meet transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6015–6026 (2023)
36. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 3–19. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01234-2\\_1](https://doi.org/10.1007/978-3-030-01234-2_1)
37. Ch'ng, C.K., Chan, C.S.: Total-text: a comprehensive dataset for scene text detection and recognition. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 935–942. IEEE (2017)
38. 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)
39. Liu, W., Rabinovich, A., Berg, A.C.: ParseNet: looking wider to see better. arXiv preprint [arXiv:1506.04579](https://arxiv.org/abs/1506.04579) (2015)
40. 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)
41. 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)
42. Yu, W., Liu, Y., Hua, W., Jiang, D., Ren, B., Bai, X.: Turning a CLIP model into a scene text detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6978–6988 (2023)
43. McCartney, E.J.: Optics of the atmosphere: scattering by molecules and particles, New York (1976)





# Breaking Boundaries: Enhancing Script Identification Using a Learnable MULLER Resizer

Souhaila Djaffal<sup>1</sup> (✉) , Yasmina Benmabrouk<sup>1</sup> , Chawki Djeddi<sup>2,3</sup> ,  
and Moises Diaz<sup>4</sup> 

<sup>1</sup> Laboratory of Mathematics, Informatics and Systems (LAMIS), Echahid Cheikh Larbi Tebessi University, Tebessa, Algeria

souhaila.djaffal@univ-tebessa.dz

<sup>2</sup> Laboratoire de Vision et d'Intelligence Artificielle (LAVIA), Université Echahid Cheikh Larbi Tebessi, Tébessa, Algérie

<sup>3</sup> Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes (LITIS), Université de Rouen-Normandie, Rouen, France

<sup>4</sup> Instituto Universitario para el Desarrollo Tecnológico y la Innovación en Comunicaciones, Universidad de Las Palmas de Gran Canaria, Campus de Tafira, Las Palmas de Gran Canaria, Spain

**Abstract.** Effective script identification is pivotal in document analysis and recognition, especially when dealing with hybrid multiscript documents at fine-grained levels. Recent advancements in deep learning models have revolutionized document analysis and recognition tasks, leveraging script document images. However, these images often require resizing, which can result in data loss. This paper explores the impact of MULLER, a learnable resizer on a hybrid word-level script identification task using the Multi-lingual and Multi-script Documents In the Wild (MDIW-13) dataset. Our approach integrates MULLER resizer with a pre-module that employs k-means clustering to determine the optimal target size. When jointly trained with MobileNet, it achieves an impressive average accuracy of 98.16%. In summary, our findings underscore the potential of the MULLER resizer to outperform conventional resizers, thereby evaluating script classification performance.

**Keywords:** Hybrid word level Script identification · MULLER resizer · conventional resizers · MDIW-13 dataset · MobileNet model

## 1 Introduction

Automatic script identification is a critical component of document image analysis. This process involves identifying a document's script (writing system) before

---

This research was made possible using funding from the Algerian Ministry of Higher Education and Scientific Research through PRFU project No. C00L07UN120120190001 and the Spanish Ministry of Economy and Competitiveness through project No. PID2019-109099RB-C41.

the Optical Character Recognition (OCR) process. In a multiscript environment, document handling necessitates a script identifier and a collection of OCRs. The process involves detecting the document’s script and then employing the appropriate OCR (one OCR per script) [1]. Script identification can provide benefits to various industries, including scene understanding [2], multi-lingual machine translation [3], and video script identification [4].

Documents can be classified into three categories based on the type of information they contain [5]. These include machine-printed documents, which consist of typed and printed-out text, handwritten documents, which contain handwritten text, and hybrid documents, which feature both printed and handwritten text blocks on the same page in many practical documents, such as bank checks, forms, and letters [6, 7]. Handling hybrid documents effectively requires adaptable systems due to their multifaceted content.

Script identification can be performed at different levels: page, block, line, and word. However, script identification at the word level is particularly challenging as it provides less information than at other levels. This level requires identifying individual words, which can be particularly difficult in cases where words are closely connected or overlap. The success of the script identification process at this level depends heavily on the accuracy of the segmentation algorithm; any segmentation errors can significantly impact the accuracy of the results. Another challenge arises from the quality of the input image can significantly impact the accuracy of word-level script identification, with factors such as low resolution, poor lighting, and skewing potentially compromising the reliability of the results [8].

Script images contain a significant amount of fine-grained valuable information in tiny connected components (characters, stress, diacritics, etc.). Still, their large and varying spatial size can pose a challenge for many computer-assisted models [9].

Convolutional neural networks have significantly transformed the field of computer vision in recent years [10]. However, one crucial aspect that has been surprisingly overlooked is the impact of image size on the accuracy of the tasks being trained for. Typically, for efficiency reasons, input images are resized to a small spatial resolution (e.g.  $224 \times 224$ ), and both training and inference phases [10]. Considering memory limitations, training CNN models with high or arbitrary resolutions might not be practical, requiring image resizing to a uniform size to adapt deep learning models. Consequently, resizing is usually essential to successfully implement deep learning models for script document images. Conventional resizing techniques like nearest neighbor interpolation [11], bilinear [11], and bicubic [11] could lead to data loss and artifacts, as they do not sustain the fine-grained details of the original image [12].

To overcome this limitation, researchers have proposed learned resizers that leverage deep neural networks to learn image resizing directly from data, yielding improved performance on several tasks. For example, the authors in [13] proposed a residual CNN module for downscaling and jointly trained it with an image compression network to generate “compression-friendly” representa-

tions. Hossein and Peyman [10] introduced a CNN-based learned resizer that is jointly trained with classification models to improve their performance. It handles any arbitrary scaling factor, including up and down-scaling. This allows to explore the resolution versus batch size trade-off and as a result to find the optimal resolution for the task in hand. Similarly, the idea of learned rescaling has been applied to other computer vision applications [14] showing improved performance in detection and recognition.

However, one of the main challenges with these learned resizers is that they often require a large number of parameters, and high computational overhead during training and inference. To mitigate this issue, authors in [12] introduce an incredibly lightweight learned resizer called MULLER, that operates on the multilayer Laplacian decomposition of images. MULLER requires very few parameters and does not incur any extra training cost, outperforming existing methods in terms of computational efficiency, parameter efficiency, and transferability. MULLER resizer only learns four parameters and is more effective than previous complex ones using deep residual blocks.

This paper presents an adaptive resizer-based transfer learning framework for classifying the word-level MDIW-13 dataset with hybrid documents incorporating the use of Kmeans clustering algorithm to determine the optimal target size, along with a lightweight learned resizer module, known as the MULLER trained conjointly with the baseline MobileNet [23] architecture. The effectiveness of the MULLER resizer is evaluated against three conventional resizing methods: Nearest Neighbor Interpolation, Bilinear, and Bicubic on various resolutions. Our results provide valuable insights into the performance of MULLER as a learned resizing model on this dataset and underscore the importance of adopting an adaptive resizer to enhance the model’s precision.

The novelty of this work lies in the integration of the MULLER resizer with a pre-module employing k-means clustering for optimal target size determination. This approach provides a studied target size that offers guidance to the resizing module, mitigating the issue of image degradation that typically arises from resizing all images to a single, and generalized target size. Specifically applied to the hybrid word-level script identification task, this method is the first to utilize a learnable resizer within this context, significantly enhancing script classification performance.

The rest of the paper is organized as follows: Section 2 provides a general overview of the prior literature on hybrid word-level script identification and learned resizing. Section 3 describes the proposed adaptive resizer-based transfer learning framework for hybrid word-level script identification. Section 4 discusses the outcomes derived from applying the proposed approach compared to the Conventional resizing technique, providing a comprehensive analysis of the approach’s performance. Finally, Sect. 5 wraps up the paper by summarizing key findings and providing suggestions for future research directions.

## 2 Related Works

### 2.1 Hybrid Word-Level Script Identification

Most research in the field of script identification concerns printed or handwritten documents. However, since several documents may contain text blocks with both printed and handwritten texts, some research is now addressing hybrid documents. Despite the progress made in this field, further investigation and the development of more accurate and efficient approaches for word-level script identification from hybrid documents still need to be further investigated.

Asma Saidani et al. [15] presented an approach for Arabic and Latin script identification based on Histogram of Oriented Gradients (HOG), Pyramid HOG, and co-occurrence matrices of HOG descriptors, along with a genetic algorithm to select the combinations of the informative features. Experimental results show a good classification rate of 99.07% using Bayes-based classifier.

By using new structural features which are intrinsic features, a successful attempt was made by [16] to identify the Arabic or Latin script. Experiments have been conducted with 1320 handwritten and printed words, covering a wide range of fonts, achieving a correct classification rate of 98.4% using Bayes classifier.

Authors in [17] proposed an accurate system based on a steerable pyramid transform for Arabic and Latin script identification at word-level. The S.P parameters tested are *sp0filter*, *sp3filter*, and *sp5filter* with respectively 2, 4, and 6 orientations and 1, 2, 3, and 4 levels, with *sp3filter* with 4 orientations and 2 levels being the best. The overall correct identification rate obtained is about 97.5% using the K nearest neighbors classifier with  $K = 5$ .

The study [18] focuses on the problem by designing a dual-branch structured deep convolutional neural network (CNN). For the training stage, a two-stage multi-task learning strategy to learn robust shared features. Furthermore, three CNN networks of different scales (small, medium, and large) are evaluated to determine the best CNN architecture. The accuracy achieved by the two-stage multi-task CNN is 95%.

In [19] a deep learning method is proposed that features a set of optimized convolutional layers followed by recurrently connected layers to identify the script of any word sample. The experiments were conducted on MDIW-13 and PHDIndic-11 datasets, achieving a correct identification rate of 97.57% on MDIW-13 dataset and 96.15% on PHDIndic-11 dataset.

While earlier researches [15–17] have made progress in this field, it has been limited to Arabic and Latin scripts using handcrafted features to achieve promising results. In more recent studies [18, 19], deep learning techniques have been employed, achieving even stronger performance across a broader range of scripts. All previously mentioned works use conventional resizing methods. This is a significant breakthrough because script document images are often challenging due to the multitude of tiny connected components, including characters, stress, diacritics, and more, each containing valuable fine-grained information and varying spatial sizes.

## 2.2 Learned Resizing

Learned resizers have been evaluated for several domains and scopes. Some recent works have explored the use of learning-based methods of image downscaling to enhance the desired content in the resized images from training data.

Xupeng and Yuehui [20] applied the learnable resizer proposed in [10] to COVID-19 lung CT image classification, jointly training the MobileNet model with the learnable resizer. The researchers compared their results with those of different models, namely VGG19, Resnet50\_v2, MobileNet, Inception\_v3, and Densenet169. The jointly trained MobileNet model with the learnable resizer achieved an accuracy of 96.9%, sensitivity of 98.3%, and specificity of 95.3%, outperforming other models. Notably, the jointly trained MobileNet model with the learnable resizer only had 30,000 more parameters than the MobileNet model.

Authors in [21] illustrated the influence of the learnable adaptive resizer [10] on breast cancer classification using the BreakHis dataset. The proposed approach incorporates the adaptive resizer with various convolutional neural network models, including VGG16, VGG19, MobileNetV2, InceptionResnetV2, DenseNet121, DenseNet201, and EfficientNetB0. Despite producing visually less appealing images, the learnable resizer effectively improves classification performance. DenseNet201, when jointly trained with the adaptive resizer achieves the highest accuracy of 98.96% for input images of  $448 \times 448$  resolution.

This work [22] introduces a zero-shot diffusion-based video generator aiming to accurately produce animal animations while preserving the background. AnimateZoo includes two steps: first improving appearance feature extraction by integrating a Laplacian detail booster and a prompt-tuning identity extractor while maintaining low computational overhead and preserving detailed information using a trainable Laplacian resizer, MULLER [12]. Extensive experiments showcase the outstanding performance of the proposed method in cross-species action following tasks, demonstrating exceptional shape adaptation capability.

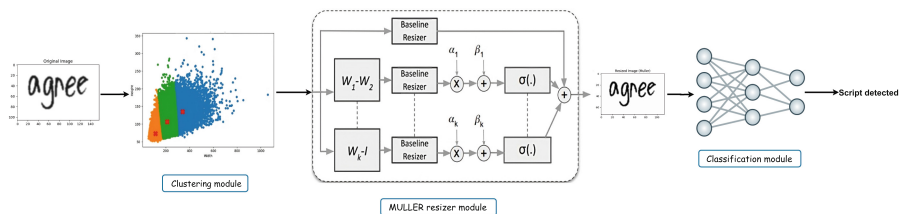
The above works show that image pre-processing, in terms of resolution, downscaling, and joint resizing has a great impact on the performance of vision models. However, these methods have not yet proven their effectiveness in the field of script identification and document analysis tasks. To the best of our knowledge, the current academic discourse has not extensively examined the impact of adaptively learned resizers, especially MULLER resizer [12], on the performance of any document analysis tasks precisely script identification. Therefore, our research aims to address this gap by investigating the effects of MULLER resizer on the effectiveness of hybrid word-level script identification tasks.

## 3 Learned Resizer-Based Transfer Learning Framework for Hybrid Word-Level Script Identification

In the field of script identification, actual systems need to collect images from the wild with various sizes. However, they must be resized to a standard size

to fit deep-learning training tools. This is an inevitable compromise for computational resources and the training framework, as the resizing procedure causes information loss, negatively affecting model performance. To address this issue, this study proposes an adaptive learnable resizer-based transfer learning framework, introduced in detail in this section.

The proposed hybrid word-level script identification system comprises a clustering module using Kmeans algorithm, a resizing module based on the MULLER resizer, and a lightweight classifier using MobileNet (see Fig. 1). Initially, we preceded the resizing module with a clustering step using the Kmeans algorithm with  $K$  set to 3. Because of the extreme variations in image sizes (see Fig. 2 and Fig. 3), resizing all images to a general size would cause additional loss and deformation of essential information. Therefore, using a clustering module to group the images into heterogeneous groups regarding height and width and defining the most suitable size for each group is considered a plus to enhance the overall pipeline.



**Fig. 1.** The architecture of the proposed system. The resizer decomposes the input image into multiple layers of Laplacian residuals and then adds them back to the default resized image. After choosing the most suitable target size using Kmeans, the MULLER resizer is trained along with the MobileNet model to get the script category.

After that, the resizing module used is based on MULLER resizer which consists of two layers, a Gaussian kernel size of 5, and a standard deviation of 1, with a bilinear resizer as the base resizer method. MULLER is adopted to boost the detail quality of low-resolution images. It combines various filters of distinct frequencies to boost the sample ratio of crucial frequencies for the task at hand, such as edge, detail, and sharpness information. This adjustment involves merely four trainable parameters as weights and biases, rendering the trainable resizer can be applied to our pipeline almost without cost.

Finally, the Resizer module is trained in conjunction with a lightweight classifier MobileNet [23] model pre-trained on ImageNet, which takes the resized image as input. This network focuses on neural network models for tasks on mobile devices, the output layer comprises a dense layer with 13 nodes, equipped with a softmax activation function to predict the final probability of the script

class. During the training phase, we used Adam optimizer with a lower learning rate ( $1e-4$ ) to preserve the valuable features learned during pre-training (see Fig. 1).

### 3.1 Dataset

The MDIW-13 [1, 24] is the largest publicly available multi-lingual and multi-script dataset for script identification, comprising 113 printed and handwritten documents that were scanned from the local newspaper and handwritten letters and notes, segmented to over 13979 lines and 86655 words from a gigantic variety of widely used scripts (13 scripts), namely Arabic, Bengali (Bangla), Gujarati, Gurmukhi, Devanagari, Japanese, Kannada, Malayalam, Oriya, Roman, Tamil, Telugu, and Thai. Nevertheless, the dataset’s arbitrary image sizes present several challenges, making it difficult to identify scripts properly.

MDIW-13 provides a comprehensive evaluation framework for our proposed method. By leveraging this extensive dataset, we are able to demonstrate the effectiveness and robustness of the MULLER resizer in real-world scenarios involving diverse and complex script documents.

As shown in Fig. 2 and Fig. 3, The dataset has a significant variation in image size, making it difficult to find a one-size-fits-all solution for all 13 scripts included in the dataset due to the unique nature of each script’s small components. It is important to note that resizing or padding images can distort specific small components unique to each script, adversely affecting the identification system’s performance [9].

### 3.2 Clustering Module: Kmeans

Our experimental dataset comprises images of arbitrary sizes as depicted in Fig. 2. This can lead to the loss of discriminative information if all images are downsampled or upsampled to the same target size. K-means clustering was applied to group the images into three clusters to mitigate this issue. These clusters correspond to three different target sizes, as shown in Fig. 3.

In our study, we employed the K-means clustering algorithm to determine the most appropriate target size for each group of images, to obtain a carefully considered target size, addressing the extreme variations observed in word image sizes and mitigating the potential loss of distinctive word forms that could occur with a fixed general target size. We set the number of clusters,  $k$ , to 3, based on manual observation of the word images, categorizing them into three groups: too small, medium, and large word image size.

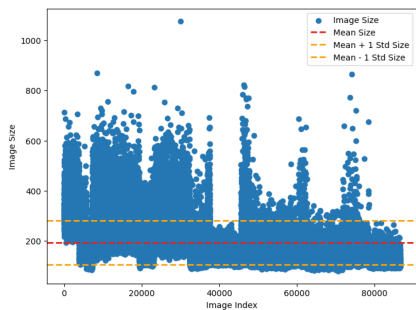


Fig. 2. Images sizes distribution.

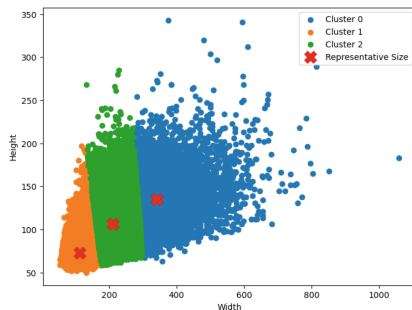


Fig. 3. Clustering of image sizes.

### 3.3 Learned Resizing Module: MULLER

We employ a compact multilayer Laplacian resizer MULLER introduced by [12] that converts images with higher input resolution into the baseline network input. The purpose of MULLER is to dynamically adjust the input image size while decomposing the input image into various Laplacian residual layers, improving image contrast and emphasizing its details and textures, as shown in Fig. 1. It has the following form:

$$z = R(x) + \sum_{l=1}^k \sigma(\alpha_l (R((W_l - W_{l+1})X) + \beta_l)) \quad (1)$$

where  $R$  denotes the base resizer (e.g. bilinear) and  $W_1, W_2, \dots, W_k$  represents the low-pass filter basis. We define  $W'$  as a positive row-stochastic matrix of size  $n \times n$ , with  $n$  representing the number of pixels in the vectorized input image  $x$ . Note that we assume  $W_{k+1} = I$ , where  $I$  is the identity matrix. In Eq. (1), each layer uses a variety of filters to decompose the image into distinct detail layers through bandpass filtering, as shown in Fig. 1.

The MULLER resizer model employs a base resizer to handle input images and applies filters at different scales to capture details across various frequencies. This iterative filtering process selectively enhances specific frequency components, boosting the overall level of detail in the image. Using multilayer Laplacian decomposition, MULLER targets specific frequency ranges like edges and textures to enhance image quality and informativeness. By concentrating on frequency subbands, MULLER improves the quality of resized images used in the identification task.

In our experiments, we used MULLER with two layers, a Gaussian kernel size of 5, and a standard deviation of 1, with a bilinear resizer as the base resizer method. In particular, the MULLER module is trained in conjunction with the recognition model to better adapt to the model architecture and get excellent results. The resizer receives an image from the data pipeline and adjusts its size before it is input into the model. MULLER is designed to address the challenge of resizing images while preserving their content and maintaining visual quality.





Fig. 4. Resizing resulting image per cluster.

It is crucial to sustain as much detail as possible when dealing with script text images, especially for fine-grained levels like word level, which include important little details. Traditional resizing of images may lead to the loss of fine details and bad image quality, which can be problematic when working with lower levels and with handwritten word images (see Fig. 4). MULLER resizer approaches this drawback by conserving the image’s unique characteristics guaranteeing critical details are preserved during the resizing operation.

As demonstrated in Fig. 4, MULLER maintains finer details and text clarity while producing smoother transitions and fewer artifacts in both downsampling and upsampling scenarios. This exceptional ability to preserve image quality makes the MULLER method ideal for demanding tasks such as script text identification.

## 4 Results and Discussion

Our research findings, as summarized in Tables 1, 2, and 3 demonstrate the effectiveness of our proposed approach across different clusters. Specifically:

1. **Cluster 1 (Large Target Sizes: (135,341))**: Achieved an impressive accuracy rate of 98.57% with a precision rate of 99%. In this context, Compared to traditional methods, MULLER’s performance shows moderate percentage improvements, suggesting that while it handles large resizing tasks effectively, the gap between MULLER and traditional methods is not as wide. This indicates that all methods are fairly robust when resizing to larger dimensions, but MULLER maintains a consistent edge.
2. **Cluster 2 (Small Target Sizes: (73,114))**: Maintained high accuracy at 98.48% with a precision rate of 98.68%. The improvements in percentage are more pronounced for MULLER with a larger improvement margin suggesting that as the target resizing size decreases, MULLER’s advantages become more apparent. Traditional methods seem to struggle slightly more with smaller dimensions, whereas MULLER’s advanced algorithms better preserve image quality and details.
3. **Cluster 3 (Medium Target Sizes: (106,212))**: Showed robust performance with an accuracy rate of 97.44% accompanied by a precision rate of 98.02%. In Cluster 03, MULLER still performs strongly, this indicates a consistent but slightly reduced margin of improvement compared to Cluster 02. The intermediate size may offer a balance where traditional methods perform relatively well, but MULLER still provides noticeable enhancements.

**Table 1.** Resizers performance comparison for Cluster 01.

Resizer	Accuracy	Precision	Recall	F1-score
Nearest Neighbor	98%	98.61%	98%	98.26%
Bilinear	98.28%	98.64%	98.28%	98.44%
Bicubic	98.34%	98.82%	98.34%	98.56%
MULLER	<b>98.57%</b>	<b>99%</b>	<b>98.57%</b>	<b>98.77%</b>

In Table 1, the performance of different resizers is compared for Cluster 01. The MULLER resizer outperforms traditional methods across all metrics. Specifically, the MULLER resizer achieves an accuracy of 98.57%, which is higher than Nearest Neighbor 0.58%, Bilinear by 0.30%, and Bicubic by 0.23%. The precision, Recall, and F1-score follow the same trend, with MULLER attaining the highest values of 99%, 98.57%, and 98.77% respectively. This demonstrates MULLER’s ability to maintain a balance between precision and recall, ensuring better overall performance in resizing script images without losing critical details.

Table 2 presents the performance metrics for Cluster 02. Here, MULLER again shows superior performance compared to traditional resizers. With an accuracy of 98.48%, MULLER surpasses Nearest Neighbor by 1.37%, 0.06% over Bilinear, and 1.39% over Bicubic. The F1-score of 98.58% further confirms MULLER’s advantage in balancing precision and recall showing an improvement of 1.45% over Nearest Neighbor, 0.06% over Bilinear, and 1.47% over Bicubic.

**Table 2.** Resizers performance comparison for Cluster 02.

Resizer	Accuracy	Precision	Recall	F1-score
Nearest Neighbor	97.15%	97.70%	97.15%	97.37%
Bilinear	98.42%	98.61%	98.42%	98.50%
Bicubic	97.13%	97.73%	97.13%	97.40%
MULLER	<b>98.48%</b>	<b>98.68%</b>	<b>98.48%</b>	<b>98.58%</b>

These percentages indicate MULLER’s effectiveness in preserving image quality and details, particularly in more challenging scenarios.

**Table 3.** Resizers performance comparison for Cluster 03.

Resizer	Accuracy	Precision	Recall	F1-score
Nearest Neighbor	97.02%	97.69%	97.02%	97.35%
Bilinear	96.20%	97.02%	96.20%	96.53%
Bicubic	96.76%	97.49%	96.76%	97.08%
MULLER	<b>97.44%</b>	<b>98.02%</b>	<b>97.43%</b>	<b>97.72%</b>

For Cluster 03, as shown in Table 3, MULLER continues to lead. The accuracy achieved by MULLER is 97.44%, which is higher than Nearest Neighbor by 0.43%, 1.29% compared to Bilinear, and 0.70% compared to Bicubic. MULLER also records the highest precision at 98.02%, showing an enhancement of 0.73% over Nearest Neighbor, 1.58% over Bilinear, and 0.99% over Bicubic. With an F1-score of 97.72%, MULLER again proves to be superior in maintaining the equilibrium between precision and recall. F1-score improvements are 0.58% over Nearest Neighbor, 1.42% over Bilinear, and 0.85% over Bicubic.

MULLER consistently outperforms traditional methods across all target sizes. However, the performance advantage of MULLER is more significant with smaller target resizing sizes. This suggests that MULLER’s learned resizing model is particularly effective in scenarios requiring high-detail preservation in smaller dimensions, where traditional methods tend to lose more critical information.

**Table 4.** Comparison of Misclassified Samples Across Different Resizing Techniques.

Cluster	Total	Train	Test	MULLER	Nearst	Bilinear	Bicubic
1	24403	19523	4880	122 (2.5%)	155 (3.18%)	185 (3.79%)	158 (3.24%)
2	8761	7009	1752	25 (1.42%)	35 (2.00%)	30 (1.71%)	29 (1.65%)
3	53491	42793	10698	153 (1.43%)	304 (2.48%)	169 (1.58%)	307 (2.87%)

Table 4 illustrates the number of misclassified samples compared to the total number of test samples for the Muller resizer, and three conventional resizing techniques: nearest neighbor interpolation, bilinear interpolation, and bicubic interpolation. The analysis is segmented into three distinct clusters, providing insight into the performance of each resizing method across different data distributions.

The analysis clearly shows:

1. **Performance Consistency:** MULLER consistently shows the lowest misclassification rates across all clusters, indicating that is more effective compared to conventional methods.
2. **Cluster Sensitivity:** The performance gap between MULLER and the conventional methods varies by cluster. For example, in Cluster 2, the differences are relatively small, while in Cluster 3 and Cluster 1, the differences are more pronounced. This suggests that the effectiveness of learned resizing can be influenced by the number of data in each cluster.

#### 4.1 Comparison with The-State-of-Art Methods

Additionally, we conducted a thorough comparison with state-of-the-art methods. Specifically, we evaluated our system against competitors in the ICDAR 2021 Competition on Script Identification in the Wild [24], focusing on the third task related to hybrid word-level script identification. The data in this competition is a subset of a large dataset since they were randomly selected from the MDIW-13 multiscript document database [1, 24].

Table 5 summarizes the participating groups and their methods compared to our proposed system.

**Table 5.** Summary of participants and submitted approaches to SIW 2021 compared with our proposed approach. The table lists the abbreviations of the models, as used in the experimental section. PR = Pre-trained models, EX = External data, HC = Hand-crafted features, AL = Detection and alignment, EM = Ensemble models, DM = Differentiate models, Pre = Pre-processing, post = Post-processing, ✓ = Yes, × = No. [24]

Team	PR	EX	HF	AL	EM	DM	Pre	Post	Score
Ambilight	✓	✓	×	✓	×	×	✓	✓	99.84%
DLVC-Lab	✓	✓	×	×	✓	×	×	×	98.87%
NAVER Papago	✓	×	×	×	×	×	✓	×	97.17%
UIT MMLab	✓	×	×	×	✓	✓	×	✓	97.09%
CITS	✓	✓	×	×	×	×	✓	✓	94.79%
Larbi Tebessi	×	×	✓	×	×	×	×	×	83.83%
<b>Ours</b>	×	×	×	×	×	×	✓	×	98.16%

Table 5 outlines the methodologies employed by different teams, indicating whether they used pre-trained models, external data, hand-crafted features, detection and alignment techniques, ensemble models, and pre- or post-processing techniques. We can observe, that the results of the participants ranged from 99.84% to 83.83%, where we've remained third with an average accuracy of 98.16%. Ambilight, which achieved the highest accuracy of 99.84%, utilized an extensive array of methods including pre-trained models, external data, detection and alignment, and both pre- and post-processing techniques, contributing to their superior performance. Similarly, DLVC-Lab, with an accuracy of 98.87%, employed pre-trained models and ensemble techniques, demonstrating the effectiveness of combining multiple advanced methods. NAVER Papago and UIT MMLab focused on pre-processing and ensemble models respectively, achieving accuracy scores of 97.17% and 97.09%. In contrast, Larbi Tebessi, relying solely on hand-crafted features, scored the lowest at 83.83%, indicating the limitations of using fewer techniques.

Our method achieved a notable accuracy of 98.16% with the use of a single learned resizing technique as a pre-processing step. This performance is significant given the simplicity and lower computational requirements compared to more complex methods. The core strength of MULLER lies in its ability to feed non-square, varying size images directly into a resizer jointed to a classification module, eliminating the need for resizing, padding, or patching, which often leads to a loss of essential information and decreased performance. This capability is revolutionary in domains where maintaining the integrity of the original image is crucial. Our proposed approach also highlights the potential for further enhancement by addressing dataset imbalances, implementing fine-tuning, and incorporating more sophisticated modules, which could lead to even higher accuracy levels.

## 5 Conclusion and Future Work

This paper introduces a novel approach for word-level script identification for hybrid documents, leveraging the synergy between the learned MULLER resizer module and the MobileNet model. From input images of varying sizes, our approach employs K-means clustering to determine the optimal target. The results are promising: our approach achieves a remarkable average accuracy of 98.16%, surpassing the original model's performance that relied on an unlearned resizer. The success of our approach can be attributed to the learned resizing capability of the MULLER module, which preserves essential details and quality of images, making it feasible to process non-square, varying size images effectively. This is revolutionary in several domains where traditional resizing, padding, or patching solutions often lead to severe loss of critical information, causing a decrease in performance.

By addressing dataset imbalances, implementing fine-tuning, investigating other Script identification Levels, and incorporating more sophisticated modules, our method could achieve even higher accuracy levels. Such improvements could

further solidify the advantage of using learned resizing in diverse and complex image analysis tasks.

## References

1. Ferrer, M.A., Das, A., Diaz, M., Morales, A., Carmona-Duarte, C., Pal, U.: MDIW-13: a new multi-lingual and multi-script database and benchmark for script identification. *Cogn. Comput.* **16**(1), 131–157 (2024)
2. Yuan, Z., Wang, H., Wang, L., Lu, T., Palaiiahnakote, S., Tan, C.L.: Modeling spatial layout for scene image understanding via a novel multiscale sum-product network. *Exp. Syst. Appl.* **63**, 231–240 (2016)
3. Toselli, A.H., Romero, V., Pastor, M., Vidal, E.: Multimodal interactive transcription of text images. *Pattern Recogn.* **43**(5), 1814–1825 (2010)
4. Phan, T.Q., Shivakumara, P., Ding, Z., Lu, S., Tan, C.L.: Video script identification based on text lines. In: 2011 International Conference on Document Analysis and Recognition, pp. 1240–1244. IEEE (2011)
5. Ubul, K., Tursun, G., Aysa, A., Impedovo, D., Pirlo, G., Yibulayin, T.: Script identification of multi-script documents: a survey. *IEEE Access* **5**, 6546–6559 (2017)
6. Zagoris, K., Pratikakis, I., Antonacopoulos, A., Gatos, B., Papamarkos, N.: Distinction between handwritten and machine-printed text based on the bag of visual words model. *Pattern Recogn.* **47**(3), 1051–1062 (2014)
7. Zheng, Y., Li, H., Doermann, D.: Machine printed text and handwriting identification in noisy document images. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(3), 337–353 (2004)
8. Sinwar, D., Dhaka, V.S., Pradhan, N., Pandey, S.: Offline script recognition from handwritten and printed multilingual documents: a survey. *Int. J. Doc. Anal. Recogn. (IJ DAR)* **24**(1), 97–121 (2021)
9. Djaffal, S., Djeddi, C., Diaz, M., Hannousse, A.: A robust analysis of local image descriptors using bag of visual words model for multi-level script identification in a multi-script environment. SSRN: <https://ssrn.com/abstract=4912105>
10. Talebi, H., Milanfar, P.: Learning to resize images for computer vision tasks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 497–506 (2021)
11. Patil, M.S.M.M.: Interpolation techniques in image resampling. *Int. J. Eng. Technol.* **7**, 567–570 (2018)
12. Tu, Z., Milanfar, P., Talebi, H.: MULLER: multilayer Laplacian resizer for vision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6877–6887 (2023)
13. Chen, L.-H., Bampis, C.G., Li, Z., Krasula, L., Bovik, A.C.: Estimating the resize parameter in end-to-end learned image compression. arXiv preprint [arXiv:2204.12022](https://arxiv.org/abs/2204.12022) (2022)
14. Riad, R., Teboul, O., Grangier, D., Zeghidour, N.: Learning strides in convolutional neural networks. arXiv preprint [arXiv:2202.01653](https://arxiv.org/abs/2202.01653) (2022)
15. Saidani, A., Echi, A.K., Belaid, A.: Arabic/Latin and machine-printed/handwritten word discrimination using HOG-based shape descriptor. *ELCVIA Electron. Lett. Comput. Vision Image Anal.*, 1–23 (2015)
16. Saïdani, A., Echi, A.K., Belaid, A.: Identification of machine-printed and handwritten words in Arabic and Latin scripts. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 798–802. IEEE (2013)

17. Benjelil, M., Mullet, R., Alimi, A.M.: Language and script identification based on steerable pyramid features. In: 2012 International Conference on Frontiers in Handwriting Recognition, pp. 716–721. IEEE (2012)
18. Feng, Z., Yang, Z., Jin, L., Huang, S., Sun, J.: Robust shared feature learning for script and handwritten/machine-printed identification. *Pattern Recogn. Lett.* **100**, 6–13 (2017)
19. Jindal, A.: Script identification in handwritten and printed documents using convolutional recurrent connection. *Multimedia Tools Appl.*, 1–15 (2024)
20. Han, X., Chen, Y.: COVID-19 classification using CT scan images with resize-MobileNet. In: 2021 International Conference on Intelligent Computing, Automation and Systems (ICICAS), pp. 286–289. IEEE (2021)
21. Duzyel, O., Catal, M.S., Kayan, C.E., Sevinc, A., Gumus, A.: Adaptive resizer-based transfer learning framework for the diagnosis of breast cancer using histopathology images. *Sig. Image Video Process.* **17**(8), 4561–4570 (2023)
22. Xu, Y., et al.: AnimateZoo: zero-shot video generation of cross-species animation via subject alignment. arXiv preprint [arXiv:2404.04946](https://arxiv.org/abs/2404.04946) (2024)
23. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint [arXiv:1704.04861](https://arxiv.org/abs/1704.04861) (2017)
24. Das, A., et al.: ICDAR 2021 competition on script identification in the wild. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12824, pp. 738–753. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-86337-1\\_49](https://doi.org/10.1007/978-3-030-86337-1_49)



# Arbitrary-Shaped Scene Text Recognition with Deformable Ensemble Attention

Shuo Xu<sup>1</sup>, Zeming Zhuang<sup>1</sup>, Mingjun Li<sup>1</sup>, and Feng Su<sup>2</sup>

State Key Laboratory for Novel Software Technology, Nanjing University,  
163 Xianlin Road, Nanjing, China  
{xushuo, zmzhuang, limingjun}@smail.nju.edu.cn, suf@nju.edu.cn

**Abstract.** Scene text recognition (STR) is a challenging task that aims to automatically localize and recognize text in varied natural scenes. Although the performance of STR methods has been significantly improved, the STR problem is far from being solved, especially when dealing with text with complex shapes and intricate backgrounds. To increase the accuracy of the STR model for arbitrary-shaped text and robustness to interferences such as noises and adjacent objects, we propose a novel deformable ensemble attention model and a scene text recognition network DEATR<sub>N</sub> based on it. The attention model combines the flexibility of an ensemble of deformable 2D local attentions for retrieving discriminative features of characters and the constraints on the regularity of the overall shape of a text depicted by its parametric centerline, which effectively enhances the text recognition performance of DEATR<sub>N</sub>. We also propose effective text geometry-based loss terms to improve the accuracy of attention. The experimental results show the superiority of DEATR<sub>N</sub> in recognizing arbitrary-shaped text in real scenarios.

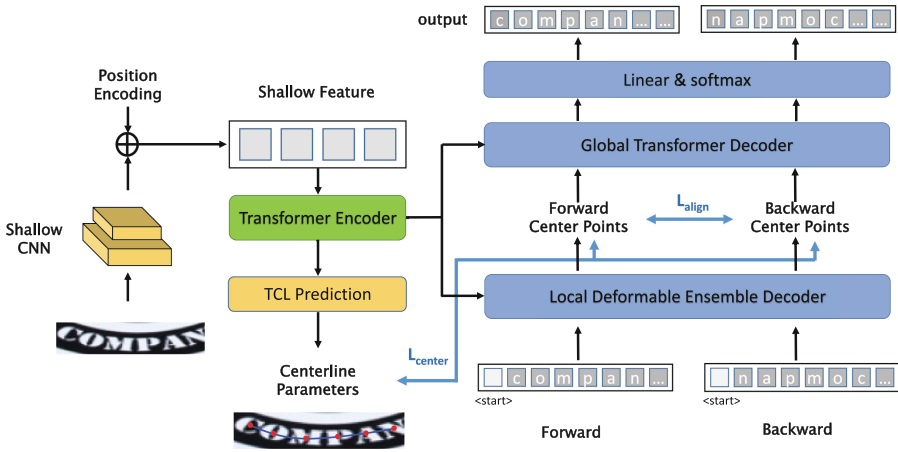
**Keywords:** Scene text recognition · Local attention · Deformable sampling · Text centerline

## 1 Introduction

Scene text is an important class of visual objects in natural images, which contains a wealth of valuable semantic information for various applications. As a critical step for retrieving textual information from the image, scene text recognition (STR) has attracted considerable research interests. However, recent studies [9] indicate that the problem of scene text recognition is far from being solved in complex real-world scenarios.

Most contemporary STR approaches employ an encoder-decoder architecture. The encoder extracts visual features from a text image, and the decoder transforms the features into the final character label sequence using some sequence model, in which attention mechanisms are often employed to adaptively align features with characters in decoding.





**Fig. 1.** The overall architecture of the proposed text recognition network DEATR.

To recognize text in arbitrary shapes, some methods [15, 22] introduce text shape rectification mechanisms that transform the input text into a more regular shape before recognition. Another line of research [13, 26] enhances the decoder with 2D attention mechanism, which adaptively selects features corresponding to each character in 2D feature maps, avoiding compressing the feature map in height dimension before decoding like some conventional recurrent neural network (RNN) based models, so that important spatial clues of irregular-shaped text can be preserved and exploited for recognition. As Transformer models [24] are widely used in STR methods, the two dimensional representations of the text used throughout the attention model significantly increase the capability and flexibility to learn spatial clues of the text.

Despite the greatly improved performance of STR methods on experimental benchmarks, few attention models used in previous STR work explicitly model and utilize the text shape clues and struggle with enforcing effective constraints on locations of characters, which reduces the robustness of the attention model to ambiguities or interferences caused by noises, adjacent objects, and intricate backgrounds in real-world scenarios.

To improve the accuracy and robustness of the attention model to arbitrary shapes of the text and various contextual interferences, we propose a scene text recognition network DEATR with a novel 2D attention mechanism, which combines the flexibility of an ensemble of deformable 2D local attentions to align characters with discriminative features and the constraints on the regularity of the overall layout of characters depicted by the parametric text centerline to enhance the text recognition accuracy. The overall architecture of DEATR is shown in Fig. 1.

The main contributions of our work are summarized as follows:

- We introduce the parametric description of the text shape in the attention model to help reliably locate character features in the face of contextual interferences.
- We propose a flexible feature selection and alignment mechanism for the decoder, which adaptively aggregates features at an ensemble of deformable local attention positions that effectively capture discriminative visual characteristics of the character in varied shapes and styles.
- We devise effective label generation schemes for the text geometry and character regions, requiring no character-level annotation information. We further propose effective loss terms to supervise the training of the model to obtain as accurate attentions as possible.
- Our method achieves leading performance on several STR benchmarks, demonstrating the effectiveness of the proposed attention and recognition model.

## 2 Related Work

Numerous methods for scene text recognition have emerged in the past few years. From the perspective of processing flow, most methods can be divided into two broad categories: character-oriented and word-oriented. The former typically follows a bottom-up pipeline [8, 25], starting by classifying individual characters and then grouping recognized characters into words. To overcome the problem of character segmentation errors encountered in the former approach, word-oriented methods recognize the character sequence of a word as a whole through certain sequence modeling mechanism, integrating character feature extraction and classification into the text recognition process.

Most of recent text recognition methods [3, 14, 19, 21, 32] leverage some sequence models like RNN to capture relations between character features and linguistic knowledge and recognize the text in a word-oriented manner. Inspired by speech recognition, these methods often employ an encoder-decoder framework, where the text image is first encoded into a feature sequence using usually some convolutional and recurrent networks and then decoded into character via sequence models, and variant attention mechanisms are exploited to adaptively align characters with related features in the sequence for classification [3, 26].

**Arbitrary-Shaped Text Recognition.** Early scene text recognition studies mostly focused on regular-shaped horizontal text, which is usually encoded into an one-dimensional feature sequence without loss of important information for character classification. However, as increasing research interests turn to arbitrary, especially irregularly shaped text which has a two-dimensional layout, representing the text image with traditional 1D sequence models becomes insufficient to keep key text clues for recognition and may introduce ambiguities and noises. Accordingly, some methods [15, 22, 32] transform the input irregular text

into a canonical (horizontal straight) shape using spatial transformation networks (STN) or specific layout rectification mechanisms, so that the text with regularized shape can be recognized using 1D sequence models.

Besides introducing text shape normalization as a pre-processing step for recognition, some methods [5, 12, 13, 23, 28] employ 2D attention to adaptively attend to certain regions in the two-dimensional feature maps during the decoding process, which better captures the spatial information of the text and enables more accurate extraction of character features. For example, some 2D attention mechanisms [13] take the hidden state of RNN as the query to sequentially attend to neighboring positions in the 2D feature map to select effective features for character recognition.

Benefiting from the context modeling capability of Transformer, some recent methods [12, 28] exploit Transformer to learn spatial dependencies of characters and sequential relations within 2D feature representation, which is then utilized to retrieve relevant features of the characters given previous decoding output.

Our work also adopts a Transformer-based 2D attention framework, but unlike most previous methods which capture global (word-level) dependencies only, we further introduce character-level spatial attention to adaptively retrieve local discriminative features of individual characters based on explicit modeling and exploitation of the geometric shape clues of the text, which effectively increases the accuracy of attention in the presence of various interferences such as noises and adjacent text instances.

**Performance Evaluation.** A common paradigm [1] in scene text recognition is to train the model on large synthetic datasets [7, 8] and then evaluate it on six standard real-world benchmarks, namely, IC13 [11], IIIT5k [16] and SVT [25] for regular text and IC15 [10], SVT-P [18], and CUTE [20] for irregular text. The performance gains that state-of-the-art methods achieve on six benchmarks are decreasing, which, however, does not mean that the challenges in STR have been largely addressed. To more comprehensively measure the capabilities of STR methods, some new datasets such as Union14M [9] and WordArt [28] have been proposed, which contain a large variety of challenging scene text samples in real scenarios with great diversity and complexity. The performance of existing STR methods on these new benchmarks is often poorer than that on the six conventional benchmarks, showing that the latter is not sufficient for fully exploring the challenges of the STR problem. In this work, we focus on the new and more comprehensive Union14M dataset and the WordArt dataset while also providing the results on the conventional benchmarks for comparison.

### 3 Methodology

As shown in Fig. 1, the proposed text recognition network DEATRNN consists of a Transformer encoder and a local-global hybrid decoder. Given a text image, the encoder extracts its feature representation, which is fed to the decoder and on the other hand is used to predict the geometric parameters of the centerline of

the text. The local decoder takes the text embeddings as the input and outputs features of the character obtained through the proposed deformable ensemble attention mechanism. The global decoder further refines the features by capturing relationships between features of different characters, and the resulting embeddings are then used for character classification. For higher accuracy, the decoding is performed in both forward and backward directions. We'll take a look at each component of the network in the following sections.

### 3.1 Feature Extraction

To obtain the feature representation of an input text image  $X \in \mathcal{R}^{H \times W \times 3}$  ( $W$  and  $H$  are image width and height), a shallow convolutional neural network (CNN) is first employed to extract the initial visual features  $F_i \in \mathcal{R}^{H/4 \times W/4 \times C}$  ( $C$  is the number of channels) of the image, which are further flattened into the size  $R^{HW/16 \times C}$ . The resulting features are then fed into the Transformer encoder proposed in SATRN [12] and get refined by self-attention. The output feature representation  $\mathbf{f}_e \in \mathcal{R}^{HW/16 \times C}$  is input into the decoder while also being used to predict the geometric parameters of the text centerline.

### 3.2 Text Centerline Regression

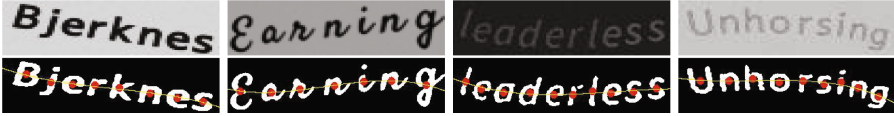
Different from previous attention mechanisms used in the decoder which do not explicitly exploit the shape characteristics of the text, in our work, the geometric description of the text's centerline, which depicts the overall spatial layout of the text as shown in the text example in Fig. 1, is utilized as a constraint on the attention positions of the characters to improve the attention accuracy and suppress attention drift caused by noise, background and adjacent text, which is especially useful for irregular-shaped text.

The centerline of a text is formulated as an  $n$ -th order polynomial:

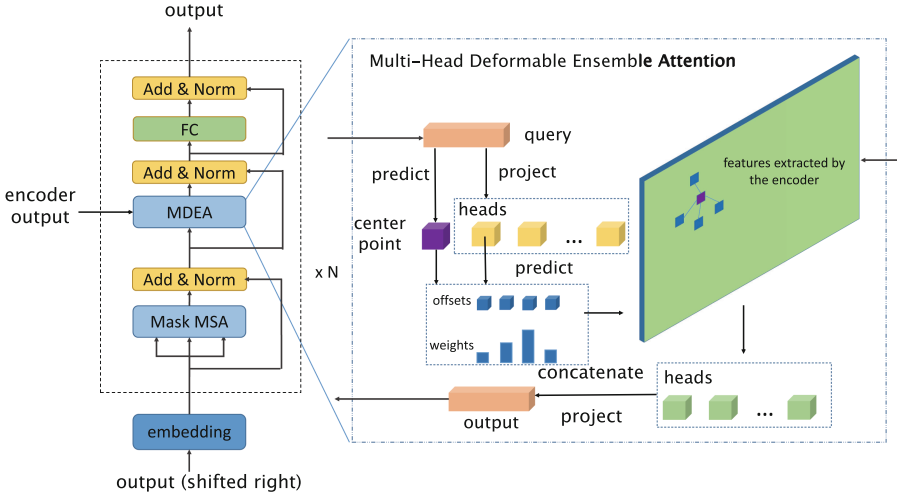
$$y = f_{\mathbf{a}}(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \quad (1)$$

where  $(x, y)$  are the coordinates of a point on the polynomial curve,  $\mathbf{a} = [a_n, a_{n-1}, \dots, a_0]$  are the coefficients of polynomial terms.

Given the features output by the encoder, a text centerline (TCL) prediction module is used to predict the parameters  $\mathbf{a}$  of the polynomial centerline of the text. To generate the ground-truth label  $\bar{\mathbf{a}}$  for a text image to train the parameter prediction network, we first apply the K-means algorithm on the image pixels with the number of clustering centers set to 2 to generate a binary label for each pixel which indicates it belonging to the text region or the background. This yields a segmentation map  $S_{gt}$  of the text image. Next, we apply the K-means clustering algorithm on the coordinates of the text pixels in  $S_{gt}$ , with the number of clusters set to  $L$  (the number of characters in the image). The resulting  $L$  cluster centers  $(x_{gt,i}, y_{gt,i})_{i \in [1, L]}$  are taken as a set of center points of the text. We then obtain the label  $\bar{\mathbf{a}}$  for the parameters of the polynomial centerline by fitting it to the center points using the least squares method. Figure 2 show some



**Fig. 2.** Examples of the labels generated for text centerline regression. The first row shows the training text image. The second row shows the generated labels of the text segmentation map, center points (denoted by red dots), and the approximated text centerline (denoted by the yellow line). (Color figure online)



**Fig. 3.** The local deformable ensemble decoder

examples of the generated labels of the text segmentation map, center points, and the polynomial centerline of the text.

As slight changes in the high order coefficients of the polynomial curve can lead to tremendous variations in the curve shape, we do not use  $\bar{\mathbf{a}}$  to supervise the prediction of  $\mathbf{a}$  directly. Instead, we uniformly sample  $T$  (10 in this work) values for the curve variable  $\{x_i\}_{i \in [1, T]}$  between the variable values of the ground-truth start point and end point (i.e.,  $x_{gt,1}$  and  $x_{gt,L}$ ) of the centerline, and compute the L2 centerline approximation loss  $L_{cline}$  between the predicted and ground-truth curve function values as follows:

$$L_{cline} = \sum_{i=1}^T L2(f_{\mathbf{a}}(x_i), f_{\bar{\mathbf{a}}}(x_i)) \tag{2}$$

### 3.3 Local Deformable Ensemble Decoder

Given the features output by the encoder, we propose a local deformable ensemble decoder (LDED) to accurately obtain aligned features of each character to

be recognized. To achieve this goal, the decoder integrates three effective mechanisms. 1) As there are usually multiple features at different locations that play an important role in the classification of a character, we propose to locate an ensemble of  $K$  sampling points ( $K = 16$  in this work) in the feature map where the features are selected for decoding the character. 2) Considering that the same character may exhibit varied appearances with different fonts and styles, instead of using a fixed, regular sampling grid, we allow the sampling points to have a deformable distribution over the feature map so as to adaptively search for effective features for the character. 3) Since the sampled features may capture different aspects of useful information for character classification, we employ a multi-head attention mechanism to allow the decoder to jointly attend to information from different representation subspaces of the sampled features.

Figure 3 shows the architecture of our proposed local deformable ensemble decoder. Similar to other auto-regressive decoder, LDED takes the embedding of the previous character as the initial query, which goes through a mask multi-head self-attention (MSA) layer, a multi-head deformable ensemble attention (MDEA) layer and a fully connected (FC) layer to obtain the feature of the current character. Specifically, given the initial query for the current character to be decoded, LDED employs the query mechanism similar to that used in D-DETR [33] to predict the center point of the current character (as shown in Fig. 4) based on the query and feed the projected query to a set of  $M$  attention heads. Each of the  $M$  heads of MDEA adaptively localizes  $K$  sampling points (in terms of  $K$  predicted offsets relative to the center point) in the output feature map  $\mathbf{f}_e$  of the encoder. At the same time, each head predicts a weight  $w_k$  ( $k \in [1, K]$ ) for each sample point, which is then used to aggregate the features at the sampling points to yield the aligned feature for the current character. The process can be formatted as follows:

$$\mathbf{s}_i = MSA(\mathbf{e}_{i-1} + \mathbf{p}_i) \quad (3)$$

$$\mathbf{h} = \mathbf{s}_i \mathbf{W}_P \quad (4)$$

$$\mathbf{pos}_{cen} = Linear_1(\mathbf{s}_i), \quad \mathbf{pos}_{off} = Linear_2(\mathbf{h}), \quad \mathbf{w} = Linear_3(\mathbf{h}) \quad (5)$$

$$\mathbf{pos}_{samp} = \mathbf{pos}_{cen} \oplus \mathbf{pos}_{off} \quad (6)$$

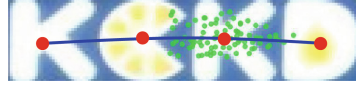
$$\mathbf{f}_i = \sum_{k=1}^K \mathbf{w}_k \cdot \mathbf{f}_e[\mathbf{pos}_{samp,k}] \quad (7)$$

where  $\mathbf{e}_{i-1} \in \mathcal{R}^C$  is the embedding of the previous character,  $\mathbf{p}_i$  is the position encoding,  $\mathbf{s}_i$  is the output of MSA,  $\mathbf{W}_P \in \mathcal{R}^{C \times C/M}$  is the projection matrix and  $\mathbf{h}$  is the projected feature.  $\mathbf{pos}_{cen} \in \mathcal{R}^{1 \times 2}$  denotes the position of the center point,  $\mathbf{pos}_{off} \in \mathcal{R}^{K \times 2}$  denotes the offsets of the sampling points relative to the center point,  $\oplus$  denotes matrix addition with broadcasting, and  $\mathbf{pos}_{samp} \in \mathcal{R}^{K \times 2}$  denotes the positions of the sampling points.  $\mathbf{w} = [w_1, \dots, w_K]$  are the predicted aggregation weights for sampling points,  $\mathbf{f}_e[\mathbf{pos}_{samp,k}]$  denotes the feature sampled from the encoder’s output feature map  $\mathbf{f}_e$  at the  $k$ th sampling point, and  $\mathbf{f}_i$  is the feature of the current character output by one head of MDEA.

Next, the features  $\{\mathbf{f}_i^m \in \mathcal{R}^{C/M}\}_{m \in [1, M]}$  extracted by all  $M$  attention heads are concatenated and projected into the feature  $\mathbf{g}_i$  of the current character:

$$\mathbf{g}_i = \text{Concat}(\mathbf{f}^1, \dots, \mathbf{f}^M) \mathbf{W}_H \quad (8)$$

where  $\mathbf{W}_H \in \mathcal{R}^{C \times C}$  is the projection matrix. The feature  $\mathbf{g}_i$  is finally fed to an FC layer to produce the output of the local deformable ensemble decoder. As shown in Fig. 3, the model stacks  $N$  LDEDs for improved accuracy.



**Fig. 4.** Illustration of the center point (red) and sampling points (green) of a character and the text centerline (blue). (Color figure online)

**Loss on Character Attention Position.** Accurate attention is crucial to the effectiveness and discriminability of character features extracted. We propose three losses, the center loss, the bidirectional alignment loss, and the dispersion loss, for training the model to accurately infer the attention positions for character features.

The center loss  $L_{center}$  brings the center points of characters closer to the polynomial centerline to reduce attention drift. Given the x-coordinates of the center points  $\mathbf{x}^{cen}$ , their y-coordinates  $\mathbf{y}^{cen}$  should be close to the values of the polynomial function Eq. 1 at  $\mathbf{x}^{cen}$ , which can be formulated as follows:

$$L_{center} = \text{MSE}(f_{\mathbf{a}}(x^{cen}), y^{cen}) \quad (9)$$

where  $\mathbf{a}$  is the predicted coefficients of the polynomial centerline, and MSE denotes the mean square error.

The bidirectional alignment loss  $L_{align}$  further fine-tunes the x-coordinates of the center points in a self-supervised way on the basis of bidirectional decoding. Specifically, given the two groups of center points obtained in the forward and backward decoding processes respectively, the bidirectional alignment loss is formulated as follows:

$$L_{align} = \max(Pos - Neg, 0) \quad (10)$$

$$Pos = \sum_{i=1}^L \text{Dist}(x_i^f, x_{L-i+1}^b), \quad Neg = \min_{i \neq j} (\text{Dist}(x_i^f, x_j^b)) \quad (11)$$

where the function  $\text{Dist}(\cdot, \cdot)$  calculates the distance between two inputs, which is the square distance in this paper.  $x^f$  and  $x^b$  are the x-coordinates of a center point obtained in the forward and backward decoding respectively.  $L$  is the number of characters in the image.  $Pos$  denotes the positive distance and  $Neg$  denotes the hardest negative distance.

The dispersion loss  $L_{disp}$  constrains the sampling points to distribute near the character's center point, which helps to suppress the influence of interfering objects such as noises and other characters located at a distance. We define the dispersion loss as follows:

$$L_{disp} = \sum_{i=1}^R \sum_{k=1}^K o_{i,k}, \quad o_{i,k} = \max\left(\left(\frac{O_x}{W/4}\right)^2 + \left(\frac{O_y}{H/4}\right)^2 - \tau, 0\right) \quad (12)$$

where  $R$  is the maximum number of characters that one text may contain and is set to 25 in this work.  $K$  is the number of sampling points associated with a center point.  $\tau$  is a hyperparameter that controls how the distance between a sampling point and the center point is counted in the loss.  $o_x$  and  $o_y$  are the offsets between the  $i$ th center point and its  $k$ th sampling point in x and y coordinates, respectively.

### 3.4 Global Decoder

Given the output features of the local decoder, which focuses on the adjacent features related to each character, we employ a global Transformer decoder shown in Fig. 5 to model the relationship between the features of all characters in the text. The features extracted by the local decoder are fed into a series of Transformer decoder blocks, each consisting of a self-attention layer, a cross-attention layer which takes the encoder output features as the key and value while taking the output features of the self-attention layer as the query, and a linear layer. The output of the final decoder block is fed to a fully connected layer and a softmax layer to predict the character label, and the labels with the highest word probability obtained in the forward and backward decoding processes are selected as the final output characters.

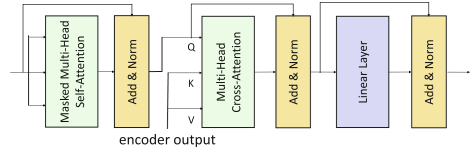


Fig. 5. The global decoder.

### 3.5 Loss Function

The whole loss function of the model is defined as follows:

$$\mathcal{L} = \lambda_1 L_{rec} + \lambda_2 L_{cline} + \lambda_3 L_{center} + \lambda_4 L_{align} + \lambda_5 L_{disp} \tag{13}$$

where  $L_{rec}$  is the cross-entropy loss for character classification, and  $\lambda_{1..5}$  are the balance weights which are set to 1.0, 10.0, 1.0, 1.0 and 1.0, respectively.

## 4 Experiments

### 4.1 Datasets

The proposed text recognition network DEATR<sub>N</sub> is trained on two synthetic text datasets, MJSynth (MJ) and SynthText (ST). MJSynth [8] contains nine million text images generated from a set of 90k common English words. SynthText [7] is created for the text detection task, whose samples are generated in a way similar to MJSynth. We crop the words in the image according to its ground-truth bounding box for the recognition task.

To evaluate the performance of DEATR<sub>N</sub>, the following eight scene text recognition benchmarks are employed in the experiments:



- **Union14M** [9] covers scene text in a broad range of real scenarios, including 900 artistic text (**ART**), 2426 curved text (**CUV**), 1495 contextless text (**CTX**), 1369 multi-oriented text (**MOR**), 892 multi-words text (**MWD**), 1585 salient text (**SAL**), and 400000 general text (**GEN**) samples.
- **WordArt** [28] contains 1511 artistic text images collected from posters, greeting cards, covers, billboards, handwriting, etc.
- **IIIT5k** [16] contains 3000 cropped word images collected from the web.
- **SVT** [25] comprises 647 testing word images cropped from Google Street View, many of which are severely corrupted by noise and blur and may have very low resolutions.
- **IC13** [11] provides 857 cropped testing word images after filtering out images containing non-alphanumeric characters or less than three characters.
- **SVT-P** [18] contains 639 cropped testing word images picked from side-view angle snapshots in Google Street View and many images contain severe perspective distortions.
- **IC15** [10] comprises 1811 cropped testing word images captured by Google Glass. A large number of text instances are irregularly shaped such as arbitrarily oriented, perspective distorted, or curved.
- **CUTE** [20] includes 288 cropped word images for testing, in which many of the text is curved.

We employ *word recognition accuracy* as the performance metric, which is defined as  $\frac{|O|}{|G|}$  where  $O$  and  $G$  are the set of correctly recognized words and the set of ground-truth words, respectively.

## 4.2 Implementation Details

We implement the proposed DEATRNN based on PyTorch. The encoder, local decoder and global decoder in the model have 12, 3 and 3 layers, respectively. The feature dimension  $C$  is 512, and the number of attention heads  $M$  is 8.

The model is trained on four NVIDIA Tesla V100 GPUs with the AdamW optimizer and a batch size of 384 for 6 epochs. The learning rate is warmed up from  $3e \times 10^{-7}$  to  $3e \times 10^{-4}$  in the first 3000 iterations of the first epoch, and then a cosine scheduler is applied in the subsequent iterations, gradually decreasing the learning rate until it reaches a final value of  $3e \times 10^{-6}$ . The bidirectional alignment loss is not used in the first 4 epochs for the stability of the training.

The character set contains 90 character classes including uppercase and lowercase English letters, numbers from 0 to 9 and 28 special symbols. All text images are resized to  $32 \times 128$ . We adopt the same data augmentation strategy as used in ABINet [5], including rotation, perspective deformation, blurring, color jittering, Gaussian noise, and so on.

## 4.3 Comparisons with State-of-the-Arts Methods

In Table 1, we compare our proposed DEATRNN with some state-of-the-art scene text recognition methods which are similarly trained on the ST and MJ datasets

**Table 1.** Recognition accuracy on Union14M and WordArt benchmarks. In each column, the best result is shown in bold.

Method	CUV	MOR	ART	CTX	SAL	MWD	GEN	WordArt
CRNN [21]	7.5	0.9	20.7	25.6	13.9	25.6	32.0	47.5
ASTER [22]	34.0	10.2	27.7	33.0	48.2	27.6	39.8	57.9
MORAN [15]	8.9	0.7	29.4	20.7	17.9	23.8	35.2	-
SAR [13]	44.3	7.7	42.6	44.2	44.0	51.2	50.5	63.8
DAN [26]	26.7	1.5	35.0	40.3	36.5	42.2	42.1	52.4
RobustScanner [31]	43.6	7.9	41.2	42.6	44.9	46.9	39.5	61.3
SEED [19]	-	-	-	-	-	-	-	60.1
SCATTER [14]	-	-	-	-	-	-	-	64.0
SATRn [12]	51.1	15.8	48.0	45.3	62.7	52.5	58.5	65.7
ABINet [5]	59.5	12.7	43.3	38.3	62.0	50.8	55.6	67.4
VisionLAN [27]	57.7	14.2	47.8	48.0	64.0	47.9	52.1	-
SRN [30]	63.4	25.3	34.1	28.7	56.5	26.7	46.3	-
SVTR [4]	63.0	<b>32.1</b>	37.9	44.2	67.5	49.1	52.8	-
CornerTransformer [28]	-	-	-	-	-	-	-	70.8
MATRn [17]	63.1	13.4	43.8	41.9	66.4	53.2	57.0	-
<b>DEATRn</b>	<b>69.2</b>	22.8	<b>59.1</b>	<b>59.3</b>	<b>71.7</b>	<b>65.7</b>	<b>61.9</b>	<b>72.1</b>

only. For the fairness of comparison, all methods are not fine-tuned on the target scene text datasets. We cite the results on the Union14M and WordArt datasets reported in [9, 28] respectively for corresponding methods in Table 1.

DEATRn achieved the best performance on seven out of eight benchmarks – Curve, Artistic, Contextless, Salient, Multi-Word, General, and WordArt, surpassing the second best by 5.8%, 11.1%, 11.3%, 4.2%, 12.5%, 4.9%, and 1.3%, respectively. The results show that the proposed geometry-constrained deformable attention mechanism can effectively recognize text of various appearances in real scenarios. Some examples of scene text recognition results are shown in Fig. 6.

We also evaluate DEATRn on the six conventional STR benchmarks IIIT5k, IC13, SVT, IC15, SVT-P, and CUTE in Table 2. DEATRn achieves competitive performance compared to other state-of-the-art methods. It is worth noting that as we take the Union14M and WordArt benchmarks as the primary optimization target of DEATRn and the same trained model is used for evaluation on the six conventional benchmarks, which have significantly fewer complex and irregularly shaped text samples than Union14M/WordArt, DEATRn’s strengths in recognizing complex scene text are more pronounced on the new Union14M and WordArt benchmarks compared to the conventional ones.



**Fig. 6.** Examples of text recognition results obtained by DEATR.

**Table 2.** Recognition accuracy on six scene text benchmarks. In each column, the best result is shown in bold.

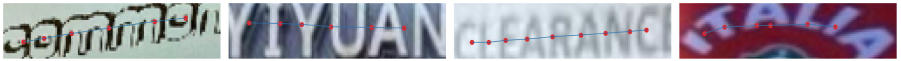
Method	Regular Text			Irregular Text		
	IIIT5k	SVT	IC13	SVT-P	IC15	CUTE
ASTER [22]	93.4	89.5	91.8	78.5	76.1	79.5
TRBA [1]	87.9	87.5	92.3	79.2	77.6	74.0
MORAN [15]	91.2	88.3	92.4	76.1	68.8	77.4
SAR [13]	91.5	84.5	91.0	76.4	69.2	83.3
ESIR [32]	93.3	90.2	91.3	79.6	76.9	83.3
DAN [26]	94.3	89.2	93.9	80.0	74.5	84.4
RobustScanner [31]	95.3	88.1	94.8	79.5	77.1	90.3
SEED [19]	89.6	92.8	93.0	81.4	80.0	83.6
SCATTER [14]	93.2	90.9	94.1	86.2	82.0	84.8
SATRN [12]	92.8	91.3	94.1	86.5	79.0	87.8
ABINet [5]	96.2	93.5	97.4	89.3	86.0	89.2
SVTR [4]	96.3	91.7	97.2	88.4	86.6	<b>95.1</b>
CornerTransformer [28]	95.9	94.6	96.4	<b>91.5</b>	86.3	92.0
MATRn [17]	96.6	95.0	<b>97.9</b>	90.6	86.6	93.5
SIGA [6]	96.6	95.1	97.8	90.5	86.6	93.1
LISTER [2]	<b>96.8</b>	93.5	97.7	89.5	<b>87.2</b>	89.6
OTE/SVTR [29]	96.4	<b>95.5</b>	97.4	89.6	<b>87.2</b>	92.4
<b>DEATRn</b>	95.6	93.2	<b>97.9</b>	90.2	85.9	94.8

#### 4.4 Ablation Study

We evaluate the effectiveness of the main components of the proposed text recognition model by ablation experiments. For simplicity, all models in the ablation experiments employ a single-directional decoder, unless specifically stated.

**Table 3.** Effectiveness of text centerline constraint on character attention.

Model	CUV	MOR	ART	CTX	SAL	MWD	GEN	WordArt
w/o centerline	<b>56.0</b>	<b>15.3</b>	48.8	52.9	62.1	54.2	57.4	65.1
w. centerline	54.1	15.0	<b>51.3</b>	<b>55.2</b>	<b>63.0</b>	<b>57.3</b>	<b>58.2</b>	<b>67.5</b>



**Fig. 7.** Illustrations of the predicted text centerline (blue) and character centers (red). (Color figure online)

**Text Centerline Constraint on Character Attention.** We compare the text recognition accuracy with and without the constraint of the geometry of the text centerline on character attention in Table 3. Specifically, in the contrasting model, the distribution of the center points of the characters is no longer constrained by the geometry of the text centerline, i.e., the center loss is omitted in this model. It can be seen that the centerline constraint helps to improve the recognition accuracy on six of the eight datasets, showing the effectiveness of the centerline constraint in improve the accuracy of attention. Figure 7 shows some examples of the predicted text centerline and character centers. Note that some degree of inaccuracy in the centerline and center point prediction can be effectively compensated for by the proposed deformable ensemble attention mechanism with dynamically predicted feature sampling positions.

**Table 4.** Effectiveness of the dispersion loss.

$\tau$	CUV	MOR	ART	CTX	SAL	MWD	GEN	WordArt
0.1	<b>59.4</b>	<b>16.2</b>	50.3	57.6	64.9	61.3	58.6	67.4
0.3	58.6	15.9	<b>51.6</b>	<b>58.4</b>	<b>65.9</b>	<b>62.9</b>	58.5	<b>68.7</b>
0.5	57.7	16.1	50.1	56.0	65.1	58.9	<b>58.7</b>	66.7
$\infty$	54.1	15.0	51.3	55.2	63.0	57.3	58.2	67.5

**Dispersion Loss.** We verify the effectiveness of the proposed dispersion loss in Table 4, which compares the text recognition accuracy using different values for  $\tau$  in Eq. 12. According to the definition of the dispersion loss, the larger the value of  $\tau$ , the more freely the sampling points are distributed in the feature map. Particularly,  $\tau = \infty$  means that the center point has no constraint on the spatial distribution of the sampling points, i.e., the dispersion loss is omitted.

As shown in the data in Table 4, too large or too small  $\tau$  are both not conducive to text recognition. We guess too much constraint on the sampling point position resulting from small  $\tau$  values limits the flexibility of the attention model. In contrast, excessive freedom in the attention position may make the model more vulnerable to interferences like noises, resulting in degraded performance. Accordingly, we adopt  $\tau = 0.3$  in our model. Compared to  $\tau = \infty$  (i.e. without the dispersion loss), the results also show the effect of the proposed dispersion loss in improving the recognition performance.

**Local Deformable Ensemble Decoder.** We verify the effectiveness of the proposed local deformable ensemble decoder in Table 5. Models ‘Global’ and ‘Local’ employ only a 6-layer global and local decoder for character prediction, respectively. Model ‘Hybrid’ combines the local and global decoders as proposed, which consists of 3 local decoder layers and 3 global decoder layers.

As shown in Table 5, compared to the standard Transformer-based global decoder, the proposed local deformable ensemble decoder achieves higher or equal recognition accuracies on all benchmarks. By combining the local and global decoders, the hybrid decoder achieves the best results on four benchmarks and a higher average performance.

**Table 5.** Effectiveness of the local deformable ensemble decoder.

Model	CUV	MOR	ART	CTX	SAL	MWD	GEN	WordArt
Global	51.1	15.8	48.0	45.3	62.7	52.5	58.5	65.7
Local	58.6	15.9	<b>51.6</b>	<b>58.4</b>	65.9	<b>62.9</b>	58.5	<b>68.7</b>
Hybrid	<b>63.1</b>	<b>19.4</b>	51.0	55.3	<b>66.9</b>	60.0	<b>59.6</b>	68.1

**Table 6.** Effectiveness of bidirectional decoding with the alignment loss.

Model	CUV	MOR	ART	CTX	SAL	MWD	GEN	WordArt
Single-Dir	63.1	19.4	51.0	55.3	66.9	60.0	59.6	68.1
Bi-Dir	67.7	20.5	58.3	<b>59.3</b>	70.2	65.3	61.4	71.8
Bi-Dir + Align. Loss	<b>69.2</b>	<b>22.8</b>	<b>59.1</b>	<b>59.3</b>	<b>71.7</b>	<b>65.7</b>	<b>61.9</b>	<b>72.1</b>

**Bidirectional Decoding and Bidirectional Alignment Loss.** To verify the effectiveness of the bidirectional decoding mechanism and the proposed bidirectional alignment loss, we compare the performance of three variants of the recognition model, single-direction decoding and bidirectional decoding without and with the alignment loss. As shown in Table 6, the performance of bidirectional decoding is much better than that of single-direction decoding, and the introduction of the alignment loss further improves the text recognition accuracy.

## 4.5 Limitations

Figure 8 shows some examples of the failure cases of DEATR. Most incorrect recognition results are caused by character-like object, heavy distortion of character, complex image background, and ambiguous text orientation.



**Fig. 8.** Examples of the failure cases of DEATR. Incorrect recognition results are displayed in red text, and the ground truth is shown in blue text in parentheses. (Color figure online)

## 5 Conclusion

We propose a novel 2D attention model for arbitrary-shaped scene text recognition. The model adaptively aggregates discriminative features selected by an ensemble of deformable local attentions to generate character feature for classification. We further introduce a parametric modeling of the text centerline and associated loss terms as spatial constraints to improve the accuracy and robustness of attention in the face of various interferences. The proposed attention model effectively enhances the performance of the recognition network for scene text with varied shapes and appearances.

## References

1. Baek, J., et al.: What is wrong with scene text recognition model comparisons? Dataset and model analysis. In: ICCV, pp. 4714–4722 (2019)
2. Cheng, C., Wang, P., Da, C., Zheng, Q., Yao, C.: LISTER: neighbor decoding for length-insensitive scene text recognition. In: ICCV, pp. 19541–19551, October 2023
3. Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., Zhou, S.: Focusing attention: towards accurate text recognition in natural images. In: ICCV, pp. 5086–5094, October 2017
4. Du, Y., et al.: SVTR: scene text recognition with a single visual model. In: IJCAI, pp. 884–890 (2022)

5. Fang, S., Xie, H., Wang, Y., Mao, Z., Zhang, Y.: Read like humans: autonomous, bidirectional and iterative language modeling for scene text recognition. In: CVPR, pp. 7094–7103 (2021)
6. Guan, T., et al.: Self-supervised implicit glyph attention for text recognition. In: CVPR, pp. 15285–15294 (2023)
7. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. CoRR abs/1406.2227 (2014)
8. Jaderberg, M., Vedaldi, A., Zisserman, A.: Deep features for text spotting. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 512–528. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10593-2\\_34](https://doi.org/10.1007/978-3-319-10593-2_34)
9. Jiang, Q., Wang, J., Peng, D., Liu, C., Jin, L.: Revisiting scene text recognition: a data perspective. In: ICCV (2023)
10. Karatzas, D., et al.: ICDAR 2015 competition on robust reading. In: ICDAR, pp. 1156–1160 (2015)
11. Karatzas, D., et al.: ICDAR 2013 robust reading competition. In: ICDAR, pp. 1484–1493 (2013)
12. Lee, J., Park, S., Baek, J., Oh, S.J., Kim, S., Lee, H.: On recognizing texts of arbitrary shapes with 2D self-attention. In: CVPRW, pp. 2326–2335 (2020)
13. Li, H., Wang, P., Shen, C., Zhang, G.: Show, attend and read: a simple and strong baseline for irregular text recognition. In: AACL, vol. 33, pp. 8610–8617, July 2019
14. Litman, R., Anshel, O., Tsiper, S., Litman, R., Mazor, S., Manmatha, R.: SCATTER: selective context attentional scene text recognizer. In: CVPR, pp. 11959–11969 (2020)
15. Luo, C., Jin, L., Sun, Z.: MORAN: a multi-object rectified attention network for scene text recognition. PR **90**, 109–118 (2019)
16. Mishra, A., Alahari, K., Jawahar, C.V.: Scene text recognition using higher order language priors. In: BMVC, pp. 1–11 (2012)
17. Na, B., Kim, Y., Park, S.: Multi-modal text recognition networks: interactive enhancements between visual and semantic features. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) CCV 2022. LNCS, vol. 13688, pp. 446–463. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-19815-1\\_26](https://doi.org/10.1007/978-3-031-19815-1_26)
18. Phan, T.Q., Shivakumara, P., Tian, S., Tan, C.L.: Recognizing text with perspective distortion in natural scenes. In: ICCV, pp. 569–576 (2013)
19. Qiao, Z., Zhou, Y., Yang, D., Zhou, Y., Wang, W.: SEED: semantics enhanced encoder-decoder framework for scene text recognition. In: CVPR, pp. 13525–13534 (2020)
20. Risnumawan, A., Shivakumara, P., Chan, C.S., Tan, C.L.: A robust arbitrary text detection system for natural scene images. ESA **41**(18), 8027–8048 (2014)
21. 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 TPAMI **39**(11), 2298–2304 (2017)
22. Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: ASTER: an attentional scene text recognizer with flexible rectification. IEEE TPAMI **41**(9), 2035–2048 (2019)
23. Tan, Y.L., Kong, A.W.K., Kim, J.J.: Pure transformer with integrated experts for scene text recognition. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13688, pp. 481–497. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-19815-1\\_28](https://doi.org/10.1007/978-3-031-19815-1_28)
24. Vaswani, A., et al.: Attention is all You need. In: NeurIPS, pp. 5998–6008 (2017)

25. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: ICCV, pp. 1457–1464 (2011)
26. Wang, T., et al.: Decoupled attention network for text recognition. In: AAAI, vol. 34, pp. 12216–12224, April 2020
27. Wang, Y., Xie, H., Fang, S., Wang, J., Zhu, S., Zhang, Y.: From two to one: a new scene text recognizer with visual language modeling network. In: ICCV, pp. 14174–14183 (2021)
28. Xie, X., Fu, L., Zhang, Z., Wang, Z., Bai, X.: Toward understanding WordArt: corner-guided transformer for scene text recognition. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13688, pp. 303–321. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-19815-1\\_18](https://doi.org/10.1007/978-3-031-19815-1_18)
29. Xu, J., Wang, Y., Xie, H., Zhang, Y.: OTE: exploring accurate scene text recognition using one token. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 28327–28336, June 2024
30. Yu, D., et al.: Towards accurate scene text recognition with semantic reasoning networks. In: CVPR, pp. 12110–12119 (2020)
31. Yue, X., Kuang, Z., Lin, C., Sun, H., Zhang, W.: RobustScanner: dynamically enhancing positional clues for robust text recognition. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12364, pp. 135–151. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58529-7\\_9](https://doi.org/10.1007/978-3-030-58529-7_9)
32. Zhan, F., Lu, S.: ESIR: end-to-end scene text recognition via iterative image rectification. In: CVPR, pp. 2054–2063, June 2019
33. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: deformable transformers for end-to-end object detection. In: ICLR (2021)





# Primary Key Free Watermarking for Numerical Tabular Datasets in Machine Learning

Xin Che<sup>1</sup>, Mohammad Akbari<sup>2</sup>, Shaoxin Li<sup>1</sup>, David Yue<sup>3</sup>,  
Yong Zhang<sup>2</sup>, and Lingyang Chu<sup>1</sup> 

<sup>1</sup> McMaster University, Hamilton, ON, Canada  
{chex5, li2018, chu19}@mcmaster.ca

<sup>2</sup> Huawei Technologies Canada Co., Ltd., Burnaby, BC, Canada  
{mohammad.akbari, yong.zhang3}@huawei.com

<sup>3</sup> University of Toronto, Toronto, ON, Canada  
david.yue@mail.utoronto.ca

**Abstract.** High-quality tabular datasets are often traded by their owners as valuable digital assets due to their scarcity and usefulness in training machine learning models. A pivotal concern when trading the datasets is their ownership, which is seriously threatened by piracy due to the simplicity of reselling illegal copies. This produces an urgent demand for an effective watermarking method to demonstrate the ownership of the dataset. Existing database watermarking methods rely on either a primary key or a virtual primary key to watermark a tabular dataset. These methods cannot work well in the context of machine learning, because a primary key can be easily modified without affecting the machine learning utility of a tabular dataset, and a virtual primary key is often not robust against watermark-removing attacks. How to watermark a tabular dataset without using a primary key or virtual primary key is a challenging task that has not been systematically studied before. In this paper, we tackle this task by a novel primary key free method that embeds a sinusoidal signal as the watermark into a discrete-time signal constructed from the tabular dataset. We conduct an in-depth theoretical analysis on the exceptional robustness of our watermark against five challenging attacks, and further validate the robustness through comprehensive experiments on two real-world datasets.

**Keywords:** Primary key free · Tabular dataset watermarking · Robust

## 1 Introduction

Artificial intelligence powered by machine learning has brought significant benefits to the modern society. In many successful applications, large machine learning models are fueled by huge amount of tabular datasets, such as marketing data [8], healthcare data [4], environment & climate data [13], and sensor

data [16]. Due to the enormous efforts and cash invested in data collection and management, these datasets are often regarded as high value digital assets of their owners. However, when these datasets are traded in the market, their safety is usually threatened by piracy due to the simplicity of creating and reselling illegal copies. This produces an urgent demand for an effective watermarking method that embeds a detectable watermark into a dataset to demonstrate the ownership of the dataset and its copies.

As discussed later in Sect. 2, existing methods are mostly not robust to watermark removing attacks because they embed their watermarks based on the primary key (PK) [2, 3, 15, 37] or virtual primary key (VPK) [3, 9, 10, 25]. The use of PK and VPK significantly weaken the robustness of existing methods against attacks, because both PK and VPK can be easily modified by an attacker to remove the watermark without significantly decreasing the machine learning utility of the dataset. As far as we know, how to watermark a tabular dataset without using a primary key (or a virtual primary key) is a novel and challenging task that has not been well studied in the literature.

In this paper, we systematically tackle this task by formulating and solving a novel problem named primary key free watermarking for numerical tabular datasets. We make the following contributions. First, we propose the novel task of primary key free watermarking for tabular datasets. The goal is to embed and detect watermarks on tabular datasets without using primary key (or a virtual primary key) while achieving good robustness against watermark-removing attacks. Second, we successfully tackle the problem with a carefully designed watermarking method. The key idea is to first map the data instances in a tabular dataset to a discrete-time signal, and then embed a watermark by adding a sinusoidal signal to the discrete-time signal. The watermark can be accurately detected by checking the existence of the sinusoidal signal. Last, we conducted extensive experiments on two real-world datasets to compare the performance of our method with five state-of-the-art baseline methods. The experimental results demonstrate the superior robustness of our watermark against six challenging watermark-removing attacks.

## 2 Related Work

Many existing works [3, 10, 12, 15, 19, 25, 33–35] have been proposed to embed and detect watermarks in tabular datasets. Our work is related to the following methods.

The **primary key methods** [2, 3, 17, 35] rely on the primary key of a tabular dataset to embed and detect watermarks. Most primary key methods [2, 3, 12, 35, 37] use a primary key to uniquely identify watermarked data instances in order to accurately detect watermark. Some other works [5, 15, 20, 26, 32–35] use primary key to organize data instances in groups to embed and detect watermarks. These methods work well when the primary key of a watermarked tabular dataset stays unchanged. However, they cannot accurately detect the watermark if an attacker modifies the primary key. Since modifying the primary key of a

tabular dataset often does not reduce its machine learning utility in training good machine learning models, the resale value of the tabular dataset in the application areas of machine learning is not affected. Therefore, the primary key methods cannot effectively protect tabular datasets against piracy, because an attacker can easily modify the primary key of a watermarked tabular dataset to create an illegal copy, which successfully escapes watermark detection and also preserves the resale value.

The **virtual primary key methods** [3,9,10,25] compute a virtual primary key (VPK) from data instances and use it as a substitution of primary key to embed and detect watermarks. For example, Agrawal et al. [3] use the most significant bits of an attribute to compute VPK. Li et al. [25] select multiple attributes to compute VPK. By using a VPK, existing primary key methods can be extended to watermark a tabular dataset. These methods are robust to primary key modification, because they do not use primary key and the VPK is secretly computed from data instances. However, a watermark embedded by using VPK is often not robust against watermark-removing attacks [3,22,34,35], because modifying the data instances changes the values of VPK [10,11,34]. As a result, by slightly modifying the tabular dataset, an attacker can remove a VPK-based watermark without causing much damage to the machine learning utility of the dataset.

To the best of our knowledge, our work is the first in the literature to watermark a numerical tabular dataset without using a primary key (or a virtual primary key) while achieving outstanding robustness against watermark-removing attacks. This makes our work particularly effective in protecting numerical tabular datasets against piracy in the application areas of machine learning.

### 3 Task Definition

In this section, we first introduce a typical application example of the proposed numerical tabular dataset watermarking task in Fig. 1. Then, we give the formal definition of our task.

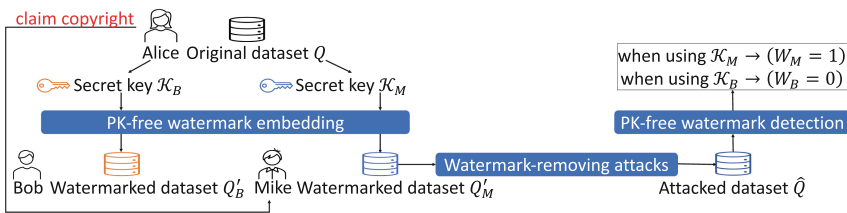


Fig. 1. A typical application scenario.

*Example 1 (A typical application).* As shown in Fig. 1, Alice owns a valuable numerical tabular dataset  $Q$  and she wants to sell it to Bob and Mike. Before

selling  $Q$ , Alice uses two different secret keys  $\mathcal{K}_B$  and  $\mathcal{K}_M$  to embed two different watermarks in  $Q$ . The watermarked dataset  $Q'_B$  produced by  $\mathcal{K}_B$  is sold to Bob. The watermarked dataset  $Q'_M$  produced by  $\mathcal{K}_M$  is sold to Mike. Mike uses watermark-removing attacks to modify  $Q'_M$  into  $\hat{Q}$  and put  $\hat{Q}$  on the market for sale. Alice uses each of  $\mathcal{K}_B$  and  $\mathcal{K}_M$  to detect watermark from  $\hat{Q}$ , which produces  $W_B = 0$  and  $W_M = 1$ , respectively. Since  $W_M = 1$  means  $\hat{Q}$  is watermarked by  $\mathcal{K}_M$ , Alice demonstrates her ownership on  $\hat{Q}$  and she also knows  $\hat{Q}$  comes from the copy she sold to Mike.

Next, we first define some related concepts and then introduce the formal definition of our task, which includes two parts such as primary key (PK)-free watermark embedding and PK-free watermark detection.

**Definition 1 (Numerical tabular dataset).** *A numerical tabular dataset is represented by a matrix, denoted by  $Q \in \mathbb{R}^{n \times d}$ , where each row stores one data instance and each column corresponds to one attribute of the data instances. A tabular dataset is also called a “dataset” in short.*

**Definition 2 (Machine learning utility).** *Given a numerical tabular dataset  $Q$ , the machine learning utility (MLU) of  $Q$ , denoted by  $\text{MLU}(Q)$ , indicates the effectiveness of  $Q$  in training good machine learning models. It is measured by the performance of a machine learning model trained on  $Q$  [19, 24].*

**Definition 3 (Secret key).** *Denote by  $Q$  an original dataset that is not embedded with a watermark and by  $Q'$  the watermarked dataset that is embedded with a watermark. A secret key, denoted by  $\mathcal{K}$ , is a cryptographic key that is used to embed and detect watermark from  $Q'$ .*

The secret key  $\mathcal{K}$  is completely different from a primary key (PK) or a virtual primary key (VPK). In typical watermarking systems [3],  $\mathcal{K}$  often consists of a set of variables storing the information to generate and identify a watermark. While PK and VPK are unique identifiers of data instances. Different watermarking systems use different types of secret keys. Leaking  $\mathcal{K}$  exposes the watermark embedded in  $Q'$ , which makes it vulnerable to watermark-removing attacks. Thus,  $\mathcal{K}$  is often kept secret by the owner of the dataset  $Q$ .

**Definition 4 (PK-free watermark embedding).** *Given  $Q$ ,  $\mathcal{K}$  and a positive threshold  $\gamma$ , the process of watermark embedding produces  $Q'$  by modifying the data instances in  $Q$ . This process should satisfy: (1) no primary key is used; (2)  $Q'$  carries a watermark that can be verified by  $\mathcal{K}$ ; and (3)  $|\text{MLU}(Q) - \text{MLU}(Q')| \leq \gamma$ .*

In the above conditions, (1) requires the embedded watermark to be independent from primary key, which improves the robustness of watermark against primary key modification. (2) means  $Q'$  is watermarked by  $\mathcal{K}$ . (3) establishes an MLU constraint, which limits the damage on  $\text{MLU}(Q)$  caused by the modification on  $Q$  when embedding the watermark. This preserves the resale value of  $Q'$  because  $\text{MLU}(Q')$  is close to  $\text{MLU}(Q)$ .

**Definition 5 (PK-free watermark detection).** Given  $\mathcal{K}$  and a suspicious dataset  $\hat{Q}$  that may or may not be watermarked by  $\mathcal{K}$ , the process of watermark detection verifies whether  $\hat{Q}$  is watermarked by  $\mathcal{K}$ . This process returns a binary variable  $W \in \{0, 1\}$ , where  $W = 1$  means  $\hat{Q}$  is watermarked by  $\mathcal{K}$  and  $W = 0$  means  $\hat{Q}$  is not watermarked by  $\mathcal{K}$ . This process should satisfy: (1) no primary key is used; (2) the original dataset  $Q$  is not used; and (3) the value of  $W$  cannot be flipped without significantly modifying  $\hat{Q}$ .

In the above conditions, (1) requires the watermark detection to be primary key free, which mitigates the influence of primary key modification on watermark detection. (2) is a classic requirement of blind watermark detection [15]; it reduces the risk of unauthorized access to the original dataset  $Q$ , because the watermark can be detected without revealing  $Q$  [3, 15, 34]. (3) means an attacker has to significantly modify  $\hat{Q}$  in order to escape watermark detection. Since a larger modification on  $\hat{Q}$  causes more damage to  $\text{MLU}(\hat{Q})$ , it reduces more resale value of  $\hat{Q}$ , which lowers the interest of the attacker in attacking  $\hat{Q}$ .

## 4 PK-Free Watermark Embedding

The **key idea** of PK-free watermark embedding is to first map the data instances to a discrete-time signal in a two-dimensional space, and then add a sinusoidal signal with a specific frequency to the discrete-time signal by modifying the data instances. This embeds the sinusoidal signal as a watermark into the dataset.

### 4.1 Mapping Dataset to Discrete-Time Signal

To map the data instances in a tabular dataset to a discrete-time signal in a two-dimensional space, we design a pair of mapping functions, denoted by  $\phi_x(\cdot)$  and  $\phi_y(\cdot)$ , where  $\phi_x(\cdot)$  maps a data instance to a real-valued  $x$ -coordinate and  $\phi_y(\cdot)$  maps the same data instance to a real-valued  $y$ -coordinate. This maps each data instance to a pair of  $x$  and  $y$  coordinates, which represents a point in a two-dimensional space. Then, the points of all the data instances are summarized in groups to form the discrete-time signal.

Denote by  $Q_{i,:}$  the  $i$ -th data instance in a tabular dataset  $Q \in \mathcal{R}^{n \times d}$ , and by  $\mathbf{e}_x, \mathbf{e}_y \in \mathbb{R}^d$  a random pair of *orthogonal vectors* with L2-norm equal to one. The mapping function  $\phi_y(\cdot)$  that maps  $Q_{i,:}$  to a  $y$ -coordinate  $y_i$  is defined as

$$y_i = \phi_y(Q_{i,:}) = Q_{i,:} \mathbf{e}_y^\top, \quad (1)$$

which is simply the projection of  $Q_{i,:}$  on  $\mathbf{e}_y$ . The mapping function  $\phi_x(\cdot)$  that maps  $Q_{i,:}$  to an  $x$ -coordinate  $x_i$  is defined as

$$x_i = \phi_x(Q_{i,:}) = \left\lfloor \frac{Q_{i,:} \mathbf{e}_x^\top}{b} \right\rfloor \tau, \quad (2)$$

where  $b \in \mathbb{R}^+$  and  $\tau \in \mathbb{R}^+$  are positive real-valued hyperparameters, and  $\lfloor \cdot \rfloor$  is the flooring operator that rounds a real number down to the closest integer.

**Algorithm 1.** Watermark embedding**Input:** An original dataset  $Q$  and a secret key  $\mathcal{K}$ .**Output:** A watermarked dataset  $Q'$ .

- 1: Initialize  $Q'$  as a zero matrix in the same size as  $Q$ .
- 2: **for** each data instance  $Q_{i,:}$  in  $Q$  **do**
- 3:     Compute:  $x_i = \phi_x(Q_{i,:})$ .
- 4:     Update:  $Q'_{i,:} = Q_{i,:} + \lambda \sin(2\pi\theta x_i) * \mathbf{e}_y$ . (See Equation [4])
- 5: **end for**
- 6: Return  $Q'$ .

The numerator  $\lfloor \frac{Q_{i,:} \mathbf{e}_x^\top}{b} \rfloor$  in Eq. (2) conducts a binning operation that projects  $Q_{i,:}$  into a bin and returns the index of the bin. The hyperparameter  $b$  is the *bin width* of each bin. By dividing the index of bin over  $\tau$ , Eq. (2) maps the index of bin into an  $x$ -coordinate, where  $\frac{1}{\tau}$  is the *step size* between the  $x$ -coordinates of neighbouring bins.

The tuple  $(x_i, y_i)$  mapped from  $Q_{i,:}$  represents a *point* in a two-dimensional space. By mapping each  $Q_{i,:}$  in  $Q$  to a point, we obtain a set of points, denoted by  $Z = \{(x_i, y_i) \mid x_i = \phi_x(Q_{i,:}), y_i = \phi_y(Q_{i,:}), i \in \{1, \dots, n\}\}$ . The points in  $Z$  cannot form a discrete-time signal because some points may have the same  $x$ -coordinates and different  $y$ -coordinates due to the binning operation in Eq. (2). To convert the points in  $Z$  into a discrete-time signal, we first group each subset of points with the same  $x$ -coordinates into a *bin*, denoted by

$$B_h = \{(x_i, y_i) \mid x_i = h, (x_i, y_i) \in Z\}, \quad (3)$$

where  $h$  is the value of the  $x$ -coordinates of the points in  $B_h$ . Then, we summarize the points in  $B_h$  into a *mean point*, denoted by  $(\bar{x}, \bar{y})$ , where  $\bar{x} = h$  is the mean of the  $x$ -coordinates of the points in  $B_h$ , and  $\bar{y}$  is the mean of the  $y$ -coordinates of the points in  $B_h$ . By doing the above summarization for each possible value of  $h$ , we convert the points in  $Z$  into a set of mean points with distinct  $x$ -coordinates. This set of mean points forms the *discrete-time signal*, denoted by  $T$ . This maps  $Q$  to the discrete-time signal  $T$ , which is written as  $\bar{T} = \varphi(Q)$ .

## 4.2 Adding Sinusoidal Signal

In this section, we introduce how to add a sinusoidal signal with a specific frequency to the discrete-time signal  $T$  by slightly modifying the data instances in  $Q$ . The *sinusoidal signal* is denote by  $y = \lambda \sin(2\pi\theta x)$ , where  $\lambda$  is the amplitude of the signal, and  $\theta$  is the frequency of the signal. To add the sinusoidal signal into  $T$ , we first map  $Q_{i,:}$  to  $x_i = \phi_x(Q_{i,:})$ , and then update  $Q_{i,:}$  by

$$Q'_{i,:} = Q_{i,:} + \lambda \sin(2\pi\theta x_i) * \mathbf{e}_y. \quad (4)$$

By applying Eq. (4) on every  $Q_{i,:}$  in  $Q$ , we modify  $Q$  into a watermarked dataset  $Q'$ , where the sinusoidal signal is embedded as a watermark.

We summarize the method to generate  $Q'$  in Algorithm 1, where the secret key, denoted by  $\mathcal{K} = \{\mathbf{e}_x, \mathbf{e}_y, \theta, b, \tau\}$ , contains the necessary variables to verify the watermark (i.e., the sinusoidal signal). The time complexity of Algorithm 1 is  $O(nd)$ , where  $n$  and  $d$  are the numbers of rows and columns in  $Q$ , respectively.

**Theorem 1.** *If  $Q'$  is obtained by Algorithm 1, then  $T' = \varphi(Q')$  contains a component of the sinusoidal signal with frequency  $\theta$ .*

*Proof.* We prove this theorem by showing that, for every mean point  $(\bar{x}', \bar{y}') \in T'$ , the analytical form of  $\bar{y}'$  contains a sinusoidal term  $\lambda \sin(2\pi\theta\bar{x}')$ . Without loss of generality, we assume  $\bar{x}' = h$  and derive the analytical form of  $\bar{y}'$  as follows.

Since  $\bar{x}' = h$ , we know  $(x', y')$  is summarized from the bin

$$B'_h = \{(x'_i, y'_i) \mid x'_i = h, (x'_i, y'_i) \in Z'\}, \quad (5)$$

where  $Z' = \{(x'_i, y'_i) \mid x'_i = \phi_x(Q'_{i,:}), y'_i = \phi_y(Q'_{i,:}), i \in \{1, \dots, n\}\}$ .

For each point  $(x'_i, y'_i) \in B'_h$ , we can derive from Eq. (4) and  $\mathbf{e}_x \mathbf{e}_y^\top = 0$  that  $Q'_{i,:} \mathbf{e}_x^\top = Q_{i,:} \mathbf{e}_x^\top$ . Then, we can derive from Eq. (2) that

$$x'_i = \phi_x(Q'_{i,:}) = \frac{\lfloor \frac{Q'_{i,:} \mathbf{e}_x^\top}{b} \rfloor}{\tau} = \frac{\lfloor \frac{Q_{i,:} \mathbf{e}_x^\top}{b} \rfloor}{\tau} = \phi_x(Q_{i,:}) = x_i. \quad (6)$$

Since  $(x'_i, y'_i) \in B'_h$ , we know  $x'_i = h$  by the definition of  $B'_h$ . Thus,

$$x'_i = x_i = h. \quad (7)$$

Since  $\mathbf{e}_y \mathbf{e}_y^\top = 1$ , we can derive from Eqs. (1) and (4) that

$$y'_i = Q'_{i,:} \mathbf{e}_y^\top = Q_{i,:} \mathbf{e}_y^\top + \lambda \sin(2\pi\theta x_i) = y_i + \lambda \sin(2\pi\theta x_i). \quad (8)$$

By plugging Eq. (7) into the above equation, we have

$$y'_i = y_i + \lambda \sin(2\pi\theta h), \quad (9)$$

which holds for every point  $(x'_i, y'_i) \in B'_h$ .

Since  $\bar{y}'$  is the mean of the  $y$ -coordinates of all the points  $(x'_i, y'_i) \in B'_h$ , we can derive the analytical form of  $\bar{y}'$  as

$$\bar{y}' = \frac{1}{|B'_h|} \sum_{(x'_i, y'_i) \in B'_h} y'_i = \left( \frac{1}{|B'_h|} \sum_{(x'_i, y'_i) \in B'_h} y_i \right) + \lambda \sin(2\pi\theta h). \quad (10)$$

Since  $\bar{x}' = h$ , the analytical form of  $\bar{y}'$  is

$$\bar{y}' = \left( \frac{1}{|B'_h|} \sum_{(x'_i, y'_i) \in B'_h} y_i \right) + \lambda \sin(2\pi\theta \bar{x}'), \quad (11)$$

which contains the sinusoidal term  $\lambda \sin(2\pi\theta \bar{x}')$  with frequency  $\theta$ .

---

**Algorithm 2.** Watermark detection

---

- Input:** A suspicious dataset  $\hat{Q}$ , a confidence level  $p \in [0, 1]$  and  $\mathcal{K}$ .  
**Output:** The detection result  $W \in \{0, 1\}$ .  
 1: Obtain the discrete-time signal  $\hat{T} = \varphi(\hat{Q})$ .  
 2: Obtain the spectrum power  $\hat{P}(\theta)$  from  $\hat{T}$  by LSP [36].  
 3: Use LSP to estimate the threshold  $\eta_p$  from  $p$ .  
 4: **If**  $\hat{P}(\theta) \geq \eta_p$ , **then** return  $W = 1$ . (*Watermark is detected*)  
 5: **If**  $\hat{P}(\theta) < \eta_p$ , **then** return  $W = 0$ . (*Watermark is not detected*)
- 

According to Theorem 1,  $T'$  contains the sinusoidal signal with frequency  $\theta$ , which means  $Q'$  is successfully embedded with the sinusoidal signal by Algorithm 1.

**Theorem 2.** Denote by  $\eta$  the maximum absolute value of all the entries in  $\mathbf{e}_y$  and by  $\Delta_{i,j} = |Q_{i,j} - Q'_{i,j}|$  the absolute modification made on the  $j$ -th attribute of  $Q_{i,:}$  when embedding the watermark. If  $Q'$  is obtained by Algorithm 1, then  $\Delta_{i,j} \leq \lambda\eta$ .

*Proof.* Since each  $Q'_{i,:}$  is obtained by modifying  $Q_{i,:}$  using Eq. (4) and  $-1 \leq \sin(2\pi\theta x_i) \leq 1$ , we have  $\Delta_{i,j} \leq \lambda\eta$ .

## 5 PK-Free Watermark Detection

In this section, we introduce how to detect a watermark from a suspicious dataset  $\hat{Q}$  that may or may not be watermarked by a secret key  $\mathcal{K}$ .

Denote by  $\hat{T}$  the discrete-time signal of  $\hat{Q}$ . We obtain  $\hat{T}$  by mapping the data instances in  $\hat{Q}$  in the same way as how we map  $Q$  to  $T$ , that is,  $\hat{T} = \varphi(\hat{Q})$ . This process requires to know the variables  $\mathbf{e}_x$ ,  $\mathbf{e}_y$ ,  $b$  and  $\tau$  in  $\mathcal{K}$ . Then, we check whether  $\hat{T}$  contains the sinusoidal signal with the frequency  $\theta \in \mathcal{K}$  by checking the spectrum power of  $\hat{T}$  at the frequency  $\theta$ , denoted by  $\hat{P}(\theta)$ .

We use Lomb-Scargle Periodogram (LSP) [27, 36] to compute the spectrum power  $\hat{P}(\theta)$  of  $\hat{T}$ . LSP provides a probabilistic method to determine whether a sinusoidal signal is a true signal in  $\hat{T}$  [36]. Denote by  $p \in [0, 1]$  the probability of a sinusoidal signal being a true signal in  $\hat{T}$ , LSP estimates a threshold  $\eta_p$  based on  $p$ . If  $\hat{P}(\theta) \geq \eta_p$ , then the probability of  $\hat{T}$  containing the sinusoidal signal with frequency  $\theta$  is at least  $p$ . This allows us to use  $p$  as a confidence level when detecting watermark from  $\hat{T}$ . For example, we can set  $p = 0.99$  and compare  $\hat{P}(\theta)$  with the threshold  $\eta_{0.99}$ . If  $\hat{P}(\theta) \geq \eta_{0.99}$ , then  $\hat{T}$  is watermarked by  $\mathcal{K}$  at the confidence level of 0.99. Otherwise,  $\hat{T}$  is not watermarked by  $\mathcal{K}$  at the confidence level of 0.99. Algorithm 2 summarizes how to detect watermark. The time complexity is  $O(n(d + 1))$ , where  $n$  and  $d$  are the number of rows and columns of  $\hat{Q}$ , respectively.



## 6 Threat Model and Attacks

In this section, we provide a comprehensive discussion of the four requirements of the threat model and delve into six typical watermark-removing attacks that an attacker might employ.

**Threat Model.** Following the literature [14, 21, 29, 30, 34], we consider a typical threat model consisting of four requirements: **(1)** the attacker can access the watermarked dataset  $Q'$ ; **(2)** the attacker cannot access the original dataset  $Q$ ; **(3)** the attacker cannot access the secret key  $\mathcal{K}$ ; and **(4)** the attacker cannot change the feature space of the original attributes in  $Q'$ . Here, the requirement (4) is practical because the semantic meaning (e.g., meta-data) carried by the original attributes of  $Q'$  is a valuable part of  $Q'$ . Moreover, if the feature space of an attacked dataset, denoted by  $\tilde{Q}$ , is different from the feature space of  $Q'$ , then a machine learning model trained on  $\tilde{Q}$  cannot generalize to new data instances represented by the original attributes of  $Q'$ .

**Watermark-Removing Attacks.** We consider the following typical attacks in the literature. **(1) Uniform alteration (UA)** [15, 30, 34] adds uniform noise sampled from  $U[-\rho_{ua}, \rho_{ua}]$  to the attributes of all the data instances in  $Q'$ . A larger  $\rho_{ua}$  implies a stronger attack. **(2) Row deletion (RD)** [15, 21, 30, 34] deletes uniformly sampled data instances of  $Q'$ . Denote by  $\rho_{rd}$  the proportion of the deleted data instances in  $Q'$ , a larger  $\rho_{rd}$  implies a stronger attack. **(3) Row insertion (RI)** [15, 21, 30, 34] inserts noise data instances to  $Q'$ . For each noise data instance, the  $j$ -th entry is sampled from a uniform distribution  $U[\mu_j - \sigma_j, \mu_j + \sigma_j]$ , where  $\mu_j$  and  $\sigma_j$  are the mean and standard deviation of  $j$ -th attribute of  $Q'$ . Denote by  $\rho_{ri}$  the proportion of inserted noise data instances, a larger  $\rho_{ri}$  implies a stronger attack. **(4) Column deletion (CD)** deletes uniformly sampled columns in  $Q'$ . Denote by  $\rho_{cd}$  the proportion of the deleted columns, a larger value of  $\rho_{cd}$  implies a stronger attack. **(5) PCA attack (PCA)** modifies the data instances in  $Q'$  by using principal component analysis (PCA) [1] to perform dimensionality reduction. We map the data instances back to the original feature space after discarding  $k$  dimensions in the feature space spanned by eigenvectors. Denote by  $\rho_{pca} = \frac{k}{d}$  the proportion of discarded dimensions, a larger  $\rho_{pca}$  implies a stronger attack. **(6) Re-watermarking (RE)** [18] attacks the original watermark in  $Q'$  by embedding a new watermark into  $Q'$ . We use the proposed watermarking method to embed the new watermark. Denote by  $\rho_{re}$  the amplitude of the new sinusoidal signal  $y = \rho_{re} * \sin(2\pi\theta x)$  embedded into  $Q'$ , a larger  $\rho_{re}$  implies a stronger attack.

## 7 Experiments

In this section, we conduct comprehensive experiments on two real-world datasets to study the performance of our method and five baseline methods. We focus on answering two questions: **(1)** How robust are the watermarks of each watermarking method against the attacks? **(2)** How is the machine learning utility of a watermarked dataset affected by the attacks? All the experiments were

conducted on a desktop with an Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz and 64 gigabytes of RAM.

**Table 1.** Information of datasets.

Dataset	#Instances	#Attributes	#Classes
Forest cover type (FCT) dataset [29]	581,012	54	7
Gas sensor array drift (GSAD) dataset [7]	13,910	128	6

**Datasets.** We use the two real-world datasets FCT<sup>1</sup> and GSAD<sup>2</sup> in Table 1. Since the original FCT dataset is too big for the baseline methods OBT [34] and IP [19] to finish the experiments in practical time, we uniformly sample 50% of instances from the original FCT dataset to do our experiments. Following the setting of [20], we use the top-4 (top-20) attributes with the largest information gain to embed watermark in FCT (GSAD). Denote by  $m$  the number of attributes used to embed watermark. Embedding watermark in this way increases the cost of conducting column deletion attack, because deleting a column with a larger information gain causes more damage to the machine learning utility of the dataset.

**Machine Learning Utility (MLU).** We evaluate the MLU of a dataset by the testing accuracy of a machine learning model. Each dataset is uniformly split into a training set and a testing set with a ratio of 4:1. The training set is used to embed/detect watermarks and train the machine learning model. The testing set is used to evaluate the machine learning model’s testing accuracy, which is regarded as the MLU of the training dataset. We evaluate MLU by two machine learning models, such as multi-class logistic regression (LR) model [23] and multi-class support vector machine (SVM) [38].

**Baseline Methods.** We use five baseline methods, such as NR [33], OBT [34], IP [19], GAHSW [15], and SCPW [30]. These methods need a primary key or a virtual primary key to work properly. Since an attacker can easily modify the primary key to remove a watermark without damaging the dataset’s machine learning utility, we develop two versions of implementations for each baseline method by using two state-of-the-art virtual primary key generation methods. One version uses M-Scheme [25] to implement the baselines as NR-M, OBT-M, IP-M, GAHSW-M and SCPW-M. The other version uses HQR [10] to implement the baselines as NR-H, OBT-H, IP-H, GAHSW-H and SCPW-H.

**Secret Keys.** For each of the compared methods, we use 10 independent secret keys, denoted by  $\mathcal{K}_{(1)}, \dots, \mathcal{K}_{(10)}$ , to embed watermarks. The baseline methods use a sequence of bits as a secret key and we use 16 bits as the default length of each secret key. For each baseline method, we use 10 secret keys with maximum pairwise hamming distance. For our method, the sampled values of

<sup>1</sup> <https://archive.ics.uci.edu/dataset/31/coverttype>.

<sup>2</sup> <https://archive.ics.uci.edu/dataset/224/gas+sensor+array+drift+dataset>.

$\theta, b$  and  $\tau$  in the 10 secret keys are listed in Table 2. The vectors  $\mathbf{e}_x$  and  $\mathbf{e}_y$  for each secret key are randomly sampled as a pair of orthogonal vectors with L2-norm equal to one.

**Watermark Strength.** The watermark strength in a dataset refers to the intensity of the embedded watermark signal [28]. A stronger watermark enhances robustness against removal attacks but also increases dataset modification, reducing MLU [31]. Thus, increasing watermark strength trades MLU for robustness. To fairly compare the robustness of all methods, we allow each to trade up to 0.01 MLU for robustness, setting  $\gamma = 0.01$  for the MLU constraint in condition (3) of Definition 4. Based on this constraint, we set the default value of  $\lambda$  for our method to 20 for GSAD and 0.8 for FCT unless otherwise specified.

**Table 2.** The  $\theta, b$  and  $\tau$  of the 10 secret keys used for FCT and GSAD.

Dataset	FCT										GSAD									
Secret keys	$\mathcal{K}_{(1)}$	$\mathcal{K}_{(2)}$	$\mathcal{K}_{(3)}$	$\mathcal{K}_{(4)}$	$\mathcal{K}_{(5)}$	$\mathcal{K}_{(6)}$	$\mathcal{K}_{(7)}$	$\mathcal{K}_{(8)}$	$\mathcal{K}_{(9)}$	$\mathcal{K}_{(10)}$	$\mathcal{K}_{(1)}$	$\mathcal{K}_{(2)}$	$\mathcal{K}_{(3)}$	$\mathcal{K}_{(4)}$	$\mathcal{K}_{(5)}$	$\mathcal{K}_{(6)}$	$\mathcal{K}_{(7)}$	$\mathcal{K}_{(8)}$	$\mathcal{K}_{(9)}$	$\mathcal{K}_{(10)}$
$\theta$	30	27	30	32	29	34	37	40	38	39	30	32	34	33	35	36	38	29	35	37
$b (\times 10^{-2})$	0.8	0.5	0.5	2.0	3.0	0.8	2.0	2.0	1.0	1.0	0.2	0.1	0.5	1.0	0.2	2.0	1.0	2.0	0.4	0.8
$\tau (\times 10^4)$	1.0	1.0	0.3	0.5	1.0	0.5	0.9	0.8	0.6	0.8	0.5	0.4	1.0	0.6	0.8	1.0	0.3	0.3	0.5	0.3

### 7.1 How to Evaluate Performance?

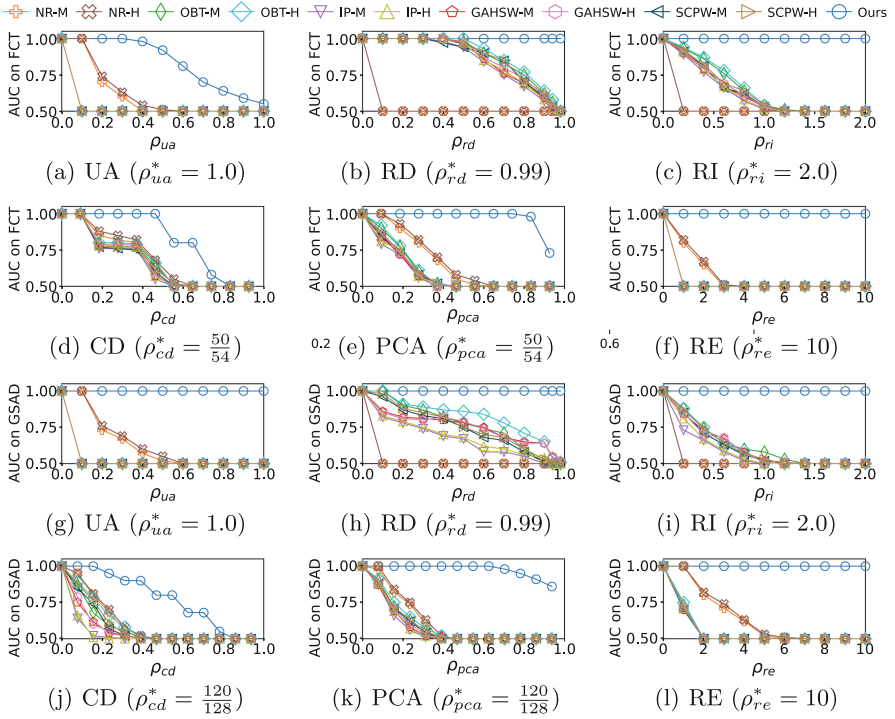
We evaluate the performance of a watermarking method by performing the following steps.

**Step1: Embedding Watermarks.** Denote by  $Q_{(0)}$  the training set of an original dataset that is not watermarked. We use each of the secret keys  $\mathcal{K}_{(1)}, \dots, \mathcal{K}_{(10)}$  to embed a watermark in  $Q_{(0)}$ . This produces 10 watermarked datasets, denoted by  $Q'_{(1)}, \dots, Q'_{(10)}$ , where  $\mathcal{K}_{(i)}$  is the ground truth secret key of  $Q'_{(i)}$ . The ground truth secret key of  $Q_{(0)}$  is denoted by  $\mathcal{K}_0$ , which means no watermark is embedded in  $Q_{(0)}$ . In this way, we construct a collection of datasets denoted by  $\mathcal{C} = \{Q_{(0)}, Q'_{(1)}, \dots, Q'_{(10)}\}$ .

**Step 2: Conducting Attacks.** We attack the datasets in  $\mathcal{C}$  before detecting the watermarks. For each attack, we produce a collection of attacked datasets, denoted by  $\tilde{\mathcal{C}} = \{\tilde{Q}_{(0)}, \tilde{Q}_{(1)}, \dots, \tilde{Q}_{(10)}\}$ .

**Step 3: Detecting Watermarks.** We use each of  $\mathcal{K}_{(1)}, \dots, \mathcal{K}_{(10)}$  to detect watermark from the datasets in  $\tilde{\mathcal{C}}$ . Denote by  $\text{Detect}(\tilde{Q}_{(j)}, \mathcal{K}_{(i)}) \rightarrow W$  the process of using  $\mathcal{K}_{(i)}$  to detect watermark from  $\tilde{Q}_{(j)}$ . If  $\text{Detect}(\tilde{Q}_{(j)}, \mathcal{K}_{(i)}) = 1$ , then we have a positive detection. If  $\text{Detect}(\tilde{Q}_{(j)}, \mathcal{K}_{(i)}) = 0$ , then we have a negative detection. A positive detection is a true positive if  $\mathcal{K}_{(i)}$  is the ground truth secret key of  $\tilde{Q}_{(j)}$ ; otherwise, it is a false positive. A negative detection is a true negative if  $\mathcal{K}_{(i)}$  is not the ground truth secret key of  $\tilde{Q}_{(j)}$ ; otherwise, it is a false negative. Last, we compute the true positive rate and false positive rate.

**Step 4: Evaluating Performance.** We evaluate the performance of a watermarking method by the area under curve (AUC) of the receiver operating characteristic (ROC) curve [6]. A larger AUC means a better performance. For each compared method, the ROC curve is obtained by changing the value of the threshold that decides whether a watermark exists or not. Different methods use different thresholds, for our method, the threshold is  $\eta_p$  in Algorithm 2.



**Fig. 2.** The AUC on FCT shown in (a)-(f) and GSAD shown in (g)-(l). The  $\rho_{ua}^*, \rho_{rd}^*, \rho_{ri}^*, \rho_{cd}^*, \rho_{pca}^*$  and  $\rho_{re}^*$  are the maximum values of each parameter.

### 7.2 How Robust Are the Watermarks?

In this section, we analyze the robustness of watermark against the attacks listed in Sect. 6. Figure 2 shows the AUC of each watermarking method on FCT and GSAD. The  $y$ -axis shows the AUC and the  $x$ -axis shows the strength of each attack controlled by  $\rho_{ua}, \rho_{rd}, \rho_{ri}, \rho_{cd}, \rho_{pca}$  and  $\rho_{re}$  in Sect. 6.

We can see in Fig. 2 that the AUC of most methods drops when the attack strength increases. A slower dropping speed of AUC means a better performance, because it implies the watermarking method is more robust to withstand

a stronger attack. Since the AUC on FCT and GSAD show similar trends, we focus on explaining the results on FCT.

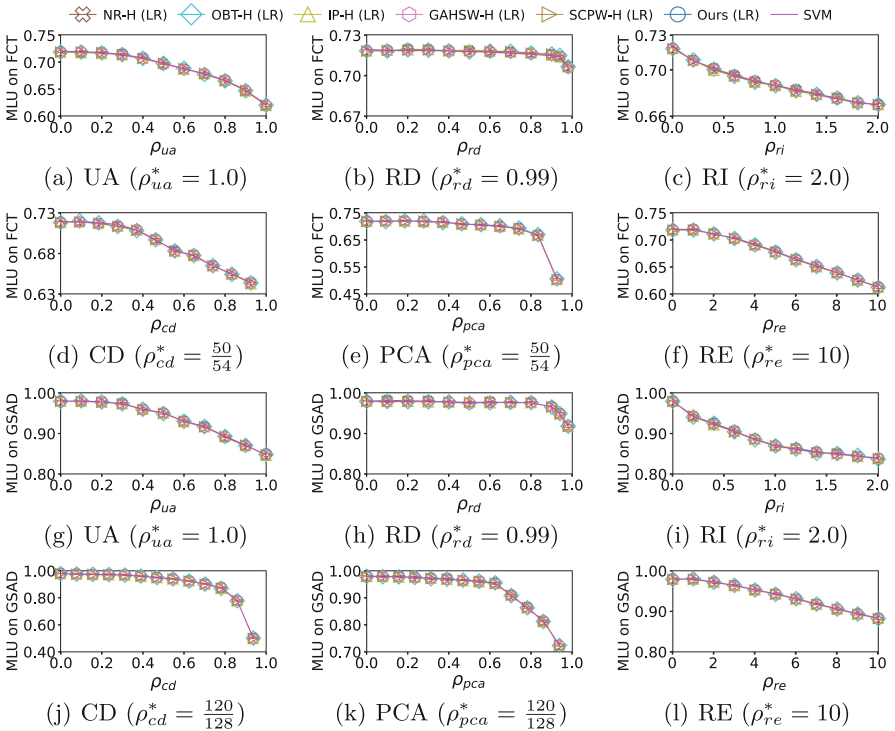
**The AUC of Baseline Methods.** As shown in Fig. 2, the AUC of the baseline methods are inferior to our method because they are affected by attacks due to the following reasons. First, since these methods use virtual primary keys to identify the watermarked groups of data instances, their watermarks are not robust to uniform alteration, column deletion, PCA attack and re-watermark. This is because such attacks change the values of virtual primary key, which corrupts the identification of the watermarked groups. Second, the watermarks of OBT, IP, GAHSW and SCPW are affected by row deletion (RD) and row insertion (RI), because RD removes watermarked data instances from watermarked groups and RI adds noise data instances into watermarked groups. Both the effects weakens the watermark signal. NR is affected by row deletion and row insertion because it relies on the accurate alignment between the data instances before and after attack. The alignment between data instances is disrupted by RD and RI because they change the number of data instances.

**The AUC of Our Method.** As shown in Fig. 2, our method achieves the best AUC, which demonstrates the outstanding robustness of our watermark. In the following, we discuss the performance of our method against each attack. **(1) Uniform alteration (UA).** In Figs. 2(a) and 2(g), our method is robust to UA because: **i)** it is primary key free, enhancing noise robustness; **ii)** the  $x$ -axis binning operation stabilizes the discrete time signal against noise; and **iii)** the Lomb-Scargle Periodogram (LSP) [27,36] is noise-resistant. UA has a bigger impact upon our method on FCT than on GSAD because GSAD has more attributes than FCT, thus embedding a watermark causes less damage to the MLU of GSAD than FCT. This allows our method to embed a stronger watermark on GSAD without violating the MLU constraint in Definition 4. **(2) Row deletion (RD) and row insertion (RI).** In Figs. 2(b),2(c),2(h) and 2(i), our method achieves outstanding AUC against RD and RI. Because RD removes watermarked points from  $B'_h$  and RI adds noise points in  $B'_h$ , but neither changes the remaining watermarked points in  $B'_h$ . Thus, the watermark signal is largely retained. **(3) Column deletion (CD).** In Figs. 2(d) and 2(j), CD impacts our method's AUC because deleting one of the  $m$  watermarked columns (i.e., attributes) in  $Q'$  removes  $\frac{1}{m}$  of the watermark signal. However, since our watermark is embedded in multiple columns that are unknown to the attacker, we still achieve high AUC even when half the columns of  $Q'$  are randomly deleted. **(4) PCA attack.** In Figs. 2(e) and 2(k), the impact of PCA attack is smaller than column deletion. This is because the dimensions discarded by PCA attack often have small information gain but we embed our watermark in the columns with large information gain. **(5) Re-watermarking (RE).** In Figs. 2(f) and 2(l), the AUC of our watermark is robust against RE. This is because embedding a new watermark adds random noise to  $Q'$ , which has a similar effect to uniform alteration attack. Therefore, our watermark is robust against RE.

### 7.3 How Is MLU Affected by the Attacks?

Figure 3 shows the MLU of each watermarked dataset under different strengths of attacks. The  $x$ -axis is defined in the same way as Fig. 2 and the  $y$ -axis shows the MLU of the attacked watermarked datasets.

Since drawing all the MLU curves is too crowded, we simplify the view as follows. First, for MLU measured by LR, we only draw the MLU of our method and each baseline method using HQR [10] as VPK, marked with a “(LR)” suffix. We omit the baseline methods using M-Scheme [25] as VPK because their absolute MLU difference is at most 0.004. Second, for MLU measured by SVM, we draw one curve marked “SVM” in Fig. 3. This curve with an error bar represents the mean and standard deviation (std) of the MLU of all methods, including baseline methods using both VPK versions and our method. The std is at most 0.003 in all cases.



**Fig. 3.** The MLU on FCT shown in (a)-(f) and GSAD shown in (g)-(l) The  $\rho_{ua}^*$ ,  $\rho_{rd}^*$ ,  $\rho_{ri}^*$ ,  $\rho_{cd}^*$ ,  $\rho_{pca}^*$  and  $\rho_{re}^*$  are the maximum values used for each parameter.

**Why are the MLU Curves Close?** This is due to the small constraint of  $\gamma = 0.01$ . Since  $Q'$  from different watermarking methods is computed from the

same dataset  $Q$ , the MLU of different  $Q'$  are almost identical. Thus, when these  $Q'$  undergo the same attack, they exhibit similar MLU curves.

**What did We Learn from the MLU Curves?** The MLU of the attacked watermarked dataset decreases with stronger attacks. This means an attacker cannot indefinitely increase the strength of attack to remove a watermark, because a stronger attack will reduce more MLU, which causes more damage to the resale value of the dataset. Since all watermarking methods have similar MLUs, the method with the slowest AUC drop as attack strength increases offers the best protection. Our method provides the best protection, because its AUC drops the slowest in Fig. 2.

## 8 Conclusion

In this paper, we propose a novel primary key free watermarking method for tabular datasets. Different from many existing watermarking methods, our method does not use a primary key to embed and detect watermarks. This makes it particularly suitable for watermarking tabular datasets used for machine learning, because such datasets often do not come with a primary key and an existing primary key can be easily modified without degrading the machine learning utility of the dataset. As demonstrated by extensive experiments, our method achieves outstanding robustness against many watermark-removing attacks, which provides strong protection on watermarked datasets.

## References

1. Abdi, H., Williams, L.J.: Principal component analysis. *Comput. Stat.* **2**, 433–459 (2010)
2. Agrawal, R., Haas, P.J., Kiernan, J.: Watermarking relational data: framework, algorithms and analysis. *VLDB J.* **12**, 157–169 (2003)
3. Agrawal, R., Kiernan, J.: Watermarking relational databases. In: *VLDB*, pp. 155–166 (2002)
4. Anand, A., Singh, A.K.: Watermarking techniques for medical data authentication: a survey. *Multimedia Tools Appl.* **80**, 165–197 (2021)
5. Bhattacharya, S., Cortesi, A.: A distortion free watermark framework for relational databases. In: *ICSOFT*, pp. 229–234 (2009)
6. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recogn.* **30**, 1145–1159 (1997)
7. Das, P., Manna, A., Ghoshal, S.: Gas sensor drift compensation by ensemble of classifiers using extreme learning machine. In: *ICSGE*, pp. 197–201 (2020)
8. Even, A., Shankaranarayanan, G., Berger, P.D.: Economics-driven data management: an application to the design of tabular data sets. *IEEE TKDE* **19**, 818–831 (2007)
9. Gort, M.L.P., Díaz, E.A., Uribe, C.F.: A highly-reliable virtual primary key scheme for relational database watermarking techniques. In: *CSCI*, pp. 55–60 (2017)
10. Gort, M.L.P., Feregrino-Uribe, C., Cortesi, A., Fernández-Peña, F.: HQR-scheme: a high quality and resilient virtual primary key generation approach for watermarking relational data. *Expert Syst. Appl.* **138**, 795–825 (2019)



11. Gort, M.L.P., Feregrino-Uribe, C., Cortesi, A., Fernández-Pena, F.: A double fragmentation approach for improving virtual primary key-based watermark synchronization. *IEEE Access* **8**, 504–516 (2020)
12. Gupta, G., Pieprzyk, J.: Database relation watermarking resilient against secondary watermarking attacks. In: *ICISS*, pp. 222–236 (2009)
13. Hashim, H.: Hybrid warehouse model and solutions for climate data analysis. *J. Comput. Commun.* **8**, 75–98 (2020)
14. Hu, D., Wang, Q., Yan, S., Liu, X., Li, M., Zheng, S.: Reversible database watermarking based on order-preserving encryption for data sharing. *ACM Trans. Database Syst.* **48**, 1–25 (2023)
15. Hu, D., Zhao, D., Zheng, S.: A new robust approach for reversible database watermarking with distortion control. *IEEE TKDE* **31**, 1024–1037 (2018)
16. Hülsmann, J., Traub, J., Markl, V.: Demand-based sensor data gathering with multi-query optimization. In: *VLDB*, vol. 13, pp. 2801–2804 (2020)
17. Iftikhar, S., Kamran, M., Anwar, Z.: RRW-a robust and reversible watermarking technique for relational data. *IEEE TKDE* **27**, 1132–1145 (2014)
18. İşler, D., Cabana, E., Garcia-Recuero, A., Koutrika, G., Laoutaris, N.: FreqWM: frequency watermarking for the new data economy. *arXiv preprint [arXiv:2312.16547](https://arxiv.org/abs/2312.16547)* (2023)
19. Kamran, M., Farooq, M.: An information-preserving watermarking scheme for right protection of EMR systems. *IEEE TKDE* **24**, 1950–1962 (2011)
20. Kamran, M., Farooq, M.: A formal usability constraints model for watermarking of outsourced datasets. *IEEE TIFS* **8**, 1061–1072 (2013)
21. Kamran, M., Farooq, M.: A comprehensive survey of watermarking relational databases research. *arXiv preprint [arXiv:1801.08271](https://arxiv.org/abs/1801.08271)* (2018)
22. Kumar, S., Singh, B.K., Yadav, M.: A recent survey on multimedia and database watermarking. *Multimedia Tools Appl.* **79**, 149–197 (2020)
23. Kwak, C., Clayton-Matthews, A.: Multinomial logistic regression. *Nurs. Res.* **51**, 404–410 (2002)
24. Li, Q., et al.: Database watermarking algorithm based on decision tree shift correction. *IEEE Internet Things J.* **9**, 24373–24387 (2022)
25. Li, Y., Swarup, V., Jajodia, S.: Constructing a virtual primary key for fingerprinting relational data. In: *Proceedings of the ACM Workshop on Digital Rights Management*, pp. 133–141 (2003)
26. Liu, Q., Xian, H., Zhang, J., Liu, K.: A random reversible watermarking scheme for relational data. In: *SecureComm*, pp. 413–430 (2022)
27. Lomb, N.R.: Least-squares frequency analysis of unequally spaced data. *Astrophys. Space Sci.* **39**, 447–462 (1976)
28. Podilchuk, C.I., Delp, E.J.: Digital watermarking: algorithms and applications. *IEEE Signal Process. Mag.* **18**, 33–46 (2001)
29. Rani, S., Halder, R.: Comparative analysis of relational database watermarking techniques: an empirical study. *IEEE Access* **10**, 970–989 (2022)
30. Ren, Z., et al.: A robust database watermarking scheme that preserves statistical characteristics. *IEEE TKDE* (2023)
31. Šarčević, T., Mayer, R., Rauber, A.: Adaptive attacks and targeted fingerprinting of relational data. In: *ICBD*, pp. 5792–5801 (2022)
32. Sebé, F., Domingo-Ferrer, J., Castella-Roca, J.: Watermarking numerical data in the presence of noise. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **14**, 495–508 (2006)



33. Sebé, F., Domingo-Ferrer, J., Solanas, A.: Noise-robust watermarking for numerical datasets. In: Torra, V., Narukawa, Y., Miyamoto, S. (eds.) MDAI 2005. LNCS (LNAI), vol. 3558, pp. 134–143. Springer, Heidelberg (2005). [https://doi.org/10.1007/11526018\\_14](https://doi.org/10.1007/11526018_14)
34. Shehab, M., Bertino, E., Ghafoor, A.: Watermarking relational databases using optimization-based techniques. *IEEE TKDE* **20**, 116–129 (2007)
35. Sion, R., Atallah, M., Prabhakar, S.: Rights protection for relational data. In: *SIGMOD*, pp. 98–109 (2003)
36. VanderPlas, J.T.: Understanding the Lomb-Scargle periodogram. *Astrophys. J. Suppl. Ser.* **236**, 1–16 (2018)
37. Wang, C., Li, Y.: A copyright authentication method balancing watermark robustness and data distortion. In: *CSCWD*, pp. 1178–1183 (2023)
38. Wang, Z., Xue, X.: Multi-class support vector machine. In: *Support Vector Machines Applications*, pp. 23–48 (2014)



# Offline Handwritten Signature Verification Using a Stream-Based Approach

Kecia Gomes de Moura<sup>(✉)</sup>, Rafael Menelau O. Cruz, and Robert Sabourin

École de technologie supérieure - Université du Québec, Montreal, Québec, Canada  
kecia.gomes-de-moura.1@ens.etsmtl.ca,  
{rafael.menelau-cruz, robert.sabourin}@etsmtl.ca

**Abstract.** Handwritten Signature Verification (HSV) systems distinguish between genuine and forged signatures. Traditional HSV development involves a static batch configuration, constraining the system's ability to model signatures to the limited data available. Signatures exhibit high intra-class variability and are sensitive to various factors, including time and external influences, imparting them a dynamic nature. This paper investigates the signature learning process within a data stream context. We propose a novel HSV approach with an adaptive system that receives an infinite sequence of signatures and is updated over time. Experiments were carried out on GPDS Synthetic, CEDAR, and MCYT datasets. Results demonstrate the superior performance of the proposed method compared to standard approaches that use a Support Vector Machine as a classifier. Implementation of the method is available at [https://github.com/kdMoura/stream\\_hsv](https://github.com/kdMoura/stream_hsv).

**Keywords:** Offline signature · biometric authentication · handwritten signature · data stream · dissimilarity data · adaptive classifier

## 1 Introduction

Handwritten Signature Verification (HSV) systems aim to automatically distinguish between genuine signatures, belonging to the claimed individual, and forgeries. In offline HSV, signatures are represented as digital images captured after the writing process is completed, as opposed to online systems that analyze the signing dynamics [7].

Offline HSV systems can be categorized into two approaches: writer-dependent (WD) and writer-independent (WI). In WD systems, a unique classifier is trained for each enrolled user, offering potentially higher accuracy. However, this approach requires individual training data for each new user. Conversely, WI systems utilize a single classifier for all users, hence being more

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78119-3\\_19](https://doi.org/10.1007/978-3-031-78119-3_19).

scalable [5]. In this case, the classification is performed on a dissimilarity space where the pattern recognition is reduced to a 2-class problem by using differences between claimed and reference signatures through a Dichotomy Transformation (DT) [5].

In general, HSV development entails two distinct datasets: a development set utilized for training and an exploitation set employed during the testing phase [6]. Each set comprises the signatures of enrolled users. While more samples generally lead to better generalization models, real-world applications often face data availability limitations regarding both user count and sample volume [5]. Therefore, the system's capability to model signature variations is constrained to the present available data. After the training process, the resulting HSV is expected to achieve generalization to the whole set of existing users and their signatures.

Nonetheless, by relying on a training process with a finite dataset, the current literature does not account for the inherent variability and changing behavior of handwriting signatures. Signatures exhibit the highest intra-class variability compared to other biometric traits [5]. Additionally, signature patterns are time-sensitive as they evolve as we age. Besides, diverse factors can impact the signing process, including emotional states, stress levels, fatigue, and influences from substances like alcohol or drugs [1]. Writing results are intrinsically related to cognitive-motor and neuromotor conditions, being affected by any minor impairment [1].

Given these challenges, we pose the following general research question: *how can a signature verification system adapt to the inherent variability and evolving nature of handwritten signatures over time, maintaining high verification performance while mitigating the problem of limited data?* To answer this question, we propose a framework to handle signature verification in an adaptive manner, where the input data is processed as a stream of offline signatures rather than a batch mode.

In the proposed framework, incoming signatures are first tested and then used to improve the system by updating its current state. In this approach, *SigNet-S* [20], one of the state-of-the-art representation models, is employed to extract features of incoming claimed signatures. These feature vectors are then compared to corresponding reference vectors stored in the database to create dissimilarity samples via a stream dichotomy transformation. Lastly, the adaptive WI-classifier is updated based on the dissimilarity vectors. To the best of our knowledge, no prior work has considered signatures in an open-set, stream-based configuration.

The main contributions of this article are as follows:

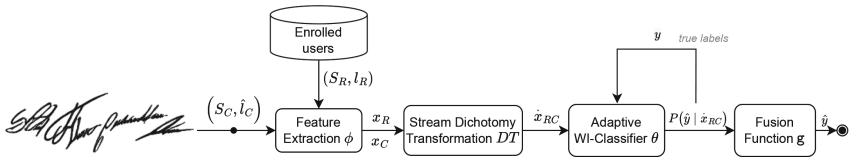
1. Stream HSV: we propose a novel HSV framework that adapts over time. This framework treats signatures as an infinite data stream, enabling continuous learning and improvement.
2. A stream dichotomy transformation: we introduce a stream dichotomy transformation process to facilitate adaptive learning from the incoming signature stream and address the challenge of imbalanced data ratios commonly encountered in stream scenarios.

- Signature stream generation method: to facilitate evaluation using standard batch configurations, we introduce a method for generating signature streams based on the existing HSV evaluation protocol.

## 2 Stream Handwriting Signature Verification (SHSV)

Streaming systems are designed to continuously process and analyze data as it arrives, delivering updated results based on the most recent characteristics of the information. In the context of handwriting signature verification, this approach enables adaptive signature authentication, accounting for the inherent variability of this biometric modality. As the system evolves to accommodate variations in handwriting patterns, it ensures accurate and reliable verification.

In light of this, we propose a framework for signature verification under a data stream context. Specifically, we introduce a system that takes as input a sequence of signatures  $S$  of claimed users  $\hat{l}$ , denoted as  $Stream = \{(S, \hat{l})_1, (S, \hat{l})_2, \dots, (S, \hat{l})_\infty\}$ , for verification against the signatures of enrolled users. As new signature samples arrive (from new or enrolled users), the system incorporates new information into its base knowledge and delivers updated results on the next verification. The system is depicted in Fig. 1, and notation is synthesized in Table 1.



**Fig. 1.** Stream HSV system.  $(S_C, \hat{l}_C)$  denotes a claimed signature from the stream, and  $(S_R, l_R)$  a reference signature of the corresponding user. Signatures, after being preprocessed, have their features extracted by a representation model  $\phi$ . The stream dichotomy transformation is applied to the pair of features vectors  $DT(\mathbf{x}_R, \mathbf{x}_C)$  and passed to the adaptive classifier  $\theta$ , which outputs a prediction. At the end, a fusion function is employed considering all reference signatures to deliver a final result. If true labels are available, the classifier is updated with all new dissimilarities information.

The SHSV is a general framework that comprises fundamental components that enable the system to work in a writer-independent (WI) manner. They are following described.

### 2.1 Representation Model $\phi(\cdot)$

The representation model is a previously well-trained model capable of extracting relevant features from the signature images. To this end, the SHSV employs the SigNet Synthetic (*SigNet-S*) developed by [20]. This model is a variant of the

**Table 1.** SHSV notation.

Sym.	Description	Sym.	Description
$S$	Signature image	$\tilde{\mathbb{X}}$	Set of dissimilarity vectors
$l$	Writer label	$\mathcal{D}$	Development set
$\phi(\cdot)$	Feature extractor	$n\mathcal{D}$	Number of writers in $\mathcal{D}$
$DT(\cdot, \cdot)$	Dichotomy transformation	$n\mathcal{D}_G$	Number of genuine sig. per writer in $\mathcal{D}$
$\theta$	WI-classifier	$\mathcal{E}$	Exploitation set
$\mathbf{g}(\cdot)$	Fusion function	$n\mathcal{E}$	Number of writers in $\mathcal{E}$
$\mathbf{x}$	Feature vector $\mathbf{x} = \phi(S)$	$n\mathcal{E}_R$	Number of ref. sig. per writer in $\mathcal{E}$
$\dot{\mathbf{x}}$	Dissimilarity vector $\dot{\mathbf{x}} = DT(\mathbf{x}_1, \mathbf{x}_2)$	$n\mathcal{E}_C$	Number of G, RF, and SK claimed sig. per writer in $\mathcal{E}$
$y$	Dissimilarity label	$\mathbb{T}$	Stream obtained from $\mathcal{E}$
$G$	Genuine signature	$\mathbb{T}_k$	Chunk $k$ of arriving signatures from $\mathbb{T}$
$RF$	Random forgery signature	$S_G^{i,j}$	$j$ -th genuine signature of writer $i$
$SK$	Skilled forgery signature	$\dot{x}_G^{i,k,j}$	Dissimilarity vector between reference $k$ and genuine signature $j$ of writer $i$
$R$	Reference signature	$\tilde{\mathbb{X}}_G^{i,j}$	Set of dissimilarities from all references and genuine signature $j$ of writer $i$
$C$	Claimed signature	$c_{\text{size}}$	Chunk size for model update
$\mathbb{S}$	Set of signatures	$w_{\text{size}}$	Window size for stream evaluation
$\mathbb{X}$	Set of feature vectors	$w_{\text{step}}$	Step size for evaluation frequency

original *SigNet* proposed by [6]. While the original *SigNet* was trained using signature examples obtained from the GPDS-960 Grayscale [18], which is no longer publicly available due to the General Data Protection Regulation (EU) 2016/679, *SigNet-S* was trained using synthetic GPDS data [4].

*SigNet-S* leverages Deep Convolutional Neural Networks (DCNNs) to learn signature representations by capturing the most discriminative characteristics that distinguish different writers. *SigNet-S* employs a writer-independent training strategy, utilizing only genuine signatures for model development. This enables its application to new incoming writers. During feature extraction for new users' signatures, the network performs feed-forward propagation until the fully connected layer before Softmax, which outputs feature vectors with a dimensionality of 2048. In this work, these vectors represent the feature space of each arriving signature. Formally, given a signature image  $S_C$  of a claimed user, its feature vector is defined by  $\mathbf{x}_C = \phi(S_C)$ .

## 2.2 Stream Dichotomy Transformation $DT(\cdot, \cdot)$

An essential part of the SHSV system is the dichotomy transformation  $DT(\cdot, \cdot)$  [2]. It transforms a multi-class problem into a 2-class problem. This enables the implementation of a writer-independent approach for the classification task,

which is crucial for the stream context. The binary-problem result is achieved by computing the absolute distance between each feature of two feature vectors, i.e., the dissimilarity between two samples. Suppose  $(\mathbf{x}_R, l_R)$  and  $(\mathbf{x}_C, l_C)$  are the feature vector ( $\mathbf{x}$ ) and label ( $l$ ) of two data samples, where  $l$  refers to the author's ID. With  $\mathbf{x}_R = \{f_k^R\}_{k=1}^K$  and  $\mathbf{x}_C = \{f_k^C\}_{k=1}^K$ , where  $K$  is the number of features  $f$ . The dissimilarity vector between  $\mathbf{x}_R$  and  $\mathbf{x}_C$  is given by  $\hat{\mathbf{x}}_{RC} = DT(\mathbf{x}_R, \mathbf{x}_C) = \{|f_k^R - f_k^C|\}_{k=1}^K$ , where  $|\cdot|$  represents the absolute value of the difference. The vector  $\hat{\mathbf{x}}_{RC}$  has the same dimensionality as  $\mathbf{x}_R$  and  $\mathbf{x}_C$ .

The resulting dissimilarity set after applying  $DT(\cdot)$  on  $(\mathbf{x}_R, l_R)$  and  $(\mathbf{x}_C, l_C)$  is given by  $(\hat{\mathbf{x}}_{RC}, y_{RC})$  where  $y$  denotes the new label. If  $l_R = l_C$ , i.e.,  $\hat{\mathbf{x}}_{RC}$  is obtained from signatures of the same writer, it is labeled as *positive* ( $y = +$ ). Otherwise, it will be labeled as *negative* ( $y = -$ ), i.e.,  $l_R \neq l_C$ . When the claimed and the reference signature are similar, the corresponding dissimilarity vector is expected to be located near the origin. In contrast, the negative samples are expected to have a sparse distribution in space [16].

In this work,  $DT(\cdot)$  is applied to the streaming of claimed signatures against each correspondent reference sample stored in the database of enrolled users. Specifically, consider  $\mathbb{S}_R^i = \{S_R^{i,1}, S_R^{i,2}, \dots, S_R^{i,M}\}$  the set of  $M$  reference signatures of user  $i$ , and  $S_C$  a claimed signature of same user. Then,  $DT(\phi(\mathbb{S}_R^i), \phi(S_C))$  results in the correspondent dissimilarity set  $\hat{\mathbb{X}}_{RC}^i = \{\hat{\mathbf{x}}_{RC}^{i,1}, \hat{\mathbf{x}}_{RC}^{i,2}, \dots, \hat{\mathbf{x}}_{RC}^{i,M}\}$ , which is passed to the adaptive WI-classifier for training and testing procedure.

The dichotomy transformation also helps mitigate the common imbalance ratio issues in streaming data. For each incoming genuine signature, it is always possible to generate the same amount of negative dissimilarities by utilizing the user's stored reference signatures and selecting the necessary number of random forgery samples. This approach consistently produces an equal number of positive and negative examples.

### 2.3 Adaptive WI-Classifier $\theta$

In the proposed SHSV approach, the core component is the adaptive verification process, which enables the system to update its base knowledge over time. In static HSV systems, Support Vector Machines (SVM) are a popular choice for the verification step [7]. Nonetheless, an adaptive classifier is required for the present work. Many methods have been developed to adapt the traditional SVM to handle evolving data [22]. An efficient optimization method is applying Stochastic Gradient Descent (SGD) to linear models to minimize the loss function [10]. To mimic the SVM behavior with adaptive capability, we adopt the SGD classifier with a hinge loss function. We follow similar works [13, 15, 21] that employed SGD to minimize the loss function in the primal formulation directly. This approach is more efficient than employing Lagrangian methods as it avoids the need to compute and store dual variables, which can become computationally expensive and memory-intensive as the number of data points and features increases. The loss function, described in Eq. 1, aims to minimize the norm of the weight vector  $\mathbf{w}$  while penalizing misclassifications (quantified

by the hinge loss term  $\max(0, 1 - y_i(w \cdot \mathbf{x}_i + b))$ , where  $C$  is the regularization parameter that controls the trade-off between maximizing the margin and minimizing the hinge loss.

$$\min_{\mathbf{w}, b} \frac{C}{2} \|\mathbf{w}\|^2 + \frac{1}{N} \sum_{i=1}^N \max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + b)) \quad (1)$$

SGD processes individual samples or small batches from the dataset, iteratively updating the model parameters based on the loss function. In SHSV, the WI-classifier is updated with dissimilarity vectors obtained from a chunk of incoming signatures. This update process occurs after the classifier’s prediction on the input chunk. For the present work, we assume that all true labels are available immediately after the classifier’s estimation.

## 2.4 Fusion Function $g(\cdot)$

In SHSV, the WI-classifier’s output is determined by the dissimilarity vector’s distance to its decision hyperplane. When there is a set of reference signatures  $\mathbb{S}_R$  for a claimed signature  $S_C$ , the system delivers a distance for each pair of dissimilarity vectors between  $\mathbb{S}_R$  and  $S_C$ . These hyper-plane distances are combined through a fusion function  $g(\cdot)$  [14]. Results in [17] reveal that better verification performance is achieved when the Max fusion function is chosen to combine hyper-plane distances output. This work employs the maximum distance to deliver a final decision.

## 3 Experimental Setup

**Datasets.** There are a few publicly available datasets for offline HSV Systems. In this work, we adopt datasets used in related works [17, 20] summarized in Table 2.

**Table 2.** Commonly used datasets for Offline Signature Verification

Ref	Dataset Name	Language	Users	Genuine signatures	Forgeries
[9]	CEDAR	Western	55	24	24
[3]	GPDS Synthetic	Western	10000	24	30
[11]	MCYT-75	Western	75	15	15

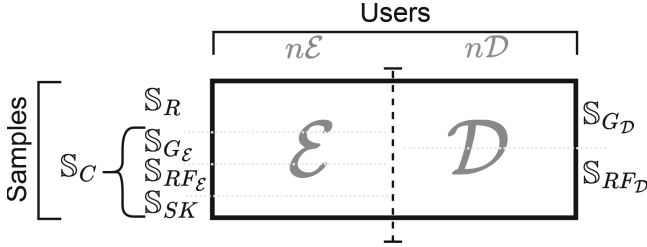
**Preprocessing.** As *SigNet-S* is adopted as the backbone for feature extraction, to ensure the reported performance of the model, we have adhered to the same initial preprocessing steps as described in [6, 20]. First, images are centered on a large canvas, with dimensions determined by the maximum size encountered

within each dataset. Next, the image’s background is removed using Otsu’s algorithm [12], which transforms background pixels to white and foreground pixels to grayscale. Subsequently, the image is inverted to set the background to zero value. Then, all images are resized to the size of 170 pixels in height and 242 pixels in width, and finally, a center crop of size  $150 \times 220$  is taken.

**Classifier.** For classifier comparison in batch context, we employ soft margin SVM with Radial Basis Function (RBF) kernel, following the experimental protocol defined in [6, 17, 20]. The SVM regularization parameter is given by 1.0, and the RBF kernel coefficient hyper-parameter is given by  $2^{-11}$ .

**Types of Signatures.** In this work, there are three types of signatures: genuine (G), which belongs to the claimed user; random forgery (RF), which is a genuine signature that belongs to a user different from the claimed one; and skilled forgery (SK), which belongs to the claimed user but it was produced by a forger. The set of all genuine signatures in a dataset is given by  $\mathbb{S}_G = \{\mathbb{S}_G^1, \mathbb{S}_G^2, \dots, \mathbb{S}_G^N\}$ , where  $N$  is the number of users and  $\mathbb{S}_G^i$  refers to the set of genuine signatures of user  $i$ . Specifically,  $\mathbb{S}_G^i = \{S_G^{i,1}, S_G^{i,2}, \dots, S_G^{i,K}\}$ , where  $K$  denotes the number of user’s signatures. Likewise, there are sets of random forgeries  $\mathbb{S}_{RF}$  and skilled forgery  $\mathbb{S}_{SK}$  signatures for each user.

**Data Segmentation and Generation.** Following [17, 19], datasets are split into two disjoint subsets of users: the development  $\mathcal{D}$ , employed for training, and the exploitation set  $\mathcal{E}$ , employed for testing models as shown in Fig. 2. Considering a dichotomy transformation  $DT(\cdot)$ , a representation model  $\phi(\cdot)$ , the sets are generated as follows:



**Fig. 2.** Data segmentation into development  $\mathcal{D}$  and exploitation  $\mathcal{E}$  sets. To generate  $\mathcal{E}$ , a set of references  $\mathbb{S}_R$  and claimed signatures  $\mathbb{S}_C$  are randomly selected for all  $n\mathcal{E}$  users.  $\mathbb{S}_C$  contains genuine, random forgery, and skilled forgery samples. To generate  $\mathcal{D}$ , a set of genuine  $\mathbb{S}_{G\mathcal{D}}$  and random forgery  $\mathbb{S}_{RF\mathcal{D}}$  are randomly chosen for all  $n\mathcal{D}$  users. Selected samples are utilized to perform dissimilarity transformations as defined in Eqs. 2, 3, 4, and 5.

- **Development set  $\mathcal{D}$ :** For each user  $i$  in  $\mathcal{D}$ ,  $n\mathcal{D}_G$  genuine signatures are randomly selected forming the set  $\mathbb{S}_{G\mathcal{D}}^i$ . The genuine signatures in  $\mathbb{S}_{G\mathcal{D}}^i$  are paired to form dissimilarity vectors of *positive* class as defined in Eq. 2:



**Positive set:**

$$\dot{\mathbb{X}}_{\mathcal{D}^+}^i = \bigcup_{k=1}^{n\mathcal{D}_G-1} \bigcup_{j=k+1}^{n\mathcal{D}_G} DT(\phi(S_G^{i,k}), \phi(S_G^{i,j})) \quad (2)$$

with  $S_G^{i,*} \in \mathbb{S}_{G_{\mathcal{D}}}^i$ .

Additionally, For each user  $i$  in  $\mathcal{D}$ ,  $n\mathcal{D}_G/2$  random forgeries signatures are randomly selected forming the set  $\mathbb{S}_{RF_{\mathcal{D}}}^i$ . Then,  $n\mathcal{D}_G-1$  genuine signatures in  $\mathbb{S}_{G_{\mathcal{D}}}^i$  are paired with all random signatures in  $\mathbb{S}_{RF_{\mathcal{D}}}^i$ , resulting in dissimilarity vectors of *negative* class as defined in Eq. 3:

**Negative set:**

$$\dot{\mathbb{X}}_{\mathcal{D}^-}^i = \bigcup_{k=1}^{n\mathcal{D}_G-1} \bigcup_{j=1}^{n\mathcal{D}_G/2} DT(\phi(S_G^{i,k}), \phi(S_{RF}^{i,j})) \quad (3)$$

with  $S_G^{i,k} \in \mathbb{S}_{G_{\mathcal{D}}}^i$  and  $S_{RF}^{i,j} \in \mathbb{S}_{RF_{\mathcal{D}}}^i$ .

The final set is formed by the union of  $\dot{\mathbb{X}}_{\mathcal{D}^+}^i$  and  $\dot{\mathbb{X}}_{\mathcal{D}^-}^i$  for all users, consisting of an equal number of positive (+) and negative (-) dissimilarity samples from  $n\mathcal{D}$  users.

- **Exploitation set  $\mathcal{E}$ :** For each user  $i$  in  $\mathcal{E}$ ,  $n\mathcal{E}_R$  reference (genuine) signatures are randomly selected, forming the set  $\mathbb{S}_R^i$ . Then,  $n\mathcal{E}_C$  signatures of each type (G, RF, SK) are randomly selected resulting in the claimed set  $\mathbb{S}_C^i$  formed by the union of  $\mathbb{S}_{G_{\mathcal{E}}}^i$ ,  $\mathbb{S}_{RF_{\mathcal{E}}}^i$ , and  $\mathbb{S}_{SK}^i$  sets of signatures. After that, dissimilarities between all samples in  $\mathbb{S}_R^i$  and  $\mathbb{S}_C^i$  are computed, and the result comprises the exploitation set. This process is defined in Eq. 4 and 5:

$$\dot{\mathbb{X}}_{\mathcal{E}}^i = \bigcup_{k=1}^{n\mathcal{E}_R} \bigcup_{j=1}^{n\mathcal{E}_C} \{\dot{\mathbf{x}}_G^{i,k,j}, \dot{\mathbf{x}}_{RF}^{i,k,j}, \dot{\mathbf{x}}_{SK}^{i,k,j}\} \quad (4) \quad \dot{\mathbb{X}}_{\mathcal{E}} = \bigcup_{i=1}^{n\mathcal{E}} \dot{\mathbb{X}}_{\mathcal{E}}^i \quad (5)$$

Where:

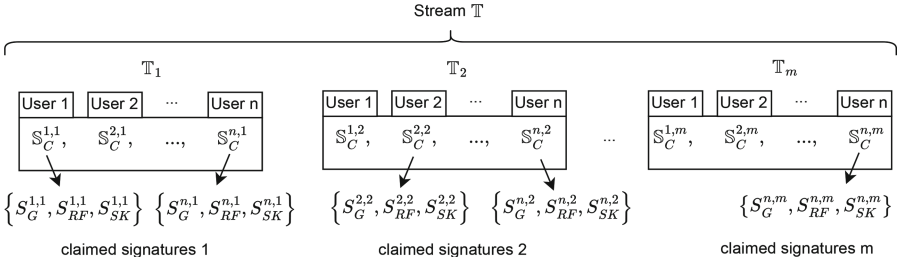
- $\dot{\mathbf{x}}_G^{i,k,j} = DT(\phi(S_R^{i,k}), \phi(S_G^{i,j}))$  *positive* dissimilarity vector (genuine)
- $\dot{\mathbf{x}}_{RF}^{i,k,j} = DT(\phi(S_R^{i,k}), \phi(S_{RF}^{i,j}))$  *negative* dissimilarity vector (random forgery)
- $\dot{\mathbf{x}}_{SK}^{i,k,j} = DT(\phi(S_R^{i,k}), \phi(S_{SK}^{i,j}))$  *negative* dissimilarity vector (skilled forgery)

For the present work, dataset segmentation is shown in Tables 3a and 3b.

**Stream Generation.** In order to provide a comparable evaluation between batch and stream settings, the batch data generation previously described is extended to an equivalent stream configuration. To generate the stream, the exploitation set  $\mathcal{E}$  is converted into a timeline where the whole stream of samples is given by  $\mathbb{T} = \{\mathbb{T}_1, \mathbb{T}_2, \dots, \mathbb{T}_\infty\}$ , where each  $\mathbb{T}_j$  is a set of arriving signatures as shown in Fig. 3.

**Table 3.** Data segmentation for each dataset

(a) Development set ( $\mathcal{D}$ ).					
Data	#Users ( $n\mathcal{D}$ )	#G sig. ( $n\mathcal{D}_G$ )	Neg. class Dissimilarity between:	Pos. class	#Samples $n\mathcal{D} \cdot 66 \cdot 2$
GPDS-S {5, 10, 50, 581 (all)} $\times$ {2, 6, 12} 11G & 6RF 12G of each user					
(b) Exploitation set ( $\mathcal{E}$ ).					
Data	#Users ( $n\mathcal{E}$ )	#Ref. signatures ( $n\mathcal{E}_R$ )	Claimed signatures #( $n\mathcal{E}_C$ )	Set	Stream $\mathbb{T}$ $\mathbb{T}_j$ size $j$ value
GPDS-S	300	{1, 2, 3, 5, 10, 12}	10	10G, 10RF, 10SK	$n\mathcal{E} \cdot 3$ $n\mathcal{E}_C$
CEDAR	55	{10}	10	10G, 10RF, 10SK	
MCYT	75	{10}	5	5G, 5RF, 5SK	



**Fig. 3.** Stream  $\mathbb{T}$  of claimed signatures obtained from the exploitation set  $\mathcal{E}$ . The number of users  $n\mathcal{E}$  is represented by  $n$ , while the number of claimed signatures  $n\mathcal{E}_C$  is denoted by  $m$ .  $S_C^{i,j}$  is the  $j$ -th set of claimed signature of user  $i$ , where  $S_C^{i,j} = \{S_G^{i,j}, S_{RF}^{i,j}, S_{SK}^{i,j}\}$ , with  $S_G^{i,j} \in \mathbb{S}_{G\mathcal{E}^i}^i$ ,  $S_{RF}^{i,j} \in \mathbb{S}_{RF\mathcal{E}^i}^i$  and  $S_{SK}^{i,j} \in \mathbb{S}_{SK}^i$ . G: genuine, RF: random forgery, SK: skilled forgery signature.

From the exploitation set  $\mathcal{E}$ , the set of claimed signatures  $\mathbb{S}_C$  is transformed in a stream comprised of  $n\mathcal{E}_C$  chunks. Each chunk  $\mathbb{T}_j$  contains all users in  $\mathcal{E}$  requesting the verification of three samples: a genuine, a random forgery, and a skilled forgery signature.

After generating stream  $\mathbb{T}$ , it is employed as input to the SHSV system (Fig. 1). First, each signature has its features extracted and passed to the stream dichotomy transformation. For this step, the set of reference signatures  $\mathbb{S}_R$  from the exploitation set  $\mathcal{E}$  is retrieved, then features are extracted, and  $DT(\cdot)$  is applied on each corresponding pair of feature vectors. That is, the stream  $\mathbb{T}$  results in a stream of dissimilarity sets as defined in Eqs. 6 and 7:

$$\hat{\mathbb{T}}_j = \bigcup_{i=1}^{n\mathcal{E}} \{\dot{\mathbb{X}}_G^{i,j}, \dot{\mathbb{X}}_{RF}^{i,j}, \dot{\mathbb{X}}_{SF}^{i,j}\} \quad (6)$$

$$\hat{\mathbb{T}} = \bigcup_{j=1}^{n\mathcal{E}_C} \hat{\mathbb{T}}_j \quad (7)$$

Where:

- $\dot{\mathbb{X}}_G^{i,j} = DT(\phi(\mathbb{S}_R^i), \phi(S_G^{i,j}))$  set of *positive* dissimilarity vectors (genuine)

- $\dot{\mathbb{X}}_{RF}^{i,j} = DT(\phi(\mathbb{S}_R^i), \phi(S_{RF}^{i,j}))$  set of *negative* dissimilarity vectors (random)
- $\dot{\mathbb{X}}_{SF}^{i,j} = DT(\phi(\mathbb{S}_R^i), \phi(S_{SF}^{i,j}))$  set of *negative* dissimilarity vectors (skilled)

Stream  $\dot{\mathbb{T}}$  is then sent to the WI-classifier for testing, and a decision using the fusion function  $\mathbf{g}(\cdot)$  is performed for each signature request. For the present work, stream configuration regarding datasets is shown in Table 3b.

**Model Initialization.** Models are initialized using all  $\mathcal{D}$  sets in Table 3a.

**Model Update and Evaluation.** In this work, we employ the prequential evaluation approach (also known as test-then-train), the most common method for evaluating data streams [8]. This approach involves continuously testing a predictive model with new arriving samples and then using those same samples to update the model. We assume all instance labels are available after testing and do not employ skilled forgeries for classifier updating.

Given a stream of arriving signatures,  $\mathbb{T}$ , we define three hyperparameters for model update and evaluation:

- **Chunk size** ( $c_{size}$ ): The number of signatures the system waits for before updating the model.
- **Window size** ( $w_{size}$ ): The number of most recently tested signatures used to compute performance metrics.
- **Window step** ( $w_{step}$ ): The frequency (number of new signatures) at which the metrics are assessed.

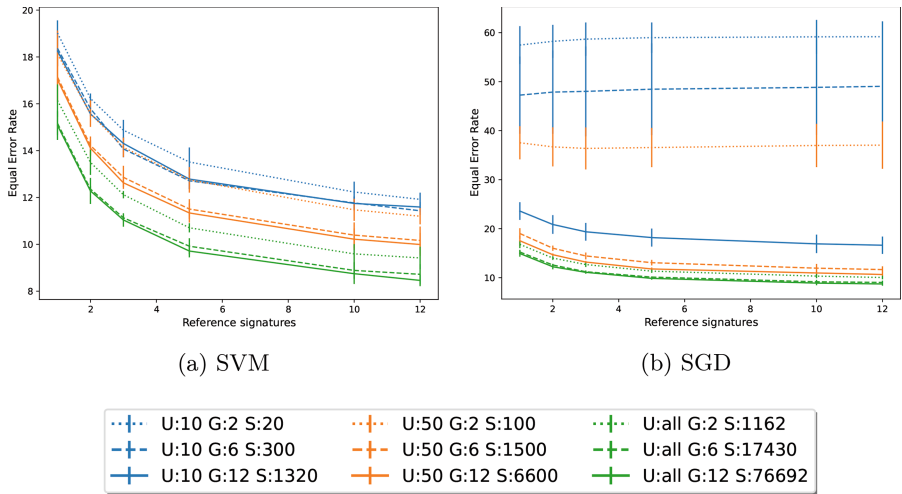
The stream evaluation employed in this paper is summarized in Table 4. While the entire block  $\mathbb{T}_k$  is tested, only the genuine and random forgery signatures are sent to update the classifier. A window smaller than the chunk size is employed to observe the evolution more frequently.

With the test results, the Equal Error Rate (EER) using a global threshold is measured at every window. The experiments are repeated five times, and the average results and standard deviation are computed.

**Table 4.** Stream evaluation.

Stream	Chunk size		Window size and step
	Test	Training	
GPDS-S	900 ( $n\mathcal{E} \cdot 3$ )	600 ( $n\mathcal{E} \cdot 2$ )	400
CEDAR + MCYT	300	200	200
Type of sig.:	G, RF, and SK	G and RF	G and SK

## 4 Experimental Results

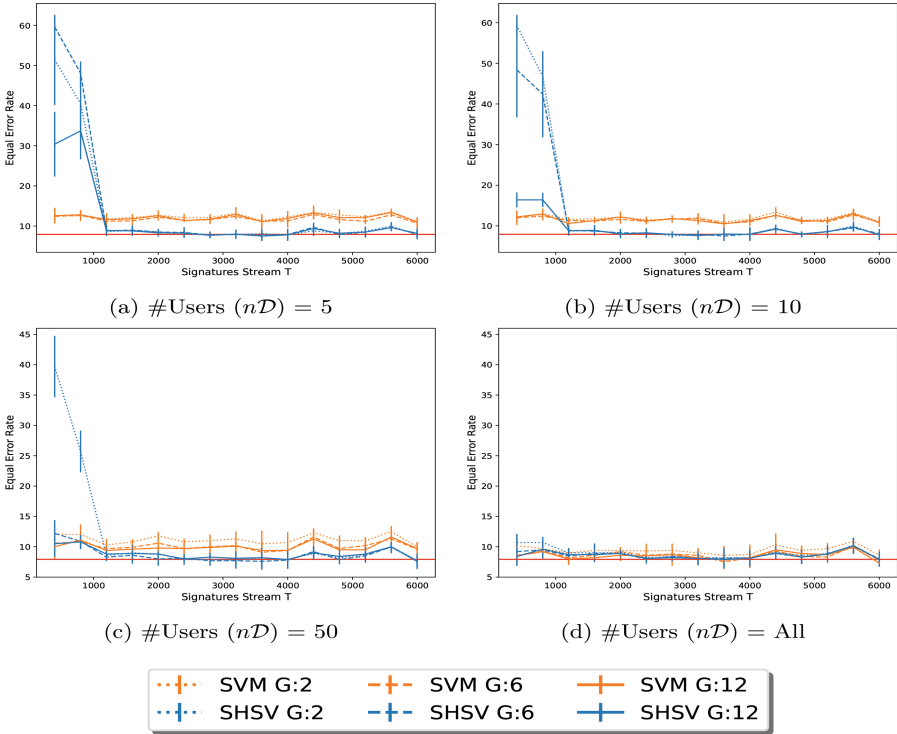


**Fig. 4.** Skilled forgery detection in batch settings for different development sets.  $\#U$  and  $\#G$  denote, respectively, the number of users  $n\mathcal{D}$  and genuine signatures  $n\mathcal{D}_G$  employed during training.  $\#S$  denotes the resulting number of samples.

**Batch Evaluation.** Figure 4 presents results for batch-trained SVM and SGD considering different number of users  $n\mathcal{D}$  ( $\#U$ ) and genuine signatures  $n\mathcal{D}_G$  ( $\#G$ ) (Table 3a). Interestingly, increasing the number of training samples does not necessarily guarantee improved performance. For example, the configuration with  $U = 50$  users and  $G = 2$  genuine signatures per user (orange dotted line) during training, resulting in 100 samples ( $S:100$ ), achieves better results than the configuration with  $U = 10$  users and  $G = 12$  signatures (blue solid line,  $S:1320$ ), despite having fewer signatures overall. This suggests that the number of users available during the development phase plays a crucial role in achieving system generalization. Conversely, when the number of users remains constant, increasing the number of samples per user during training leads to improved performance.

Furthermore, SGD exhibits greater sensitivity to limited initial data than SVM, although both achieve comparable performance when trained with all users and samples. Overall, the results indicate a trend of decreasing error rates with an increase in the number of reference signatures. Please refer to Table 1 in the supplementary material for a comprehensive set of results.

**Stream Evaluation.** Figure 5 presents the performance comparison between the SVM and the proposed SHSV method when the exploitation set is transformed into a continuous stream of incoming signatures. At a certain point,



**Fig. 5.** Stream evaluation of skilled forgery detection on GPDS Synthetic using SVM and the Stream HSV (SHSV). The evaluation considers 12 reference signatures with a Max fusion for decision-making. SHSV is updated after every training chunk, employing only genuine and random forgery signatures, and evaluated on every window (Table 4).  $\#Users (nD)$  and  $\#G$  denote the number of users and genuine signatures used in the initial training, respectively. The horizontal red line shows the result (7.93) reported in [20] for 12 reference signatures,  $\#G = 12$ , and  $nD = 2000$ .

SHSV surpasses SVM for all initial training configurations, being more pronounced when there is a limited number of users and signatures for pre-training models. Recently, [20] reported an EER of  $7.93 \pm 0.30$  for a writer-independent approach using global thresholds on GPDS Synthetic with 12 reference signatures. In their study, the authors utilized *SigNet-S* as the feature extractor, selected 2000 users for training an SVM classifier, and conducted tests on the GPDS-S-300 dataset. In contrast, SHSV achieves comparable results while requiring significantly fewer users for initial training. This finding highlights the effectiveness of SHSV in real-world scenarios with restricted sample availability. Moreover, SHSV accommodates the dynamic nature of handwriting signatures by enabling a continuous adaptation of the system, leading to improved performance over time.

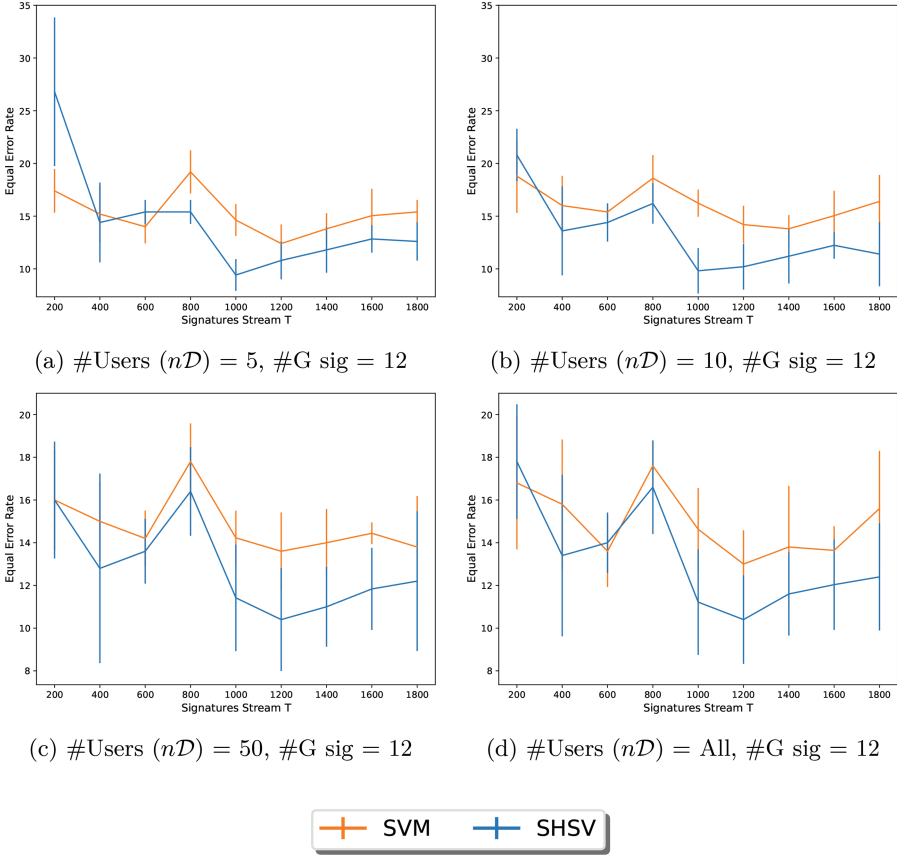
**Table 5.** EER on the last chunk of signatures stream. Results show skilled forgery detection on GPDS Synthetic using SVM and Stream HSV (SHSV).  $n\mathcal{D}$  and  $n\mathcal{D}_G$  refer to the initial training setup (Table 3a).  $n\mathcal{E}_R$  denotes the number of reference signatures (Table 3b). Stream evaluation performed according to Table 4.

$n\mathcal{D}$	$n\mathcal{E}_R$	#G sig. ( $n\mathcal{D}_G$ ) = 2			#G sig. ( $n\mathcal{D}_G$ ) = 12		
		$SVM_{last}$	$SHSV_{last}$	$\Delta$	$SVM_{last}$	$SHSV_{last}$	$\Delta$
2	1	19.20 (2.73)	14.00 (2.09)	5.20	19.00 (2.26)	15.00 (1.41)	4.00
	2	17.40 (2.22)	12.00 (2.40)	5.40	17.00 (1.12)	12.90 (1.98)	4.10
	5	14.20 (1.48)	9.20 (1.20)	5.00	14.10 (1.08)	9.00 (1.27)	5.10
	12	12.10 (0.55)	7.70 (1.15)	4.40	11.30 (0.57)	7.80 (1.10)	3.50
10	1	18.50 (2.12)	13.70 (2.05)	4.80	17.70 (2.17)	13.80 (2.08)	3.90
	2	16.00 (1.41)	12.20 (2.61)	3.80	15.20 (2.05)	12.10 (2.88)	3.10
	5	13.00 (1.37)	9.20 (1.35)	3.80	11.90 (1.08)	9.10 (1.19)	2.80
	12	11.10 (1.29)	7.70 (1.15)	3.40	10.80 (1.04)	7.80 (1.30)	3.00
50	1	17.90 (1.67)	14.00 (2.26)	3.90	16.60 (1.78)	14.20 (2.05)	2.40
	2	14.60 (1.14)	12.20 (1.89)	2.40	14.90 (1.39)	12.10 (1.78)	2.80
	5	12.30 (1.15)	9.20 (1.15)	3.10	11.30 (1.04)	9.30 (1.30)	2.00
	12	10.00 (0.71)	7.60 (0.96)	2.40	9.70 (0.76)	7.70 (0.84)	2.00
All	1	16.00 (1.77)	14.50 (2.21)	1.50	14.40 (2.19)	13.90 (2.43)	0.50
	2	13.30 (1.57)	11.90 (1.98)	1.40	12.70 (1.25)	11.80 (1.92)	0.90
	5	10.70 (1.52)	9.30 (1.30)	1.40	9.00 (1.62)	9.10 (1.52)	0.10
	12	8.60 (0.96)	7.80 (0.84)	0.80	7.30 (0.57)	7.90 (1.19)	0.60

Table 5 presents the performance of SVM and SHSV on the final chunk of the signature stream, evaluated across different numbers of reference signatures used for the fusion function. Consistent with the batch setting results, performance improves with an increasing number of stored signatures per user. Findings also highlight the discrepancy between SVM and SHSV performance, which becomes more pronounced as the number of users in the training phase decreases. Unlike SVM, SHSV consistently exhibits improved performance over time, achieving better or comparable results regardless of the initial development configuration.

SHSV is particularly interesting for handling signatures from unknown distributions due to its ability to learn over time. Figure 6 shows the performance of models pre-trained on GPDS Synthetic data when they receive signatures coming randomly from CEDAR stream and MCYT stream. While initially affected by the change, SHSV outperforms SVM, especially when few users are available at the beginning. Please see Table 2 in the supplementary material for a detailed set of results.

In summary, the proposed SHSV system consistently demonstrates superior performance and adaptability compared to the traditional SVM approach in various scenarios. Its resilience to limited initial training data and its continuous adaptation capabilities make SHSV particularly well-suited for real-world



**Fig. 6.** Stream evaluation of skilled forgery detection on signatures randomly coming from the CEDAR and MCYT streams using SVM and the Stream HSV (SHSV). The evaluation considers 10 reference signatures with a Max fusion for decision-making. SHSV is updated after every training chunk and evaluated on every window (Table 4). #Users ( $n\mathcal{D}$ ) and #G denote the number of users and genuine signatures used in the initial training.

handwriting signature verification tasks where data availability and signature variability pose significant challenges.

## 5 Conclusion

This work proposes a novel handwriting signature verification approach called SHSV. SHSV treats signatures as continuous data streams and updates the system dynamically. To achieve this, we introduce a stream generation approach compatible with standard batch evaluation settings.

Experimental results in batch settings demonstrated that having a high number of users is more crucial than the sheer volume of signatures, indicating an

overall improvement in performance when more users are available at initial training. Results also showed that SHSV overcame the problem of limited training data by incorporating new information over time, demonstrating superior performance compared to the SVM approach across different scenarios under stream configuration.

Future work may include using partially labeled data to explore scenarios where labels are not available for all test samples, as well as investigating the trade-off between adapting the representation model and the WI-classifier over time.

## References

1. Diaz, M., Ferrer, M.A., Impedovo, D., Malik, M.I., Pirlo, G., Plamondon, R.: A perspective analysis of handwritten signature technology. *ACM Comput. Surv.* **51**(6) (2019)
2. Eskander, G., Granger, E.: Dissimilarity representation for handwritten signature verification, vol. 1022 (2013)
3. Ferrer, M.A., Diaz, M., Carmona-Duarte, C., Morales, A.: A behavioral handwriting model for static and dynamic signature synthesis. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1041–1053 (2017)
4. Ferrer, M.A., Diaz-Cabrera, M., Morales, A.: Static signature synthesis: a neuro-motor inspired approach for biometrics. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(3), 667–680 (2015)
5. Hafemann, L.G., Sabourin, R., Oliveira, L.S.: Offline handwritten signature verification - literature review, pp. 1–8. *IEEE* (2017)
6. Hafemann, L.G., Sabourin, R., Soares de Oliveira, L.: Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recogn.* **70** (2017)
7. Hameed, M., Ahmad, R., Mat Kiah, M.L., Murtaza, G.: Machine learning-based offline signature verification systems: a systematic review. *Sig. Process. Image Commun.* **93**, 116139 (2021)
8. Haug, J., Tramontani, E., Kasneci, G.: Standardized evaluation of machine learning methods for evolving data streams (2022)
9. Kalera, M.K., Srihari, S., Xu, A.: Offline signature verification and identification using distance statistics. *Int. J. Pattern Recognit Artif Intell.* **18**(07), 1339–1360 (2004)
10. Losing, V., Hammer, B., Wersing, H.: Incremental on-line learning: a review and comparison of state of the art algorithms. *Neurocomputing* **275**, 1261–1274 (2018)
11. Ortega-Garcia, J., et al.: MCYT baseline corpus: a bimodal biometric database. *IEE proc vis image signal process spec issue biom internet*. In: *IEE Proceedings - Vision Image and Signal Processing*, pp. 395–401 (2003)
12. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979)
13. Panagiotakopoulos, C., Tsampouka, P.: The stochastic gradient descent for the primal L1-SVM optimization revisited. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) *ECML PKDD 2013. LNCS (LNAI)*, vol. 8190, pp. 65–80. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-40994-3\\_5](https://doi.org/10.1007/978-3-642-40994-3_5)



14. Rivard, D., Granger, E., Sabourin, R.: Multi-feature extraction and selection in writer-independent off-line signature verification. *Int. J. Doc. Anal. Recogn. (IJ DAR)* **16**(1), 83–103 (2013)
15. Shalev-Shwartz, S., Singer, Y., Srebro, N., Cotter, A.: Pegasos: primal estimated sub-gradient solver for SVM. *Math. Program.* **127**(1), 3–30 (2011)
16. Souza, V.L.F., Oliveira, A.L.I., Cruz, R.M.O., Sabourin, R.: Characterization of handwritten signature images in dissimilarity representation space. In: Rodrigues, J.M.F., et al. (eds.) *ICCS 2019. LNCS*, vol. 11538, pp. 192–206. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-22744-9\\_15](https://doi.org/10.1007/978-3-030-22744-9_15)
17. Souza, V.L.F., Oliveira, A.L.I., Sabourin, R.: A writer-independent approach for offline signature verification using deep convolutional neural networks features. In: 2018 7th Brazilian Conference on Intelligent Systems (BRACIS), pp. 212–217 (2018)
18. Vargas, F., Ferrer, M., Travieso, C., Alonso, J.: Off-line handwritten signature GPDS-960 corpus. In: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), vol. 2, pp. 764–768 (2007)
19. Viana, T.B., Souza, V.L., Oliveira, A.L., Cruz, R.M., Sabourin, R.: Contrastive learning of handwritten signature representations for writer-independent verification. In: 2022 International Joint Conference on Neural Networks (IJCNN), pp. 01–09 (2022)
20. Viana, T.B., Souza, V.L., Oliveira, A.L., Cruz, R.M., Sabourin, R.: A multi-task approach for contrastive learning of handwritten signature feature representations. *Expert Syst. Appl.* **217**, 119589 (2023)
21. Zhai, T., Gao, Y., Wang, H., Cao, L.: Classification of high-dimensional evolving data streams via a resource-efficient online ensemble. *Data Min. Knowl. Disc.* **31**(5), 1242–1265 (2017)
22. Zhou, X., Zhang, X., Wang, B.: Online support vector machine: a survey. In: Kim, J.H., Geem, Z.W. (eds.) *Harmony Search Algorithm. AISC*, vol. 382, pp. 269–278. Springer, Heidelberg (2016). [https://doi.org/10.1007/978-3-662-47926-1\\_26](https://doi.org/10.1007/978-3-662-47926-1_26)



# OCR4HSV: A Multi-task Learning Approach for Handwritten Signature Verification

Chao-Qun Lin<sup>1,2</sup>, Da-Han Wang<sup>1,2(✉)</sup>, Yan-Fei Su<sup>1,2</sup>, De-Wu Ge<sup>3</sup>,  
and Xu-Yao Zhang<sup>4</sup>

<sup>1</sup> School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China

[lincq@stu.xmut.edu.cn](mailto:lincq@stu.xmut.edu.cn), [{wangdh,suyanfei}@xmut.edu.cn](mailto:{wangdh,suyanfei}@xmut.edu.cn)

<sup>2</sup> Fujian Key Laboratory of Pattern Recognition and Image Understanding, Xiamen 361024, China

<sup>3</sup> Xiamen KEYTOP Communication Technology Co., Xiamen 361024, China

<sup>4</sup> State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation of Chinese Academy of Sciences, Beijing 100190, China  
[xyz@nlpr.ia.ac.cn](mailto:xyz@nlpr.ia.ac.cn)

**Abstract.** Handwritten signature verification (HSV) models are notably recognized for their ability to discern whether a signature is forged in an offline document. Recently, HSV technology has made significant development. However, these methods have primarily focused on the shape features of the text image and overlooked the textual feature information inherent in the text itself, which makes the HSV model overfit. In this paper, we propose a novel network model named OCR4HSV to solve the above shortcomings. The proposed OCR4HSV first attempts to combine OCR and HSV to learn textual features in a multi-task learning manner. The model employs a dual-parameter approach, combining shared parameters and independent parameters. Specifically, Within the shared parameters, the Laplace attention module (LAM) is incorporated for edge information extraction. For independent parameters, CN-Mamba is utilized for sequence feature extraction in OCR, and the multi-scale global fusion block (MGFB) is designed to enhance the distinction between reference and test sample pairs. Leveraging OCR-related information and these architectural enhancements can fully mine the inherent textual feature information and significantly mitigate overfitting in the HSV task, thereby boosting verification accuracy. Our model has achieved state-of-the-art performance on the ChiSig and HanSig datasets.

**Keywords:** Handwritten signature verification · Optical character recognition · Multi-task learning · CN-Mamba

# 1 Introduction

In contemporary society, Handwritten signature verification (HSV) is a crucial forensic tool widely employed across various domains such as law, insurance, and culture. This technology aims to compare a given signature image with a reference signature to determine if the same individual authors them. Hence, the innovation of an exact and efficient HSV framework is of utmost significance.

How to extract robust and discriminative signature features is the essential problem of HSV. With the development of deep learning, more and more studies [2, 11, 15, 17, 28], extract signature features using deep neural networks. Compared to the traditional hand-crafted features [22], these methods significantly improve the performance of the signature verification system. All these methods target the basic features of offline images, such as the text edge information, the stroke information of the text, and the structure information of the overall signature. However, these models are prone to overfitting during the training. To solve the overfitting problem, [28] proposed an inverse discriminant network to extract effective information from the sparse stroke pixel part; [15] proposed a 2-channel-2-logit network that outputs two logits and uses the distance between the two outputs as the similarity of the two input images. This information is important for HSV tasks, and many achievements have been made. However, these methods ignore the fact that the text itself also has text feature information. As a result, the performance of these methods is often unsatisfactory.

Optical character recognition (OCR) is commonly employed to extract textual features from images. OCR is the process of converting text in an image into a digital text sequence. In response to the question of whether the length of a character is determined, researchers have done an amount of work [18, 27]. Currently, there are numerous technologies based on OCR and new technologies built upon OCR, such as document analysis [33], table recognition [34], and key information extraction [30]. Both the input for HSV and OCR consist of images containing text; hence, employing OCR technology to extract text features for use in HSV is a viable approach.

The popularity of multi-task learning (MTL) extends across computer vision [29] and natural language processing [32]. Motivated by the principles of MTL, we simultaneously train the OCR and HSV tasks to learn abundant textual feature information. This dual training approach, enhanced by cross-task knowledge transfer, enhances generalization and mitigates overfitting while maintaining the distinctiveness of each task. Training a unified model for multiple tasks demonstrates greater parameter efficiency than individual task modeling. In this paper, we propose a new offline handwritten signature verification network called OCR4HSV based on MTL. The OCR4HSV model does not distinguish between OCR and HSV and trains them simultaneously. Leveraging OCR, we extract textual information from signature images. Through shared parameters and independent parameters, we optimize the parameters of the HSV task, reducing overfitting and boosting accuracy.

In this model, OCR and HSV shared parameters CNN-Blocks give these blocks the ability to extract signature verification discriminative information

when processing OCR tasks. To enable the model to capture text edges and stroke information when extracting text information, in the CNN-Blocks, we design the Laplace attention module (LAM), allowing the model to focus more on stroke information during training. After CNN-Blocks, OCR and HSV used independent parameters, we introduced a multi-scale global fusion block (MGFB) module to discriminate between reference and test signature pairs. Regarding OCR, Mamba [8] has been proven to be highly efficient for sequence processing, and we employ the Mamba module for extracting sequential features. To the best of our knowledge, this work pioneers the use of the MTL approach for HSV. The reliability of our proposed model has been validated through experiments conducted on the ChiSig [31] and HanSig [12] databases.

The main contributions of this paper are as follows:

- 1) We proposed the OCR4HSV model, which is based on MTL. It provides a new benchmark for multi-task-based signature verification methods. In the OCR4HSV model, we proposed CN-Mamba architecture for OCR tasks.
- 2) We proposed the Laplace attention module in CNN-Block of OCR4HSV. This module uses a fixed Laplace operator convolution kernel to extract stroke edge information.
- 3) We proposed the multi-scale global fusion block module in HSV, which enables the model to extract global feature information at different scales of the fused reference-test sample pairs.
- 4) The OCR4HSV model achieved impressive results on different styles of Chinese datasets ChiSig and HanSig.

## 2 Related Work

### 2.1 Handwritten Signature Verification

HSV has been studied for many years [10]. In recent years, deep learning methods [2, 28] have gradually surpassed traditional two-stage methods due to the advantages of end-to-end architecture and the ability to extract powerful features. [11] uses convolutional neural networks to extract features for HSV tasks. [2] proposes a network to metric distances between signature pairs. [28] designed a four-stream network and a multi-path attention mechanism to validate signatures in a binary classification paradigm. [16] introduces a static-to-dynamic interaction method for offline signature verification tasks. [23] proposes a region-based metric learning method for solving writer-independent and writer-dependent signature verification tasks. [15] uses a dual-channel fusion, dual-logit output supervised learning approach for HSV. [17] proposes a signature verification method using Transformer. [24] proposes a HSV framework based on dual channels and dual Transformer.

Although deep learning methods have shown significant results, they focus on the shape of text features while overlooking the textual features inherent in the text itself. Therefore, we propose the OCR4HSV network, which uses the text information features of signatures for HSV.

## 2.2 Optical Character Recognition

Embedded in the legacy of telegraphy and enriched by innovations for the visually impaired [1], OCR technology has matured from primitive character-reading devices [3] to a sophisticated tool for detailed text analysis. Early approaches framed text recognition as detection and classification [14], labeling English words akin to image categories. Yet, these methods struggled with variable-length sequences. Sequence-based techniques, leveraging connectionist temporal classification (CTC) [6] and attention [5], emerged as versatile solutions for sequence labelling, aligning image blocks with character sequences.

Recent trends favor sequence-based frameworks for their adaptability and labeling ease. These architectures feature a dual-module design: a feature encoder for visual text representation and a sequence decoder for character sequence mapping, optionally aided by linguistic context. [27] exemplify this integration, utilizing a CNN-RNN-CTC architecture for end-to-end OCR training. The CNN extracts feature sequences, RNN (bi-LSTM) predicts based on these sequences, and CTC outputs character sequences, collectively forming the CRNN network, trainable under a unified loss function.

Recently, state space sequence models (SSMs) [7], particularly structured SSMs such as S4, have emerged as efficient building blocks (or layers) for constructing deep networks, achieving state-of-the-art performance in continuous long sequence data analysis [9]. Mamba [8] further enhances S4 by incorporating a selection mechanism, enabling the model to selectively attend to input-dependent relevant information. Coupled with hardware-aware implementations, Mamba surpasses Transformers on dense modalities like language and genomics. Given that image patches and features can be transformed into sequences [4], the appealing traits of SSMs motivate us to explore the potential of using Mamba modules for OCR. In this paper, we propose the CN-Mamba architecture, which employs two convolutional layers for feature encoding followed by a Mamba block and utilizes CTC for decoding.

## 2.3 Multi-task Learning

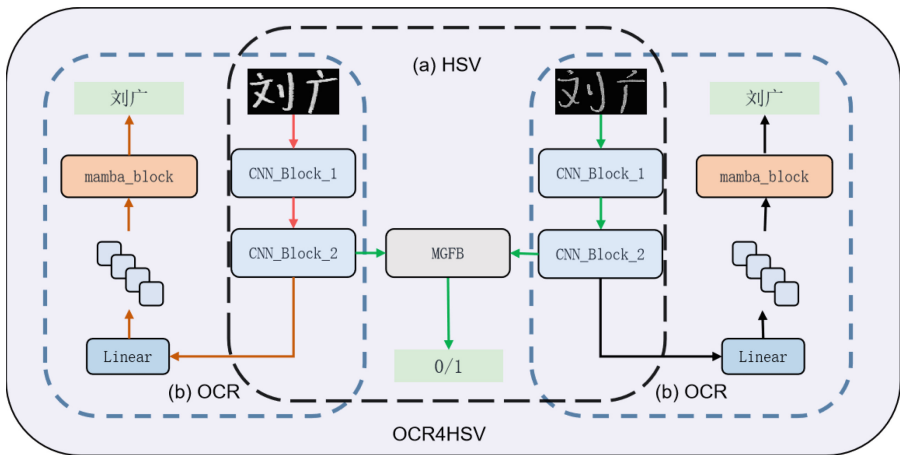
MTL aims to concurrently learn multiple tasks by sharing knowledge and computation. In the realm of computer vision, there exist two classic paradigms of multitasking. The first category pertains to dense scene understanding (DSU) multitasking, which encompasses tasks such as semantic segmentation, surface normal estimation, and saliency detection for each input sample. Presently, research in DSU multitasking predominantly centers around the innovation of decoder architectures [19]. The second paradigm involves cross-domain classification multitasking, where input data comprises multiple datasets with domain shifts. Owing to the involvement of multiple domains, current studies emphasize learning shared and private information across domains [26]. An efficacious multitask network should balance both the shared feature aspects and task-specific components, necessitating the learning of generalized representations

across tasks to prevent overfitting while also capturing the unique characteristics of each task to avert underfitting.

Based on the extent of network parameter sharing when addressing different tasks, MTL methodologies can be categorized architecturally into 1) Shared Parameter, where the main body of the model shares parameters while output structures are task-independent; and 2) Independent Parameter, where distinct tasks employ independent models with parameters constrained with one another. Inspired by the second task, we propose an end-to-end HSV model that, for the first time, combines OCR and HSV tasks into a joint learning framework.

### 3 Method

In this section, we first summarise our methods and detail the proposed model. Then, we delve into the various details of the proposed respectively.

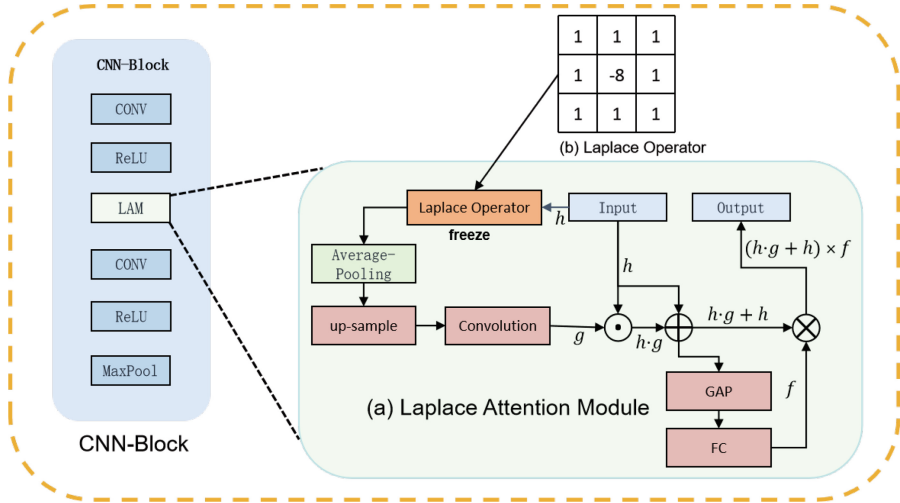


**Fig. 1.** The architecture of the OCR4HSV. (a) The HSV component of OCR4HSV serves as the primary architecture during testing. (b) Employing the CN-Mamba framework, the OCR unit manifests a bespoke design for precise recognition tasks.

#### 3.1 Overview

The OCR4HSV model is illustrated in Fig. 1. OCR4HSV is an end-to-end model and this network model adopts a Siamese network as the basic architecture. The OCR model, depicted in the red and brown data flow in Fig. 1(b), shares weights due to its Siamese characteristic. Therefore, the OCR4HSV model has to be used twice OCR train in every single train. The HSV model, shown in Fig. 1(a), extracts features from reference-test image pairs inputted into the network, performs channel fusion and finally discerns using the MGF module.

The input for both OCR and HSV tasks is the same. Hence, we adopt the concept of MTL and design neural networks with shared and independent parameters. Compared to standard single-task learning on HSV, the joint MTL of OCR and HSV incorporates feature information required for the HSV task during the OCR task, thereby enhancing the robust capability of the HSV task. This approach mitigates the overfitting issue arising from insufficient features needed for the HSV task, consequently improving the accuracy of the HSV task.



**Fig. 2.** The structure of CNN-Block. (a) The intricate structure constituting the Laplace Attention Module. (b) Utilizing the Laplace operator with frozen parameters throughout the training phase.

### 3.2 Shared Parameters

In the OCR4HSV model, CNN-Block is designed by shared parameters. The CNN-Block module is divided into two parts: CNN-Block1 and CNN-Block2. As the convolutional modules for both OCR and HSV modes, these two parts share the same structure and different parameters. Each CNN-Block contains two convolutional layers (the kernel size is  $3 \times 3$  and the strip is 1) activated by the ReLU function and one max-pooling layer (the kernel size is  $2 \times 2$  and the strip is 2), CNN-Block1 comprises 64 channels, and CNN-Block2 is 128 channels. As depicted in Fig. 2. In the OCR4HSV model, the CNN-Block serves as a shared convolutional module for OCR and HSV, responsible for extracting detailed features from offline signature images, encompassing all the features required by OCR and HSV. In CNN-Block, after the first ReLU function, we design a Laplace attention module to extract stroke edge information.

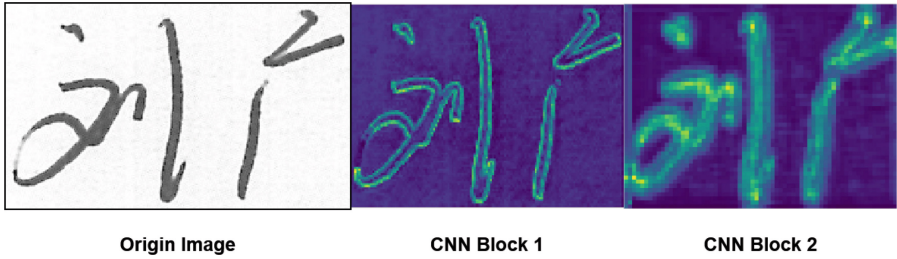


Fig. 3. Feature of Laplace attention module

**Laplace Attention Module.** The Laplace operator, as one of the traditional edge detection operators in image processing, belongs to the commonly used integral transforms in engineering mathematics, the same as the Sobel operator [25]. It is a spatial sharpening filtering operation. The Laplace operator is the simplest isotropic second-order differential operator, possessing rotational invariance. According to the properties of function differentiation, where the second-order differential of the pixel value is zero, are considered edge points. For a two-dimensional image function  $f(x, y)$ , the expression for the second-order Laplace operator is:

$$\begin{aligned} \nabla^2 f(x, y) &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\ &= f(x, y + 1) + f(x, y - 1) + f(x + 1, y) \\ &\quad + f(x - 1, y) - 4f(x, y) \end{aligned} \tag{1}$$

According to this formula, we can get this filter mask:

$$G_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{2}$$

Expanding formula Eq. (2) yields the following filter mask, as shown in Fig. 2(b):

$$G_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{3}$$

Since Laplace operator can achieve edge detection in traditional image processing, it can highlight the edge information of the text. Therefore, this paper adopts the Laplace operator as the foundation and designs the Laplace attention module. Specifically shown in Fig. 2(a). In the CNN-Block, assuming the input feature for the attention module is  $h$ , edge features are extracted using the Laplace operator, and blur processing uses average pooling. Then, upsampling is performed using the nearest neighbor algorithm, and after convolution operations with Sigmoid activation, the output is  $g$ . The process multiplies  $h$  by the



elements of  $g$  and then adds  $h$  to produce the intermediate attention measurement  $h \cdot g + h$ , where “ $\cdot$ ” denotes dot product. The subsequent global average pooling layer and the fully connected layer with Sigmoid activation receive the intermediate attention measurement and output the feature vector  $f$ . Multiplying each intermediate of the channel attention measurement by each element of  $f$  generates the final attention  $(h \cdot g + h) \times f$ , which is then output to the convolution module for feature extraction. To preserve the edge feature extraction effect of the Laplace operator, the parameters of the Laplace operator are frozen.

Figure 3 shows the feature maps of the Laplace attention module, illustrating that after the first layer of attention, the model focuses on the edge of text information. After the second layer of attention, the focus shifts to the text itself, likely due to the edges of the image becoming increasingly blurred from prior processing. However, the input image has a pure black background, which focuses on the text of the model.

### 3.3 Independent Parameters

After the CNN-Block, the OCR4HSV module adopts a design with independent parameters, connecting different blocks for the OCR task and the HSV task. For the HSV module, we propose the MGF module, which draws on the attention mechanism of pyramid networks that captures coarse-grained and fine-grained information along with global context at various scales in a top-down manner [13]. The design of this module, through a progressive convolution strategy of global attention mechanisms across different scales, facilitates focusing on the features that distinguish between the two images at varying scales after fusion, effectively enhancing discriminative capabilities. For the OCR module, this study is the first to apply the recent advancements in SSM, namely Mamba, to OCR. Mamba and its underlying SSM have been demonstrated to bring substantial performance enhancements to dense models in areas like language and genomics due to their selectivity. After passing through the CNN-Block, when the image is transformed into a 1-dimensional sequence, Mamba can extract textual information embedded throughout the entire long sequence. The CN-Mamba architecture, by integrating CNN with Mamba, introduces the capability of capturing local information to Mamba, which is advantageous for highlighting feature information crucial during decoding, thereby making the interpretation of intrinsic data information particularly beneficial.

**Multi-scale Global Fusion Block.** The problem of accurately and quickly distinguishing the differences between two handwriting images has consistently been a focal point of academic research. These images share a lot of similar information, whereas the distinguishing features are scarce and abstract. Excessive convolutional processing tends to obscure the very details crucial for differentiation. To address this, we propose the MGF module. This module employs multi-scale feature extraction to gather characteristics across various scales and utilizes global-local encoding to accentuate the fused features, thereby enabling

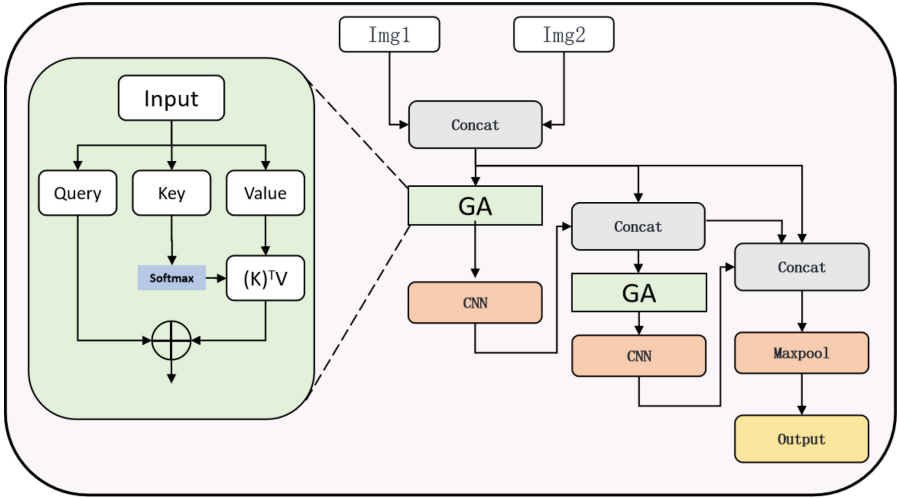


Fig. 4. The structure of multi-scale global fusion block

precise discrimination following the integration of reference and test handwriting images.

The MGFB comprises a combination of two convolutional layers and one pooling layer, employing  $3 \times 3$  convolutions and max-pooling. First, two images,  $Img_1$  and  $Img_2$ , are concatenated along the channel dimension as input. The output is then split into three pathways. In the first path, the fused feature vector is fed into a global attention (GA) mechanism to capture global feature information, followed by a convolutional layer for local feature extraction. Subsequently, the output from the CNN is recombined with the second path and again processed through both GA and CNN modules. Lastly, the outputs from the two CNN operations are combined with the third path, undergoing max-pooling before the final output. The process is shown in Fig. 4 and expressed as:

$$\begin{aligned}
 Fusion_1 &= CAT([Img_1, Img_2]) \\
 Output_{CNN1} &= CNN(GA(Fusion_1)) \\
 Fusion_2 &= CAT([Output_{CNN1}, Fusion_1]) \\
 Output_{CNN2} &= CNN(GA(Fusion_2)) \\
 Output &= MaxPool(CAT([Output_{CNN2}, Fusion_2, Fusion_1]))
 \end{aligned} \tag{4}$$

where,  $Img_1$  and  $Img_2$  are means input features,  $CAT(\cdot)$  is the concatenation,  $Maxpool(\cdot)$  is the maxpool layer,  $CNN(\cdot)$  is the  $3 \times 3$  convolution and  $GA(\cdot)$  represents the Global feature extraction module, represented by Eq 5:

$$G_i(Q, K, V) = \frac{Q}{\sqrt{d}}(SoftMax(K)^T V) \tag{5}$$

where,  $Q, K, V \in R^{N \times C}$  is linearly projected of self attention,  $N = H \times W$ ,  $C$  denoted as Channel dimension,  $d$  is bias constant.

### 3.4 Loss Function

The primary objective of OCR4HSV is to calculate the accuracy of the signature sample. For the HSV method, the loss function used is focal loss function [21]. Focal Loss is a loss function used to solve the problem of class imbalance, which is formulated as follows:

$$Loss(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (6)$$

where  $p_t$  represents the model's predicted probability of the sample. For the HSV task, it is the binary classification problem, give a sample of reference-test pair, if the sample belongs to the positive class, then  $p_t = p$  (the probability of being predicted as the positive class); conversely, if the sample belongs to the negative class, then  $p_t = 1 - p$ .  $\alpha_t$  is the class balance weight, which is used to manually adjust the importance of positive and negative samples.  $\gamma$  is a focus parameter. The larger its value, the smaller the penalty for easily classified samples, and the model will focus more on misclassified or difficult-to-classified samples.

For the OCR loss function, we use CTC [6] for decoding. The CTC decoder was initially used for speech recognition, and researchers have applied it to OCR tasks with great success. CTC is designed to tackle sequence-to-sequence learning problems where the length of the input sequence may not match the length of the target sequence. It allows the model to learn a probability distribution that can map input sequences to output sequences without prior alignment. Given an input sequence  $X$ , it computes the conditional probability  $P(Y|X)$  of the output sequence  $Y$ . The CTC loss function is defined as follows:

$$Loss = -\log P(Y|X) \quad (7)$$

where the calculation of  $P(Y|X)$  is performed by summing the probabilities of all possible alignment sequences  $\pi$ , which is formulated as follows:

$$P(Y|X) = \sum_{\pi \in A_{Y,X}} P(\pi|X) \quad (8)$$

$A_{Y,X}$  represents the set of all possible paths that align the input sequence  $X$  to the output sequence  $Y$ . The probability  $P(\pi|X)$  of each path  $\pi$  is calculated using the probability distribution obtained from the Softmax function over the input sequence.

Therefore, we have two losses of OCR and a loss of HSV and defined finally loss as follows:

$$Loss = loss_{OCR1} + \alpha \times loss_{HSV} + loss_{OCR2} \quad (9)$$

where  $\alpha$  is a hyper-parameter.

## 4 Experiment

We train the model based on Pytorch 1.13 platform with NIDIA 3090 and i7-8700 CPU. We use the minibatch Adam with a base learning rate  $10^{-4}$ , and this study tests our method exclusively on the ChiSig dataset and HanSig dataset. We use False Rejection Rate (FRR), False Acceptance Rate (FAR), Equal Error Rate (EER), Area Under the Curve (AUC) and Accuracy (Acc) to comprehensively evaluate our approach and compare it with other existing approaches.

### 4.1 Dataset

**ChiSig.** The ChiSig [31] dataset covers all tasks related to signature detection, recovery, and verification. For this dataset, we randomly selected 250 signatures as the training set and used the remaining 250 signatures as the test set. For each name, signatures from the same volunteer were considered as real sample pairs, while signatures from different volunteers were considered as forged sample pairs. Specifically, forged data were only used as forged sample pairs, not as real sample pairs. To ensure data balance between real sample pairs and forged sample pairs, redundant sample pairs were removed from the study.

**HanSig.** HanSig is a comprehensive, large-scale offline Chinese handwritten signature dataset designed to address the nuances of signature authentication [12]. A unique feature of HanSig is its incorporation of real-world variability, achieved through the inclusion of signatures in three distinct styles per writer, reflecting the natural intra-writer variations seen in everyday life. We randomly split HanSig into a training set and a test set. The training set comprises 795 names signed by 213 writers, while the test set includes 90 names signed by 25 writers. From the training set, 20 writers’ signatures (78 names) are randomly selected for validation. For each name in the test set, we follow a similar procedure used in the CEDAR and BHSig to form 190 positive pairs and 190 negative pairs. The final test data of HanSig consists of 34,200 signature pairs.

We tested the OCR4HSV model in ChiSig and HanSig and the datasets exhibited distinct stylistic characteristics, as illustrated in Table 1.

### 4.2 Comparison with Previous Methods

To verify the effectiveness of the model and keep up with the development of the current handwriting recognition task, this paper selects the latest deep learning

**Table 1.** Detailed information on the ChiSig dataset and the HanSig dataset.

Dataset name	Script	names	writers	Samples	Train/Test	Genuine/Forger	Dict
ChiSig	CHS	500	102	10,242	250/250	-/-	525
HanSig	CHS, CHT	885	238	35,400	795/90	20/20	805

models for comparison, including SigNet (2017arXiv) [2], IDN (2019CVPR) [28], InceptionResnet (2022CVPR) [31] and MCFFN (2024AAS) [20]. Among them, InceptionResnet is the baseline model provided by the dataset paper. For the HanSig dataset, because the dataset is relatively new, there is no mature model yet. Therefore, this paper uses the baseline provided by the dataset paper for comparison.

**Table 2.** Comparison on ChiSig dataset (%).

Model	FRR	FAR	Acc
InceptionResnet	–	–	93.6
SigNet	–	–	82.28
IDN	10.46	17.91	84.82
MCFFN	5.34	5.34	95.23
OCR4HSV (ours)	<b>5.26</b>	<b>2.89</b>	<b>95.92</b>

In Table 2, experimental results show that OCR4HSV outperforms current mainstream offline handwriting identification algorithms, achieving an accuracy of 95.92%. It also demonstrated superiority in comparisons of FRR and FAR, achieving optimal results in both. Compared to the baseline, OCR4HSV demonstrates an accuracy increase of 2.32%, and it improved by 0.69% compared to the latest algorithm (MCFFN). This sufficiently proves the superiority of the OCR4HSV model proposed in this paper.

For the HanSig dataset, we conformed to previous methodologies in designing our experiments, adhering to the framework outlined by the authors of HanSig. Specifically, we utilized metrics such as FRR, FAR, EER, and AUC to evaluate our experimental outcomes.

**Table 3.** Comparison on HanSig dataset (%).

Model	FRR	FAR	EER	AUC
Simple Baseline (pre-trained VGG-16)	32.43	19.66	26.31	80.94
VGG-16 with triplet loss	15.60	22.40	19.07	89.47
VGG-16 with co-tuplet loss	14.20	16.21	15.26	92.60
MS-SigNet with triplet loss	9.99	10.82	10.44	95.92
MS-SigNet with co-tuplet loss	<b>7.69</b>	11.85	9.93	96.38
OCR4HSV (ours)	10.79	<b>7.87</b>	<b>9.14</b>	<b>96.97</b>

The experimental results presented in Table 3 demonstrate that OCR4HSV outperforms prevailing handwriting recognition algorithms, achieving an EER of 9.14% and an AUC of 96.97%. Compared to MS-SigNet, the algorithm proposed by the authors, OCR4HSV shows improvements of 0.79% in EER and 0.59% in AUC, both reaching state-of-the-art levels. This substantiates the superiority of the OCR4HSV model introduced in this paper.

### 4.3 Ablation Study

To validate the effectiveness of each module, we also conducted ablation experiments, with specific results presented in Table 4:

**Table 4.** Ablation study result (%) of OCR4HSV.

Method	ChiSig			HanSig		
	EER	AUC	Acc	EER	AUC	Acc
baseline	5.89	98.76	93.46	11.18	95.69	88.76
+LAM	5.72	98.77	94.29	10.90	95.91	89.04
+MGFB	5.05	99.03	94.78	10.07	96.37	89.64
+OCR	6.31	98.57	93.68	11.01	95.89	88.95
+LAM+MGFB	4.30	99.25	95.38	10.10	96.53	89.86
+LAM+OCR	5.24	98.94	94.69	10.05	96.53	89.55
+MGFB+OCR	4.91	99.11	95.11	9.82	96.74	90.12
<b>OURS</b>	<b>4.21</b>	<b>99.31</b>	<b>95.92</b>	<b>9.14</b>	<b>96.97</b>	<b>90.67</b>

Insights from Table 4 reveal that the introduction of the LAM, OCR, and MGFB modules each contributes distinct features to the model. Focusing on individual contributions, the MGFB module stands out due to its extraction of multi-scale global features, achieving impressive outcomes. Specifically, on the ChiSig dataset, it boosts performance by 1.32%. When considering the synergistic effect of two modules combined, the pairing of LAM and OCR exhibits a significant improvement, enhancing ChiSig performance by 1.23% and HanSig by 0.79%. This is because when textual feature information is introduced, the LAM strategically emphasizes edge information to prevent the model from over-focusing on text features, thereby mitigating overfitting. Ultimately, the OCR4HSV method surpasses alternative approaches across all evaluation metrics.

## 5 Conclusion

In this paper, we integrate the OCR task with the HSV task, employing an MTL approach to introduce the OCR4HSV model for HSV independent of the author. OCR4HSV introduces OCR features for text information extraction within the

HSV framework, mitigating overfitting in HSV tasks and thereby enhancing validation accuracy. Extensive experiments on the ChiSig and HanSig datasets demonstrate the efficacy of our proposed OCR4HSV. In the future, we will focus on extending the application of the model to signatures in various languages.

**Acknowledgement.** This work is supported by National Natural Science Foundation of China (61773325, 62222609, 62076236), Unveiling and Leading Projects of Xiamen (No. 3502Z20241011), Open Project of the State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS2024101), Natural Science Foundation of Xiamen (3502Z202373058), and Fujian Key Technological Innovation and Industrialization Projects (2023XQ023).

## References

1. Chaudhuri, A., et al.: Optical Character Recognition Systems. Springer (2017)
2. Dey, S., Dutta, A., Toledo, J.I., Ghosh, S.K., Lladós, J., Pal, U.: Signet: convolutional siamese network for writer independent offline signature verification. arXiv preprint [arXiv:1707.02131](https://arxiv.org/abs/1707.02131) (2017)
3. Dhavale, S.V.: Advanced Image-Based Spam Detection and Filtering Techniques. IGI Global (2017)
4. 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)
5. 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)
6. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376 (2006)
7. Gu, A.: Modeling Sequences with Structured State Spaces. Stanford University (2023)
8. Gu, A., Dao, T.: Mamba: linear-time sequence modeling with selective state spaces. arXiv preprint [arXiv:2312.00752](https://arxiv.org/abs/2312.00752) (2023)
9. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. arXiv preprint [arXiv:2111.00396](https://arxiv.org/abs/2111.00396) (2021)
10. Guerbai, Y., Chibani, Y., Hadjadji, B.: The effective use of the one-class svm classifier for handwritten signature verification based on writer-independent parameters. *Pattern Recogn.* **48**(1), 103–113 (2015)
11. Hafemann, L.G., Sabourin, R., Oliveira, L.S.: Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recogn.* **70**, 163–176 (2017)
12. Huang, F.H., Lu, H.M.: Multiscale feature learning using co-tuplet loss for offline handwritten signature verification. Available at SSRN 4677183
13. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 603–612 (2019)
14. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. *Int. J. Comput. Vision* **116**, 1–20 (2016)






15. Li, C., Lin, F., Wang, Z., Yu, G., Yuan, L., Wang, H.: Deepshv: user-independent offline signature verification using two-channel CNN. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 166–171. IEEE (2019)
16. Li, H., Wei, P., Hu, P.: Static-dynamic interaction networks for offline signature verification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 1893–1901 (2021)
17. Li, H., Wei, P., Ma, Z., Li, C., Zheng, N.: Transosv: offline signature verification with transformers. *Pattern Recogn.* **145**, 109882 (2024)
18. Li, M., et al.: Trocr: transformer-based optical character recognition with pre-trained models. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 13094–13102 (2023)
19. Liang, X., Niu, M., Han, J., Xu, H., Xu, C., Liang, X.: Visual exemplar driven task-prompting for unified perception in autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9611–9621 (2023)
20. Lin, C., et al.: Offline handwriting verification based on siamese network and multi-channel fusion. *Acta Automat. Sinica* **50**(AAS-CN-2023-0777), 1 (2024). <https://doi.org/10.16383/j.aas.c230777>
21. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
22. Liu, C., Liu, Y., Dai, R.: Writer identification by multichannel decomposition and matching (1997). <http://www.aas.net.cn/article/id/17080>
23. Liu, L., Huang, L., Yin, F., Chen, Y.: Offline signature verification using a region based deep metric learning network. *Pattern Recogn.* **118**, 108009 (2021)
24. Ren, J.X., Xiong, Y.J., Zhan, H., Huang, B.: 2c2s: a two-channel and two-stream transformer based framework for offline signature verification. *Eng. Appl. Artif. Intell.* **118**, 105639 (2023)
25. Ren, X., Lai, S., et al.: Medical image enhancement based on laplace transform, sobel operator and histogram equalization. *Acad. J. Comput. Inf. Sci.* **5**(6) (2022)
26. Shen, J., Zhen, X., Worring, M., Shao, L.: Variational multi-task learning with gumbel-softmax priors. *Adv. Neural. Inf. Process. Syst.* **34**, 21031–21042 (2021)
27. 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)
28. Wei, P., Li, H., Hu, P.: Inverse discriminative networks for handwritten signature verification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5764–5772 (2019)
29. Xin, Y., Du, J., Wang, Q., Lin, Z., Yan, K.: Vmt-adapter: parameter-efficient transfer learning for multi-task dense scene understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 16085–16093 (2024)
30. Xu, Y., et al.: Layoutxlm: multimodal pre-training for multilingual visually-rich document understanding. arXiv preprint [arXiv:2104.08836](https://arxiv.org/abs/2104.08836) (2021)
31. Yan, K., et al.: Signature detection, restoration, and verification: a novel chinese document signature forgery detection benchmark. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE (2022). <https://doi.org/10.1109/cvprw56347.2022.00564>
32. Yang, E., et al.: Adatask: a task-aware adaptive learning rate approach to multi-task learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 10745–10753 (2023)



33. Yang, Z., et al.: Focal and global knowledge distillation for detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4643–4652 (2022)
34. 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)



# Learning Explicit Radical Representations for Zero-Shot Chinese Character Recognition

Song-Liang Pan<sup>1,2</sup> , Da-Han Wang<sup>1,2</sup> , Nanfeng Jiang<sup>1,2</sup> ,  
Xu-Yao Zhang<sup>3</sup> , and Shunzhi Zhu<sup>1,2</sup> 

<sup>1</sup> School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China

2222031144@stu.xmut.edu.cn, {wangdh,szzhu}@xmut.edu.cn

<sup>2</sup> Fujian Key Laboratory of Pattern Recognition and Image Understanding, Xiamen 361024, China

<sup>3</sup> State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation of Chinese Academy of Sciences, Beijing 100190, China  
xyz@nlpr.ia.ac.cn

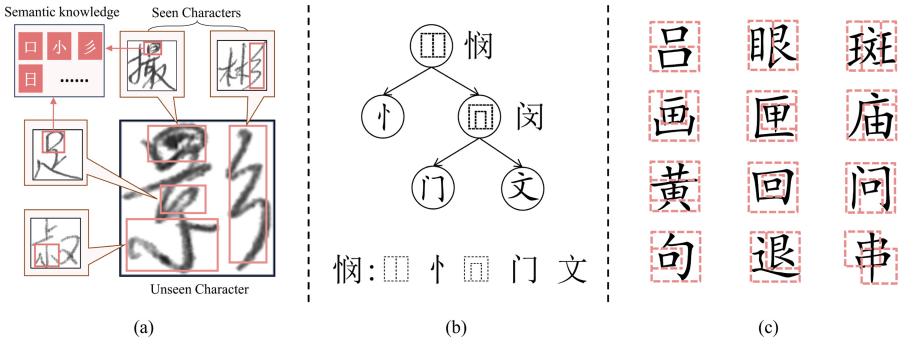
**Abstract.** Zero-shot Chinese character recognition (ZSCCR) aims to recognize unseen Chinese characters by learning the semantic knowledge of seen characters. Radical-based methods treat Chinese characters as combinations of radicals, recognizing characters by predicting the radicals in the images. Existing radical-based methods have a closed radical parsing process that cannot be intervened in mid-course, relying only on semantic labels for constraints. However, semantic embedding vectors are usually manually designed and lack alignment with visual features, making it extremely difficult for the model to learn and locate discriminative radical representations from visual features. This paper proposes a ZSCCR network called Learning Explicit Radical Representations (LERRNet). LERRNet introduces learnable attribute hint vectors to guide the model in locating discriminative radicals and learning explicit representations of images. Specifically, we introduce a Radical Relevance Enhanced Encoder (RREE) to enhance the correlation of local radicals by augmenting the relationships between grid regions in visual features. Guided by attribute hint vectors, LERRNet employs a Radical Representation Decoder (RRD) to locate the most relevant regions of each radical in the given image and learn explicit radical representations. Extensive experiments demonstrate that LERRNet outperforms state-of-the-art radical/stroke-based methods across three ZSCCR benchmarks.

**Keywords:** Chinese character recognition · Zero-shot learning · Radicals representation learning

## 1 Introduction

Zero-Shot Chinese Character Recognition (ZSCCR) has received extensive attention due to the pressing need for recognizing unseen Chinese characters in various

application scenarios, such as open-set text recognition [10, 20]. ZSCCR rethinks and changes the traditional Chinese Character Recognition (CCR) paradigm, which regards Chinese characters as a more granular target (radical/stroke description) rather than a unified whole (character). Since radicals/strokes are shared attributes among Chinese characters, ZSCCR methods can acquire the capability to recognize unseen character categories through exploring and learning the semantic knowledge (as shown in Fig. 1 (a)). Thus, the key purpose for ZSCCR is learning and locating discriminative semantic representations from visual features. This benefits the model by allowing them to recognize unseen characters effectively.



**Fig. 1.** (a) The semantic knowledge of seen characters can be transferred to the unseen characters. (b) Radicals tree and IDS description of the character “惘”. (c) Twelve Chinese character structures and corresponding character examples.

Early ZSCCR methods [2, 16, 23] usually decompose Chinese characters into sequences and predict characters by decoding visual features into radical/stroke sequences. However, due to common issues like repetition, errors, and omissions in sequence prediction, these methods exhibit relatively low fault tolerance. To address this problem, on the one hand, many researchers propose to embed characters based on ideographic description sequences (IDS) and then transform visual features into the same semantic space (*e.g.*, hierarchical decomposition [1], crucial radicals [12], CLIP alignment [20]) for character recognition. Although these methods have achieved progressive improvements, they implicitly decode global visual features and lack cueing information, which makes it difficult for the model to focus on critical radical regions in the image, as these radicals are often present in only a tiny portion of the visual image. On the other hand, the IDS embedding vectors are mostly manually designed with limited information representing the characters. This will lead to redundancy when directly transforming visual features into semantic representations.

To tackle the above challenges, we propose a novel ZSCCR framework called Learning Explicit Radical Representations (LERRNet), which improves the relevance of regional radicals in visual features and localizes discriminative explicit

radical representations in the target image. Specifically, LERRNet consists of a Radicals Relevance Enhancement Encoder (RREE), a Radical Representation Decoder (RRD), and a Semantic Matching Network (SMN). Firstly, to mitigate the drop in radical relevance due to flattened features, RREE uses the relative geometric embedding features to augment the visual features and improve the transferability of visual features to semantic features. Then, we process radicals and structures into the attribute hint vectors and use them to guide RRD in locating the most relevant image region for each radical/structure to obtain the explicit radicals representations. Finally, SMN uses the radicals-represented visual features and IDS semantic vectors to achieve character recognition. Extensive experiments show that LERRNet outperforms the current state-of-the-art radicals/strokes-based methods in the ZSCCR benchmark test.

The main contributions of this paper are summarized as follows:

- We introduce a novel ZSCCR method, Learning Explicit Radical Representations (LERRNet), which utilizes learnable attribute hint vectors to learn and localize explicit radical representations.
- We introduce a radicals relevance enhancement encoder to mitigate the loss of radicals' relevance due to the flattening of visual features, thus improving the transferability of visual features to semantic features.
- Extensive experiments validate that the proposed LERRNet model outperforms state-of-the-art radicals/strokes-based methods on three ZSCCR benchmarks.

## 2 Related Work

### 2.1 Traditional Chinese Character Recognition

Early CCR methods mainly relied on handcrafted features [8], which had limited performance due to insufficient representative capacity [4]. With the development of deep learning and big data, deep neural networks have replaced traditional methods. MCDNN [5] is one of the pioneering attempts to apply CNN to CCR. This method achieves performance close to the human level in large handwritten Chinese character recognition vocabulary. Subsequently, Zhong *et al.* [27] combined a simplified version of GoogLeNet with traditional feature extraction methods, surpassing human performance. Zhang *et al.* [24] proposed a method combining directMap with CNN to reduce the reliance of the CCR model on data augmentation and model ensembles. They introduced an adaptation layer to address the issue of varied handwriting styles. Although the methods above have achieved excellent performance in traditional Chinese character recognition, they operated under the closed-world assumption [25], meaning they cannot recognize Chinese characters that do not appear in the training set. Since there are many uncommon Chinese characters and emerging new characters in reality [10], these models must be annotated and retrained to recognize these new categories, incurring a significant cost and thus limiting their practicality. Therefore, this paper primarily discusses CCR research under the zero-shot learning setting [15].

## 2.2 Zero-Shot Chinese Character Recognition

ZSCCR aims to recognize unseen Chinese characters by learning from seen Chinese character classes and auxiliary information. Based on the auxiliary information, current ZSCCR methods can mainly be categorized into radical-based, stroke-based, and glyph-based.

Radical-based and stroke-based methods treated Chinese characters as combinations of radicals/strokes, predicting characters based on their compositional structure. DenseRAN [16] interpreted Chinese characters using a specific structure of radical sequences and proposed a GRU-based attention decoding structure to generate radical sequences. Subsequently, a series of methods based on radical sequence prediction have been proposed [17, 23]. Although these methods can recognize unseen Chinese characters, the performance of recognition is compromised due to errors in radical prediction. To address these issues, HDE [1] proposed a radical embedding method. Chinese characters were decomposed hierarchically and embedded into a semantic space, enabling recognition through direct interaction between visual and semantic embedding vectors. SLD [2] introduced a stroke-level decomposition method, utilizing a Transformer [14] to decode Chinese character stroke sequences for recognition, resolving the zero-shot radical problem. STAR [22] simultaneously considered information at both radical and stroke levels, reducing recognition ambiguity. ACPM [28] proposed using three types of decomposition knowledge, character, radical, and stroke, for joint matching. RSST [19] represented Chinese characters as stroke trees and organized them based on their radical structures, fully leveraging the advantages of radicals and strokes. Recently, SIR [12] discovered varying contributions of different radicals in character discrimination and improved previous sequence matching and embedding methods using radical self-information.

Glyph-based methods prepared a template image for each category (usually a computer-generated print image). By learning from template image knowledge, models can recognize unseen classes of Chinese characters with different styles and types. For instance, CCR-CLIP [20] introduced a CLIP-like framework aligning template images with IDS for character recognition, achieving impressive performance. SideNet [9] proposed joint learning of character-level representations with the assistance of radicals and template images. In this work, we only utilize radical information as auxiliary knowledge. Therefore, for a fair comparison, we select methods based on radicals, strokes, and partial glyphs to design our comparative experiments. Additionally, we report results without employing template images in these glyph-based methods.

## 3 Methodology

### 3.1 Preliminaries

**Auxiliary Information.** Unlike English, Chinese characters are diverse and complex. According to the Chinese national standard GB18030-2005<sup>1</sup>, there are

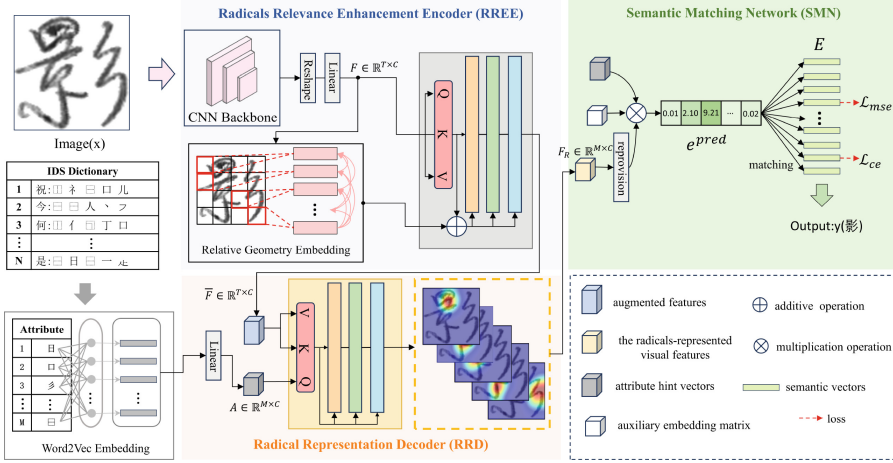
<sup>1</sup> [https://zh.wikipedia.org/wiki/GB\\_18030](https://zh.wikipedia.org/wiki/GB_18030).

70,244 categories of Chinese characters, of which only 3,755 are commonly used. Despite the intricate nature of Chinese characters, human beings can swiftly grasp numerous characters through acquired learning, even discerning characters they have never encountered before. This phenomenon arises because all Chinese characters share common attributes, such as strokes or radicals, enabling humans to swiftly memorize characters by understanding them and their meanings. The current ZSCCR method emulates the human process of learning Chinese characters based on inferring radical/stroke compositions within images to predict characters.

In ZSCCR, the attributes of Chinese characters serve as prior knowledge to aid in training and recognition. This auxiliary information can be categorized into three types: radical, stroke, and glyph. This work primarily focuses on radical information. As illustrated in Fig. 1 (b), each Chinese character can be represented as a radical tree, where parent nodes denote the structure of radical combinations, and leaf nodes represent radicals, with their IDS descriptions obtained through depth-first traversal. It is noted that the commonly used 3,755 characters are composed of 514 radicals and 12 basic structures (as shown in Fig. 1 (c)). In this paper, besides utilizing IDS embedding (*e.g.*, RIE [12]) to obtain class-level semantic vectors to represent Chinese characters, we additionally employ word2Vec [13] to acquire each radical/structural word vector to assist LERRNet in learning radical representations.

**Overview.** As illustrated in Fig. 2, our LERRNet comprises a Radicals Relevance Enhancement Encoder (RREE), a Radical Representation Decoder (RRD), and a Semantic Matching Network (SMN). LERRNet first utilizes RREE to alleviate the problem of radical relevance loss in the flattened features, then employs RRD to learn explicit radical representations and their localization in visual features guided by attribute hint vectors. Finally, the radicals-represented visual features are input into SMN for visual semantic interaction to realize ZSCCR.

**Problem Definition.** We denote seen Chinese character data as  $S = \{x_s, y_s\}$ , where  $x_s \in X$  denotes the image of seen characters and  $y_s \in Y^s$  represents the corresponding class labels from the seen class set  $Y^s$ . Let the set of unseen character classes be represented as  $U = \{x_u, y_u\}$ , where the image data  $x_u$  is unavailable. Given  $S$  and  $U$ , a non-intersecting constraint exists on their character class sets, *i.e.*,  $Y^s \cap Y^u = \emptyset$ .  $S$  and  $U$  collectively contain  $N = \text{len}(Y^s \cup Y^u)$  Chinese characters. Their IDS sequences are known and can be represented by  $M$  radical-level attributes (*i.e.*, radicals/structures). We utilize word2vec [13] to obtain the word vectors for each attribute to initialize the attribute hint vectors  $A \in \mathbb{R}^{M \times C}$ , which will assist our RRD in learning the explicit radical representations in visual features. Note that  $A$  can be optimized through training. Additionally, we use RIE [12] to obtain the class-level semantic vectors for each class to aid in the final character matching, with the set of semantic vectors denoted as  $E = \{e_1, e_2, \dots, e_N\} \in \mathbb{R}^{N \times M}$ .



**Fig. 2.** The architecture of the proposed LERRNet model. LERRNet comprises a radicals relevance enhancement encoder, a radical representation decoder, and a semantic matching network.

### 3.2 Radicals Relevance Enhancement Encoder

For the radical-based approaches, different radicals contribute to recognition to varying degrees [12]. Meanwhile, the critical radicals may occupy only a tiny portion of the image (for example, distinguishing between “戊” and “戌” hinges on “一”). ZSCCR methods usually rely on character coherence information to analyze their corresponding radical sequences. However, visual features usually need to be flattened for decoding, which leads to the entanglement of representations in different image regions and destroys the relevance between radicals, indirectly making it challenging for the model to focus on critical radicals. To solve this problem, we propose radical relevance enhancement scaled dot-product attention, which strengthens visual features by enhancing the relative relationships between grid regions of characters.

**Relative Geometry Embedding.** Initially, we employ the CNN backbone to process input images  $x \in \mathbb{R}^{H \times W \times Q}$ , yielding a two-dimensional flattened feature map  $F \in \mathbb{R}^{T \times C}$ , where  $T = H \times W$ . To extract the relative geometric features from the image [3, 7, 26], we then acquire 2-D positional coordinate pairs for each grid. For the  $p$ -th grid, its coordinate pairs is  $\{(u_p^{min}, v_p^{min}), (u_p^{max}, v_p^{max})\}$ . Through these coordinate pairs, we can compute the relative central coordinates  $(cu_p, cv_p)$  of the  $p$ -th grid:

$$(cu_p, cv_p) = \left( \frac{u_p^{min} + u_p^{max}}{2}, \frac{v_p^{min} + v_p^{max}}{2} \right), \tag{1}$$

$$[w_p, h_p] = [(u_p^{max} - u_p^{min}) + 1, (v_p^{max} - v_p^{min}) + 1], \tag{2}$$

where  $(u_p^{min}, v_p^{min})$  and  $(u_p^{max}, v_p^{max})$  represent the coordinates of the upper-left and lower-right corners of the grid  $p$ , additionally,  $w_p$  and  $h_p$  represent the width and height of the grid  $p$ .

Subsequently, we can construct the relative geometry relationship between the grid features. Given the grids  $p$  and  $q$ , the region geometry features  $G_{pq}$  of grid  $p$  relative to  $q$  are as follows:

$$G_{pq} = ReLU(w_g^T FC(r_{pq})), \quad r_{pq} = \begin{pmatrix} \log \left( \frac{|cu_p - cu_q|}{w_p} \right) \\ \log \left( \frac{|cv_p - cv_q|}{h_p} \right) \end{pmatrix}, \quad (3)$$

where  $FC$  denotes the linear layer activated by the  $ReLU$  function,  $r_{pq}$  is the relative geometry relationship and  $w_g^T$  is the learnable hyperparameter matrix.

**Radicals Relevance Enhancement.** Finally, we extract regional geometric features from visual features and input them into the Transformer [14] encoder to learn attention features. The formula is:

$$Q^F = FW_q^F, K^F = FW_k^F, V^F = FW_v^F, \quad (4)$$

$$\bar{F} \leftarrow F + softmax\left(\frac{Q^F K^{F^T}}{\sqrt{d_k^F}} + G\right)V^F, \quad (5)$$

where  $F \in \mathbb{R}^{T \times C}$  denotes the visual features,  $W_q^F, W_k^F, W_v^F$  represent learnable hyperparameter matrices.  $Q^F, K^F$ , and  $V^F$  denote the query, key, and value derived from the weighted encapsulated features,  $d_k^F$  is a scaling factor,  $G$  is the relative geometric feature, and  $\bar{F} \in \mathbb{R}^{T \times C}$  is the augmented features.

### 3.3 Radical Representation Decoder

We adopt the Transformer [14] decoder to construct RRD to learn more accurate radical representations from visual features. RRD consists of multi-head attention layers and feed-forward networks. Guided by the attribute hint vectors  $A \in \mathbb{R}^{M \times C}$ , the decoder can gradually learn radical representations from visual information and effectively locate the image regions most relevant to various radicals/structures in the target image. We use the output  $\bar{F}$  of RREE as keys and values, the learnable attribute hint vectors  $A$  as the query input for multi-head self-attention layers to learn radical representations in visual features, formulated as:

$$Q_i = AW_{q_i}, K_i = \bar{F}W_{k_i}, V_i = \bar{F}W_{v_i}, \quad (6)$$

$$head_i = softmax\left(\frac{Q_i K_i^T}{\sqrt{d_k^R}}\right), \quad (7)$$

$$\bar{F}_{att} = (head_1 \oplus head_2 \oplus \dots \oplus head_i)W_o, \quad (8)$$



where  $W_{q_i}, W_{k_i}, W_{v_i}$  are learnable hyperparameter matrices,  $d_k^R$  is a scaling factor,  $W_o$  are learnable hyperparameter matrices, and  $\oplus$  denotes the concatenate operation.

Then, the attention features of multi-head attention input are fed into the feed-forward network to obtain  $F_R$ :

$$F_R = \text{ReLU}(\overline{F}_{att}W_1 + bias_1)W_2 + bias_2, \quad (9)$$

where  $F_R \in \mathbb{R}^{M \times C}$  the radicals-represented visual features.

### 3.4 Semantic Matching Network

After decoding visual features into the radical representation, we map them into the semantic vector  $e$  to match Chinese characters. Specifically, the radicals-represented visual features  $F_R$ , combined with  $A$ , are input into the semantic matching head to obtain semantic vector  $e^{\text{pred}}$ . It is defined as:

$$e^{\text{pred}} = \left[ \sum_{j=1}^M (A \cdot W \cdot F_R^T)_{ij} \right]_{i=1}^M, \quad (10)$$

where  $A \cdot W \cdot F_R^T \in \mathbb{R}^{M \times M}$  is the semantic matching matrix,  $W \in \mathbb{R}^{C \times C}$  is an auxiliary embedding matrix, and  $\top$  indicates transposition. Then, according to  $e^{\text{pred}}$ , we can match Chinese characters based on all semantic vectors:

$$y^{\text{pred}} = \arg \max_{y \in Y} e^{\text{pred}} \cdot e^y, \quad (11)$$

where  $y^{\text{pred}} \in Y$  denotes the predicted class label,  $e^y \in E$  denotes the candidate semantic vector, and  $Y = Y^s \cup Y^u$ .

### 3.5 Loss Functions

We adopt a cross-entropy loss  $\mathcal{L}_{ce}$ , a regression loss  $\mathcal{L}_{mse}$  to optimize LERRNet.

When certain radicals are present in an image, their associated image embedding distributions are closer to the corresponding semantic vectors  $e^y$ . Therefore, to obtain better compatibility scores, we adopt a radical embedding cross-entropy loss to optimize our model. Given a batch of training data with batch size  $B$ ,  $\mathcal{L}_{ce}$  is defined as follows:

$$\mathcal{L}_{ce} = -\frac{1}{B} \sum_{i=1}^B \left( e_i^{\text{pred}} \cdot e_i^y - \log \sum_{y' \in Y^s} \exp \left( e_i^{\text{pred}} \cdot e^{y'} \right) \right), \quad (12)$$

where  $e^{y_i}$  represents the corresponding ground truth semantic vector of the  $i$ -th sample.

We also introduce the radical regression loss constraint. Specifically, we treat visual-semantic mapping as a regression problem, optimizing by minimizing the

mean square error between the ground truth semantic vectors and the predicted vectors for each sample.  $\mathcal{L}_{mse}$  is defined as follows:

$$\mathcal{L}_{mse} = \frac{1}{B} \sum_{i=1}^B \|e_i^{\text{pred}} - e_i^y\|^2. \quad (13)$$

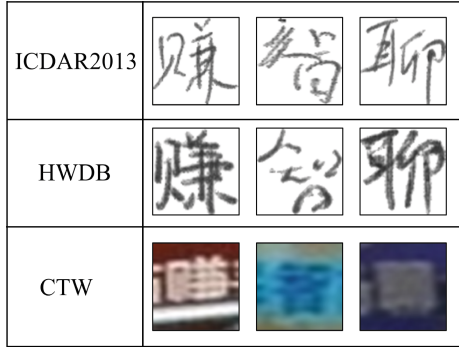
Finally, the overall LERRNet optimization function is:

$$\mathcal{L}_{total} = \mathcal{L}_{ce} + \mu\mathcal{L}_{mse}, \quad (14)$$

where  $\mu$  is a hyperparameter that balances the losses.

## 4 Experiments

### 4.1 Experimental Setup



**Fig. 3.** Some examples of Chinese characters from the dataset used in our experiments.

**Dataset.** We evaluate the proposed method on the HWDB1.0–1.1 [11], ICDAR 2013 [18], and CTW [21] datasets. Some examples of samples from these datasets are shown in Fig. 3. The HWDB 1.0 dataset consists of 4,037 character classes with 1,680,258 samples, while the HWDB 1.1 dataset includes 3,926 Chinese character classes with 1,172,907 samples. We selected the 3,755 commonly used first-level Chinese characters for our evaluation. This ICDAR2013 test dataset contains 3,755 handwritten Chinese character classes with 224,419 samples written by 60 authors. This CTW dataset contains 812,872 Chinese character samples (760,107 images for training and 52,765 for testing) extracted from scene text images and covers 3,580 Chinese character classes. The CTW dataset poses significant challenges for Chinese character recognition due to the occlusion, low resolution, diverse styles, etc. For fair comparisons, we followed the dataset partition proposed by SLD [2] for ZSCCR experiments.

**Implementation Details.** We implemented the model using PyTorch, and all experiments are deployed on an NVIDIA RTX 4090 24GB GPU. Our CNN backbone is the same as SLD [2]. The batch size and input image size are 64 and 128-128. We use the Adam optimizer with a learning rate of 0.001, and the weight decay is 0.0001. Based on experience, we set  $\mu$  to 0.05 for all datasets. In the transformer, the head number, the scaling factor, the channel number, and the dropout are set to 4, 300, 512, and 0.4, respectively. We adopt Character Accuracy (CACC) as the evaluation metric. Meanwhile, we follow the traditional approach by combining the training and test sets to construct the candidate set [16].

## 4.2 Comparing with State-of-the-Art

**Experiments on Zero-Shot Settings.** This section compares our method with the state-of-the-art ZSCCR method on handwritten and scene character datasets. The handwritten character datasets include HWDB1.0-1.1 [11] and ICDAR2013 [18]; the scene dataset is CTW [21]. For a fair comparison, we follow the settings of SLD [2] and conduct experiments using the character zero-shot setting and radical zero-shot setting.

For the character zero-shot setting, we respectively selected the first  $m$  classes of samples from the HWDB1.0-1.1 and the CTW as the training sets for handwritten and scene experiments, where the range of  $m$  is  $\{500, 1000, \dots, 2755\}$  and  $\{500, 1000, \dots, 3150\}$ . The test set for handwritten experiments consists of all samples from the last 1000 classes of the ICDAR2013, while the test set for scene experiments comprises samples from the first 500 classes of the CTW test set. Additionally, due to some radicals in the test characters that do not exist in seen characters, we inevitably need to address the radical zero-shot problem [1, 2]. Therefore, we conducted experiments under the radical zero-shot setting to evaluate our method comprehensively. The radical zero-shot experimental setup includes two steps: (1) calculating the frequency of occurrence of each radical in the candidate character set, and (2) if a character’s IDS contains radicals appearing fewer than  $n$  times, where  $n \in \{50, 40, 30, 20, 10\}$ , the character is assigned to the test set; otherwise, it is assigned to the training set.

The experimental results in Table 1 indicate that our method performs excellently on handwritten datasets. In the character zero-shot setting, our method outperforms state-of-the-art methods by an average of 8.38%. Notably, as  $m$  decreases, our method significantly improves, suggesting its ability to learn radical knowledge from a few samples efficiently. Although our performance is slightly lower than stroke-based methods, *e.g.*, STAR [22], in the radical zero-shot setting, our proposed method is significantly better than previous radical-based methods, demonstrating its capability to learn better radical representations with strong generalization, enabling inference of unseen radicals from seen ones. The results of scene character experiments are shown in Table 2, where our method achieves the best performance in both settings. In the character and radical zero-shot settings, it outperforms the current state-of-the-art methods by an average of 9.29% and 0.47%, respectively, which indicates our method’s

**Table 1.** Results of character zero-shot (left) and radical zero-shot (right) tasks on Handwritten characters. (%)

Handwritten	$m$ for Character Zero-Shot Setting					$n$ for Radical Zero-Shot Setting				
	500	1000	1500	2000	2755	50	40	30	20	10
DenseRAN [16]	1.70	8.44	14.71	19.51	30.68	0.21	0.29	0.25	0.42	0.69
HDE [1]	4.90	12.77	19.25	25.13	33.49	3.26	4.29	6.33	7.64	9.33
SLD [2]	5.60	13.85	22.88	25.73	37.91	5.28	6.87	9.02	14.67	15.83
ACPM [28]	9.72	18.50	27.74	34.00	42.43	4.29	6.20	7.85	10.36	12.51
STAR [22]	7.54	19.47	27.79	35.53	43.86	6.95	12.28	14.74	<b>18.37</b>	<b>23.23</b>
SIR [12]	7.43	15.75	24.01	27.04	40.55	–	–	–	–	–
RSST [19]	11.56	21.83	35.32	39.22	47.44	7.94	11.56	15.13	15.92	20.21
SideNet* [9]	5.1	16.2	33.8	44.1	50.3	–	–	–	–	–
CCR-CLIP* [20]	21.79	42.99	55.86	62.99	72.98	<b>11.15</b>	<b>13.85</b>	<b>16.01</b>	16.76	15.96
LERRNet(Ours)	<b>32.73</b>	<b>55.36</b>	<b>66.01</b>	<b>70.81</b>	<b>73.59</b>	8.35	11.22	14.81	15.98	19.37

\* This result does not use glyph auxiliary information

**Table 2.** Results of character zero-shot (left) and radical zero-shot (right) tasks on Scene characters. (%)

Scene	$m$ for Character Zero-Shot Setting					$n$ for Radical Zero-Shot Setting				
	500	1000	1500	2000	3150	50	40	30	20	10
DenseRAN [16]	0.15	0.54	1.60	1.95	5.39	0	0	0	0	0.04
HDE [1]	0.82	2.11	3.11	6.96	7.75	0.18	0.27	0.61	0.63	0.90
SLD [2]	1.54	2.54	4.32	6.82	8.61	0.66	0.75	0.81	0.94	2.25
ACPM [28]	3.44	6.18	10.65	15.40	21.29	0.54	0.70	0.74	0.78	0.89
STAR [22]	1.19	3.77	8.04	11.00	11.27	2.16	2.33	2.76	4.81	5.35
RSST [19]	1.41	2.53	4.95	9.32	13.02	1.21	1.29	1.89	2.90	3.88
CCR-CLIP* [20]	3.55	7.70	9.48	17.15	24.91	0.95	1.77	2.36	2.59	4.21
LERRNet(Ours)	<b>6.84</b>	<b>15.92</b>	<b>20.95</b>	<b>31.55</b>	<b>34.20</b>	1.92	<b>2.56</b>	<b>3.63</b>	<b>4.90</b>	<b>6.77</b>

\*This result does not use glyph auxiliary information

better ability to handle issues such as low resolution and occlusion in scene samples.

**Experiments on General CCR Settings.** We also conducted experiments on general CCR settings to evaluate the performance of our method in recognizing only seen characters. For handwritten character experiments, we used HWDB1.0–1.1 as the training set and the ICDAR2013 dataset as the test set. The experimental results are shown in Table 3, where our method achieved the second-best performance in handwritten and scene character experiments, second only to ACPM [28], which benefits from multiple decomposition information of Chinese characters. Our method only learns radical representations for

CCR, outperforming other methods based on radicals and strokes. Such results once again demonstrate the effectiveness of our approach.

**Table 3.** Performance comparison in benchmark ICDAR2013 and CTW with general CCR settings. (%)

Method	Decomposition Level	ICDAR2013	CTW
ResNet [6]	Character	96.83	79.46
DenseRAN [16]	Radical	96.66	85.56
RAN [23]	Radical	93.79	81.80
HDE [1]	Radical	97.14	89.25
SLD [2]	Stroke	96.28	85.29
ACPM [28]	Radical, Stroke and Character	<b>97.80</b>	<b>91.48</b>
STAR [22]	Radical and Stroke	97.11	85.43
CCR-CLIP [20]	Radical	97.18	85.78
LERRNet(Ours)	Radical	97.39	89.37

### 4.3 Ablation Study

To evaluate the performance of the proposed RREE and RRD, we conducted ablation experiments on handwritten and scene character datasets. For this purpose, we set up simplified pipeline versions as baselines. For RRD, we use a linear layer instead. The results of the ablation experiments are shown in Table 4, where the proposed RREE and RRD both achieved significant progress. It is worth noting that using only RREE resulted in negative gains on CTW. Still, when combined with RRD, more considerable improvements were obtained, indicating that these two modules can complement each other.

**Table 4.** Results of ablation study. (%)

Module	Handwritten		Scene	
	1500	2755	1500	3150
Baseline	47.77	57.82	15.60	18.19
Baseline + RREE	48.67	60.90	12.29	13.22
Baseline + RRD	61.75	70.03	16.91	28.56
Baseline + RREE + RRD	<b>66.01</b>	<b>73.59</b>	<b>20.95</b>	<b>34.20</b>

#### 4.4 Visualization of Attention Maps

We conducted visualization on LERRNet to demonstrate its effectiveness in learning and locating key radical representations from visual features. Figure 4 shows the attention maps of the top 6 radical representations with the highest contribution to recognition in  $F_R$ . The results demonstrate that our LERRNet accurately focuses on the crucial radical regions in Chinese characters, which validates the effectiveness of our approach.

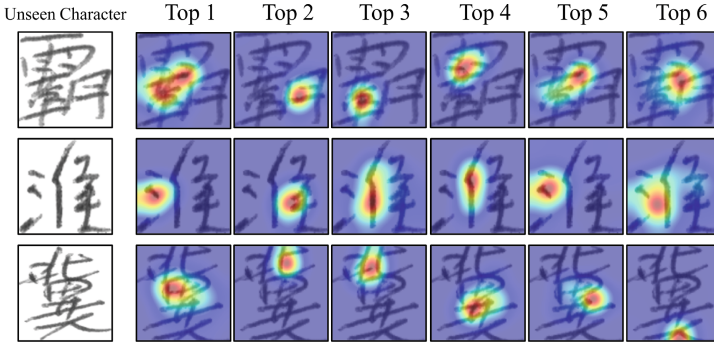


Fig. 4. Visualisation of attention maps for Ours LERRNet.

## 5 Conclusion

This paper proposes a novel ZSCCR framework called Learning Explicit Radical Representations (LERRNet). Firstly, our LERR adopts a Radicals Relevance Enhancement Encoder (RREE) to mitigate the loss of radicals' relevance due to the flattening of visual features, thus improving the transferability of visual features to semantic features. Then, we introduce a Radical Representation Decoder (RRD) to learn the most relevant regions for each radical in the image. Finally, a Semantic Matching Network (SMN) facilitates the interaction between the radicals-represented visual features and semantic vectors, thus recognizing unseen character classes. Extensive experiments on three popular benchmark datasets demonstrate the effectiveness of this approach. Our future work will incorporate more effective printed character images as auxiliary information to enhance the robustness of our framework and explore new semantic embedding methods to improve radical representation.

**Acknowledgements.** This work is supported by National Natural Science Foundation of China (61773325, 62222609, 62076236), Unveiling and Leading Projects of Xiamen (No. 3502Z20241011), Open Project of the State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS2024101), Natural Science Foundation of Xiamen (3502Z202373058), and Fujian Key Technological Innovation and Industrialization Projects (2023XQ023).

## References

1. Cao, Z., Lu, J., Cui, S., Zhang, C.: Zero-shot handwritten chinese character recognition with hierarchical decomposition embedding. *Pattern Recogn.* **107**, 107488 (2020)
2. Chen, J., Li, B., Xue, X.: Zero-shot chinese character recognition with stroke-level decomposition. In: Zhou, Z.H. (ed.) *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 615–621. International Joint Conferences on Artificial Intelligence Organization (2021)
3. Chen, S., et al.: Transzero++: cross attribute-guided transformer for zero-shot learning. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022)
4. Chen, X., Jin, L., Zhu, Y., Luo, C., Wang, T.: Text recognition in the wild: a survey. *ACM Comput. Surv.* **54**(2), 1–35 (2021)
5. Cireşan, D., Meier, U.: Multi-column deep neural networks for offline handwritten Chinese character classification. In: *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6. IEEE (2015)
6. 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)
7. Herdade, S., Kappeler, A., Boakye, K., Soares, J.: Image captioning: transforming objects into words. *Adv. Neural Inf. Process. Syst.* **32** (2019)
8. Jin, L.W., Yin, J.X., Gao, X., Huang, J.C.: Study of several directional feature extraction methods with local elastic meshing technology for HCCR. In: *Proceedings of the Sixth International Conference for Young Computer Scientist*, pp. 232–236 (2001)
9. Li, Z., Huang, Y., Peng, D., He, M., Jin, L.: Sidenet: learning representations from interactive side information for zero-shot Chinese character recognition. *Pattern Recogn.* **148**, 110208 (2024)
10. Liu, C., Yang, C., Yin, X.C.: Open-set text recognition via character-context decoupling. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4523–4532 (2022)
11. Liu, C.L., Yin, F., Wang, D.H., Wang, Q.F.: CASIA online and offline Chinese handwriting databases. In: *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pp. 37–41. IEEE (2011)
12. Luo, G.F., Wang, D.H., Du, X., Yin, H.Y., Zhang, X.Y., Zhu, S.: Self-information of radicals: a new clue for zero-shot Chinese character recognition. *Pattern Recogn.* **140**, 109598 (2023)
13. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space. In: *International Conference on Learning Representations* (2013). <https://api.semanticscholar.org/CorpusID:5959482>
14. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017)
15. Wang, W., Zheng, V.W., Yu, H., Miao, C.: A survey of zero-shot learning: settings, methods, and applications. *ACM Trans. Intell. Syst. Technol.* **10**(2), 1–37 (2019)
16. Wang, W., Zhang, J., Du, J., Wang, Z.R., Zhu, Y.: Denseran for offline handwritten Chinese character recognition. In: *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 104–109. IEEE (2018)
17. Wu, C., Wang, Z.R., Du, J., Zhang, J., Wang, J.: Joint spatial and radical analysis network for distorted Chinese character recognition. In: *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 5, pp. 122–127. IEEE (2019)

18. Yin, F., Wang, Q.F., Zhang, X.Y., Liu, C.L.: ICDAR 2013 Chinese handwriting recognition competition. In: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), pp. 1464–1470. IEEE (2013)
19. Yu, H., Chen, J., Li, B., Xue, X.: Chinese character recognition with radical-structured stroke trees. *Mach. Learn.* 1–21 (2023)
20. Yu, H., Wang, X., Li, B., Xue, X.: Chinese text recognition with a pre-trained clip-like model through image-ids aligning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 11943–11952 (2023)
21. Yuan, T.L., Zhu, Z., Xu, K., Li, C.J., Mu, T.J., Hu, S.M.: A large chinese text dataset in the wild. *J. Comput. Sci. Technol.* **34**, 509–521 (2019)
22. Zeng, J., Xu, R., Wu, Y., Li, H., Lu, J.: Zero-shot Chinese character recognition with stroke-and radical-level decompositions. In: 2023 International Joint Conference on Neural Networks (IJCNN), pp. 1–9. IEEE (2023)
23. Zhang, J., Du, J., Dai, L.: Radical analysis network for learning hierarchies of chinese characters. *Pattern Recogn.* **103**, 107305 (2020)
24. Zhang, X.Y., Bengio, Y., Liu, C.L.: Online and offline handwritten Chinese character recognition: a comprehensive study and new benchmark. *Pattern Recogn.* **61**, 348–360 (2017)
25. Zhang, X.Y., Liu, C.L., Suen, C.Y.: Towards robust pattern recognition: a review. *Proc. IEEE* **108**(6), 894–922 (2020)
26. Zhang, X., et al.: Rstnet: captioning with adaptive attention on visual and non-visual words. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15465–15474 (2021)
27. Zhong, Z., Jin, L., Xie, Z.: High performance offline handwritten Chinese character recognition using googlenet and directional feature maps. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 846–850. IEEE (2015)
28. Zu, X., Yu, H., Li, B., Xue, X.: Chinese character recognition with augmented character profile matching. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 6094–6102 (2022)





# Deep Learning for Arabic Word Classification: Leveraging Transfer Learning and Grad-CAM for Morphological Analysis

Mohamed Hjaiej<sup>1,3(✉)</sup>, Imen Ben Cheikh<sup>1,2</sup>, and Heithem Abbas<sup>1,3</sup>

<sup>1</sup> Latice Laboratory, University of Tunis, Tunis, Tunisia

<sup>2</sup> ISAMM, Mannouba University, Manouba, Tunisia

<sup>3</sup> FST, University of Tunis El Manar, Tunis, Tunisia

mohamed.hjaij@etudiant-fst.utm.tn

**Abstract.** We propose a deep learning approach to deal with large Arabic lexicon recognition. We propose a combination of methods aimed at learning and recognizing written decomposable Arabic words, trying to simulate writing human reading. The training is based on derivational and inflectional characteristics specific to Arabic. Word recognition process begins with an initial phase of inflection and derivation classifications using six CNN-BiLSTM models that have been transfer learned and fine-tuned. Subsequently, we employ the Grad-CAM technique on these classifiers to localize and extract word prefixes, infixes, and suffixes. Following this, a root extraction phase utilizes image pre-processing techniques to isolate characters that belong to the root, serving as preparation for root classification. Ultimately, we aim to utilize the classifier results to reconstruct words based on the different feature of the Arabic words morphology.

**Keywords:** Arabic Word Classification · Deep Learning · Explainable AI

## 1 Introduction

Morphological analysis and feature extraction of Arabic words using artificial intelligence pose significant challenges due to the language's rich and complex morphology. Arabic is characterized by its root-and-pattern system, extensive inflection, and derivational morphology, making accurate word classification a complex task. This complexity necessitates advanced techniques for effectively analyzing and interpreting Arabic words.

## 2 Morphological Linguistic Knowledge in Arabic

Understanding the morphological structure of Arabic is crucial for improving NLP models tailored to the language. Arabic morphology is rich and complex,

characterized by root-based word formation and extensive use of inflections and derivations. By dissecting the morphological processes in Arabic, we can gain insights into how words are formed, modified, and understood within their linguistic context. This section will delve into the key aspects of Arabic morphology, including derivational, inflectional, and agglutinative processes, providing examples to illustrate each type.

### 2.1 Derivational Morphology

Derivational Morphology involves the creation of new words by adding prefixes, suffixes, or other meaningful units (morphemes) to a base word (the root). This process often changes the word’s meaning and sometimes its grammatical category. The Fig. 1 depict an example with the Root: **كتب** (k-t-b) which mean “write.”

**Example 1:** Derivational form: **كاتب** (kātib), meaning “writer” (noun derived from the verb “write”) The derivational form **كاتب** (kātib) is formed by adding the vowel pattern **ا** (a) after the first character of the root, changing the meaning to “writer,” which is a noun derived from the verb “write.”

**Example 2:** The derivational form **مكتبة** (maktaba) is formed by adding the prefix **م** (ma-) and the suffix **ة** (-a) to the root, changing the meaning to “library,”

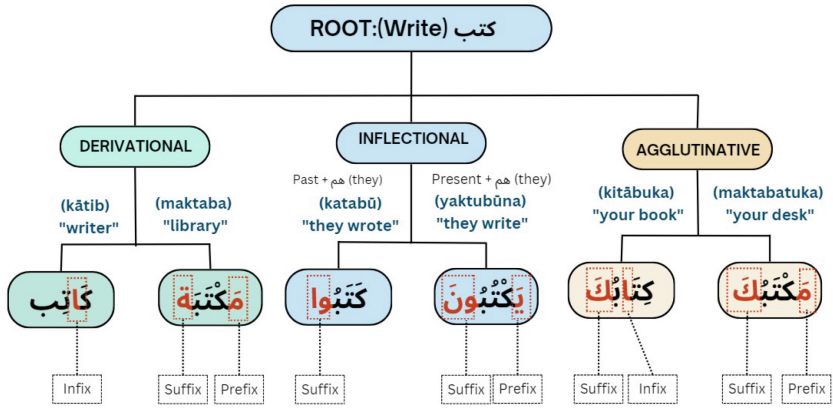


Fig. 1. Model Morphology

## 2.2 Inflectional Morphology

Inflectional morphology deals with the modification of a word to express different grammatical features such as tense, mood, number, case, and gender. Unlike derivational morphology, inflectional changes do not create new words but rather change the form of the same word to convey different grammatical information.

**Example 1:** Inflectional form: يَكْتُبُونَ (yaktubūna), meaning “they write” (present tense, third person plural) The inflectional form يَكْتُبُونَ (yaktubūna) is formed by adding the prefix يَ (ya-) and the suffix وَنَ (-ūna) to the root, indicating the present tense and third person plural, changing the meaning to “they write.”

**Example 2:** Inflectional form: كَتَبُوا (katabū), meaning “they wrote” (past tense, third person plural). The inflectional form كَتَبُوا (katabū) is formed by adding the suffix وَا (-ū) to the root, indicating the past tense and third person plural, changing the meaning to “they wrote.”

## 2.3 Agglutinative Morphology

While Arabic is not primarily an agglutinative language, it does use a form of agglutination in some contexts, particularly with pronouns and prepositions.

**Example 1:** Agglutinative form: مَكْتَبِكَ (maktabuka), meaning “your desk” (masculine singular). The agglutinative form مَكْتَبِكَ (maktabuka) is formed by adding the possessive suffix كَ (ka) to the base word مكتب (maktab), indicating “your desk” (masculine singular).

**Example 2:** Agglutinative form: كِتَابِكَ (kitābuka), meaning “your book” (masculine singular) The agglutinative form كِتَابِكَ (kitābuka) is formed by adding the possessive suffix كَ (ka) to the base word كتاب (kitāb), indicating “your book” (masculine singular).

## 3 NLP Within Humain-Reading Based Models

Pattern recognition is the automation of artificial perception tasks performed by the human sensory system and brain. It aims to classify entities into categories on the basis of observations made on them. In this respect, psychoanalytic studies show that humans memorize letters standing for the whole word instead

of sequences of separate letters; analogically to the fact that in the relatively long-term memory, learning by sentence writing - so semantic elaboration - is more effective than memorizing by learning by word lists.

The “Word Superiority Effect” concept, proposed by Mc Clelland and Rumelhart [McClelland, Belaid], was inspired by this human reading perception. It permits a layered representation of the word, from the local to the global layer and vice versa. According to [14], this model is applicable to Arabic script recognition on condition that an intermediary global level, known as the level of the pseudo word (PAW= Piece of Arabic Word), is added. Therefore, word recognition can be guided by the meaning of different possibilities of PAW combinations. The importance of the “Pseudo-Word Superiority Effect” derives not only from the primary function of PAWs in the Arabic language, but also from the analogy with the “Word Superiority Effect”.

On another hand, the integration of linguistic information in the recognition process is one of the most promising research approaches. In this context, interesting results were yielded in [11] study, which focused on the “Word Derivation Effect” by integrating linguistic information such as roots and patterns in the Arabic word recognition process. Accordingly, they generally estimate that word recognition can occur at several independent, but complementary, levels, such as root recognition, pattern recognition, agglutination recognition, recognition of conjugation elements, etc.

Then, in this work, we assume too that the human-reading process is also characterized - in addition to the “Word or Pseudo-Word superiority effect”, from a scriptural point of view- by the concept of “Word Morphology Effect” which includes principally derivational and flexional layers, from a linguistic point of view. To substantiate this, we will experiment the incorporation of morphological linguistic knowledge into a perceptual model while benefiting from the power of deep learning in text recognition.

## 4 Related Work

In the literature, we can find many types of techniques that take advantage of advancements in the field of artificial intelligence, particularly deep learning, to enhance text and word recognition. Optical Character Recognition (OCR) is a fundamental research area in text recognition. Many techniques have been applied to achieve higher performance and enhance results by considering newer challenges, such as poor quality and noisy images, while trying to cover many languages and their different characteristics.

### 4.1 Language-Free Methods

Language-free methods refer to techniques or approaches that do not rely on linguistic knowledge or language models to perform tasks. These methods typically depend on visual, structural, or statistical features rather than semantic or syntactic information from a language. For instance, Connectionist Temporal

Classification (CTC) is a type of neural network output layer used for sequence modeling, particularly in tasks where the alignment between the input and output sequences is unknown. CTC is widely used in speech recognition, handwriting recognition, and scene text recognition. This technique has been applied to several languages, such as English [4] and Arabic [6] based on CNN-LSTM network. There is a widely adopted technique in the field of text recognition that combines segmentation-recognition strategy with Connectionist Temporal Classification (CTC). This method comprises three essential steps: image regularization, segmentation, and recognition. The first step, image regularization, involves applying various preprocessing techniques to enhance image quality by addressing challenges such as low resolution, blur, perspective distortion, and nonuniform lighting. The second step, segmentation, involves dividing the regularized text image into subimages, each containing a single character or language element. Finally, the recognition step processes these subimages using character recognition techniques. This step employs various classifiers, ranging from traditional machine learning methods like Support Vector Machines (SVM) to advanced deep learning approaches, to identify and classify the characters[3].

## 4.2 Language-Based Methods

Linguistic-based approaches generally refer to techniques or methodologies that heavily leverage linguistic knowledge or language models to address various tasks or challenges. These methods utilize linguistic rules, semantics, syntax, and other linguistic features to enhance or guide the process of data analysis, modeling, or decision-making within a specific context. The approach proposed by [5] integrates two distinct models. The vision model comprises a backbone network and a position attention module. This model utilizes ResNet and transformer units for feature extraction and sequence modeling, respectively. Given an input image, the model outputs character probabilities. The language model, as the second component, operates independently for spelling correction. It takes probability vectors of characters as input and produces probability distributions of expected characters. To integrate visual and linguistic features effectively, a final fusion technique is employed.

## 5 Approach

To facilitate Arabic word recognition, we propose a compositional approach based on four steps, as depicted the Fig. 2, that takes advantages from the Derivational and Inflectional arabic morphology.

### 5.1 Step 1: Features Classifications

As depicted the Fig. 2, this step employs ensemble classifiers to extract features from Arabic words, aiming to discern key characteristics such as the root pattern (Schema), gender (male or female), and person (first person, second person,

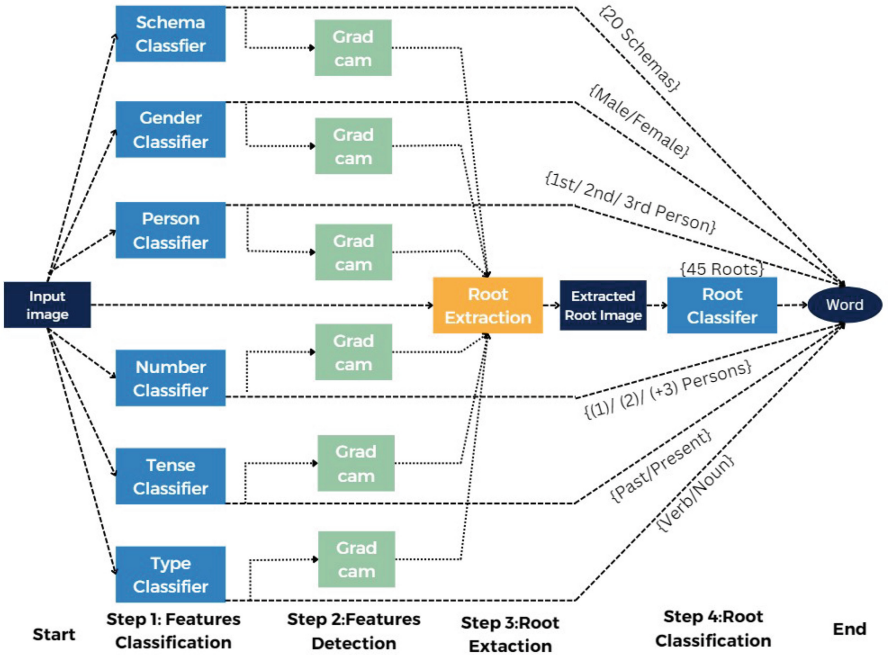
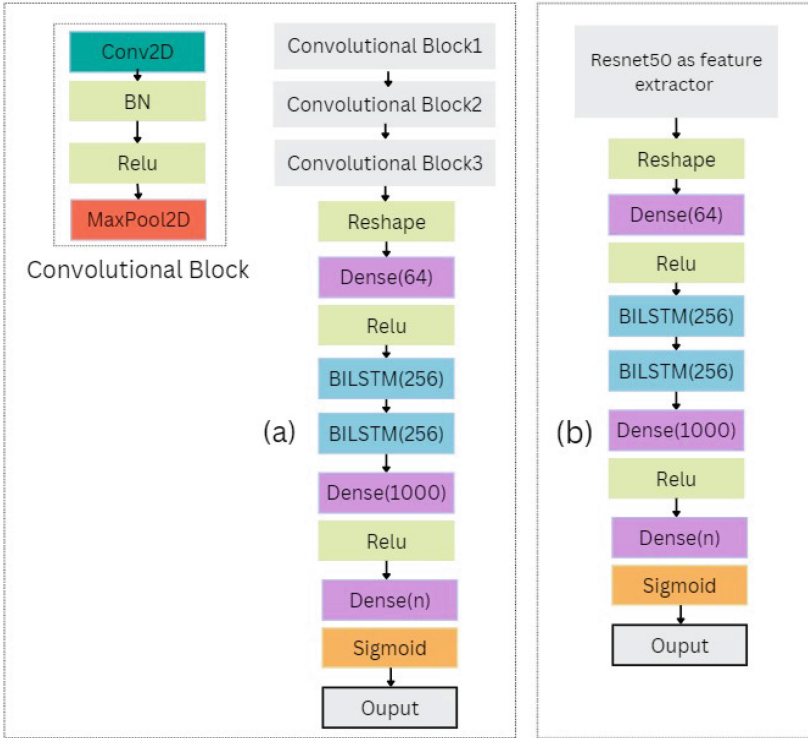


Fig. 2. Four steps based approach

or third person). In Arabic, the feature ‘number’ denotes whether a pronoun represents singular or plural form. Unlike English, Arabic includes pronouns specifically for dual entities, such as أنتما “Antoma” for the second person and هما “Homa” for the third person, used when addressing two males or two females, respectively. Our classifiers also incorporate tense recognition (past, present) and type (verb, noun).

**The Model Architecture.** The six classifiers are based on a CNN-BiLSTM network for feature extraction and sequence pattern recognition. These models share a similar classifier architecture, with differences only in the last layer, which varies depending on the number of classes. To examine our approach, we prepared a model from-scratch as depicted in Fig. 3 (a). This model employed three blocks of convolutional neural networks and Maxpooling layers for feature extraction, followed by a sequence of dense and BiLSTM layers for sequence information extraction and classification. On the other hand, recent models based on ResNet have demonstrated their performance across a wide range of classification problems. For this reason, we incorporated ResNet50 for feature extraction, as shown in Fig. 3 (b), leveraging its residual blocks and attention mechanisms. This integration ensures deeper feature extraction by effectively propagating information through the network’s residual connections.



**Fig. 3.** (a) Model from-scratch, (b) Fine-tuned ResNet50 model

### 5.2 Step 2: Features Detection

The classification models in our approach serve two primary roles. The first role is feature classification, which helps us identify different prefixes, infixes, and suffixes in Arabic words, if they exist. Simultaneously, we need to remove these additional characters to achieve more accurate root classification in the subsequent step. To accomplish this, we employ Gradient-weighted Class Activation Mapping (Grad-CAM) [12] to get the regions corresponding to the prefixes, infixes, and suffixes as depicted the Fig. 2. This allows us to better understand and isolate these features for improved root classification.

### 5.3 Step 3: Root Extraction

In this stage, the third step of the Fig. 2, we aim to employ various image processing techniques to remove the regions of interest identified by the preceding Grad-CAM step, retaining only the root characters. This approach ensures that only the essential components of the Arabic words are preserved for further analysis and classification.

## 5.4 Step 4: Root Classification

As we prepared for the second and third steps, our aim in this step, as presented in the Fig. 2, is to apply root classification to determine the corresponding root of the Arabic word in the image. The model employed has the same architecture as the previous classifier in the first step, aiming to utilize the same model architecture features.

## 5.5 Data Preparation

For training purposes, we prepared two datasets in order to examine our approach and analyze our results. The classifiers for gender, person, number, tense, and type were trained using the same dataset, which was structured to contain all classes for each classifier. The first dataset originally consisted of 1044 images, but we tailored it for each classifier to include only the corresponding classes necessary for each model. The second dataset was also derived from the APTI database [13], comprising 2510 images labeled with 20 schemas and 45 roots.

## 5.6 Training

**Loss Function.** Binary Cross-Entropy (BCE) loss, also known as log loss, is a loss function used for binary classification tasks where the model's output is a probability value between 0 and 1. It measures the difference between the true labels and the predicted probabilities, penalizing the model more heavily for confident and incorrect predictions.

$$\text{BCE} = \frac{-1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (1)$$

**Epochs.** The gender, person, number, tense and type classifier have been trained for 150 epochs, and the Schema and root classifiers have been trained from 300 epochs due the difference of the number of the data for models.

# 6 Results

The Training results for the both models showed similar interesting performances while the both models achieved over 0.98 accuracy on the train data and over 0.91 on test data for all arabic word features.

## 6.1 Transfer Learning of ResNet50

The transfer learning and fine-tuning techniques of ResNet50 as feature extractors present promising results on different image features and dataset sizes, making it the selected model for our approach as depicted in the Table 1. We achieve over 0.97 accuracy for all models on the train dataset and over 0.91 as accuracy on the test dataset.



**Table 1.** The ResNet50 model Performance Metrics

Model	Acc Train	Acc Test	Loss Train	Loss Test	Nb. images train	Nb. images test
Tense	98%	93%	0.05	0.43	621	156
Gender	98%	91%	0.06	0.6	691	173
Type	98%	96%	0.0001	0.28	815	204
Number	99%	98%	0.000006	0.008	621	156
Person	99%	91%	0.006	0.34	560	140
Schema	99%	93%	0.0001	0.03	1587	397
Root	97.62%	96.78%	0.0047	0.0054	1587	397

## 6.2 Training Results of the from-Scratch Model

The training results of the from-scratch model demonstrate comparable performance in recognizing Arabic word features, as shown in the Table 2.

**Table 2.** The from-scratch model Performance Metrics

Model	Acc Train	Acc Test	Loss Train	Loss Test	Nb. images train	Nb. images test
Tense	98.21%	91.54%	0.0431	0.4164	621	156
Gender	99%	97.82%	0.00001	0.0811	691	173
Type	99%	99%	0.0001	0.0005	815	204
Number	99%	97%	0.00001	0.0059	621	156
Person	93.03%	81.89%	0.1440	0.4925	560	140
Schema	99%	97%	0.00001	0.0212	1587	397
Root	99%	98%	0.000001	0.0007	1587	397

## 6.3 Deep Comparative Results

To advance our research and refine the results of our approach, we applied the GRAD CAM technique to determine the regions and positions of the tense characters in image words. While both models, the fine-tuned ResNet50 and the model from-scratch present interesting classification results and similar training curves, as depicted in Fig. 6, The ResNet50 showed promising results with GRAD CAM, unlike the model from-scratch, which demonstrated a lack of performance with GRAD CAM in the second step.

**Grad CAM Results with ResNet50 as Tense Classifier.** The Fig. 4 illustrates the features of each tense, showing how the model successfully identifies the prefixes and suffixes belonging to each tense, distinct from the root.

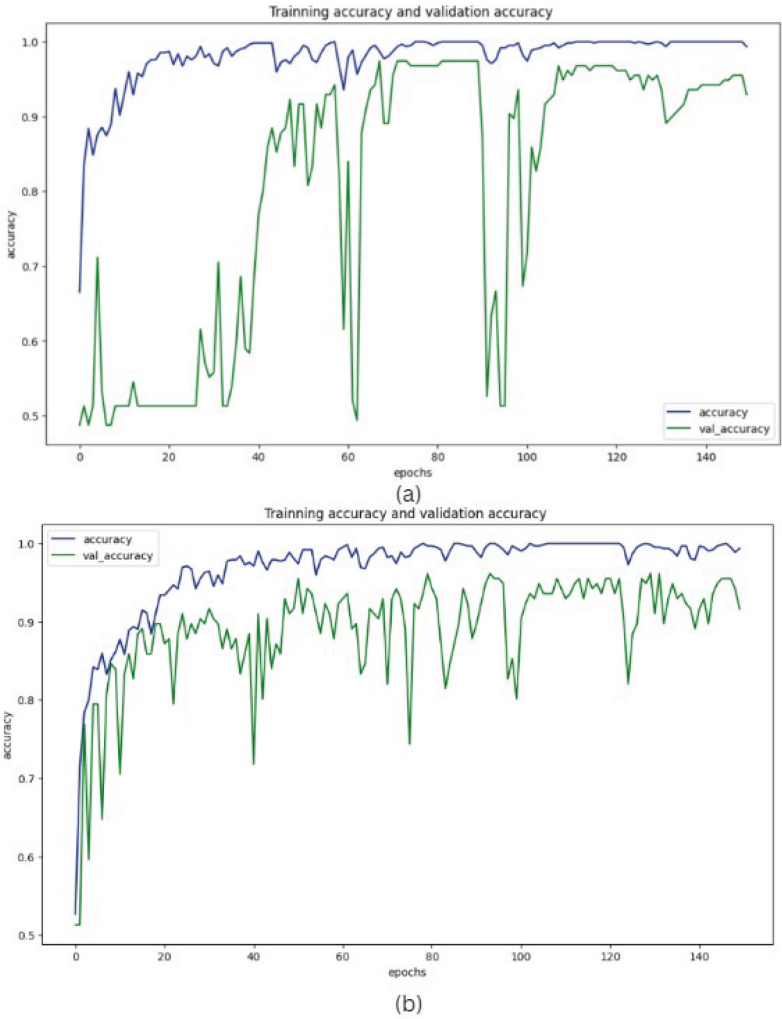


**Fig. 4.** Grad CAM results with the fine tuned ResNet50

**Grad CAM Results with the from-Scratch Model as Tense Classifier.** Unlike the ResNet model, the model from-scratch shows poor results with the Grad-CAM technique, as it does not focus on any part of the Arabic word prefixes and suffixes as present the Fig. 5.



**Fig. 5.** Grad CAM results the from-scratch model CNN-BiLSTM



**Fig. 6.** (a) Train and validation accuracy curves of the fine tuned ResNet50, (b) Train and validation accuracy curves of the from-scratch model.

## 7 Conclusion and Perspectives

Overall, our approach aims to decompose Arabic word recognition into four steps, taking advantage of the Arabic word features and morphology. To examine this approach, we prepared two types of models that leverage the advancements of deep learning techniques by employing a model architecture that combines Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory

(BiLSTM), aiming to extract both spatial and temporal Arabic word morphology. While both models show performance in Arabic word feature classification, the ResNet50 succeeded in extracting tense features, unlike the model from-scratch.

Our findings encourage further research in this field by testing other Arabic features presented in this work that have not been treated. The results highlight the importance of using techniques such as transfer learning, fine-tuning, and GRAD CAM as explainability methods. These methods help present the learned knowledge of the models and can be utilized for other research purposes in the field of artificial intelligence and in computer vision tasks.


## References

1. McClelland, J.L., Rumelhart, D.E.: An interactive activation model of context effects. *Lett. Percept. Psychol. Rev.* **88**, 375–407 (1981)
2. Belaïd, A., Choisy, Ch.: Human Reading Based Strategies for Off-line Arabic Word Recognition SACH 2006, Summit on Arabic and Chinese Handwriting, 27–28 September 2006. University of Maryland, College Park (2006)
3. Wang, H., Pan, C., Guo, X., Ji, C., Deng, K.: From object detection to text detection and recognition: a brief evolution history of optical character recognition. *Wiley Interdiscip. Rev. Comput. Statist.* **13** (2021). <https://doi.org/10.1002/wics.1547>
4. Geetha, M., et al.: A Hybrid Deep Learning Based Character Identification Model Using CNN, LSTM, and CTC To Recognize Handwritten English Characters and Numerals, vol. 1–6(2022). <https://doi.org/10.1109/ICCCI54379.2022.9740746>
5. Fang, S., Xie, H., Wang, Y., Mao, Z., Zhang, Y.: Read Like Humans: Autonomous, Bidirectional and Iterative Language Modeling for Scene Text Recognition (2021)
6. Ziadi, F., Cheikh, I., Jemni, M.: A Deep Convolutional and Recurrent Approach for Large Vocabulary Arabic Word Recognition, pp. 213–220 (2022). <https://doi.org/10.5220/0010814800003122>
7. Alothman, A., Als Salman, A.M.: Arabic morphological analysis techniques. *Int. J. Adv. Comput. Sci. Appl.* **11** (2020). <https://doi.org/10.14569/IJACSA.2020.0110229>
8. Osman, A., Shalaby, M., Soliman, M., Elsayed, K.: Ar-CM-ViMETA: arabic image captioning based on concept model and vision-based multi-encoder transformer architecture. *Int. Arab J. Inf. Technol.* **21**(3), 458–465 (2024). <https://doi.org/10.34028/iajit/21/3/9>
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 (CVPR)* (2016)
10. Vaswani, A., et al.: Attention Is All You Need (2017)
11. Ben Cheikh, I., Belaïd, A., Kacem, A.: A Novel Approach for the Recognition of a Wide Arabic Handwritten Word Lexicon, ICPR, Tampa (2008)
12. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vision* **128**(2), 336–359 (2020). <https://doi.org/10.1007/s11263-019-01228-7>

13. Slimane, F., Ingold, R., Kanoun, S., Alimi, A., Hennebert, J.: A New Arabic Printed Text Image Database and Evaluation Protocols, pp. 946–950 (2009). <https://doi.org/10.1109/ICDAR.2009.155>
14. Zouaoui, Z., Ben Cheikh, I., Jemni, M.: Combinatorial optimization approach for arabic word recognition based on adaptive simulated annealing. In: Nyström, I., Hernández Heredia, Y., Milián Núñez, V. (eds.) CIARP 2019. LNCS, vol. 11896, pp. 480–489. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-33904-3\\_45](https://doi.org/10.1007/978-3-030-33904-3_45)



# A Cost Minimization Approach to Fix the Vocabulary Size in a Tokenizer for an End-to-End ASR System

Sunil Kumar Kopparapu<sup>(✉)</sup>  and Ashish Panda

TCS Research, Mumbai, India  
{sunilkumar.kopparapu, ashish.panda}@tcs.com  
<http://www.tcs.com/>

**Abstract.** Unlike hybrid speech recognition systems where the use of tokens was restricted to phones, biphones or triphones the choice of tokens in the end-to-end ASR systems is derived from the text corpus of the training data. The use of tokenization algorithms like Byte Pair Encoding (BPE) and WordPiece is popular in identifying the tokens that are used in the overall training process of the speech recognition system. Popular toolkits, like ESPNet use a pre-defined vocabulary size (number of tokens) for these tokenization algorithms, but there is no discussion on how vocabulary size was derived. In this paper, we build a cost function, assuming the tokenization process to be a black-box to enable choosing the number of tokens which might most benefit building an end-to-end ASR. We show through experiments on LibriSpeech 100 h set that the performance of an end-to-end ASR system improves when the number of tokens are chosen carefully.

**Keywords:** sub-word tokenization · speech recognition · sentencepiece · byte pair encoding

## 1 Introduction

It was a standard practise to choose mono-phones, bi-phones, tri-phones as the tokens to train a hybrid automatic speech recognition (ASR) system [11]. The tokens to be trained was dependent on the prominent sounds in that language and the training data required the phonetic transcription of the training speech corpus. However, with the advent of end-to-end systems and the availability of phonetic transcripts, the move has been on automatically identifying the tokens from the text data of the training corpus. As a result, most current ASR systems model tokens derived from the training text rather than use unit which have relevance to the language or pronunciation.

Several tokenization algorithms, such as Byte Pair Encoding (BPE) [13], WordPiece [12], or unigram language model tokenization [4] have been researched. These algorithms, broadly work on the principle of iteratively merging frequently occurring pairs of characters or tokens to create a vocabulary of tokens that can represent the language's vocabulary efficiently. The difference

among these tokenization algorithms lies in the way characters are paired. BPE uses a pre-tokenizer to split the training data into words and creates a set of base vocabulary consisting of all symbols that occur in the set of unique words. After this, BPE learns the merge rules to create a new symbol from two symbols of the base vocabulary. This process goes on till the desired number of symbols (or tokens) are created. While BPE chooses the most frequent symbol pairs to merge, WordPiece merges the symbol pair that maximizes the likelihood of the training data until the desired number of symbols have been obtained. In contrast to BPE and WordPiece, Unigram language model tokenization starts from a large set of symbols and trims down each symbol to obtain a smaller set of symbols. SentencePiece [5] is a popular language independent sub-word tokenizer and detokenizer for Neural Text Processing. It does not need a pre-tokenizer unlike BPE and hence it is suitable for languages such as Japanese and Chinese. While the above tokenization algorithms work on training text, various efforts have been made to bring in acoustic perspective into the tokenization process. Pronunciation Assisted Subword Modeling (PASM) [16], Acoustic Data Driven Subword Modeling (ADSM) and Phonetically Induced Subwords [9] are examples of such efforts.

Most ASR systems that use sub-word tokens for training fix the number of tokens (vocabulary size) and use one of the above tokenizers to generate the tokens from the training text data. To the best of our knowledge there has been no discussion on the criteria used for determining the optimal number of tokens and they are fixed empirically. In this paper, we explore a formulation that can help identify the number of tokens best suited for ASR training. We set about the task assuming the availability of a sub-word tokenizer and use it as a black box.

We first formulate a cost function which when minimized results in an optimal number of sub-word tokens for a given training text data. We specifically use an off-the-shelf tokenizer for the purposes of demonstration but it should be kept in mind that the formulation should work for any tokenizer. We then evaluate the performance of a standard deep architecture ASR to validate the choice of the number of sub-word tokens. The main contribution of this paper is in (a) formulating a framework to identify the number of sub-word tokens and (b) building a cost function to enable identifying the optimal number of sub-word tokens and (c) evaluating the performance on a ASR to validate the need for formal identification of the number of sub-word tokens required to train an ASR system.

The rest of the paper is organized as follows. We describe the formulation of the cost function to allow identification of the optimal number of tokens given a training text corpus in Sect. 2. In Sect. 3 we experiment with LibriSpeech 100h training data set to first identify the optimal number of tokens and then use the tokens to evaluate the performance of an end-to-end ASR. We conclude in Sect. 5.

## 2 Problem Setup

Let  $\mathcal{S}$  be a text corpus consisting of  $\mathcal{S} = \{s_1, s_2, \dots, s_k\}$   $k$  sentences associated with training data which consists of  $w$  words ( $w_u \ll w$ , unique). For simplicity, we will assume the text to be English so that blank spaces represent word boundaries and newlines identify sentences. Let  $\mathcal{T}$  be a tokenization routine (for example, byte pair encoding [17]) which takes as input a variable  $n$  and operates on  $\mathcal{S}$  to produce a set of tokens  $T_n$ , namely,

$$\mathcal{T}(n, \mathcal{S}) = T_n \quad (1)$$

where  $T_n = \{t_1, t_2, \dots, t_n\}$ ,  $|T_n| = n$  is the number of tokens,  $t_i$  is the  $i^{th}$  token, and  $t_i \neq t_j$  for  $\forall i \neq j$ . Let  $enc^{\mathcal{T}}$  and  $dec^{\mathcal{T}}$  be a pair of functions associated with  $\mathcal{T}$  such that

$$enc^{\mathcal{T}}(s_i) = \bigcup_{l=1}^{\beta_i} \tau_l \quad (2)$$

acts on a sentence  $s_i$  in the corpus  $\mathcal{S}$  and represents it using  $\beta_i$  tokens  $\{\tau_l\}_{l=1}^{\beta_i} \in T$ , and  $dec^{\mathcal{T}}$  function

$$dec^{\mathcal{T}}\left(\bigcup_{l=1}^{\beta_i} \tau_l\right) = s_i \quad (3)$$

reconstructs  $s_i$  by concatenating (represented by  $\bigcup$ ) a sequence of tokens. Let

$$\theta_t = \sum_{i=1}^k \beta_i \quad (4)$$

denote the number of tokens required to span the corpus  $\mathcal{S}$ . As mentioned earlier,  $\beta_i$  represents the number of tokens required to represent the sentence  $s_i$ . Let  $\Gamma$  represent the set of all token required to span the corpus  $\mathcal{S}$ , namely,

$$\Gamma = \bigcup_{i=1}^k \left( \bigcup_{l=1}^{\beta_i} \tau_l \right) \quad (5)$$

where the total number of tokens in  $\Gamma$  is  $|\Gamma| = \sum_{i=1}^k \beta_i$ . We can compute the frequency of occurrence of token  $\tau \in T_n$  as

$$f(\tau) = \sum_{i=1}^{|\Gamma|} [\Gamma_i = \tau]. \quad (6)$$

where  $\tau \in T_n$  and  $\Gamma_i \in \Gamma$  and  $[\cdot]$  are the Iverson brackets such that  $[Q]$  is defined to be 1 if  $Q$  is true, and 0 if it is false. Further let  $f^+$  and  $f^-$  represent the average of top few most frequently occurring tokens and the top few most infrequently occurring tokens respectively. Note that (a)  $\theta_t$ , (b)  $f^+$ , and (c)  $f^-$  are a function of  $n$  as seen in (1).



We hypothesize that choosing the optimal number of tokens would be equivalent to finding an  $n^*$  which minimizes the cost function (8), namely,

$$n^* = \min_n \{C\} \quad (7)$$

where

$$C = \left\{ \alpha_1 \overbrace{n}^{t_1} + \alpha_2 \overbrace{\left( \frac{f^+}{f^-} - 1 \right)}^{t_2} + \alpha_3 \overbrace{\left( \frac{\theta_t}{w} - 1 \right)}^{t_3} \right\} \quad (8)$$

is the cost function and  $\alpha_{1,2,3}$  are weights which are chosen heuristically.

The first term,  $t_1$ , in the cost function (8) is to ensure that the total number of tokens used to represent the corpus  $\mathcal{S}$  is small because a large  $n$  means not only training for a large number of tokens which in turn requires a larger amount of training data but smaller number of tokens can result in faster training and inference times for ASR systems. With fewer tokens to model, the computational complexity of the ASR system can be reduced, leading to quicker training convergence and real-time performance during inference. The second term,  $t_2$  ensures a balance between the most frequently occurring and the least frequently occurring tokens, because an imbalance in data can lead to bias during training [1]. Note that a perfectly balanced training data would have  $\left( \frac{f^+}{f^-} \right) = 1$ . And the third terms,  $t_3$  ensures that the number of tokens required to represent the corpus  $\mathcal{S}$  is close to the number of words  $w$  in the corpus. This constraint makes sure that the cost of compute is minimum, because most major Gen AI portals charge for their services based on the number of tokens required to represent the input and generate an output. The construction of the cost function as mentioned in (8) to identify the optimal number of tokens is one of the main contributions of this paper.

### 3 Experimental Setup

We use the LibriSpeech-100 [7] (sentences:28537; words:990093) dataset in our experiments. Note that the database consists of 100 h of read English speech accompanied by the textual transcript. The training data consists of  $k = 28538$  sentences and  $w = 990093$  words of which  $w_u = 33798$  are unique. Our experimental evaluations, in this paper, are of two types. We make use of the text transcript and determine the optimal number of tokens. We also validate the usefulness of the obtained optimal number of tokens by ASR systems using the same tokens. We report the performance obtained on the test-clean [6] (sentences:2620; words: 52576) and test-other [8] (sentences:2939; words:52343) datasets.

We have used state-of-the-art conformer encoder-decoder architecture as the ASR. The conformer model is implemented using an ESPNet toolkit [15] with Librispeech 100 h (low-resource) recipe. While we changed the number of tokens in the original recipe, we used the rest if the model hyper-parameters as described below:

The encoder of the conformer has 12 layers and the decoder has 6 layers. The model dimension is 256 while number of attention heads used is 4. Models were trained using Adam optimizer [2] with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.98$  and  $\epsilon = 10^{-9}$ , which is along the lines of the optimizer proposed in [14]. Warm up steps used is 25000. The models are trained for 100 epochs and the batch size was 64. A single Nvidia RTX 3090 GPU was used.

No language model has been used for shallow fusion in this work. The  $n$  tokens extracted from the training text served as output units. Features used for the model are log mel spectrograms with 80 dimensions along with the pitch (total 81). Three way speed perturbation [3] with speed factors of 0.9, 1.0 and 1.1 and SpecAugment [10] was used in all the experiments.

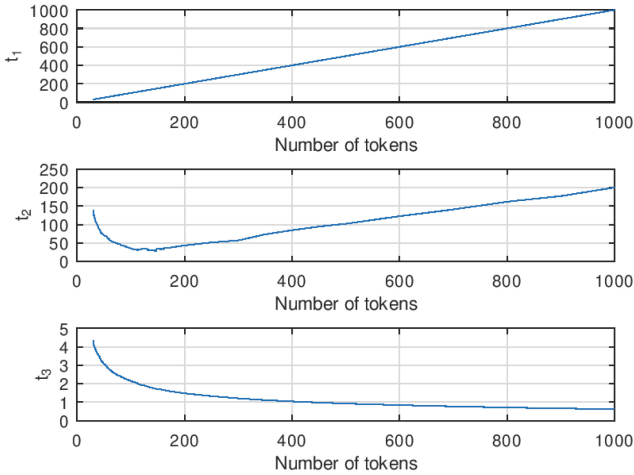
## 4 Experimental Results

We computed the cost function  $C$  (8) by varying  $n$  between 30 and 1000. We found that higher value of  $n (> 1000)$  always resulted in a higher  $C$  and  $n < 30$  resulted in the number of tokens ( $n$ ) being less than the number of characters in the training text data. We trained the SentencePiece [5], with unigram (`-vocab.size=n`, `-model.type=unigram`, `-split.by.whitespace=False`) called SentencePiece-Unigram model and with BPE (`-model.type=bpe`) called SentencePiece-BPE model.

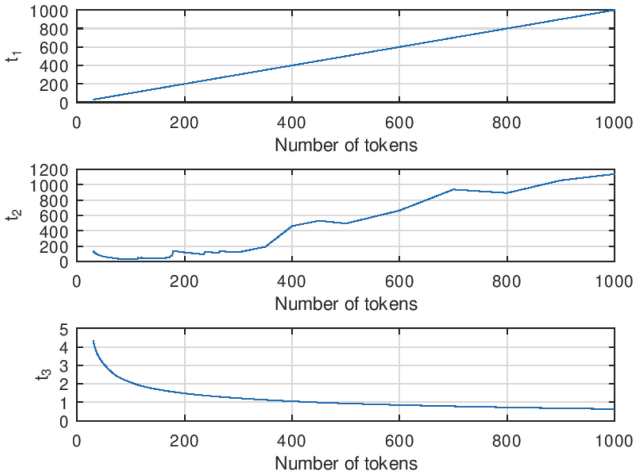
Figure 1 shows the plot of  $t_1, t_2, t_3$  (8) as a function of number of tokens (x-axis). In both Fig. 1a (SentencePiece-Unigram) and Fig. 1b (SentencePiece-BPE) it can be observed that while  $t_1$  is linearly increasing and  $t_3$  is exponentially decreasing;  $t_2$  exhibits a dip before linearly increasing as a function of  $n$ . Figure 2 (Fig. 3) shows the plot of the cost function  $C$ , for SentencePiece-Unigram (SentencePiece-BPE), for  $n = 1, \dots, 1000$  for  $\alpha_{1,2,3} = 1, 0, 0$  (Fig. 2a (3a)),  $\alpha_{1,2,3} = 0, 1, 0$  (Fig. 2b (3b)), and  $\alpha_{1,2,3} = 0, 0, 1$  (Fig. 2c (3c)) with the  $n^*$  (7) marked with a red “\*”.

Clearly the minimum value of  $C$  (8) varies with different values of  $\alpha$ 's. As expected,  $n^* = 30$  when the cost function is only a function of the number of tokens for both SentencePiece-Unigram (Fig. 2a) and SentencePiece-BPE (Fig. 3a) and  $n^* = 1000$  when  $C$  is a function of only  $t_3$  (see Fig. 2c and 3c). The most interesting aspect is observable for  $t_2$  where the  $n^* = 145$  for SentencePiece-Unigram (see Fig. 2b) and  $n^* = 97$  for SentencePiece-BPE (see Fig. 3b). Figure 4 shows the plot of the cost function  $C$  for  $\alpha_{1,2,3} = 1$ . As shown in Fig. 4a the minimum value of  $C$  occurs for  $n^* = 61$  for SentencePiece-Unigram model while  $n^* = 70$  for SentencePiece-BPE model.

The experiments above provide the optimal values for number of tokens corresponding to different values of  $\alpha$ 's. We now present the performance of the ASR systems with the number of tokens obtained above, namely  $n^* = 30, 61, 145, 1000$  for SentencePiece-Unigram and  $n^* = 30, 97, 70, 1000$  for SentencePiece-BPE, in Table 1. We compare this with the ESPNet recipe recommendation of using the number of tokens as 300 using SentencePiece-Unigram language model. Using  $n = 300$  results in an average WERs of 13.8% over the dev sets and



(a) SentencePiece-Unigram.



(b) SentencePiece-BPE.

**Fig. 1.**  $t_1, t_2, t_3$  in the cost function  $C$  (8) for  $n = 30$  to 1000

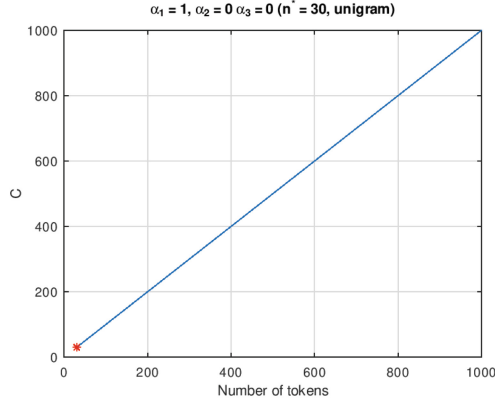
14.5% over the test sets (see Table 1). Using 30 tokens (which corresponds to  $\alpha_{1,2,3} = (1, 0, 0)$ ), the average WER over test sets reduces very slightly to 14.3%, however the average WER over the dev sets increases slightly to 13.9%. Using  $n^* = 61$  tokens and 1000 tokens (corresponding to  $\alpha_{1,2,3} = (0, 1, 0)$  and  $(0, 0, 1)$ , respectively), did not improve the performance, with  $n = 1000$  tokens performing the worst. It suggests that blindly increasing the number of tokens does not necessarily improve the performance of the ASR. When we provide equal weight

**Table 1.** WERs (in %) for various  $n$  on Librispeech 100h. “dev-avg”: average over dev-clean and dev-other. “test-avg”: average over test-clean and test-other

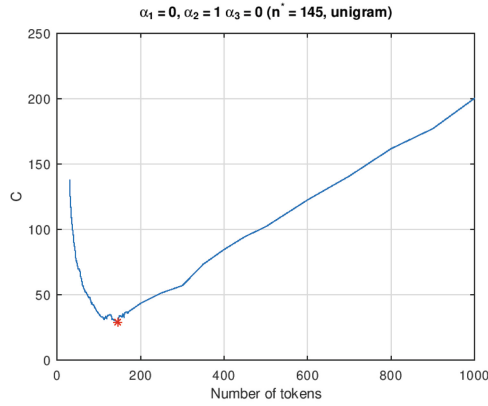
SentencePiece-Unigram						
$n(\alpha_{1,2,3})$	dev-clean	dev-other	dev-avg	test-clean	test-other	test-avg
300 (-)	7.7	20.0	13.8	8.3	20.8	14.5
30 (1, 0, 0)	7.6	20.3	13.9	7.9	20.7	14.3
145 (0, 1, 0)	8.6	20.8	14.7	9.0	21.0	15.0
1000 (0, 0, 1)	11.0	21.0	16.0	11.5	23.0	17.2
<b>61 (1, 1, 1)</b>	<b>7.2</b>	<b>19.2</b>	<b>13.2</b>	<b>7.7</b>	<b>19.6</b>	<b>13.6</b>
SentencePiece-BPE						
$n(\alpha_{1,2,3})$	dev-clean	dev-other	dev-avg	test-clean	test-other	test-avg
300 (-)	8.1	20.4	14.2	8.5	20.4	14.4
30 (1, 0, 0)	7.6	20.3	13.9	<b>7.9</b>	20.7	14.3
97 (0, 1, 0)	7.7	20.1	13.9	8.2	20.4	14.3
1000 (0, 0, 1)	7.9	20.0	13.9	8.0	20.6	14.3
<b>70 (1, 1, 1)</b>	<b>7.6</b>	<b>19.8</b>	<b>13.7</b>	8.0	<b>20.4</b>	<b>14.2</b>

of 1 to  $t_1$ ,  $t_2$  and  $t_3$  in the cost function, we get the number of tokens as  $n^* = 145$  (SentencePiece-Unigram) and  $n^* = 70$  (SentencePiece-BPE). This choice of  $n$  for SentencePiece-Unigram outperforms all the other systems resulting in an average WER of 13.2% over the dev sets and an average WER of 13.6% over the test sets. We observe similar improvement in ASR performance for SentencePiece-BPE. While the recommended  $n = 300$  tokens with ASR result in an average WERs of 14.2% and 14.4% with dev and test sets respectively, the choice of  $n^* = 70$  tokens, which correspond to  $\alpha_{1,2,3} = (1, 1, 1)$  show an improved performance of 13.7% and 14.2% WER for dev and test sets respectively. What is remarkable is that this improvement in ASR performance comes with reduced computational cost when compared to the use of recommended 300 tokens in ESPNet recipe. This set of experiments again reinforces the earlier finding that blindly increasing the number of tokens does not improve the performance of the ASR system.

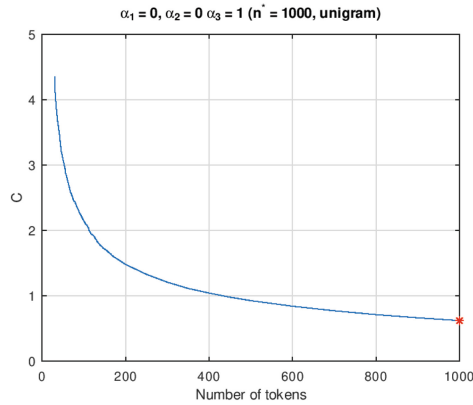
It is interesting to note that the best performance is obtained when  $\alpha_{1,2,3} = 1$ . This suggests that each of the terms  $t_1, t_2, t_3$  in the cost function has a role to play in determining the number of tokens optimal for the ASR performance. The number of tokens will change when the training text changes or the tokenizer changes and hence the cost function minimization should be performed once every time the training text or the tokenizer changes.



(a)  $\alpha_{1,2,3} = (1, 0, 0)$ ;  $n^* = 30$ .

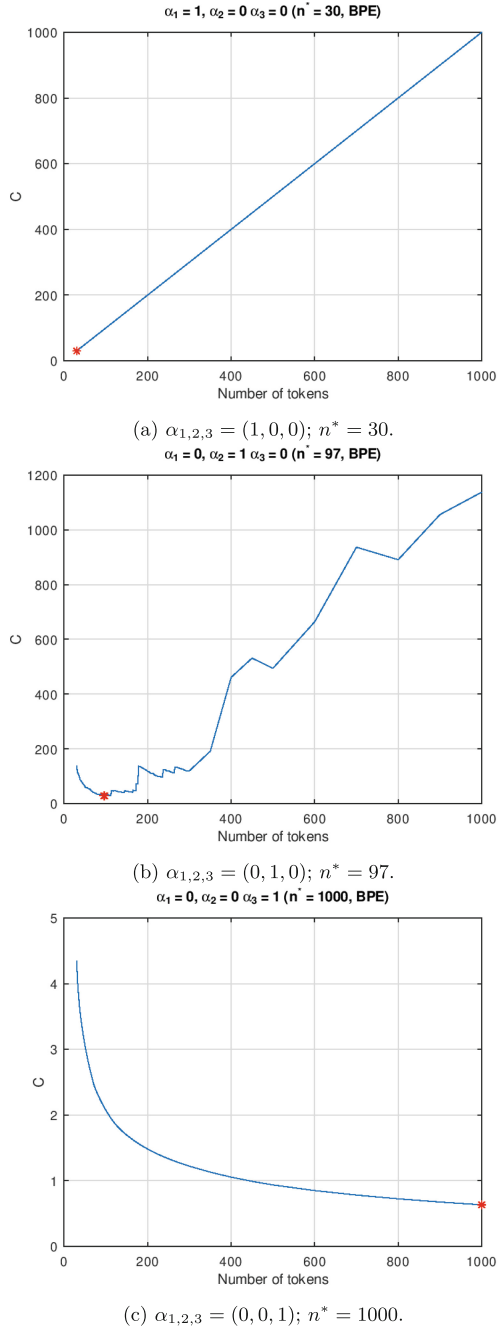


(b)  $\alpha_{1,2,3} = (0, 1, 0)$ ;  $n^* = 145$ .

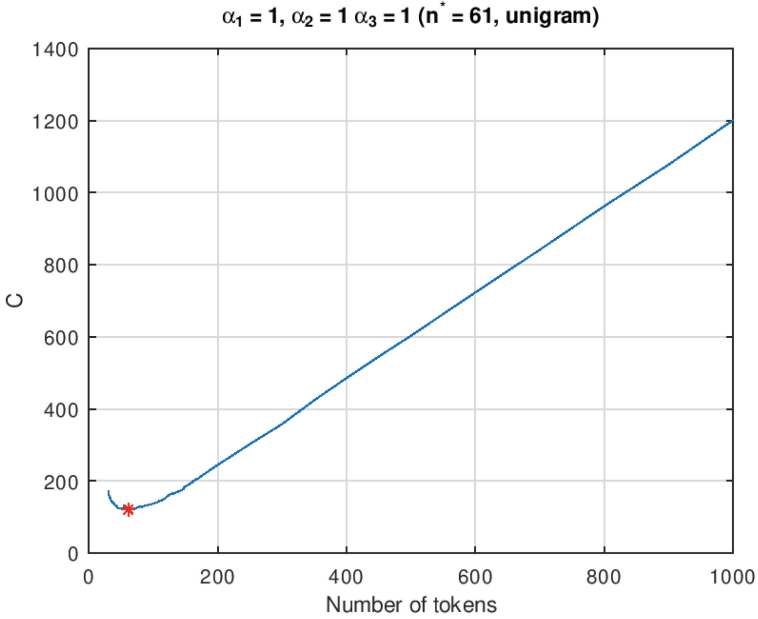


(c)  $\alpha_{1,2,3} = (0, 0, 1)$ ;  $n^* = 1000$ .

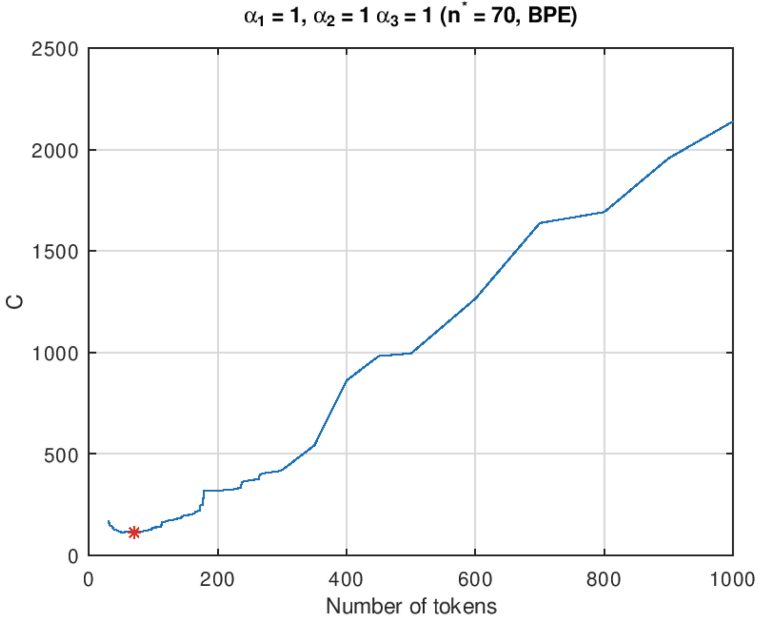
**Fig. 2.** SentencePiece-Unigram.  $n^*$  marked with a red “\*”. x-axis shows the number of tokens and y-axis the  $C$  (Color figure online)



**Fig. 3.** SentencePiece-BPE.  $n^*$  marked with a red “\*”. x-axis shows the number of tokens and y-axis the  $C$  (Color figure online)



(a) SentencePiece-Unigram.  $n^* = 61$ .



(b) SentencePiece-BPE.  $n^* = 70$ .

**Fig. 4.** Optimal number of tokens ( $n^*$ ) for  $\alpha_{1,2,3} = (1, 1, 1)$

## 5 Conclusions

In this paper we proposed a formulation based on construction of a cost function that allows for the identification of an optimal vocabulary size for the tokenizers used during the training of a speech recognition engine. The formulated cost function is based on keeping a balance between the most frequently and least frequently occurring training sub-word data to avoid bias in training as well as making sure that the number of tokens required to represent the data is minimal from the computing cost perspective. Using Librispeech 100h training set, we showed the efficacy of the approach to determining the vocabulary size of the tokenizer. In future, it would be worthwhile to look into the cost function in more detail to include other relevant factors. A robust approach to determining the values of  $\alpha$ 's would improve the performance further.

## References

1. Katare, D., Noguero, D.S., Park, S., Kourtellis, N., Janssen, M., Ding, A.Y.: Analyzing and mitigating bias for vulnerable classes: towards balanced representation in dataset. arXiv (2024)
2. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, 7–9 May 2015, Conference Track Proceedings (2015). <http://arxiv.org/abs/1412.6980>
3. Ko, T., Peddinti, V., Povey, D., Khudanpur, S.: Audio augmentation for speech recognition. In: Proceedings of the Interspeech 2015, pp. 3586–3589 (2015). <https://doi.org/10.21437/Interspeech.2015-711>
4. Kudo, T.: Subword regularization: improving neural network translation models with multiple subword candidates. In: Gurevych, I., Miyao, Y. (eds.) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 66–75. Association for Computational Linguistics, Melbourne (2018). <https://doi.org/10.18653/v1/P18-1007>
5. Kudo, T., Richardson, J.: SentencePiece: a simple and language independent subword tokenizer and detokenizer for neural text processing (2018). <https://doi.org/10.18653/v1/D18-2012>
6. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech ASR corpus: test-clean-100 (2015). <https://www.openslr.org/resources/12/test-clean.tar.gz>. Accessed 26 June 2024
7. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech ASR corpus: train-clean-100 (2015). <https://www.openslr.org/resources/12/train-clean-100.tar.gz>. Accessed 26 June 2024
8. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech ASR corpus: test-other-100 (2015). <https://www.openslr.org/resources/12/test-other.tar.gz>. Accessed 26 June 2024
9. Papadourakis, V., Mueller, M., Liu, J., Mouchtaris, A., Omologo, M.: Phonetically induced subwords for end-to-end speech recognition. In: Interspeech 2021 (2021). <https://www.amazon.science/publications/phonetically-induced-subwords-for-end-to-end-speech-recognition>



10. Park, D.S., et al.: SpecAugment: a simple data augmentation method for automatic speech recognition. In: Proceedings of the Interspeech 2019, pp. 2613–2617 (2019). <https://doi.org/10.21437/Interspeech.2019-2680>
11. Raissi, T., Beck, E., Schlüter, R., Ney, H.: Towards consistent hybrid hmm acoustic modeling. arXiv (2021)
12. Schuster, M., Nakajima, K.: Japanese and Korean voice search. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5149–5152 (2012). <https://doi.org/10.1109/ICASSP.2012.6289079>
13. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Erk, K., Smith, N.A. (eds.) Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1715–1725. Association for Computational Linguistics, Berlin (2016). <https://doi.org/10.18653/v1/P16-1162>
14. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017)
15. Watanabe, S., et al.: ESPnet: end-to-end speech processing toolkit. In: Proceedings of the Interspeech 2018, pp. 2207–2211 (2018). <https://doi.org/10.21437/Interspeech.2018-1456>
16. Xu, H., Ding, S., Watanabe, S.: Improving end-to-end speech recognition with pronunciation-assisted sub-word modeling. In: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019), pp. 7110–7114 (2019). <https://doi.org/10.1109/ICASSP.2019.8682494>
17. Zouhar, V., et al.: A formal perspective on byte-pair encoding. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Findings of the Association for Computational Linguistics (ACL 2023), pp. 598–614. Association for Computational Linguistics, Toronto (2023). <https://doi.org/10.18653/v1/2023.findings-acl.38>



# ASD-Diffusion: Anomalous Sound Detection with Diffusion Models

Fengrun Zhang<sup>✉</sup>, Xiang Xie<sup>✉</sup>, and Kai Guo<sup>✉</sup>

Beijing Institute of Technology, Beijing 100081, China  
{zhangfengrun, xiexiang, guokai}@bit.edu.cn  
<https://www.bit.edu.cn>

**Abstract.** Unsupervised Anomalous Sound Detection (ASD) aims to design a generalizable method that can be used to detect anomalies when only normal sounds are given. In this paper, Anomalous Sound Detection based on Diffusion Models (ASD-Diffusion) is proposed for ASD in real-world factories. In our pipeline, the anomalies in acoustic features are reconstructed from their noisy corrupted features into their approximate normal pattern. Secondly, a post-processing anomalies filter algorithm is proposed to detect anomalies that exhibit significant deviation from the original input after reconstruction. Furthermore, denoising diffusion implicit model is introduced to accelerate the inference speed by a longer sampling interval of the denoising process. The proposed method is innovative in the application of diffusion models as a new scheme. Experimental results on the development set of DCASE 2023 challenge task 2 outperform the baseline by 7.75%, demonstrating the effectiveness of the proposed method.

**Keywords:** anomalous sound detection · denoising diffusion probabilistic models · unsupervised learning

## 1 Introduction

The purpose of anomalous sound detection (ASD) in the industrial scene is to monitor the machine's condition by distinguishing between normal and anomalous machine-generated sounds. Detection and classification of acoustic scenes and events (DCASE) challenge and workshop is committed to advancing the field of sound event detection. Since 2020, ASD has been adopted as a new task and held every year until now in DCASE challenge [1]. Previous methods achieve better performance by using test data in development set to tune hyper-parameters of the model [2]. However, in some practical conditions, due to the diversity of operational conditions and atypical anomalies, it is challenging to collect anomalous sounds with comprehensive pattern coverage and collect anomalous data for tuning.

Considering all the above factors, a main goal of DCASE 2023 task 2 is to detect anomalous sounds when only normal sounds are given without tunable

hyper-parameters of the trained model for each machine type, which is called first-shot ASD.

A series of methods, which can be generally divided into self-supervised and unsupervised methods, have been proposed to tackle these issues. Self-supervised ASD introduces classification as an auxiliary task to calculate anomalous degree in accordance with classification confidence. However, since classification-based self-supervised approaches extremely rely on additional labels (i.e. machine ID or attribute) from metadata [3–5], effectiveness may degrade when auxiliary labels are limited or domain shifts occur [2].

Unsupervised ASD approaches minimize the negative log-likelihood or reconstruction error as the optimization objective and learn the distribution only from the acoustic features of normal sounds. Anomalies of audio are detected by the inner likelihood of the learned distribution or the reconstruction error of generated samples. A flurry of generative models have been previously explored in ASD, such as variational autoencoder (VAE) [6], generative adversarial network (GAN) [7], and normalizing flows (NF) [8]. In recent studies, denoising diffusion probabilistic model (DDPM) [9], as an emerging generative model, has attracted much attention from researchers in many fields. It has been proven that DDPMs are capable of generating samples from complex data distributions with broader pattern coverage than VAEs and GANs [10]. These properties are considered suitable for anomaly detection that lacks anomalous samples. Recent advances in computer vision also indicate that DDPM is well suited to anomaly detection tasks. Until now, DDPM has been used for anomaly detection in images. AnoDDPM [11] achieves a huge improvement over GAN-based approaches in medical image anomaly detection. DiffusionAD [12] outperforms other methods in general image anomaly detection. However, applying diffusion models to ASD remains challenging and has not been explored.

Since the high-dimensional time-frequency information in audio can be intuitively represented in the acoustic features (i.e. mel-spectrogram), employing diffusion models for anomaly detection in these acoustic features is a reasonable choice. Inspired by the works mentioned above, we propose ASD-Diffusion, a novel diffusion-based ASD approach. The main contributions of this paper can be summarized as follows:

- A diffusion-based approach to ASD. To the best of our knowledge, ASD-Diffusion is the first time that diffusion models have been applied to the field of ASD.
- A carefully designed post-processing anomalies filter (AF) algorithm, which is well suited for anomaly detection in samples reconstructed by diffusion models. Meanwhile, it can also be used for anomaly localization.
- For problems of long sampling timesteps existing in DDPM, we introduce denoising diffusion implicit model (DDIM) [13] in the inference process to accelerate sampling.

## 2 Methods

### 2.1 Diffusion Models for ASD-Diffusion

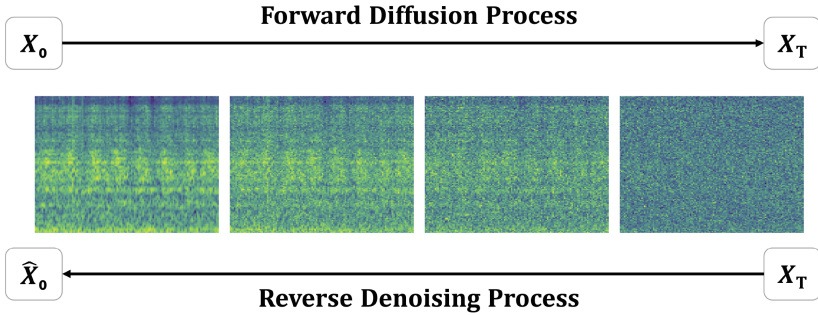


Fig. 1. Forward and reverse process of DDPM.

**DDPM.** In general, DDPM specifies a forward diffusion process and a reverse denoising process illustrated in Fig. 1. The input data are gradually disturbed by adding Gaussian noise for a few timesteps in the forward diffusion process and DDPM is guided to reconstruct target noise-free data from corrupted data in the reverse denoising process. Assume that the distribution of normal sounds is  $\phi(x)$ , the forward diffusion process is defined as

$$q(x_t | x_0) = \mathcal{N}(x_t | x_0 \sqrt{\bar{\alpha}_t}, (1 - \bar{\alpha}_t) \mathbf{I}) \quad (1)$$

$$x_t = x_0 \sqrt{\bar{\alpha}_t} + \epsilon_t \sqrt{1 - \bar{\alpha}_t}, \quad \epsilon_t \sim \mathcal{N}(0, \mathbf{I}) \quad (2)$$

where data  $x_0 \sim \phi(x)$  is transformed into noisy data  $x_t$  for  $t \in \{0, 1, \dots, T\}$  by adding noise for  $t$  timesteps to  $x_0$ . Here,  $\bar{\alpha}_t = \prod_{i=0}^t \alpha_i = \prod_{i=0}^t (1 - \beta_i)$  and  $\beta_i \in (0, 1)$  represents the noise variance schedule. This can be defined as a schedule from  $\beta_1 = 10^{-4}$  to  $\beta_T = 10^{-2}$  [9, 14, 15].

As  $x_T$  is shown in Fig. 1, the distribution of  $x_0$  is gradually disrupted and approaches Gaussian noise when  $t$  increases. A neural network  $\epsilon_\theta(x_t, t)$  is trained to predict added noise  $\epsilon$  by minimizing the training objective with mean squared error (MSE) loss:

$$\mathcal{L} = E_{t \sim [1-T], x_0 \sim q(x_0), \epsilon \sim \mathcal{N}(0, \mathbf{I})} \left( \|\epsilon - \epsilon_\theta(x_t, t)\|^2 \right) \quad (3)$$

At inference phase,  $x_{t-1}$  is reconstructed from previous step  $x_t$  in reverse process with the diffusion model  $\epsilon_\theta(x_t, t)$  according to:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \tilde{\beta}_t z \quad (4)$$

in which,  $z \sim \mathcal{N}(0, \mathbf{I})$  and  $\tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t$ .  $x_0$  is reconstructed to fit  $\phi(x)$  from  $x_t$  in a way of Markovian chain.

**DDIM.** DDIM is generalized from DDPM via a class of non-Markovian diffusion processes. In DDIM, sample  $x_{t-1}$  can be generated from sample  $x_t$  via:

$$\begin{aligned}
 x_{t-1} = & \underbrace{\sqrt{\bar{\alpha}_{t-1}} \left( \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta^{(t)}(x_t)}{\sqrt{\bar{\alpha}_t}} \right)}_{\text{“predicted } x_0\text{”}} + \\
 & \underbrace{\sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \epsilon_\theta^{(t)}(x_t)}_{\text{“direction pointing to } x_t\text{”}} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}}
 \end{aligned} \tag{5}$$

in which,  $\epsilon_t \sim \mathcal{N}(0, I)$  is standard Gaussian noise independent of  $x_t$ .

Different values of  $\sigma_t$  will lead to different generative processes. When  $\sigma_t = \sqrt{(1 - \bar{\alpha}_{t-1}) / (1 - \bar{\alpha}_t)} \sqrt{1 - \bar{\alpha}_t / \bar{\alpha}_{t-1}}$  for all  $t$ , the forward diffusion process becomes Markovian, and the generative process becomes a DDPM. When  $\sigma_t = 0$  for all  $t$ , samples are generated from latent variables with a fixed procedure (from  $x_T$  to  $x_0$ ), the process becomes DDIM. This fixed denoising procedure results in a more stable reconstruction to approach  $\phi(x)$ . The specific derivation process of Eq. 5 can be seen in [13].

In DDIM, since the forward and reverse processes are non-Markov, samples can be reconstructed with a larger sampling interval in the reverse process, saving a lot of computing resources. Meantime, the training objective is also MSE loss shown in Eq. 3, which means that there is no difference in the training process with DDPM.

## 2.2 Anomaly Detection with Diffusion Models

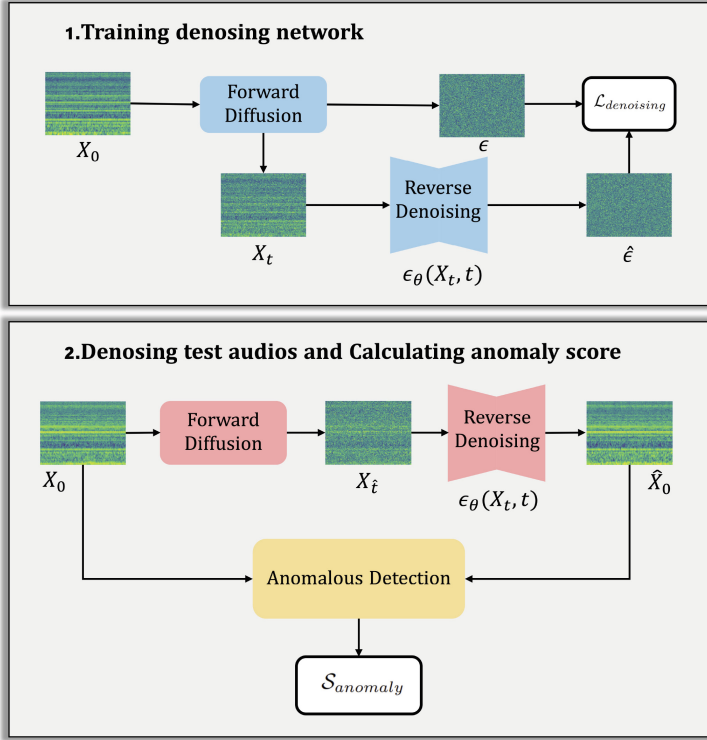
The overall architecture of ASD-Diffusion is illustrated in Fig. 2. In our work, filterbank (FBank) features extracted from waveform is chosen for anomaly detection. Since the difference between anomaly and normality can be roughly divided into frequency domain and time domain, FBank is considered suitable for anomaly detection that contains abundant time-frequency information.

During the training stage, ASD-Diffusion corrupts the FBank of normal samples  $x_0$  to  $x_t$  by adding Gaussian noise with a random parameter  $t \in \{0, 1, \dots, T\}$ , and noise scale is controlled by  $\alpha_t$  in Eq. 1. Then the denoising network  $\epsilon_\theta(x_t, t)$  predicts the added noise of  $x_t$ . The denoising loss in Eq. 3 can be simplified as:

$$\mathcal{L}_{denoising} = \|\epsilon_t - \epsilon_\theta(x_t, t)\|^2 \tag{6}$$

where  $\epsilon_\theta(x_t, t)$  learns the distribution of normal samples through minimizing  $\mathcal{L}_{denoising}$ .

During inference, since anomalous samples share distributions different from  $\phi(x)$ , an effective method is to corrupt the anomalous samples by forward diffusion and reconstruct them into their approximate normal samples in  $\phi(x)$ . Then the anomalies are detected by comparison between original and reconstructed samples. In our method, a strategy of partial diffusion is adopted. The



**Fig. 2.** The overview of ASD-Diffusion. In stage 1,  $x_t$  is obtained by adding noise  $\epsilon$  to  $x_0$  through forward diffusion. A neural network  $\epsilon_\theta(x_t, t)$  is trained to estimate the noise  $\hat{\epsilon}$  from  $x_t$ . In stage 2,  $\epsilon_\theta(x_t, t)$  reconstruct  $\hat{x}_0$  on  $x_{\hat{t}}$  after forward diffusion,  $\mathcal{S}_{anomaly}$  is then calculated by anomaly detection function.

query samples are firstly corrupted with Gaussian noise with a fixed parameter  $\hat{t}$ . The hyper-parameter  $\hat{t}$  is set to cause the anomalous regions indistinguishable from normal and retain some characteristics of the energy distribution instead of destroying the FBank into Gaussian noise totally [16]. Finally, the corrupted samples are reconstructed within  $\phi(x)$ . Anomaly detection is achieved by comparing the difference between the reconstructed and query samples. The widely used mean absolute error (MAE) calculates anomaly scores from the whole FBank, where the environmental noise is also used for calculation. Since both original and reconstructed samples contain pixel-level noise interference, which is considered redundant for anomaly detection, an AF algorithm is proposed to achieve better anomaly detection. In the experiment, we find that compared with normal sounds, most anomalous patterns appear at inappropriate frequencies, which are reflected in FBank as unreasonable energy areas with some locations. The anomaly regions are often a small part of the whole features. So we introduce AF for post-processing. The AF function proposed can filter out regions with

little difference between the reconstructed samples and the original samples. The anomaly score  $\mathcal{S}_{anomaly}$  is calculated via:

$$\mathcal{S}_{anomaly} = \frac{1}{FT} \sum_{\text{Topk}} \text{Relu}(x_{ij} - \hat{x}_{ij}) \quad (7)$$

where  $x$  represents the original sample and  $\hat{x}$  represents the reconstructed sample.  $T$  is the number of frames in the FBank and  $F$  is the number of the mel filters. The AF functions are a simple ReLU with TopK function or just a TopK function. In real practice, due to the diversity of anomalies from different machine types, we adopt multiple AF functions for different machines to get the best performance and verify the upper limit of the algorithm. Since audio commonly exhibits continuity in both spectral and temporal domains, the AF function can well filter out the anomalous regions in multiple domains by setting the appropriate K, which represents the top k largest data values.

### 3 Experiments

#### 3.1 Dataset

The experiments are carried out on the DCASE 2023 task 2 development dataset (conducted on ToyADMOS2 [17] and MIMII DG [18]) including seven machine types. Each machine type in the dataset has one section that contains data for training and testing. Each audio recording is single-channel with a duration of 6 to 18 sec and a sampling rate of 16 kHz.

Domain shift is introduced to reflect changes in the working conditions of machines. Most of the training data comes from the source domain. Each section of a machine type contains: (a) 990 clips of normal sounds in the source domain for training. (b) 10 clips of normal sounds in the target domain for training. (c) 100 clips each of normal and anomalous sounds including data from both domains for testing.

#### 3.2 Experimental Settings

The 128-dimensional FBank is extracted on 25 ms hann windows with 10 ms shifts after 1024-point Fast Fourier Transformation (FFT), then the magnitude is normalized to  $[0, 1]$ . The FBank of each audio is divided into multiple segments of  $128 \times 128$  by applying a sliding window. Since diffusion models do not require a large amount of data, the hop size of the sliding window is 128 for training and 5 for testing. The model is trained on a single NVIDIA 3090 GPU and implemented with PyTorch. As for training parameters, the U-net architecture is adopted as the denoising network. The hyper-parameters are listed in Table 1. Reverse timestep  $\hat{t}$  is chosen by experience, since the data is fully corrupted with larger  $\hat{t}$ , the reconstruction error may not be an effective detector. Similarly, if the corruption is minimal with smaller  $\hat{t}$ , it will also be useless.

**Table 1.** Hyper-parameters of diffusion, denoising U-net and training process.

Forward timestep $T$	1000
Reverse timestep $\hat{t}$	280
Noise schedule	sigmoid [15]
DDIM sampling interval	4
Channels	64
Channels multiple	(1, 2, 4, 8)
Head	4
Attention resolutions	32
Optimizer	Adam [19]
Learning rate	$1e-4$
EMA rate	0.995
Training steps	64000
Batch size	24

### 3.3 Evaluation Metrics

Area under receiver operator characteristic curve (AUC) is the most widely used metric in ASD, since anomaly detection is essentially a binary classification task. Same to the DCASE 2023 challenge, we adopt source-AUC (sAUC), target-AUC (tAUC) and partial-AUC (pAUC) as the evaluation metrics. Then the final system score is obtained by calculating the harmonic mean (hmean) for all machine domains and types. Compared with arithmetic mean, hmean is more susceptible to the influence of low values, which is adopted to evaluate the overall performance of ASD systems [20].

## 4 Results

### 4.1 Main Results

To demonstrate the effectiveness of ASD-Diffusion, other unsupervised methods are chosen for comparison. For fairness, we first compared all unsupervised methods from the top five teams in DCASE 2023, specifically those not employing machine ID or machine attribute. AE (MAHALA) [20] is the baseline provided by the challenge organizers. GAN-VAE [21] is the fourth team in the challenge. In our method, the ReLU function is used only for bearing, fan, slider, and Toy-Train, while the parameter  $K$  in the TopK function is adjusted for each machine type to achieve the best performance.

As illustrated in Table 2, ASD-Diffusion outperforms other approaches with an improvement of 7.75% and 1.04%, respectively, which means that our method ranks fourth on this dataset. The overall hmean is higher than other methods, which demonstrates that ASD-Diffusion can be better generalized to more



**Table 2.** Performance (%) comparison with unsupervised methods. Best in bold.

Machine	AE (MAHALA) [20]			GAN-VAE [21]			Ours w/o AF			Ours		
	sAUC↑	tAUC↑	pAUC↑	sAUC↑	tAUC↑	pAUC↑	sAUC↑	tAUC↑	pAUC↑	sAUC↑	tAUC↑	pAUC↑
bearing	65.16	55.28	51.37	<b>92.80</b>	<b>74.30</b>	<b>66.20</b>	79.84	66.06	54.05	83.68	70.40	54.58
fan	<b>87.10</b>	45.98	59.33	77.00	<b>73.60</b>	56.20	77.84	47.40	60.58	84.10	59.38	<b>69.05</b>
gearbox	<b>71.88</b>	<b>70.78</b>	54.34	64.40	61.50	51.80	59.78	65.14	55.79	61.38	64.98	<b>57.16</b>
slider	84.02	73.29	54.72	87.80	<b>78.80</b>	55.10	90.34	58.46	<b>61.74</b>	<b>91.98</b>	61.01	61.68
ToyCar	<b>74.53</b>	43.42	49.18	72.20	52.70	50.0	67.92	<b>56.68</b>	<b>53.26</b>	67.78	56.74	53.21
ToyTrain	55.98	42.45	48.13	61.90	45.80	48.20	60.32	54.04	50.47	<b>63.74</b>	<b>56.30</b>	<b>52.47</b>
valve	56.31	<b>51.40</b>	<b>51.08</b>	55.90	50.40	50.80	<b>56.74</b>	50.96	49.26	55.78	49.54	49.37
hmean	56.91			60.69			59.22			<b>61.32</b>		

machine types. Note that the overall tAUC is substantially superior to other methods on most machine types without any domain adaptation method, even though only 10 normal audios from the target domain are provided for training. We argue that this is due to the powerful pattern coverage ability of diffusion, that is, the distribution of the target domain can be well learned without extra domain enhancements or adaptations.

In addition, we remove the proposed post-processing algorithm (‘w/o AF’) for the ablation study. As can be seen in Table 2, the performance of ASD-Diffusion declines by 3.42%, respectively, which demonstrates the effectiveness of AF in ASD-Diffusion.

## 4.2 Comparison with Self-supervised Methods

In Table 3, our method is compared with self-supervised methods among the top three teams [22–24]. As mentioned in the introduction, self-supervised methods achieve better results due to the use of auxiliary labels for classification. In comparison, unsupervised methods are more broadly applicable and can be used even in the absence of reliable auxiliary labels. As shown in Table 3, our method is closer to the third self-supervised method [22]. Furthermore, we found that the main performance difference between self-supervised and unsupervised methods

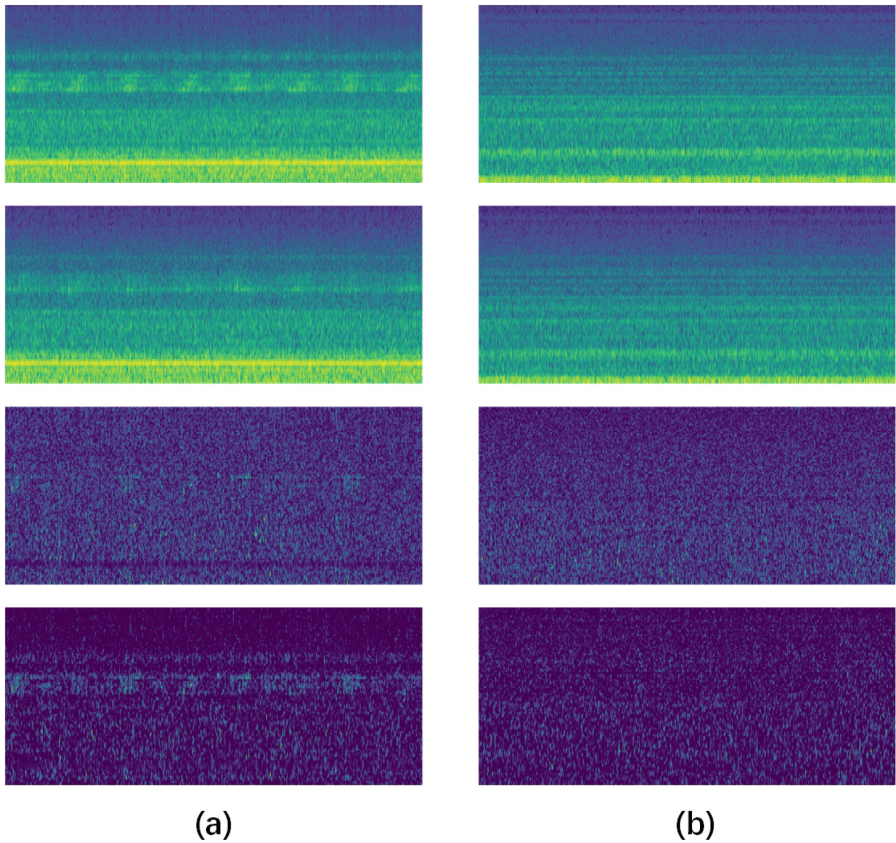
**Table 3.** Performance (%) comparison with self-supervised methods of the top three teams.

Route	Method	hmean↑
Unsupervised	GAN-VAE [21]	60.69
	Ours	61.32
Self-supervised	WSP-NFCDEE [22]	63.26
	Wav2Vec (ck2) [23]	66.56
	MDAM + knn [24]	69.25

is the valve machines, indicating that the reconstructed-based method may not well reflect the anomalous characteristics of the valve. We consider this to be the non-stationary characters of the valve sounds [25] that the reconstructed acoustic features have a large deviation from the original inputs. Therefore, even the normal sounds are reconstructed poorly, showing challenges in detecting anomalies.

### 4.3 Visualization of Anomaly Detection

The results of anomalous detection can be visualized in Fig. 3. We chose one normal and one anomalous audio from the test set of fan for comparison. The first and second rows are the original and reconstructed FBank respectively. The third and last rows are the visual detection results of MAE and the AF respectively. In the detection results, the brighter the region, the more likely it is to contain anomalies.



**Fig. 3.** Visualization of an anomalous audio (a) and a normal audio (b). First row: original FBank. Second row: reconstructed FBank. Third row: detection result of MAE. Last row: detection result with AF.

While preserving the overall energy distribution of acoustic features, details are reconstructed at a fine-grained level. Therefore, subtle anomalies in both the time domain and the frequency domain become apparent. From the comparison of the first and second rows in Fig. 3 (a), there is a clear difference in the middle channels of FBank, which may mean the existence of anomalies. However, subtracting and taking the absolute value causes possible anomalies to be slightly masked by background noise. The introduction of the AF function allows possible anomalies in the detection results to be retained and part of the noise to be removed. Whereas in Fig. 3 (b), there are virtually no conspicuous anomalous regions, which also indicates that our method can be effectively utilized for the analysis and localization of anomalies.

#### 4.4 Influence of AF Parameter

We further explore the influence of hyperparameters in AF on the performance. We choose sAUC as the evaluation metric because it is relatively better and will not be affected by domain adaptation. In other words, different AF parameters will have a more significant impact on it. We conducted experiments from two aspects: whether to use ReLU and the percentage K of the selected TopK pixels to all pixels. In our experiments, different values of K from 0 to 1 with an interval of 0.03 are tested., while the dark- or light-colored lines represent performance with or without ReLU.

As can be seen in Fig. 4, we can see that the peaks of some curves appear when K is small, which means that the abnormal locations only occupy a small part of FBank (i.e. ToyTrain with ReLU and slider with Relu). This shows that the anomaly only occurs in a shorter time and a smaller frequency range. Besides, the ReLU function in AF has a more obvious effect on some machines (i.e. fan, bearing and ToyTrain). This means that the anomalies of these machines are more likely to be manifested as missing high-energy regions on the FBank, as can be seen in Fig. 3 (a).

The result verifies the effectiveness of AF when facing various types of anomalies from different machines. However, tunable parameters for each machine are not possible in some scenes, such as first-shot ASD. We believe that AF is more useful for the analysis of anomalous characteristics. For example, when the ReLU function is more effective, the anomaly is more likely to be the lack of certain frequencies. To some extent, the size of K in TopK reveals the range of the anomaly.

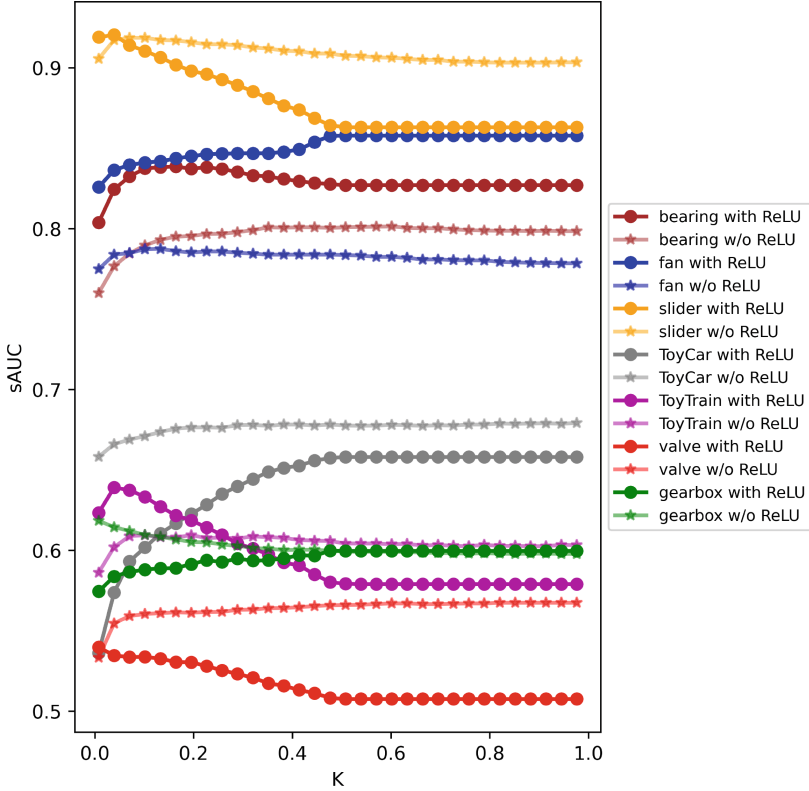


Fig. 4. Performance of the proposed method under different K or ReLU functions.

#### 4.5 Accelerating Inference Speed by DDIM

We conduct a comparative experiment between DDPM and DDIM on the bearing. It is demonstrated in Table 4 that compared with DDPM, DDIM greatly improves the inference speed with lower Real Time Factor (RTF) while maintaining better performance. Compared with DDPM, the deterministic reverse process exhibits superior consistency [13]. For the ASD task, the consistency of the reverse process is more critical than diversity. We consider this to be a difference between generative tasks and anomalous detection tasks.

Table 4. Performance over DDPM and DDIM.

Method	sAUC (%) $\uparrow$	tAUC (%) $\uparrow$	pAUC (%) $\uparrow$	RTF $\downarrow$
DDPM	79.22	67.94	<b>54.74</b>	1.17
DDIM	<b>83.68</b>	<b>70.40</b>	54.58	<b>0.29</b>

## 5 Conclusions

In this paper, we introduce diffusion models to the field of anomalous sound detection for the first time and propose a novel method named ASD-Diffusion. Our method showcases the efficacy of diffusion models for ASD. Experimental results outperform other unsupervised methods in DCASE 2023. Meanwhile, from a practical standpoint, our method achieves interpretability and localization of anomalies with the high-quality reconstruction from DDPM. In future work, we will focus on further exploring unsupervised methods and providing better anomaly localization for ASD.

## References

1. Koizumi, Y., Kawaguchi, Y., Imoto, K.: Description and discussion on DCASE2020 challenge Task2: unsupervised anomalous sound detection for machine condition monitoring. DCASE2020 Challenge, Technical report, July 2020
2. Dohi, K., Imoto, K., Noboru, H., Daisuke, N.: Description and discussion on DCASE 2023 challenge Task 2: first-shot unsupervised anomalous sound detection for machine condition monitoring. DCASE2023 Challenge, Technical report, June 2023
3. Almudévar, A., Ortega, A., Vicente, L., Miguel, A., Lleida, E.: Variational classifier for unsupervised anomalous sound detection under domain generalization. In: Proceedings of INTERSPEECH 2023, pp. 2823–2827 (2023). <https://doi.org/10.21437/Interspeech.2023-1965>
4. Hojjati, H., Armanfard, N.: Self-supervised acoustic anomaly detection via contrastive learning. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3253–3257 (2022). <https://doi.org/10.1109/ICASSP43922.2022.9746207>
5. Guan, J., Xiao, F., Liu, Y., Zhu, Q., Wang, W.: Anomalous sound detection using audio representation with machine ID based contrastive learning pretraining. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5 (2023). <https://doi.org/10.1109/ICASSP49357.2023.10096054>
6. Daniluk, P., Gozdziowski, M., Kapka, S., Kosmider, M.: Ensemble of autoencoder based systems for anomaly detection. DCASE2020 Challenge, Technical report, July 2020
7. Jiang, A., Zhang, W.-Q., Deng, Y., Fan, P., Liu, J.: Unsupervised anomaly detection and localization of machine audio: a GAN-based approach. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5 (2023). <https://doi.org/10.1109/ICASSP49357.2023.10096813>
8. Dohi, K., Endo, T., Purohit, H., Tanabe, R., Kawaguchi, Y.: Flow-based self-supervised density estimation for anomalous sound detection. In: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 336–340 (2021). <https://doi.org/10.1109/ICASSP39728.2021.9414662>
9. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Adv. Neural. Inf. Process. Syst.* **33**, 6840–6851 (2020)
10. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. *Adv. Neural. Inf. Process. Syst.* **34**, 8780–8794 (2021)

11. Wyatt, J., Leach, A., Schmon, S.M., Willcocks, C.G.: AnoDDPM: anomaly detection with denoising diffusion probabilistic models using simplex noise. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 650–656 (2022)
12. Zhang, H., Wang, Z., Wu, Z., Jiang, Y.-G.: DiffusionAD: denoising diffusion for anomaly detection. arXiv preprint [arXiv:2303.08730](https://arxiv.org/abs/2303.08730), 2023
13. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2020)
14. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning, pp. 8162–8171. PMLR (2021)
15. Jabri, A., Fleet, D., Chen, T.: Scalable adaptive computation for iterative generation. arXiv preprint [arXiv:2212.11972](https://arxiv.org/abs/2212.11972) (2022)
16. Li, A.C., Prabhudesai, M., Duggal, S., Brown, E., Pathak, D.: Your diffusion model is secretly a zero-shot classifier. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2206–2217 (2023)
17. Harada, N., Niizumi, D., Takeuchi, D., Ohishi, Y., Yasuda, M., Saito, S.: Toy-ADMOSS2: another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions. In: Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE), Barcelona, Spain, November 2021, pp. 1–5 (2021. ISBN: 978-84-09-36072-7)
18. Dohi, K., et al.: MIMII DG: sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task. In: Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022), Nancy, France, November 2022
19. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
20. Harada, N., Niizumi, D., Takeuchi, D., Ohishi, Y., Yasuda, M.: First-shot anomaly detection for machine condition monitoring: a domain generalization baseline. In: arXiv e-prints: 2303.00455 (2023)
21. Yafei, J., Jisheng, B., Siwei, H.: Unsupervised abnormal sound detection based on machine condition mixup. DCASE2023 Challenge, Technical report, June 2023
22. Jiang, A., et al.: THUEE system for first-shot unsupervised anomalous sound detection for machine condition monitoring. DCASE2023 Challenge, Technical report, June 2023
23. Lv, Z., Han, B., Chen, Z., Qian, Y., Ding, J., Liu, J.: Unsupervised anomalous detection based on unsupervised pretrained models. DCASE2023 Challenge, Technical report, June 2023
24. Jie, J.: Anomalous sound detection based on self-supervised learning. DCASE2023 Challenge, Technical report, June 2023
25. Guan, J., et al.: Transformer-based autoencoder with ID constraint for unsupervised anomalous sound detection. EURASIP J. Audio, Speech, Music Process. **2023**(1), 42 (2023)



# FCHiFi-GAN: Aggrandizing Fast Convergence with Batchwise Normalization

Ravindrakumar M. Purohit<sup>(✉)</sup>, Arushi Srivastava, and Hemant A. Patil

Speech Research Lab, DA-IICT, Gandhinagar, India  
{202321002,202215003,hemant\_patil}@daiict.ac.in

**Abstract.** In past decades, speech synthesis methods based on Deep Learning (DL) has been used to model the raw speech waveform and successfully generated natural-sounding waveforms. However, the autoregressive, non-autoregressive, flow-based, and GAN-based models that produced acoustic features using intermediate prediction of Mel spectrogram and subsequently used vocoders to generate the raw waveforms. Autoregressive and flow-based architecture requires a million of steps to achieve a realistic waveform generation capability, which is resource-intensive. In this paper, we introduce Faster Convergence HiFi-GAN (FCHiFi-GAN) exploit batch-wise normalization method to identify and capture data distribution effectively. There is a less number of architectures are capable of fast training. The proposed architecture got the real-like speech generation with only 600K steps, whether existing state-of-the-art architectures need 2.5M (16 batch size) and 580k (24 batch size) steps, respectively, for HiFi-GAN and WaveGlow. However, the number of steps varies depending on the dataset and total number of parameters. We trained the HiFi-GAN with the same setting for evaluation and got noticeably good results over the baseline architecture. To evaluate the performance of the FCHiFi-GAN, we measured and analysed generated samples with existing architecture using subjective (Mean Opinion Score (MOS)) and objective (Mel Cepstral Distance (MCD)) measures, Perceptual Evaluation of Speech Quality (PESQ), Signal-to-Noise Ratio (SNR), and Modulation Spectra Distance (MSD). FCHiFi-GAN generated samples achieve 4.42 MOS (+0.06 than baseline architecture) while reducing the required computational cost to generate high-quality samples.

**Keywords:** Generative Adversarial Networks (GANs) · High Fidelity · Batch Normalisation · Self-Supervised Learning

## 1 Introduction

Recently, the development of deep learning (DL) lead a significant impact on speech technology applications. Creating high-quality speech data that sounds-like a genuine speech is very complex due to the temporal resolution of human



speech. Mostly, the structure of speech is aperiodic on the time scale. Therefore, predicting the raw waveform is challenging because of the diverse acoustic properties, sound distortions, and variability in speech utterances. To find hidden aperiodic patterns in speech data, it is important to have high-quality data and high-end computing facilities. With lower quality speech data or limited resources, it is very difficult to preserve or identify the speech or regenerate using resource language properties of the original speech features during speech synthesis, such as speech formants,  $F_1$  to  $F_2$ . In particular, it directly impacts on model training and complicates the trade-off between quality and resource constraints.

Various signal processing strategies have been investigated, from which realistic low-dimensional speech representations can be modeled and efficiently converted back to temporal speech [1]. For example, the Griffin-Lim rules allow reconstruction of a Short-Time Fourier Transform (STFT) collection into a temporal signal [2]. However, the drawback of Griffin-Lim algorithm is that it generates noticeable *robotic artifacts* in synthesized speech. Later on, advanced representations and signal processing techniques were explored, e.g., WORLD vocoder [1]. This vocoder introduces a middleman representation for speech modeling, by incorporating functions akin to Mel spectrograms, which is coupled with a specialized signal processing algorithm to transform the intermediary representation again to the original speech. This algorithm has proven powerful impact in including textual content-to-speech synthesis. Later on, in DL, it called as Char2Wav, where features from the WORLD vocoder are modeled using bi-directional recurrent neural network [3–5]. Further, it is important to look at the fidelity of the synthesized speech, ensuring the accuracy and faithfulness of generated speech, however, it is slow to learn the patterns from the aperiodic speech, and also, ability to capture and reproduce range of speech frequencies across the spectrum, along with exactly reproducing amplitudes from soft to loud without distortion, which played a crucial role to generate high fidelity waveforms. It is also important role to maintaining the relationship between the different frequency components and spatial representation, w.r.t the multi-channel speech data, Signal-to-Noise Ratio (SNR), a reverberation of the accurate placement, and movement of the speech source contribute to the intended fidelity [6].

Generative Adversarial Networks (GANs) are state-of-the-art DL architectures for image generation using semi-supervised learning [7]. There are two networks in GANs, namely, Generator (G) and Discriminator (D). The G takes the random noise as the input and tries to create meaningful synthetic data, which looks like the real-one, such as training data. The discriminator estimates the probability on the generated sample, in terms of real *vs.* fake binary classifier. Adversarial training drives the learning process towards convergence, and finds hidden patterns from the training data through the gradient descent iteratively. The dynamic and competitive interaction between the G and the D drives progress. During training, both networks update their weights to improve their performance. They continuously adjust their parameters in an effort to



reach an equilibrium or convergence point. At this stage, the synthetic outputs generated become increasingly difficult to distinguish from real data.

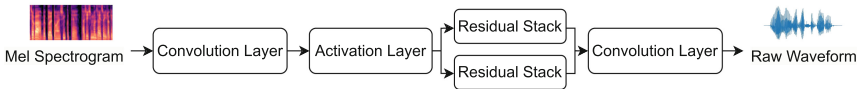
The rest of the paper is structured as follows: Sect. 2 provides an overview of the related work. Section 3 presents the architecture of the proposed FCHiFiGAN model. Section 4 gives details of the experimental setup, Sect. 5 shows the results in terms of subjective *vs.* objective measures and finally, Sect. 6 summarises the paper along with potential future directions.

## 2 Related Work

Most research on speech synthesis has focused on high-quality waveform generation. A study in WaveNet discussed an autoregressive Deep Neural Network (DNN) that can generate natural-sounding raw speech waveforms by predicting each sample based on the previous ones [8]. A WaveNet is efficient to capture the patterns of the various speech properties. Which can be written as  $p(x_{t+1} | x_1, x_2, x_3, \dots, x_t)$ , where  $t$  is the time stamp, which depends on the previous temporal properties, e.g.,  $x_{t-1}, x_{t-2}, x_{t-3}$ . Oord *et al.* used the Gated Activation Function (GAT) for creating the speech signals [8, 9]. In particular,

$$z = \tanh(W_{f,k} * x) \circ (W_{g,k} * x), \tag{1}$$

where  $\tanh$  is the hyperbolic tangent function,  $*$  denotes convolution operation, and  $\circ$  represents the element-wise product in function.  $K$  layers of residual blocks are used to enhance the ability to recognize patterns and fast convergence [10].



**Fig. 1.** Mel spectrogram for raw waveform generation using MelGAN Generator (G). After [11].

As shown in Fig. 1, The MelGAN G uses the Mel spectrogram as input rather than random noise. Residual dilated convolution has a distinct number of dilations, and fixed kernel size. WaveNet is working on the conditional probability  $p(x|h)$ , i.e.,

$$p(x_{1:T}|h) = \prod_{t=1}^T p(x_t|x_1, \dots, x_{t-1}, h). \tag{2}$$

**Global and Local Conditioning:** Global conditioning represents  $h$  (refer, Eq. (3)), which controls the output distribution for all time stamps representations. In particular,

$$z = \tanh(W_{f,k} * x + V_{f,k}^T h) \circ \sigma(W_{g,k} * x + V_{g,k}^T h), \tag{3}$$

where  $V_{*,k}$  is a trainable linear projection, and the vector  $V_{*,k}^T h$  is broadcast over the time dimension. In local conditioning, there is a two time series  $h_t$ , which potentially employs a lower sampling frequency higher than that for the speech signal, mapping it to a new time series denoted as  $y = f(h)$ , which retains the same resolution as the original speech signal. This mapping process occurs within the activation unit. In particular,

$$z = \tanh(W_{f,k} * x + V_{f,k} * y) \circ (W_{g,k} * x + V_{g,k} * y), \quad (4)$$

where  $V_{f,k} * y$  is now a  $1 \times 1$  convolution. As a substitute to utilizing a transposed convolution network, one can opt for utilizing  $V_{f,k} * h$ , with the option to replicate these values across time. In some forward-direction operations, it is prolonged due to the probabilistic approach being the major issue in the WaveNet. In 2018, ClariNet [5] flow-based speech synthesis was based on the Inverse Autoregressive Flow (IAF), where Kullback-Leibler (KL) divergence is used to make a pre-trained WaveNet [8] model that works as the teacher and generates the real-time high-fidelity (16-bits per sample) speech inference. Prenger *et al.* [12], proposed WaveGlow, a Flow-based fast and high-quality speech synthesizer, minimizing the negative likelihood or maximizing likelihood of training data. Yamamoto *et al.* [13] proposed that the Parallel WaveNet, a parallel feedforward network paradigm, 20x speeds up the speech synthesis faster than the WaveNet [8], however, this work was an initial step towards reducing both training and inference time. The log-likelihood of the spherical Gaussian is given by:

$$\log p_{\theta}(x) = \log p_{\theta}(z) + \sum_{i=1}^k \log |\det (J (f_i^{-1}))|. \quad (5)$$

where (5) penalizes the  $l_2$  norm of the transformed sample, the term is derived from the change of variables, where  $J$  is the Jacobian and log-determinant of Jacobian rewards for forward pass. The motive of WaveGlow is to multiply  $x$  terms using zero to optimize  $l_2$  and normalize the flow of the sequence. Kumar *et al.* [11] proposed the GAN architecture for the synthetic speech generation from the high-quality Mel spectrogram using fully convolution neural network. Engel *et al.* [14] generated the musical timbre by modeling the STFT magnitudes and phase angle instead of the raw waveform. Neekhara *et al.* used the GANs to learn the raw waveforms from the spectrogram. GANs learn a stochastic mapping from the perceptually informed spectrogram to the magnitude spectrogram [15]. HiFi-GAN uses conditioning to synthesize the waveform [16]. However, it is resource-consuming as well as not being able to maintain the quality under various conditions. By addressing this issue, Lee *et al.* added spectral normalization to large-scale model training 112M parameters, however, BigVGAN risks overfitting during training, especially if not properly regularized or speech data is not sufficiently diverse [17].

### 3 The FCHiFi-GAN Model

FCHiFi-GAN is built on the HiFi-GAN, which consists of one generator and two discriminators for adversarial training. There are three loss functions to guide the training process: Mel spectrogram loss, feature matching loss, and final loss. As shown in Fig. 2, FCHiFi-GAN takes input as a wave file corresponding a Mel spectrogram and generates the realistic, raw speech waveform.

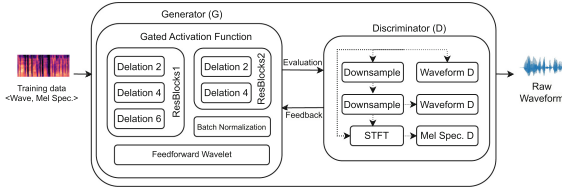


Fig. 2. Architecture of proposed FCHiFi-GAN.

#### 3.1 Generator (G)

HiFi-GAN [16] generator achieves the ability to generate the studio quality raw waveform in approximately 2.5M steps and thus, weighted normalization in the GAT extracts and generate raw speech. We noticed that the overall training is stable but includes instability at the beginning stage. However, HiFi-GAN is slow to grasp the patterns from training data due to weighted normalization, a complex speech pattern [16]. This probably occurs because weighted normalization is too slow. In such a case, the generator (G) would benefit if it grasped more information about the near-similar context around the data-like noise. However, batch-wise normalization resolves this random improvement problem [18, 19], when generated distribution and real distributions are disjoint from each other.

**In Proposed Architecture.** The G is a CNN that takes a Mel spectrogram as input and uses transposed convolutions to upsample it. This process continues up to the threshold of the generated sequence and aligns with the temporal resolution of raw waveforms. Training begins with a 1D convolution transforming the input from  $(n, 80)$  to  $(n, x)$ , followed by BatchNorm and LeakyReLU. It iteratively uses transposed convolutions (e.g.,  $(n, x)$ ) and residual ResBlocks, where the choice between ResBlocks (ResBlock1 or ResBlock2) depends on the diversity of the input data. Residual blocks ensure that the generated data output from the G has authentic and rich speech signal properties. Batchwise normalizing outputs after each iteration are shown in Algorithm 1. Finally, a 1D convolution reduces the shape to a single channel and upsampling layer, residual blocks for fine-tuning the learned features, and  $\tanh$  activation produces the output scaled between  $-1$  and  $1$ . Layer-wise architecture is shown in Table 1, where  $x$  represents *upsample\_initial\_channel*,  $k$  and  $j$  represent *kernel size* and *resblock\_dilation\_sizes*, respectively (Table 2).

**Table 1.** Architecture of FCHiFi-GAN Generator.

Layer	Input Shape	Output Shape	Parameters
Input	$(n, 80)$	$(n, 80)$	–
Conv1d	$(n, 80)$	$(n, x)$	$7 \times 80 \times x$
BatchNorm1d	$(n, x)$	$(n, x)$	$x$
LeakyReLU	$(n, x)$	$(n, x)$	–
ConvTranspose1d	$(n, x)$ (Stride = upsample_rates[i])	$(n, \frac{x}{2^{(i+1)}})$	$k \times \frac{x}{2^i} \times \frac{x}{2^{(i+1)}}$
BatchNorm1d	$(n, \frac{x}{2^{(i+1)}})$	$(n, \frac{x}{2^{(i+1)}})$	$\frac{x}{2^{(i+1)}}$
LeakyReLU	$(n, \frac{x}{2^{(i+1)}})$	$(n, \frac{x}{2^{(i+1)}})$	–
ResBlock (kernel_sizes[j], j)	$(n, \frac{x}{2^{(i+1)}})$	$(n, \frac{x}{2^{(i+1)}})$	–
Conv1d	$(n, \frac{x}{2^{(i+1)}})$	$(n, 1)$	$7 \times \frac{x}{2^{(i+1)}} \times 1$
Tanh	$(n, 1)$	$(n, 1)$	–

**Table 2.** Architecture of FCHiFi-GAN Discriminator.

Layer	Input Shape	Output Shape	Parameters
Conv1d-1	$(1, t)$	$(128, \frac{t}{2}, 1)$	$128 \times 1 \times 15$
LeakyReLU	$(128, \frac{t}{2}, 1)$	$(128, \frac{t}{2}, 1)$	–
Conv1d-2	$(128, \frac{t}{2}, 1)$	$(128, \frac{t}{4}, 1)$	$128 \times 128 \times 41$
LeakyReLU	$(128, \frac{t}{4}, 1)$	$(128, \frac{t}{4}, 1)$	–
Conv1d-3	$(128, \frac{t}{4}, 1)$	$(256, \frac{t}{8}, 1)$	$256 \times 128 \times 41$
LeakyReLU	$(256, \frac{t}{8}, 1)$	$(256, \frac{t}{8}, 1)$	–
Conv1d-4	$(256, \frac{t}{8}, 1)$	$(512, \frac{t}{32}, 1)$	$512 \times 256 \times 41$
LeakyReLU	$(512, \frac{t}{32}, 1)$	$(512, \frac{t}{32}, 1)$	–
Conv1d-5	$(512, \frac{t}{32}, 1)$	$(1024, \frac{t}{32}, 1)$	$1024 \times 512 \times 41$
LeakyReLU	$(1024, \frac{t}{32}, 1)$	$(1024, \frac{t}{32}, 1)$	–
Conv1d-6	$(1024, \frac{t}{32}, 1)$	$(1024, \frac{t}{32}, 1)$	$1024 \times 1024 \times 41$
LeakyReLU	$(1024, \frac{t}{32}, 1)$	$(1024, \frac{t}{32}, 1)$	–
Conv1d-7	$(1024, \frac{t}{32}, 1)$	$(1024, \frac{t}{32}, 1)$	$1024 \times 1024 \times 5$
LeakyReLU	$(1024, \frac{t}{32}, 1)$	$(1024, \frac{t}{32}, 1)$	–
Conv1d-8	$(1024, \frac{t}{32}, 1)$	$(1, \frac{t}{32}, 1)$	$1 \times 1024 \times 3$

**Normalization Method.** Batch normalization is a technique to accelerate training in DNN. Batch-wise normalization is implemented using the Batch-Norm2d from PyTorch [20]. During the training, FCHiFi-GAN was used to normalize each layer’s sample input, which helps to stabilize the training, and allow it to grasp the patterns from input data, as shown in Eq. (6). However, this technique is applied before and after each convolution layer and before the activation function in the G architecture in order to reduce heavy calculations during training. In particular,

$$\text{Batch Normalization}(\xi) = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta, \quad (6)$$

where  $x$  represents the input to the layer,  $\mu$  and  $\sigma$  are the mean and standard deviation of the batch, respectively,  $\gamma$  and  $\beta$  is the scale parameter and shift parameter, respectively, and  $\epsilon$  is a minor constant to avoid division by zero.

### 3.2 Discriminator (D)

In GANs, the D does the binary classification of real *vs.* fake. In HiFi-GAN, D is made MPS (Multi Period Discriminator) and MSD (Multi Scale Sub-discriminator) using existing HiFi-GAN architecture [11]. MPD (refer Fig. 3-(b)) uses multiple discriminators to discriminate the generated samples. It captures the repetitive and overlapped aperiodic patterns of the speech. MPD uses weighted normalization to do fast model training by avoiding issues [21], e.g., mode collapse and equilibrium; it also can generalize well to diverse speech patterns while reducing the computational overhead typically associated with adversarial training. MSD (refer, Fig. 3-(1)) maps the input Mel spectrogram and generated waveform using  $l_1$  loss. MSD contains three sub-discriminators, e.g., generated speech ( $x$ ),  $*2x$ , and  $*4x$ , where spectral normalization is used in the first discriminator to stabilize the Lipschitz constant and helps to achieve reliable updates during training. The second and third sub-discriminators used weighted normalization to speed up training by reparameterizing the weights, ensuring consistent learning in pooled scales [21].

---

#### Algorithm 1. Proposed FCHiFi-GAN Generator Algorithm

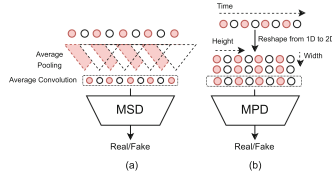
---

```

1: Input:  $x$  (Input Features), Number of kernels ( $k$ )
2: Output:  $y$  (Batch tensor  $Z \in \mathbb{R}^{B \times F \times L}$  for each step)
3:  $x \leftarrow \text{BatchNorm}(\text{Conv1d}(x))$ 
4: for  $i \leftarrow 0$  to  $k - 1$  do
5:    $x \leftarrow \text{BatchNorm}(\text{LReLU}(x, \text{LReLU}(\text{slope}(\alpha))))$ 
6:    $x \leftarrow \text{ConvTranspose1d}(\text{ups}[i])(x)$ 
7:    $xs \leftarrow \text{None}$ 
8:   for  $j \leftarrow 0$  to  $k - 1$  do
9:     if  $xs$  is None then
10:       $xs \leftarrow \text{ResBlocks}[i \times k + j](x)$ 
11:     else
12:       $xs \leftarrow xs + \text{ResBlocks}[i \times k + j](x)$ 
13:     end if
14:   end for
15:    $x \leftarrow \text{BatchNorm}(xs/k)$ 
16: end for
17:  $x \leftarrow \text{BatchNorm}(\text{LReLU}(x, \alpha))$ 
18:  $x \leftarrow \text{BatchNorm}(\text{Conv1d}(x))$ 
19:  $y \leftarrow \text{Tanh}(x)$ 
20: return  $y$ 

```

---



**Fig. 3.** Architecture of Discriminator (D) model: (a) Multi-Scale Discriminator (MSD), (b) Multi-Period Discriminator (MPD). After [16].

**Architecture.** The discriminator architecture in the FCHiFi-GAN is multi-scale and periodic [11, 22]. It has the ability to assess the diverse features of the speech signals, which is implemented over several components for speech signal evaluation. First, the periodic D processes the one-dimension tensor, which takes the input of the raw speech waveform generated by the G. The sequence of the convolution layers increases with the channels and incorporates the leaky ReLU activation function for the specified slope. At the final convolution layer, single channel output. MPD extends the discriminator to different periodicities and classifies real and generated speech signals. A spectral D is designed with a 1D input tensor backed with several convolution layers. A MSD was introduced to find the variation in the scale and employ the mean pooling layers to process the signal classification before passing through the spectral, D.

### 3.3 Dataset Used

We trained the FCHiFi-GAN model on the [LJ Speech](#) [23] and [CSTR VCTK](#) [24] datasets. Both are available under a public domain and ODC-By v1.0 license, respectively. LJ Speech contains 13,100 mono-channel samples, all sampled at 22050 Hz. The dataset contains the wave files and the corresponding transcripts. Thus, transcripts are stored in transcripts.csv with fields, such as ID (matching.wav file names), and transcription (UTF-8 words spoken). The VCTK corpus contains samples from 110 English native speakers, where each speaker reads about 400 sentences, which are taken from the rainbow passage. Each sample is quantized with 16-bits, and the sampling rate is 48 kHz. The total speech data is approximately 44 h.

### 3.4 Training Methodology

The GAN training works in the adversarial paradigm. The G tries to create synthetic speech, and the discriminator are conditioned to discriminate between authentic and synthesized speech. The training pipeline includes the distributed multi-GPU training using PyTorch’s distributed data parallel and distributed samples, AdamW optimizers, and learning rate to deal with the critical part of

the feature matching. Pytorch uses the data loader and Mel dataset classes to update the D sequentially. The different loss functions, such as the adversarial loss of the network, feature loss, and L1 loss, match the spectrogram for better evaluation. Continuous validation examinations help to improve the performance of the model. In the multi-GPU training paradigm, parallelism divides the workload into the available GPUs. The goal is to enhance the G's ability to generate high-quality synthetic data and refine the D's skill in discriminating between the real and synthetic samples, as shown in the Eq. (7); i.e.,

$$\min_G \max_D V(D, G) = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]. \quad (7)$$

In simpler terms, FCHiFi-GAN training works on the two-player minimax game [7], where the  $G$  is trained to minimize Eq. (7), while the  $D$  is trained to maximize it. This adversarial process leads the  $G$  to produce more accurate, realistic data as epochs increase.

**Training Loss Functions.** The essential goal of GANs is to generate artificial data that closely resembles real information. The characteristics of GANs accommodate critical components: the loss functions of  $G$  and  $D$ . The loss of  $G$  quantifies how effectively the  $G$  is deceiving the  $D$ , with the aim of reducing the computational cost. Mathematically,

$$G_{\text{loss}} = -\frac{1}{2} \mathbb{E}_{\mathbf{z}} [\log D(G(\mathbf{z}))], \quad (8)$$

where  $G$  and  $\mathbf{z}$  is the  $G$  and random noise vector, respectively,  $D$  is the discriminator. Conversely, loss of  $D$  measures the ability of the discriminator to differentiate between actual and generated samples. It is formulated as:

$$D_{\text{loss}} = -\frac{1}{2} \mathbb{E}_{\mathbf{x}} [\log D(\mathbf{x})] - \frac{1}{2} \mathbb{E}_{\mathbf{z}} [\log(1 - D(G(\mathbf{z})))] , \quad (9)$$

where  $\mathbf{x}$  is a real sample, and  $D(\mathbf{x})$  is the  $D$ 's output for a real sample. The overall training objective is to achieve a Nash equilibrium or convergence point [7]. In the context of the FC-HiFi GAN, we kept the loss function same as the conventional HiFi-GAN, for better comparison:

$$L_{Adv}(G; D) = E_s [(D(G(s)) - 1)^2]. \quad (10)$$

The FCHiFi-GAN uses the three loss functions to better reconstruct the speech waveform. Mel spectrogram loss, feature matching loss, and final loss.

**(1) Mel spectrogram Loss:** Mel spectrogram loss helps to improve the training performance and fidelity of the generated speech. Isola *et al.* proposed reconstructing the GAN model's loss to generate a realistic result [25]. Also, this loss function uses the  $L_1$  distance between the expected raw waveform and the original speech as per Eq. (11); i.e.,

$$L_{Mel}(G) = E(x, s) [|\phi(x) - \phi(G(s))|_1]. \quad (11)$$

**(2) Feature Matching Loss:** This loss is used to learn the similarity matrix of the pattern in the speech. As shown in Eq. (12), D will find the difference between the real and fake samples, which will help the G for the next step of training, where the D measures the  $L_1$  distance between the ground truth sample in each feature space and the conditionally generated sample:

$$L_{FM}(G; D) = E(x, s) \left[ \sum_{i=1}^T \frac{1}{N_i} \|Di(x) - Di(G(s))\|_1 \right]. \tag{12}$$

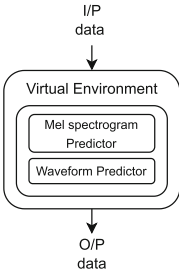
**(3) Final Loss:** Final loss describes the total loss of the Discriminator (e.g., D is divided into the two subsets, namely, MPD and MSD). Hence,

$$L_G = L_{Adv}(G; D) + \lambda_{fm}L_{FM}(G; D) + \lambda_{mel}L_{Mel}(G), \tag{13}$$

$$L_D = L_{Adv}(D; G), \tag{14}$$

where  $k$  denotes the  $k^{th}$  sub-discriminator in architecture of D.

### 3.5 Inference (I) and Simulation Setup



**Fig. 4.** Flow of inference

The inference process runs on the G model, which is loaded with pre-trained weights obtained from the training step with the lowest loss, ensuring the production of high-quality speech output. As shown in Fig. 4, the original sample is fed to the Mel spectrogram predictor, to predict the intermediate Mel spectrogram representation. Subsequently, this representation is transformed into a raw waveform by the vocoder, which produces the synthesized speech samples. As depicted in Table 3, a comparison of inference latency is presented. The FCHiFi-GAN models achieve a synthesis speed of 17.04x and 1.97x faster than real-time on the GPU (NVIDIA 1080) and CPU, respectively. Although this performance is comparatively lower than the other architectures due to the limitations of the GPU used, e.g., HiFi-GAN reaches a much higher speedup of 1,186x (NVIDIA V100), and ClariNET and BigVGAN achieve 20x (NVIDIA GTX1080Ti) and 44.72x (NVIDIA RTX 8000) faster than RTF. The dash (“-”) in the table indicates that data for specific models is unavailable.

All the experiments were conducted using Python 3.8 with Numpy 1.17.4 and Torch 1.4.0. The entire experiment was performed on a system with an Ubuntu 22.04 OS, hardware configuration included an Intel i7-12700 CPU with 32 GB RAM, and 2 \* GTX 1080 ARMOR 8 GB GPUs.



**Table 3.** The comparison of inference latency in waveform synthesis using CPU and GPU devices. All information mentioned w.r.t. RTF (Real-Time Factor).

Architectures	CPU ( $\downarrow$ )	GPU ( $\downarrow$ )
HiFi-GAN [16]	13.44x	1,186x (NVIDIA V100)
ClariNET [5]	–	20x (NVIDIA GTX1080Ti)
BigVGAN [17]	–	44.72x (NVIDIA RTX 8000)
FCHiFi-GAN (Proposed)	1.97x	17.04x (NVIDIA GTX1080)

## 4 Experiment Setup

### 4.1 Model Details

A FCHiFi-GAN model is a series of residual blocks with dilated convolutions for effective feature extraction. Two residual blocks are used, namely, ResBlock1 and ResBlock2, to define the DNN. Resblocks help remove the vanishing gradient problem from the GANs, which helps model to recognize the hidden patterns from the speech data, where ResBlock1 and ResBlock2 have 9 and 4 dilation layers, respectively. In the generator part of the GANs, it transforms the random noise features to the voice features. ‘conv\_pre,’ ‘self.ups,’ ‘self.resblocks,’ ‘conv\_post,’ and ‘forward’ methods are used to get the final transforms of the synthetic speech. Each convolution is later followed by the ‘nn.BatchNorm’ layers. The final convolution layer uses the weight normalization method followed by the ReLU activation function. As a baseline of our system, we used batch-wise normalization in the architecture of the G model, especially a neural network G of GAN [26], which works to generate the synthetic data from the training dataset using semi-supervised learning. *ResBlock1* contains 14 layers of the dilated residual convolution blocks in the G network with the three dilated cycles. The *ResBlock2* has the six layers of the dilated residual convolution blocks (Dilated ResBlocks) with the two-cycle of the deletion. The network was trained using 600K steps with the Gaussian distribution. The learning rate taken for the entire experiment is 0.001.

### 4.2 Performance Metrics Used

To evaluate the quality of generated samples’ quality using subjective and objective measures (refer Table 4), we randomly selected 50 samples from the LJ Speech dataset; these samples were inferred using the FCHiFi-GAN model. In our evaluation, each utterance pair consisted of a ground truth sample ( $x$ ) from the dataset and its corresponding generated samples ( $\hat{x}$ ) from the model, forming an utterance pair  $\langle x, \hat{x} \rangle$ . The same set of samples was used across all the

measures for meaningful performance comparison. When comparing our result to existing architectures, e.g., HiFi-GAN [16], ClariNET [5], and BigVGAN [17]. We used the pre-trained models available to create the utterance pair  $\langle x, \hat{x} \rangle$ . However, in instances, where authors did not make pre-trained models or corresponding sample pairs  $\langle x, \hat{x} \rangle$  available to the research community, we only reported accessible and derived information.

## Subjective Measures

1. **Mean Opinion Score (MOS)**: It is a subjective measure we obtained using a framework of the textbox weapp to collect user ratings [27]. In this process, a group of listeners evaluate the quality of generated speech samples on a scale from 1 to 5 based on intelligibility and quality factors. The ratings are as follows: 1 (bad), 2 (poor), 3 (fair), 4 (good), and 5 (excellent). MOS is computed as (15):

$$\text{MOS} = \frac{\sum_{n=1}^N R_n}{N}, \quad (15)$$

where  $R_n$  represents an individual rating or score, and  $N$  is the total number of ratings. We randomly selected 15 generated samples for the calculation. All participants were from non-native English regions and had no hearing disabilities. The male-to-female ratio was 3:2; the MOS test was performed in a fully controlled environment at the in-house Speech Research Lab @ DA-IICT<sup>1</sup>.

## Objective Measures

1. **Mel Cepstral Distortion (MCD)**: It is particularly relevant in the context of Mel Frequency Cepstral Coefficients (MFCC). At its essence, MCD measures the distortion between MFCC of utterance pair  $\langle x, \hat{x} \rangle$  [28], i.e.,

$$\text{MCD} = \frac{\sqrt{\sum_{t=1}^N \left( \sqrt{\sum_{i=1}^D (X(t, i) - Y(t, i))^2} \right)^2}}{N}, \quad (16)$$

where  $N$  and  $D$  are the number and dimension of feature vectors, respectively.  $X$  and  $Y$  are the reference and synthesized signals, respectively. Table 4 demonstrates the MCD score of the generated speech signal. A lower and higher MCD score indicates a closer match and higher dis-similarity between the utterance pair  $\langle x, \hat{x} \rangle$ , respectively [28].

2. **Perceptual Evaluation of Speech Quality (PESQ)**: It is a widely used tool for assessing speech quality and intelligibility [29]. It operates by mimicking the response of auditory system to speech, including its sensitivity to distortions, noise, and other impairments. In particular,

$$\text{PESQ} = 4.5 - 0.1d_{\text{sym}} - 0.0309d_{\text{asm}}. \quad (17)$$

<sup>1</sup> The speech samples are available at [Website](#).

It can influence the perceived quality of speech and psychoacoustic principles [30]. We used the utterance pair  $\langle x, \hat{x} \rangle$  in Eq. (17) to compute the PESQ score (refer Table 4). This score typically ranges from 1.0 (indicating high distortion and poor quality) to 4.5 (representing no distortion and high-quality) [29].

3. **Signal to Noise Ratio (SNR)**: It quantifies the clarity of a speech signal in the presence of noise. It represents the ratio of the power of the desired speech signal  $P_{\text{signal}}$  to the power of background noise  $P_{\text{noise}}$ , as shown in Eq. (18) with the results typically expressed in decibels (dB), i.e.,

$$\text{SNR} = 10 \log_{10} \left( \frac{P_{\text{signal}}}{P_{\text{noise}}} \right). \quad (18)$$

A higher SNR indicates that the speech signal is much stronger than the noise signal, e.g., clearer and more intelligible speech. Conversely, a lower SNR speech contains the degraded speech in terms of quality.

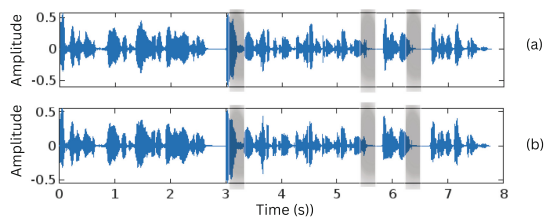
4. **Modulation Spectra Distance (MSD)**: It is a metric that measures the difference between the spectra of two speech signals. It represents the power spectrum of a speech signal transformed into the Mel frequency scale (refer to Table 4 for the results), reflecting how humans perceive pitch, giving more weight to lower frequencies [31], i.e.,

$$\text{MSD} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left( s(y)_i^t - s(y)_i^{\hat{t}} \right)^2}. \quad (19)$$

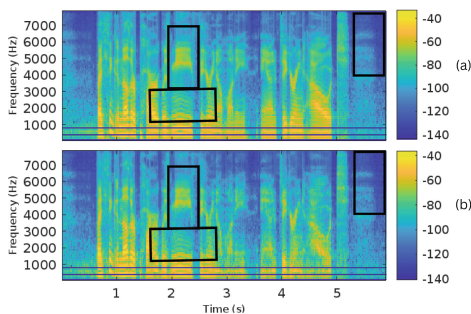
To calculate MSD, used Eq. (19) and utterance pair  $\langle x, \hat{x} \rangle$ , where lower values indicate greater similarity and higher values suggest notable differences.

## 5 Experimental Results

In Fig. 5 shows, the amplitude *vs.* time for the original speech sample and FCHiFi-GAN generated speech sample. From Fig. 5-(a) and Fig. 5-(b), it can be observed that the FCHiFi-GAN generated speech sample is nearly-similar to the original speech sample (studio-recorded speech). The visual similarity between the two plots indicates that the FCHiFi-GAN model effectively captures the temporal dynamics and nuances of the original speech signal. Rectangular boxes shows the specific regions for more detailed visual analysis in the amplitude *vs.* time plot, highlighting specific areas, where the generated signal closely follows the original speech sample.



**Fig. 5.** Amplitude vs. time plot samples were recorded from the spoken sentence in a studio setting, “and was used there with very little variation all through the sixteenth and seventeenth centuries, and indeed into the eighteenth.” Original speech signal and comparison of amp vs. acquired from the FCHiFi-GAN: (a) Original speech sample, and (b) FCHiFi-GAN generated speech sample.



**Fig. 6.** Showed Mel spectrogram comparison of the spoken sentence in ambient environment, “I think another part was me, figuring not money and figuring out” between (a) original speech sample vs. (b) generated speech sample. The rectangular box indicates that generated speech sample closely mirrors the spectral characteristics of the original speech sample.

Figure 6 illustrated the Mel spectrogram comparison between Fig. 6-(a) original speech, and Fig. 6-(b) generated Mel spectrograms. A rectangular box highlights a region of the speech where it closely generates the spectral characteristics of the original sample with the similar frequency distribution over time. However, we noticed the lower intensity in the generated sample Fig. 6-(b) as compared with the highlighted parts in the ground truth waveform sample as shown Fig. 6-(a). This decrease might be due to environmental noise of input sample.

**Table 4.** A comparison of the generated samples with the Ground Truth (GT) sample, and other existing architectures for w.r.t objective and subjective measures on LJ Speech, the Confidence Interval (CI) for the subjective evaluation Mean Opinion Score (MOS) with a 95% (A dash (‘-’) indicates that the data is not disclosed by authors or samples are not available online)

Architecture	Objective				Subjective
	PESQ (↑)	SNR (↓)	MSD [dB](↓)	MCD [dB](↓)	MOS (↑)
Ground Truth	–	–	–	–	4.58
HiFi-GAN [16]	1.92	2.95	0.26	12.44	4.36
ClariNET [5]	3.07	0.42	<b>0.24</b>	75.04	<b>4.40</b>
BigVGAN [17]	<b>3.78</b>	1.96	7.15	<b>1.06</b>	4.11
FCHiFi-GAN	2.53	<b>0.35</b>	5.70	2.41	4.21
(Proposed)	(±0.07)	(±0.11)	(±1.33)	(±0.15)	(±0.56)

## 6 Summary and Conclusions

In this paper, we presented the FCHiFi-GAN architecture in a fully convolutional neural network to accelerate the speed to achieve convergence. Batch-wise normalization is found to be positively impacting the learning capabilities. Careful parameter tuning is essential to achieve convergence with computational constraints. The resulting system synthesizes the convergence quicker than the baseline architecture, HiFi-GAN. In the future, the plan is to investigate additional techniques to enhance speech synthesis latency and quality. We look forward to setting up this model in the edge devices for real-world scenarios. Also, further research is needed to train on large-scale datasets to increase the robustness and generalizability of proposed model for unseen speakers. Further, we aim to enhance the multilingual capabilities of FCHiFi-GAN for various linguistic applications by training it on diverse language datasets, allowing the model to capture a broader linguistic spectrum. Our proposed version showcased a much higher performance w.r.t achieving the convergence speed than the traditional HiFiGAN architectures.

**Acknowledgements.** The authors sincerely thank the authorities of Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT), Gandhinagar, India, for their kind support and cooperation to carry out this research work.

## References

1. Morise, M., Yokomori, F., Ozawa, K.: WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Trans. Inf. Syst.* **99**(7), 1877–1884 (2016)
2. Griffin, D., Lim, J.: Signal estimation from modified short-time Fourier transform. *IEEE Trans. Acoust. Speech Signal Process.* (ICASSP) **32**(2), 236–243 (1984)

3. Sotelo, J., et al.: Char2Wav: end-to-end speech synthesis. In: International Conference on Learning Representations (ICLR), Toulon, France (2017)
4. Shen, J., et al.: Natural TTS synthesis by conditioning Wavenet on MEL spectrogram predictions. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4779–4783, Calgary, Alberta, Canada (2018)
5. Ping, W., Peng, K., Chen, J.: ClariNet: parallel wave generation in end-to-end text-to-speech. arXiv preprint [arXiv:1807.07281](https://arxiv.org/abs/1807.07281) (2018). Accessed 22 Feb 2024
6. Kaderavek, J.N., Justice, L.M.: Fidelity: an essential component of evidence-based practice in speech-language pathology. *Am. Speech Lang. Hear. Assoc. (ASHA)* **19**(4), 369–79 (2010)
7. Goodfellow, I., et al.: Generative adversarial networks. In: Neural Information Processing Systems (NIPS), vol. 27, Montreal, Canada (2014)
8. van den Oord, A., et al.: WaveNet: a generative model for raw audio. arXiv preprint [arXiv:1609.03499](https://arxiv.org/abs/1609.03499) (2016). Accessed 22 Feb 2024
9. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with PixelCNN decoders. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 29, pp. 4797–4805, Barcelona, Spain (2016)
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 (CVPR), pp. 770–778, Las Vegas, USA (2016)
11. Kumar, K., et al.: MelGAN: generative adversarial networks for conditional waveform synthesis. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 32, pp. 14910–14921, Vancouver, Canada (2019)
12. Prenger, R., Valle, R., Catanzaro, B.: WaveGlow: a flow-based generative network for speech synthesis. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3617–3621, Brighton, United Kingdom (2019)
13. Yamamoto, R., Song, E., Kim, J.-M.: Parallel WaveGAN: a fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6199–6203, Virtual Barcelona (2020)
14. Engel, J., Agrawal, K.K., Chen, S., Gulrajani, I., Donahue, C., Roberts, A.: GAN-Synth: adversarial neural audio synthesis. arXiv preprint [arXiv:1902.08710](https://arxiv.org/abs/1902.08710) (2019). Accessed 22 Feb 2024
15. Neekhara, P., Donahue, C., Puckette, M., Dubnov, S., McAuley, J.: Expediting TTS synthesis with adversarial vocoding. arXiv preprint [arXiv:1904.07944](https://arxiv.org/abs/1904.07944) (2019). Accessed 22 Feb 2024
16. Kong, J., Kim, J., Bae, J.: HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 17022–17033, (Virtual Conference) (2020). Accessed 22 Feb 2024
17. Lee, S., Ping, W., Ginsburg, B., Catanzaro, B., Yoon, S.: BigVGAN: a universal neural vocoder with large-scale training (2022). arXiv preprint [arXiv:2206.04658](https://arxiv.org/abs/2206.04658). Accessed 22 Feb 2024
18. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 214–223, Boston, USA (2017)
19. Bińkowski, M., et al.: High fidelity speech synthesis with adversarial networks (2019). Accessed 22 Feb 2024

20. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (ICML), pp. 448–456, Lille, France (2015)
21. Salimans, T., Kingma, D.P.: Weight normalization: a simple reparameterization to accelerate training of deep neural networks. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 29, pp. 901–909. Centre Conventions International Barcelona, Spain (2016)
22. 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 (CVPR), pp. 8798–8807, Washington D.C (2018)
23. Ito, K., Johnson, L.: The LJ Speech Dataset. <https://keithito.com/LJ-Speech-Dataset/>. Accessed 22 Feb 2024
24. Yamagishi, J., Veaux, C., MacDonald, K.: CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92) (2019). Accessed 22 Feb 2024
25. 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)
26. Cao, Y.-J., et al.: Recent advances of generative adversarial networks in computer vision. *IEEE Access* **7**, 14985–15006 (2018)
27. Streijl, R., Winkler, S., Hands, D.: Mean Opinion Score (MOS) revisited: methods and applications, limitations and alternatives. *Multimedia Syst.* **22**, 213–227 (2016)
28. Kubichek, R.: Mel-cepstral distance measure for objective speech quality assessment. In: Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing, vol. 1, pp. 125–128 (1993)
29. Rix, A.W., Beerends, J.G., Hollier, M.P., Hekstra, A.P.: Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (Cat. No. 01CH37221), vol. 2, pp. 749–752, Salt Lake City, USA (2001)
30. Fu, S.-W., Liao, C.-F., Tsao, Y., Lin, S.-D.: MetricGAN: generative adversarial networks based black-box metric scores optimization for speech enhancement. In: International Conference on Machine Learning (ICML), pp. 2031–2041, California, United States (2019)
31. Hermansky, H., Morgan, N.: RASTA processing of speech. *IEEE Trans. Speech Audio Process.* **2**(4), 578–589 (1994)



# Adaptive Enhanced Reversible Flow Model for Remote Sensing Image Super Resolution

Peishan Li, Yonghong Zhang<sup>(✉)</sup>, Junfei Wang, Guangyi Ma, and Ziwei Yuan

Nanjing University of Information Science and Technology, Nanjing, Jiangsu, China  
{zyh, jf\_wang, gyma, 202312490127}@nuist.edu.cn

**Abstract.** In recent years, convolutional neural networks (CNNs) have excelled in remote sensing image super-resolution reconstruction (RSISR) tasks, becoming the predominant algorithms in this domain. However, these models primarily leverage the dependency of high-resolution (HR) images on low-resolution (LR) counterparts during the super-resolution (SR) forward process, neglecting mutual dependencies. To address the ill-posed nature of one-to-many mappings and enhance reconstruction performance, this paper proposes Adaptive Enhanced Reversible Flow Model (AERNet), an image SR algorithm based on invertible neural networks. AERNet treats image degradation and reconstruction as invertible transformations, where LR and HR images mutually project into each other's spaces. This mutual dependency optimizes distribution mapping across LR and HR images, constraining the solution space effectively in both forward and inverse directions. Integrating a multi-path adaptive feature fusion group and a global interaction enhancement module enhances the network's adaptability, improving its capability to fuse and enhance feature information. This approach enables more accurate processing of key image details and regions. Experimental results demonstrate AERNet's superior performance on two benchmark remote sensing datasets.

**Keywords:** Deep Learning · Super-Resolution Reconstruction · Remote Sensing Images · Flow Model · Invertible Coupling

## 1 Introduction

Image Super-Resolution (SR) is a key area in computer vision focused on recovering high-resolution (HR) images from low-resolution (LR) inputs. This is crucial in both industry and academia, particularly in Remote Sensing (RS), which provides critical insights beyond visible observations. However, Single Image Super-Resolution (SISR) is challenging due to the loss of high-frequency (HF) details, resulting in multiple potential solutions for a single LR input. Obtaining quality HR images in RS is difficult, making Remote Sensing Image Super-Resolution (RSISR) essential. [1] RS images suffer more severe detail loss compared to natural images, complicating HR reconstruction. Current SISR algorithms aim within this complex solution space to identify



the correct solution. Methods are categorized into interpolation-based, reconstruction-based, [2] and learning-based approaches. [3] Interpolation methods, such as nearest-neighbor, bilinear, and bicubic, often lose HF information due to lack of external priors. Reconstruction-based methods use prior knowledge to enhance image quality but require manual parameter adjustment, slow convergence, and high computational costs, limiting their applicability in diverse RS scenarios.

With advancements in computer performance, deep learning research has rapidly developed across various application fields. Significant progress has also been made in SR algorithms based on deep neural networks. Learning-based approaches, in contrast to traditional methods, construct neural network models to establish the mapping between LR and HR remote sensing images. These methods leverage extensive LR and HR image pairs as external prior knowledge. Deep convolutional neural networks (CNNs) excel in feature representation and enable fast, end-to-end training, making them the mainstream for SISR. [4] Prominent algorithms include SRCNN [3], VDSR [5], RCAN [6], DRN [7], and SAN [9]. However, comparisons indicate that existing CNN-based SISR algorithms are nearing a bottleneck in achieving substantial quality improvements without fundamental changes. They face two primary challenges: Firstly, the ill-posed nature of one-to-many mappings in SR reconstruction limits feedforward networks in effectively constraining the solution space, leading to unrealistic reconstructions and artifacts. Secondly, relying solely on feedforward training fails to fully exploit image degradation models, restricting SISR model performance.

Flow models directly compute generation probabilities, effectively addressing RSISR by determining sample distributions in latent space without adversarial training. [10] Invertible flow models tackle one-to-many problems and model complexity. This paper proposes an adaptive-enhanced invertible flow model algorithm, integrating a multi-path adaptive feature fusion group and global interaction enhancement module to enhance feature extraction and adaptability. This enables precise capture and processing of key image details and regions, improving SR image reconstruction quality. The main contributions are as follows:

- We propose an invertible neural network based on the flow model for remote sensing image SR reconstruction. Modeling image upscaling and downscaling as inverse tasks through deliberate invertibility design, our proposed AERNet significantly alleviates the ill-posed nature of reconstructing upscaled images from downscaled LR images.
- We designed a multi-path adaptive feature fusion group to enhance the network's feature extraction and adaptability through a broader receptive field and dynamic channel weight adjustment, generating more textured and detailed information for accurate HR image reconstruction. Additionally, a global interaction enhancement module was introduced to improve feature information transmission and fusion between network layers, maintaining global information consistency and producing SR images closer to the original.
- We evaluated the proposed network on two public remote sensing benchmark datasets, NWPU-RESISC45 and AID, and compared it with 11 state-of-the-art methods. Experimental results demonstrate that our method achieves superior SR performance in terms of accuracy and visual quality.

## 2 Related Work

### 2.1 Super-Resolution of Remote Sensing Images

Given the substantial need for enhanced spatial resolution across various remote sensing (RS) applications such as scene classification, object detection, and instance segmentation, SISR has emerged as a pivotal area of research within RSI processing. Nguyen and Milanfar [11] were the first to use discrete wavelet transform (DWT) to decompose LR images, upsample the wavelet coefficients using interpolation algorithms, and subsequently apply an inverse transformation to the coefficients to produce HR images. In recent years, due to their exceptional performance and broad applicability, CNN-based methods have dominated RSISR. Lei et al. [12] introduced a SISR model for CNN-based RSI named Local-Global Combined Network (LGCNet). LGCNet utilizes an innovative ‘multi-branch’ architecture to capture multilevel representations of RSI, encompassing both local details and global context priors. Guo et al. [13] proposed a novel dense generative adversarial network (NDSRGAN) that incorporates multilevel dense networks and a matrix mean discriminator for reconstructing HR aerial images through SR. However, previous models relied too heavily on distorted LR images and often reconstructed inaccurately due to optimization-based approaches. Moreover, flow models based on image modulation have shown superior performance in image SR compared to regression-based models. Therefore, we aimed to use a generative model that can produce more detailed image semantic information.

### 2.2 Invertible Neural Network

Invertible Neural Networks (INNs) [14] feature reversible structures where input data propagated forward to produce output can be retrieved by reverse propagation, preserving information integrity. Invertible neural networks (INNs) are commonly used in generative models. Using an invertible neural network, the input originalized image  $I_x$  generates a downsized image  $I_y$  and an implicit variable  $z$ . The generation process of  $z$ , denoted as  $I_x = f_\theta(z)$ , is defined by the architecture  $f_\theta$  of the invertible neural network, where  $z$  follows a Gaussian distribution. Since the distribution of  $z$  is known, it can be omitted during image transmission. To recover the HR large-sized image, one only needs to input  $I_y$  and a randomly sampled  $z$  from this Gaussian distribution into the invertible network. This involves accessing the inverse mapping  $z = f_\theta^{-1}(I_x)$ , making inference more efficient. During the downsizing process, HF information is lost, requiring the supplementation of this lost information during the enlargement process, which can be approximated as a reversible process. INNs consist of reversible blocks, and in this study, we adopted the reversible architecture from reference [15] to construct the reversible coupling structure shown in Fig. 1. For the  $l$ -th block, the input  $g^l$  is split along the channel axis into  $g_1^l$  and  $g_2^l$ , undergoing an additive affine transformation [16]:

$$\begin{aligned} g_1^{l+1} &= g_1^l + s(g_2^l) \\ g_2^{l+1} &= g_2^l + v(g_1^{l+1}) \end{aligned} \quad (1)$$

Functions  $s(\cdot)$  and  $v(\cdot)$  represent additive coupling functions. The corresponding outputs are  $[v1, v2]$ , denoted as  $[g_1^{l+1}, g_2^{l+1}]$ . Given the output, its inverse transformation is straightforward to compute.

$$g_2^l = g_2^{l+1} - s(g_1^{l+1})$$

$$g_1^l = g_1^{l+1} - v(g_2^l)$$

Function  $t(\cdot)$  is a multiplicative coupling function. To enhance transformation capabilities, the identity branch is expanded. Functions  $s(\cdot)$ ,  $v(\cdot)$  and  $t(\cdot)$  all constitute the multi-path adaptive feature fusion group, as detailed in Sect. 3.2.

$$g_1^{l+1} = g_1^l \odot \exp(\psi(g_2^l)) + s(g_2^l)$$

$$g_2^{l+1} = g_2^l \odot \exp(t(g_1^{l+1})) + v(g_1^{l+1})$$

$$g_2^l = (g_2^{l+1} - v(g_1^{l+1})) \odot \exp(-t(g_1^{l+1}))$$

$$g_1^l = (g_1^{l+1} - s(g_2^l)) \odot \exp(-\psi(g_2^l))$$
(3)

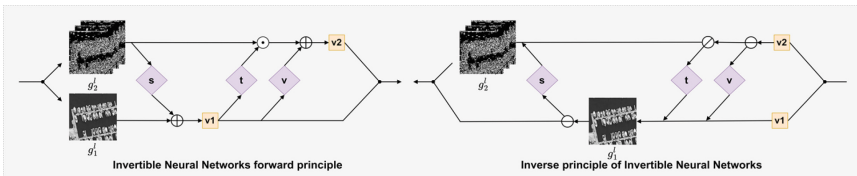


Fig. 1. Forward and inverse principles of Invertible Neural Network

### 3 Methodology

#### 3.1 Overview of AERNet

To address the ill-posed nature of upscaling, we model the distribution of lost information from HR to LR images, guided by the Nyquist-Shannon sampling theorem which indicates lost high-frequency (HF) content during downscaling. Initially, we decompose the HR image  $I_{HR}$  into low-frequency (LF)  $I_{HR}^L$  and HF components  $I_{HR}^H$  using wavelet transform. To recover these lost HF details effectively, we employ an invertible neural network during upscaling. An auxiliary latent variable  $z \sim p(z)$ , often an isotropic Gaussian distribution, models this lost information’s distribution. Therefore, retaining the LR image  $I_{LR}$  after downscaling becomes unnecessary. During upscaling,  $z$  is randomly sampled and combined with  $I_{LR}$  to reconstruct  $I_{HR}$  using the inverse mapping model.

The architecture of our proposed AERNet comprises stacked downscaling modules, where each module includes a Haar transform block, several MRBlocks (blocks based on Multi-path Adaptive Feature Fusion Groups (MAFFG)) of invertible neural networks, and a Global Interaction Enhancement Module (GIEM), illustrated in Fig. 2.

The model effectively decomposes HR image  $I_{HR}$  into a downsampled image  $I_{LR}$  and context-dependent HF information  $z$ . The Haar transform initiates each downscaling module, separating input images into LF approximation and HF coefficients in three directions [17]. These components are subsequently processed by MRBlocks. The Haar transform’s mapped features serve as input, further abstracted into LR and latent representations through a stack of MRBlocks, following the coupling layer architecture proposed in [15] by Eqs. (1–3).

Coupling layers are utilized based on two key considerations: (1) the initial decomposition of input into low-frequency (LF) and high-frequency (HF) components via Haar transform, and (2) the objective to refine LF and HF inputs through two branches of coupling layer output, ensuring a suitable appearance for low-resolution (LR) images and an independent distribution for latent HF content represented by  $g_1^l$  and  $g_2^l$  in Eq. (1). Additionally, leveraging significant findings from image scaling tasks [18] (see Fig. 1), we integrate a transformation enhancing LF with  $g_1^l$  and an augmented affine transformation for HF with  $g_2^l$ , thereby expanding model capacity. Functions  $s(\cdot)$ ,  $v(\cdot)$ , and  $t(\cdot)$ , representing multi-path adaptive feature fusion groups, are employed. The function  $t(\cdot)$  incorporates a scaling term and a central sigmoid function to stabilize computations without the  $exp(\cdot)$  function. This enhances feature extraction and adaptability with a wider receptive field and dynamic channel weight adjustment, maintaining computational efficiency and information integrity. Features then undergo processing in the GIEM module  $F_{GIEM}$ , based on residual networks integrating multi-level spatial and channel attention mechanisms. This module significantly enhances the network’s ability to handle feature information, capturing and processing critical image details and regions with enhanced precision.

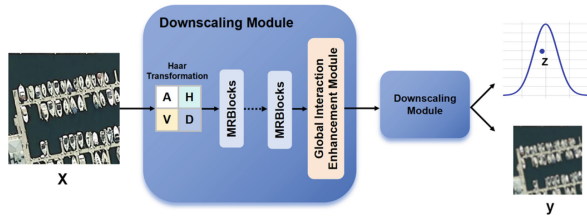


Fig. 2. Framework of the proposed AERNet

### 3.2 Multi-path Adaptive Feature Fusion Group

This paper introduces a multi-path adaptive feature fusion group that incorporates dilated convolutions and channel-domain attention within a dense connection architecture. Unlike standard convolutions, dilated convolutions [16] use a dilation rate to increase the spacing between elements in the convolution kernel, thereby addressing spatial information loss and data structure disruption caused by pooling operations. This method also expands the receptive field. If  $k$  represents the dilation rate and  $k$  the kernel size, the effective kernel size with dilation is calculated as:  $k = k + (e - 1) * (k - 1)$ .

The input data first undergoes three layers of densely connected dilated convolutions [19], each with a  $3 \times 3$  kernel size and dilation rates of 1, 2, and 4, respectively. Each layer’s output is activated by the LeakyReLU function, and the outputs of all previous layers are used as inputs for subsequent layers, ensuring feature reuse. This approach enhances the receptive field, allowing the three dense connection layers to extract ample feature information while minimizing the number of parameters in the module.

After the dense connection layers, the channel domain part of the convolutional attention module [20] is introduced. This module aggregates the spatial information of the feature maps and allocates more resources to critical feature information. The overall feature extraction network is shown in Fig. 3. The three layers of densely connected dilated convolutions extract multi-level dense feature information from the input data and input it into the attention channel domain. The channel domain calculates the scaling factors for each feature channel, which are then added to the input feature information, enabling the network to adaptively focus on key areas of feature information.

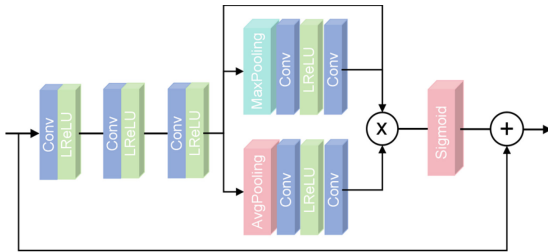


Fig. 3. Structure of Multi-path adaptive feature fusion group

### 3.3 Global Interaction Enhancement Module

To effectively enhance the transmission of feature information and better understand the correlation and importance between different regions in RSI, we designed a Global Interaction Enhancement Module (GIEM), as shown in Fig. 4. This structure is based on a residual network [6], and literature [8] has demonstrated the advantages of spatial attention mechanisms for RSI reconstruction. Consequently, this network structure also incorporates a spatial attention mechanism. Additionally, we introduced a Residual Channel Attention Mechanism (RCAM) before the spatial attention mechanism to enhance feature interaction. In Fig. 4, the parameters of Convolution 1 and Convolution 5 are denoted as  $W^1$  and  $W^5$ , respectively. Convolution 1 and Convolution 5 are used for the amplification and compression of spatial information, respectively. Convolutions 2 through 4 construct a dense connection structure, with convolution kernel parameters denoted as  $W^2$ ,  $W^3$ , and  $W^4$ . Assuming the input feature is  $x$ , the output feature of the complex network structure is  $z$ . The output feature after the first convolution is shown as follows:

$$h_1 = \varphi(W^1x) \tag{4}$$

where  $h_1$  represents the output feature of Convolution 1, and  $\varphi(\cdot)$  represents the ReLU activation function. The output feature of the residual connection can be expressed as shown as follows:

$$\begin{aligned} h_2 &= \varphi(W^2 h_1) \\ h_3 &= \varphi(W^3 \text{Con}(h_1, h_2)) \\ h_4 &= \varphi(W^4 \text{Con}(h_1, h_2, h_3)) \end{aligned} \quad (5)$$

where  $h_2$  to  $h_4$  represent the output features of the intermediate layers in the dense connection structure corresponding to their convolution kernels, and  $\text{Con}(\cdot)$  represents the feature concatenation operation.

In the RCAB, the same stage receives dense features from the previous stage and feature mappings from preceding stages. After concatenating them channel-wise, RCAB is used to enhance important features, followed by a  $1 \times 1$  convolution to fuse them. The input, with  $C$  channels, is first mapped to a vector with  $C$  channels using Global Average Pooling (GAP). Then, a  $1 \times 1$  convolution after the ReLU activation maps the vector to  $\frac{C}{r}$  channels, where  $r$  represents the reduction rate. Typically, the reduction rate is a number not less than 1. By adjusting  $r$ , the number of intermediate feature mappings can be controlled. The second  $1 \times 1$  convolution maps the vector back to  $C$  channels and uses a sigmoid function to obtain the mask  $s$ . These processes can be represented as follows:

$$s = \varphi(C_2(\delta(C_1(f_{\text{gap}}(h_4)))))) \quad (6)$$

Given that  $h_4$  is the input feature,  $f_{\text{gap}}(\cdot)$  is the GAP function,  $C_1$  and  $C_2$  are  $1 \times 1$  convolutions, and  $\delta$  and  $\varphi$  are the sigmoid and LeakyReLU functions respectively, the output can be expressed as:

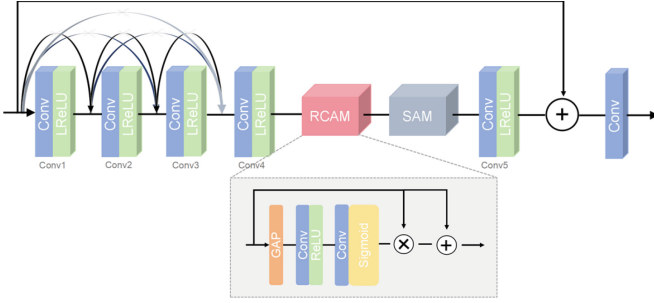
$$F_{RCAB} = s \cdot h_4 + h_4 = (1 + s) \cdot h_4 \quad (7)$$

Where  $F_{RCAB}$  represents the output feature mapping. Due to the sigmoid function, the value range of elements in the mask is  $(0,1)$ . With the addition of the skip connection, the value range of the mask is mapped to  $(1,2)$ , highlighting important features in the input while maintaining its original beneficial properties. We employ RCAB to form a fusion block to receive dense features. Specifically, we place RCAB between the connection and SAM, allowing thorough observation of feature mappings and learning various representations. The SAM[14] structure involved in the algorithm refers to the constructed spatial attention mechanism. The derivation and implementation principles of this structure can be referenced in [14]. In this section, this operation is denoted as  $F_{\text{SAM}}(\cdot)$ , thus the final output can be expressed as follows:

$$z = x + \varphi\left(W^5 F_{\text{SAM}}(F_{RCAB})\right) \quad (8)$$

### 3.4 Loss Function

The AERNet algorithm facilitates the forward process through the reversible downscaling model  $F(\cdot)$ , transforming a HR image  $I_{\text{HR}}$  into a LR image  $I_{\text{LR}}$  and data distribution  $q(I_{\text{HR}})$ . The sample cloud of HR images is denoted as  $\{I_{\text{HR}}^{(n)}\}_{n=1}^N$ . Although.



**Fig. 4.** The structure of the Global Interaction Enhancement Module (GIEM). RCAM refers to the proposed Residual Channel Attention Mechanism. SAM refers to the Spatial Attention Mechanism

The reversible downscaling task does not directly require the generated LR images to be visually appealing, we aim for them to be effective and visually pleasing. Our model's downscaling process is guided by employing the bicubic method [21]. Let  $I_{\text{LR}\downarrow\text{bic}}^{(n)}$  be the LR image corresponding to  $I_{\text{HR}}^{(n)}$  obtained by bicubic method. To ensure our model adheres to this guidance, we guide the model to produce LR images  $I_{\text{LR}}^{(n)} = F(I_{\text{HR}}^{(n)})$  that resemble  $I_{\text{LR}\downarrow\text{bic}}^{(n)}$ . The loss  $L_G$  is expressed as follows

$$L_G = \sum_{n=1}^N \ell_y \left( I_{\text{LR}\downarrow\text{bic}}^{(n)}, F \left( I_{\text{HR}}^{(n)} \right) \right) \quad (9)$$

where  $\ell_y$  represents the  $L_2$  loss.

To minimize the disparity between the reconstructed image  $I_{\text{SR}}$  and the original image  $I_{\text{HR}}$ , our algorithm utilizes the SmoothL1 loss function [22] to compute the HR reconstruction loss between  $I_{\text{SR}}$  and  $I_{\text{HR}}$ . The loss function  $L_R$  is expressed as follows,

$$L_R = \frac{1}{N} \sum_{i=1}^N \left( \frac{0.5 \|I_{\text{HR}} - I_{\text{SR}}\|^2, |I_{\text{HR}} - I_{\text{SR}}| \leq 1}{|I_{\text{HR}} - I_{\text{SR}}| - 0.5, |I_{\text{HR}} - I_{\text{SR}}| > 1} \right) \quad (10)$$

where  $N$  represents the number of images in a batch during training, and  $i$  denotes the current image being processed.

To encourage the model to capture the data distribution  $q(I_{\text{HR}})$  of the HR images, we demonstrate this using its sample cloud  $\{I_{\text{HR}}^{(n)}\}_{n=1}^N$ . The model reconstructs the SR image  $I_{\text{SR}}^{(n)}$  through  $F^{-1}(I_{\text{LR}}^{(n)}, z^{(n)})$ , where  $I_{\text{LR}}^{(n)} = F(I_{\text{HR}}^{(n)})$  is the down-scaled LR image generated by the model, and  $z^{(n)} \sim p(z)$  is a randomly sampled latent variable. To effectively traverse the cloud of real HR images  $\{I_{\text{HR}}^{(n)}\}_{n=1}^N$ , the set  $\{I_{\text{LR}}^{(n)}\}_{n=1}^N$  also forms a distribution sample cloud. We use the forward-push notation  $*$  to represent this distribution  $F_*[q(I_{\text{HR}})]$ , indicating the distribution of the transformed random variable  $F(I_{\text{HR}})$ , where the distribution  $q(I_{\text{HR}})$ ,  $I_{\text{HR}} \sim q(I_{\text{HR}})$  of the original random variable  $I_{\text{HR}}$ . Similarly, the sample cloud  $\{I_{\text{SR}}^{(n)}\}_{n=1}^N = \{F^{-1}(I_{\text{LR}}^{(n)}, z^{(n)})\}_{n=1}^N$  indicates the distribution of the SR images reconstructed by the model, denoted as  $F_*^{-1}[F_*[q(I_{\text{HR}})]p(z)]$ , since  $(I_{\text{LR}}^{(n)}, z^{(n)}) \sim F_*[q(I_{\text{HR}})] \times p(z)$  (noting independence due to the generative process of  $I_{\text{LR}}^{(n)}$  and  $z^{(n)}$ ).

The objective of distribution matching is to align the distribution reconstructed by the forward process with the target data distribution, achieved by minimizing the divergence between specific distribution metrics:

$$L_D = \ell_p(F_*^{-1}[F_*[q(I_{HR})]p(z)], q(I_{HR})) \quad (11)$$

We employ the Jensen-Shannon (JS) divergence, a metric for measuring similarity between two probability distributions, as the probability metric  $\ell_p$ . To optimize the AERNet model, we minimize the combined loss  $L_C$ , which integrates the reconstruction SR loss  $L_R$ , the LR guidance loss  $L_G$ , and the distribution alignment loss  $L_D$ :

$$L_C = \lambda_1 L_R + \lambda_2 L_G + \lambda_3 L_D \quad (12)$$

where  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are coefficients that balance the contributions of the different loss components.

## 4 Experimental Analysis

### 4.1 Dataset and Metrics

This paper utilizes two publicly available RS datasets for validation: NWPU-RESISC45 [23] and AID [24]. The datasets are partitioned into training, testing, and validation sets in a ratio of 6:3:1.

1. NWPU-RESISC45 Dataset: This RSI dataset has pixel sizes of  $256 \times 256$ . It comprises 31,500 images distributed across 45 scene categories, with 700 images per category.
2. AID Dataset: This RSI dataset has pixel sizes of  $600 \times 600$ . It includes 30 scene categories, each containing approximately 220–420 images. The dataset consists of a total of 10,000 images with resolutions ranging from 8 m to 0.5 m.

Based on HAUNet [25], we use the original images as true HR references. LR images are generated using bicubic interpolation to form HR/LR image pairs for training and evaluation. Building upon this foundation, quantitative evaluations are conducted, including 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 values, and lower SAM values indicate improved image quality. To gain a deeper understanding of the workings and behaviors of the SR network, we utilize Local Attribution Maps (LAM) [29]. LAM helps identify which input pixels significantly contribute to overall performance. For instance, in Fig. 5(b), pixels marked in red are crucial for the reconstruction process. Furthermore, Different Importance (DI) indicates the extent of pixel involvement, where higher DI reflects broader attention. Intuitively, superior network performance can be achieved by utilizing more informative pixels.



## 4.2 Implementation Details

In this study, we focused on scale factors of 2 and 4, adjusting the number of downsampling modules accordingly. The model training process was optimized using Adaptive Moment Estimation (ADAM) [30], with  $\beta_1$  and  $\beta_2$  set to 0.9 and 0.999, respectively, and  $\rho = 1e - 8$ . To enhance model stability, a series of data augmentation operations were applied to the training dataset, including rotations and flips. During training, the initial learning rate was set to  $1 \times 10^{-4}$ , with 200 iterations and a batch size of 8. The experiments used Python 3.8 and PyTorch 1.11.0, running on a server equipped with an Intel Core i9-9900K CPU, 32 GB RAM, and NVIDIA GeForce RTX 2080Ti GPU, with CUDA 11.3 and CuDNN 8.2.0 enabled.

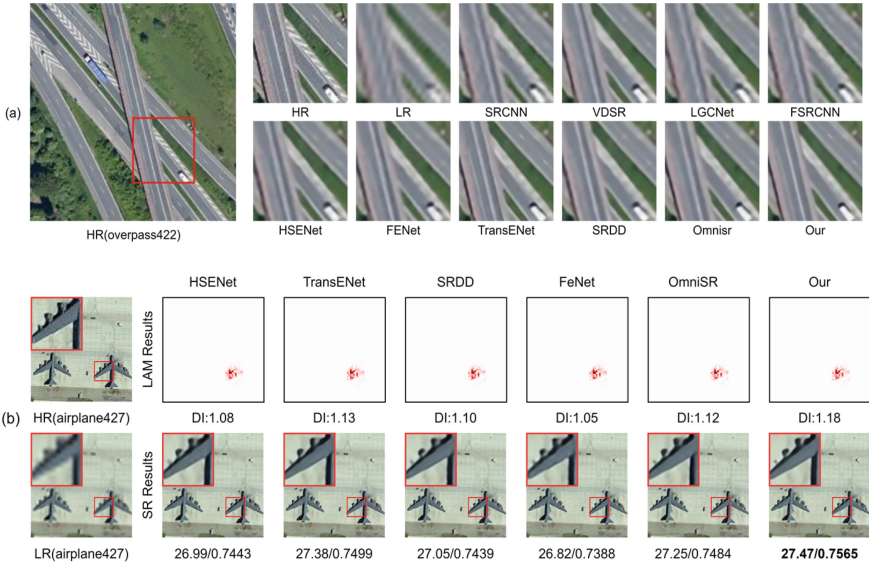
## 4.3 Evaluations with State-of-the-Art

Figure 5(a) and Fig. 5(b) present the SR results at different magnifications on a general test set, highlighting the detailed reconstruction effects of the super-resolved images through subjective visual analysis. Figure 5(a) compares the SR results on the NWPU-RESISC45 dataset at a  $4 \times$  magnification. The LR images were obtained by downsampling the test set images, and the algorithm was used alongside comparative methods for image reconstruction. Compared to other methods, the images produced by AERNet exhibit clearer shapes and edges, typically leading to outputs that are overly smooth and somewhat blurred. Specifically, the details in the locally enlarged areas (highlighted in red boxes) show that the bicubic method produces excessively blurred details, and the images from deep learning-based algorithms like FSRCNN [31] and Omnisr [32] display texture details that are not as clear and sharp as those generated by our proposed algorithm. Compared to deep learning-based remote sensing image SR algorithms such as FENet [20] and TransENet [33], our proposed algorithm provides clearer and sharper reconstructions, preserving HF information and enhancing the visual results of textures, edges, and similar content.

Figure 5(b) qualitatively compares the Local Attribution Maps (LAM) and SR results of different networks on the NWPU-RESISC45 test dataset at a  $4 \times$  magnification factor. Notably, the LAM result images of FENet [20] and HSENet [1] contain fewer informative pixels for reconstruction, leading to less detailed structural information and an inability to restore the clear edges of the airplane wings. In contrast, AERNet's attention extends along the texture directions and is more widely distributed across the entire scene, enabling it to recover richer details. Due to the distribution alignment objective, AERNet further produces clearer and more realistic images, with the visual quality and fidelity of the reconstructed HR images compared to previous state-of-the-art methods.

In addition to visual assessment, the quantitative comparison results using PSNR, SSIM, SCC, and SAM values are presented in Table 1 and Table 2. The performances ranked as best, second-best, and third-best are highlighted using red, blue, and green colors, respectively. These values represent the average PSNR, SSIM, SCC, and SAM values of the NWPU-RESISC45 and AID test sets after SR magnification by various algorithms. For the NWPU-RESISC45 dataset with a  $4 \times$  magnification factor, AERNet outperforms the classical SR method FSRCNN [31], with improvements of 0.73 dB in

PSNR and 0.0272 in SSIM. Compared to the recent novel SR method Omnisr [32], our proposed method increases PSNR and SSIM values by 0.11 dB and 0.0026, respectively. In Table 2, compared to the transformer-based SR method TransENet [33], our method improves PSNR and SSIM values by 0.31 dB and 0.0077, respectively. These results demonstrate the high performance of AERNet, highlighting its practicality in edge and fine detail recovery. MAFFG and GIEM are emphasized as practical tools for achieving more accurate recovery.

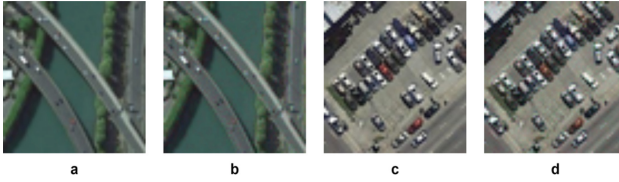


**Fig. 5.** (a) Comparisons of results using various methods on NWPU-RESISC45 datasets. (b) Evaluating  $4 \times$  Super-Resolution (SR) results (PSNR/SSIM) and LAM attribution results from different SR networks on NWPU-RESISC45. The LAM outcomes visually represent the significance of individual pixels.

We evaluated the quality of the downsampled LR images produced by AERNet. Figure 6 illustrates their visual similarity, demonstrating our accurate perception of the downsampled images, indicating that AERNet can perform SR visual tasks as effectively as Bicubic.

#### 4.4 Ablation Studies

To evaluate the impact of key modules in AERNet, we conducted ablation studies by removing specific components: 1) the Multi-Path Adaptive Feature Fusion Group (MAFFG), 2) the Global Interaction Enhancement Module (GIEM), and 3) the Combined Loss (CLoss). This study maintained consistent datasets and experimental settings across different variables. Table 3 and Fig. 7 present the quantitative and qualitative results of the ablation study conducted on the AID dataset, showcasing the best-performing results. It is important to note that removing any fundamental component leads to a significant



**Fig. 6.** Demonstration of downsampled images from the NWPU-RESISC45 and AID validation sets. The images in the left column (a, c) are downsampled using Bicubic. The images in the right column (b, d), downsampled using AERNet. They exhibit comparable visual perception.

**Table 1.** Experimental results on NWPURESISC45 dataset. *bolditalic* indicates the best performance, *italic* indicates the second best performance and **bold** indicates the third best performance.

Method	NWPURESISC45							
	$\times 2$				$\times 4$			
	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[3]	34.06	0.9202	0.6050	0.0587	28.59	0.7431	0.2073	0.1069
LGCNET[12]	34.26	0.9227	0.6080	0.0574	28.74	0.7519	0.2124	0.1052
FSRCNN[31]	34.16	0.9219	0.6116	0.0581	28.82	0.7554	0.2222	0.1044
DRN[7]	32.39	0.8878	0.4917	0.0709	27.47	0.6882	0.1249	0.1210
HSENET[1]	<b>34.62</b>	<b>0.9284</b>	<i>0.6650</i>	<b>0.0551</b>	29.20	0.7709	0.2575	0.1000
FENET[20]	34.55	0.9272	0.6340	0.0555	29.16	0.7694	0.2527	0.1006
SRDD[34]	<i>34.68</i>	<i>0.9289</i>	<b>0.6401</b>	<i>0.0546</i>	<b>29.28</b>	<b>0.7740</b>	<b>0.2666</b>	<b>0.0991</b>
OMNISR[32]	34.51	0.9266	0.5964	0.0552	<i>29.44</i>	<i>0.7800</i>	<i>0.2810</i>	<i>0.0973</i>
Ours	<b>34.89</b>	<b>0.9343</b>	<b>0.6992</b>	<b>0.0522</b>	<b>29.55</b>	<b>0.7826</b>	<b>0.2924</b>	<b>0.0960</b>

decline in evaluation metrics and visual quality. Specifically, eliminating the Combined Loss resulted in a substantial drop in PSNR (-2.09 dB), making the depiction of house lines in the image noticeably blurry. This highlights the powerful performance of the Combined Loss in enhancing SR results. Furthermore, ignoring the Global Interaction Enhancement Module not only caused a PSNR decrease of 0.14 dB on the AID dataset but also led to inadequate restoration of ground feature edges, underscoring its importance. Therefore, this study continues to adopt this approach. In summary, through comprehensive consideration, we conclude that each design element in the proposed network is indispensable for achieving satisfactory SR results.

**Table 2.** Comparison results of PSNR, SSIM, SCC and SAM on AID dataset with a scale factor of 4.

Method	AID			
	PSNR	SSIM	SCC	SAM
BICUIC	28.69	0.7334	0.1586	0.1051
SRCNN	29.76	0.7788	0.2173	0.0928
FSRCNN	29.80	0.7798	0.2074	0.0929
VDSR[5]	30.35	0.7976	0.2491	0.0871
DRN	28.48	0.7203	0.0927	0.1072
HSENet	30.44	0.8011	0.2603	0.0863
DCM[35]	<b>30.50</b>	<b>0.8032</b>	0.2685	<b>0.0857</b>
TransENet[33]	30.53	0.8048	<b>0.2655</b>	0.0853
Ours	<b>30.84</b>	<b>0.8125</b>	<b>0.2894</b>	<b>0.0826</b>

**Table 3.** Ablation study of different component combinations ( $\times 4$ )

Method			NWPURESISC45			
MAFFG	GIEM	CLoss	PSNR	SSIM	SCC	SAM
$\times$	$\checkmark$	$\checkmark$	29.48	0.7799	0.2872	0.0968
$\checkmark$	$\times$	$\checkmark$	29.41	0.7776	0.2785	0.0983
$\checkmark$	$\checkmark$	$\times$	27.46	0.6924	0.1260	0.1201
$\checkmark$	$\checkmark$	$\checkmark$	<b>29.55</b>	<b>0.7826</b>	<b>0.2924</b>	<b>0.0960</b>

**Fig. 7.** Ablation results. From top to bottom, left to right: generated LR image, original HR image, results from our proposed network, results without MAFFG, results without GIEM, results without Combined Loss.

## 5 Conclusion

In this paper, we address the challenges of one-to-many mappings and the ineffective utilization of degradation models in existing image super-resolution algorithms. Considering the complex textures and structures present in remote sensing images, we propose AERNet, an adaptive-enhanced invertible neural network-based super-resolution algorithm. AERNet integrates an invertible coupling flow computation model with remote sensing image reconstruction, combining a Multi-Path Adaptive Feature Fusion Group (MAFFG) and a Global Interaction Enhancement Module (GIEM). This combination significantly enhances feature extraction, adaptability, and the transmission, fusion, and enhancement of feature information. Using the model, high-resolution (HR) images are input to obtain super-resolved (SR) and low-resolution (LR) reconstructed images. Two loss functions are designed based on the differences between these reconstructed images and the real images. To ensure the SR images capture the distribution of HR images, the model aligns the reconstructed distribution with the target by minimizing distribution metric differences, creating a distribution matching loss function. These loss functions are weighted to form a combined loss function. Optimizing this combined loss function enhances the model's reconstruction capability in both LR and HR spaces. The model's invertible characteristics allow optimization in both forward and reverse directions. We evaluated AERNet's generalization ability on two public remote sensing datasets, comparing it with traditional bicubic interpolation and various deep learning-based super-resolution methods. Experimental results demonstrate AERNet's superiority and effectiveness, with improvements in both qualitative and quantitative performance.

## References

1. Lei, S., Shi, Z.: Hybrid-scale self-similarity exploitation for remote sensing image super-resolution [J]. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–10 (2021)
2. Yang, J., Wright, J., Huang, T.S., et al.: Image super-resolution via sparse representation [J]. *IEEE Trans. Image Process.* **19**(11), 2861–2873 (2010)
3. Dong, C., Loy, C.C., He, K., et al.: Image super-resolution using deep convolutional networks [J]. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(2), 295–307 (2015)
4. Xiao, M., et al.: Invertible image rescaling. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer International Publishing, pp. 126–144 (2020)
5. 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)
6. Zhang, Y., 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)
7. Guo, Y., 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)
8. Dosovitskiy, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)

9. Dai, T., 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)
10. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions[J]. Adv. Neural Inf. Proc. Syst. 31 (2018)
11. Nguyen, N., Milanfar, P.: A wavelet-based interpolation-restoration method for superresolution (wavelet superresolution) [J]. Circuits Syst. Signal Proc. **19**, 321–338 (2000)
12. Lei, S., Shi, Z., Zou, Z.: Super-resolution for remote sensing images via local–global combined network [J]. IEEE Geosci. Remote Sens. Lett. **14**(8), 1243–1247 (2017)
13. Guo, M., Zhang, Z., Liu, H., et al.: NDSRGAN: a novel dense generative adversarial network for real aerial imagery super-resolution reconstruction [J]. Remote Sensing **14**(7), 1574 (2022)
14. Behrmann, J., et al.: Invertible residual networks. In: International conference on machine learning. PMLR, pp. 573–582 2019
15. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp[J]. arXiv preprint [arXiv:1605.08803](https://arxiv.org/abs/1605.08803) (2016)
16. Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation[J]. arXiv preprint [arXiv:1410.8516](https://arxiv.org/abs/1410.8516) (2014)
17. Ardizzone, L., et al.: Guided image generation with conditional invertible neural networks[J]. arXiv preprint [arXiv:1907.02392](https://arxiv.org/abs/1907.02392) (2019)
18. Wang, X., et al.: Esrgan: enhanced super-resolution generative adversarial networks. In: Proceedings of the European conference on computer vision (ECCV) workshops (2018)
19. Huang, G., et al.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 4700–4708 (2017)
20. Wang, Z., Li, L., Xue, Y., et al.: FeNet: feature enhancement network for lightweight remote-sensing image super-resolution [J]. IEEE Trans. Geosci. Remote Sens. **60**, 1–12 (2022)
21. Mitchell, D.P., Netravali, A.N.: Reconstruction filters in computer-graphics[J]. ACM Siggraph Comput. Graphics **22**(4), 221–228 (1988)
22. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, 1440–1448 2015
23. Cheng, G., Han, J., Lu, X.: Remote sensing image scene classification: Benchmark and state of the art [J]. 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 [J]. IEEE Trans. Geosci. Remote Sens. **55**(7), 3965–3981 (2017)
25. Wang, J., et al.: Hybrid attention based u-shaped network for remote sensing image super-resolution [J]. 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 [J]. 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 [J]. Int. J. Remote Sens. **19**(4), 743–757 (1998)
28. Yugas, 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. Diederik, P.K.: Adam: A method for stochastic optimization [J]. (No Title) (2014)
31. 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, October 11–14, 2016, Proceedings, Part II 14. Springer International Publishing, pp. 391–407 2016.

32. Wang, H., 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
33. Lei, S., Shi, Z., Mo, W.: Transformer-based multistage enhancement for remote sensing image super-resolution [J]. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–11 (2021)
34. Maeda, S.: Image super-resolution with deep dictionary. In: European Conference on Computer Vision. Cham: Springer Nature Switzerland, 464–480 (2022)
35. Haut, J.M., Paoletti, M.E., Fernández-Beltran, R., et al.: Remote sensing single-image super-resolution based on a deep compendium model [J]. *IEEE Geosci. Remote Sens. Lett.* **16**(9), 1432–1436 (2019)



# Saliency-Based Neural Representation for Videos

Qian Cao<sup>1</sup>, Dongdong Zhang<sup>1(✉)</sup>, and Xiaolei Zhang<sup>2</sup>

<sup>1</sup> Department of Computer Science and Technology, Tongji University, Shanghai, China

{2230788, ddzhang}@tongji.edu.cn

<sup>2</sup> Department of Geotechnical Engineering, Tongji University, Shanghai, China  
Xiaolei.Zhang@tongji.edu.cn

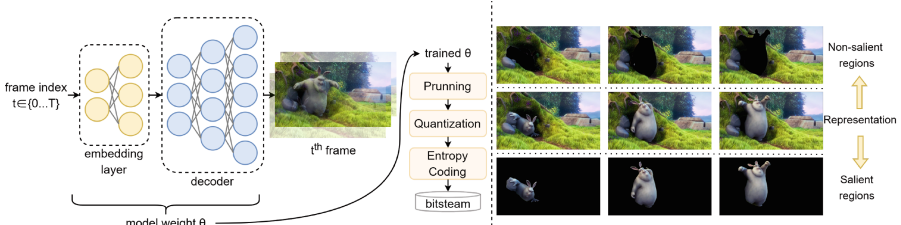
**Abstract.** Neural representation for videos (NeRV) has emerged as a promising method for video representation and compression. However, existing NeRV methods primarily focus on objective quality and overlook subjective quality. Considering the varying sensitivity of human eyes to different regions, we propose a saliency-based neural representation for videos (SNeRV). By introducing a multi-scale temporal-spatial feature grid and SNeRV blocks, we enhance the model's representation capability, improving both objective and subjective quality. Additionally, our saliency-guided training strategy enables more efficient parameter allocation, prioritizing the representation of regions of interest (ROI) for superior visual quality. On the UVG dataset, our proposed method improves objective quality by 0.3 dB to 0.5 dB PSNR compared to the state-of-the-art method and significantly enhances subjective quality, particularly in ROI areas.

**Keywords:** Implicit Neural Representations · Neural Video Compression · Saliency Detection

## 1 Introduction

The rapid growth of video content imposes significant challenges on network transmission and storage, necessitating the exploration of more efficient video coding methods. Implicit neural representations (INR) [1, 2] have demonstrated significant potential in representing and coding various signals involving image [3, 4], scene [1, 2], and videos [5, 6], presenting a promising solution to encoding tasks. The fundamental principle of INRs involves learning a function mapping coordinates to values to support the implicit reconstruction of the object. These mapping function are typically implemented using multilayer perceptrons (MLPs), which reconstruct only one point at a time, leading to slow reconstruction speeds and low quality. To address these limitations, recent Neural Representation for Videos (NeRV) methods [7–14] introduce Convolutional Neural Networks (CNNs) to reconstruct entire frame images at once, achieving higher reconstruction quality and faster decoding speeds.





**Fig. 1.** (Left) High-level diagram of NeRV methods and video compression pipeline. (Right) Our proposed SNeRV decomposes the video representation into salient and non-salient regions.

As shown in Fig. 1(left), NeRV methods typically involve an embedding layers to encode the time index  $t$ , followed by a decoder to generate frames. For video compression tasks, NeRV methods directly learn the implicit representation of video data using neural networks, eliminating the need for explicit storage of each frame. Instead, network parameters serve as compressed data, achieving video compression through model compression methods including pruning, quantization, and entropy coding.

Despite significant advancements made by these methods, they often consider only objective quality and overlook a crucial aspect of the human visual system (HVS): varying visual sensitivities across different regions within video scenes. Existing NeRV methods typically employ uniform loss functions, treating all regions within a video equally important, thus inadequately addressing the human eye’s preference for salient regions. In traditional video coding, rate control techniques allocate more bits to regions of interest (ROI) based on saliency analysis. Inspired by this, we aim to guide neural representations’ preferences, by distinguishing salient regions (typically foreground areas) from non-salient ones (typically background) and assigning different loss weights accordingly, as illustrated in Fig. 1(right). This approach achieves more efficient parameter allocation by prioritizing regions more sensitive to the human eye, resulting in better subjective visual quality.

To further improve both subjective and objective quality, we design modules for the embedding layer and decoder respectively to enhance the network’s representational capacity. In the aspect of embedding layers, although methods like FFNeRV [13] surpass Fourier-style positional embedding [7–9] and content-based feature embedding methods [11, 12] in performance, they predominantly focus on modeling the temporal dimension, neglecting the spatial dimension’s features. To address this, we introduce a multi-scale feature grid approach, covering various temporal and spatial scales, to provide a richer feature embedding. For the decoder, a challenge of NeRV methods is to enhance the model’s representational capacity within a limited parameter budget. We adopt the design [14] using bilinear interpolation for upsampling followed by deep convolutional networks. Based on advanced lightweight structure designs [15–17] and reparameterization

techniques [18], we design the SNeRV block for decoder to enhance the model’s representational capacity further.

In this paper, we propose Saliency-based Neural Representation for Videos (SNeRV), and present three contributions outlined as follows:

- We propose a saliency-guided training strategy, making the video representation more inclined towards region of interest, thus achieving better subjective visual quality.
- We design multi-scale temporal-spatial feature grids and SNeRV block for the embedding layer and decoder respectively, thereby improving the network’s representation capability.
- Through experiments, we demonstrate that our SNeRV outperforms existing NeRV methods. It exceeds the baseline model by 0.3 to 0.5 in objective metrics (PSNR), and exhibits higher reconstruction quality especially in regions of interest, resulting in better overall visual performance.

## 2 Related Work

### 2.1 Video Compression

Video compression is crucial in computer vision and multimedia processing. Traditional techniques like H.264/AVC [19] and H.265/HEVC [20] have dominated for many years. Recently, some studies have replaced local modules of traditional pipelines (e.g., motion prediction and compensation, transform coding, and entropy coding) with learning-based models. Further, DVC [21] replaces all modules with neural networks, proposing an end-to-end video encoding framework. Many methods follow this architecture and introduce optimizations, such as replacing predictive coding with conditional coding [22] and incorporating image domain operations into the feature domain [23]. However, these methods have a complex pipeline and slow decoding speeds, limiting real-time applications. INRs offer a new approach to video encoding by overfitting a small neural network to the signal and compressing it into a bitstream, achieving faster decoding speeds through a simple architecture.

### 2.2 Implicit Neural Representation

Implicit Neural Representations (INRs) are utilized for complex natural signal representation, including images [3, 4], videos [5, 6], and voxels [1, 2]. Typically, these methods employ a network (usually an MLP) to fit an implicit function mapping the coordinates to their values. Recently, NeRV [7] introduces frame-based INRs for video tasks, significantly enhancing reconstruction performance and achieving high-speed decoding. NeRV utilizes CNNs to map one-dimensional frame coordinates to entire frames, showing promise in denoising, frame interpolation, inpainting, super-resolution, and video compression tasks. Many methods [8–14] have enhanced embedding layers and decoders based on NeRV. For example, HNeRV [11] and DNeRV use encoders for content-related

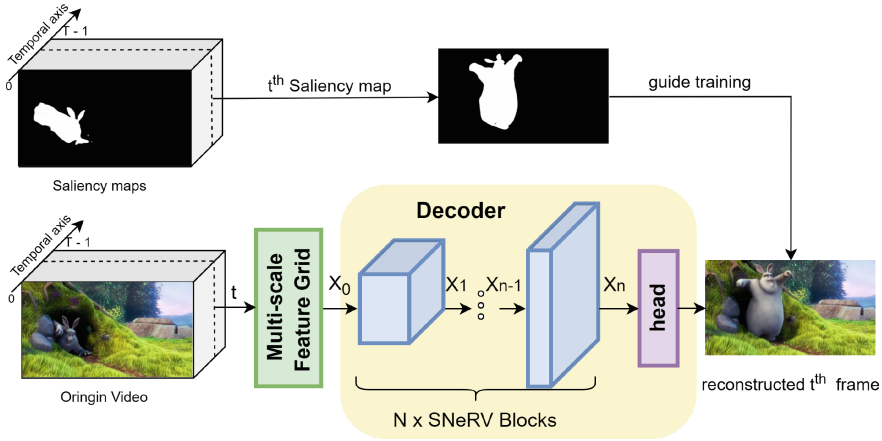


Fig. 2. The framework of SNeRV.

embeddings, while FFNeRV [13] employs feature grids, surpassing Fourier-style position encoding. We adopt FFNeRV’s [13] grid-based embedding method, integrating designs inspired by feature grids in image representation task [24, 25], proposing a multi-scale temporal-spatial feature grid. As for decoders, while most methods [7–13] use shallow CNNs and pixel shuffle layers, HiNeRV [14] introduces interpolation followed by deep CNNs structures for superior performance. Inspired by these, We integrate lightweight network structures [15–17] and reparameterization techniques [18], and propose the SNeRV block to improve decoder.

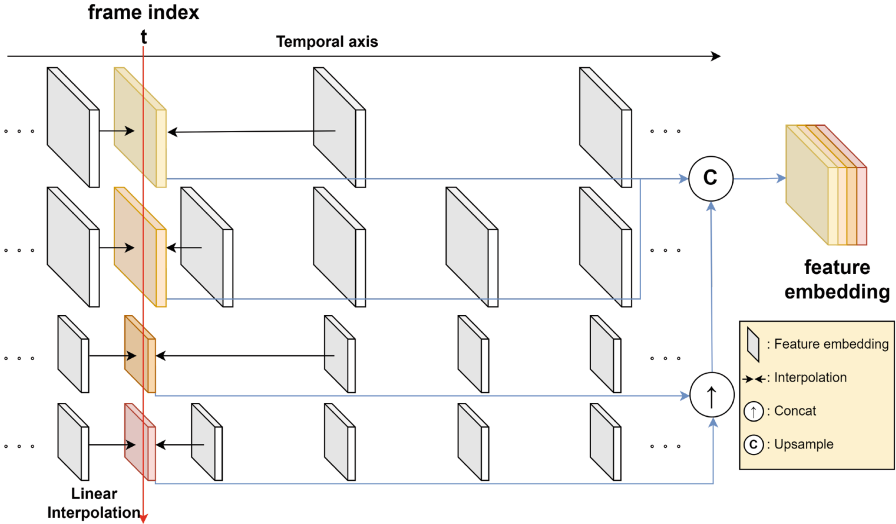
### 2.3 Video Salient Object Detection

Video Salient Object Detection is an important task aimed at identifying regions of videos that most attract human visual attention. Common methods for video salient object detection (VSOD) often utilize Long Short-Term Memory networks(LSTM) [26] or attention modules [27]. In video representation tasks, existing methods often overlook the varying sensitivity of human eyes to different regions. We propose to incorporate saliency detection to guide the network representation. In this paper, we utilize pre-trained saliency detection models [27] to obtain saliency maps for guiding the training process.

## 3 Method

### 3.1 Overview

The framework of our proposed SNeRV is shown in Fig. 2. We consider the video representation where a neural network maps the frame index to the image of that frame, and further applies it to video compression. Section 3.2 provides



**Fig. 3.** Structure of Multi-Scale Temporal-Spatial Feature Grid. For a given temporal coordinate  $t$  (indicated by the red arrow), the corresponding feature embeddings are interpolated from different feature grids. These embeddings are then upsampled to the same spatial resolution and concatenated to form the final feature embedding. (Color figure online)

a detailed introduction to the network architecture, primarily focusing on our improvements to the embedding layer and the decoder. In Sect. 3.2, we introduce Saliency-Guided Training, which prioritizes the quality enhancement of salient regions, making the reconstruction more aligned with human visual perception. Section 3.2 discusses how to compress the model after training to form the final video bitstream.

### 3.2 Architecture

As shown in Fig. 2, the frame index  $t$  is processed by the multi-scale temporal-spatial feature grid to generate initial feature embeddings  $X_0$ . The feature is progressively upsampled and processed through  $N \times$  SNeRV Blocks, resulting in the final feature  $X_N$  at the original resolution. The head layer subsequently generates the output  $Y$ .

**Multi-scale Temporal-Spatial Feature Grid.** The grid-based embedding [13] not only provides richer content-related features but also utilizes temporal correlations of frames. This method stores embeddings at intervals to construct feature grids at multiple different time resolutions. Each frame’s embedding is interpolated from the feature grids along the temporal dimension, and then the embeddings obtained from different grids are concatenated as the final

embedding. However, they only considered the temporal dimension. Inspired by grid-based INR for image [24], we additionally introduce multi-spatial resolution grids: low-resolution grids capture large-scale structure information, and high-resolution grids capture finer textures. We construct a multi-scale temporal-spatial feature grid as shown in Fig. 3. Each Grid  $G \in R^{s \times c \times h \times w}$  is a tensor, where  $s, c, h, w$  respectively denote the number of frames, channels, height, and width. The feature at time  $t$  in grid  $G$  is obtained through linear interpolation between the two nearest feature embeddings:

$$\begin{aligned} \phi(t, G) &= |\hat{t} - m| \cdot G[m] + |\hat{t} - n| \cdot G[n], \\ \hat{t} &= \frac{t \cdot s}{T}, \quad m = \lfloor \hat{t} \rfloor, \quad n = \lceil \hat{t} \rceil, \end{aligned} \quad (1)$$

where  $\phi(t, G)$  represents the embedding of the frame index  $t$  in the feature grid  $G$ ,  $t$  is the input index,  $s$  is the total number of frames in the feature grid, and  $T$  is the total frame number of the video.  $\hat{t}$  is the normalized index of  $t$  in the feature grid, and  $m$  and  $n$  are the indices of the two adjacent reference embeddings in the grid, computed using the floor  $\lfloor \cdot \rfloor$  and ceiling functions  $\lceil \cdot \rceil$  respectively.  $G[x]$  is the  $t^{\text{th}}$  feature embedding of  $G$ . We construct multiple feature grids with different temporal and spatial resolutions to encode the temporal index  $t$ . All obtained feature embeddings are upsampled to match the spatial resolution of the largest embedding and concatenated to form the final feature embedding  $E_t$ . The formula is as follows:

$$E_t = \text{Concat}(\text{Upsample}(\phi(t, G)_{k=1}^K)) \quad (2)$$

where  $K$  is the total number of feature grids, and  $k$  is the index of multiple grids. Subsequently, we employ a convolutional layer to project the embedding to the input channel number of the decoder:

$$X_0 = \text{Conv}(E_t) \quad (3)$$

**SNeRV Block.** After obtaining the initial feature map  $X_0$  through the feature grid, we proceed with  $N$  SNeRV Blocks for incremental upsampling and processing. As shown in Fig. 4(a), the SNeRV Block mainly consists of two stages: the upsampling stage and the processing stage. In the upsampling stage, we adopt the design of HiNeRV [14], where the output  $X_{n-1}$  of the previous block is first upsampled with a stride  $6$   $S_n$  using bilinear interpolation. Simultaneously, the input  $t$  is encoded through a local grid and added to the upsampled output. We primarily integrate some lightweight network designs [17] to enhance the processing stage. Initially, we adjust the channel number through one Convnext block [16] (Fig. 4(b)), and then refine it using two RepConv blocks (Fig. 4(c)). The RepConv block consists of a reparameterized depthwise convolution (repDWconv) and a feedforward network (FFN). The RepDWconv applies reparameterization [18] to depthwise convolution (DWConv). During training, RepDWConv employs a multi-branch structure with several convolutional paths,

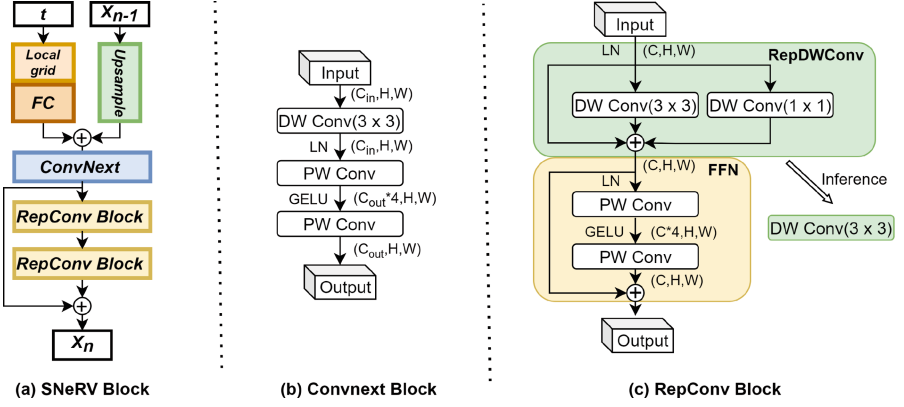


Fig. 4. Structure of SNeRV Block.

which helps in better learning and optimization. During inference, these multiple paths are re-parameterized into a single path DWConv, leading to a simple and efficient architecture similar, but with enhanced performance. This approach leads to improved training outcomes without introducing extra parameters. Subsequently, two pointwise convolutions (PWConv) are connected via residual links to form a Feedforward Network (FFN). The process of SNeRV Block can be written as:

$$X_n = F_n(\text{Upsample}(X_{n-1}) + \text{localgrid}(t)) \tag{4}$$

**Head Layer.** The output of last SNeRV Block  $X_n$  is mapped to the final output frame  $Y$  through a convolutional layer followed by a sigmoid activation function.

$$Y = \text{Sigmoid}(\text{Conv}(X_n)) \tag{5}$$

### 3.3 Saliency-Guided Training

The human eye exhibits varying degrees of visual sensitivity across different regions. For instance, the moving objects rabbit, tend to capture more attention compared to the background. Obviously, enhancing the reconstruction quality in these areas significantly improves the visual experience. However, existing NeRV methods employ a uniform weight allocation strategy for video reconstruction, overlooking this aspect. Inspired by variable bitrate techniques in traditional video coding, we utilize saliency detection technology to guide network training. This prioritizes improving the reconstruction quality in visually significant areas, enabling intelligent parameter allocation across different regions within video frames, thereby enhancing the overall visual quality (Fig. 5).

First, we preprocess the original video frames using a pre-trained saliency detection model [27] to generate saliency maps  $S$  for each frame, reflecting the

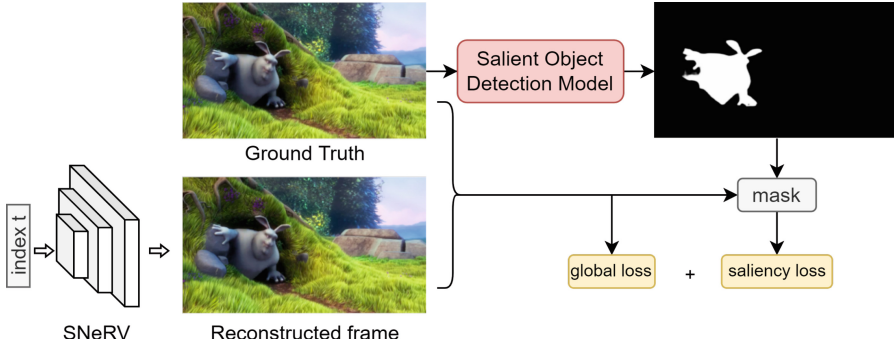


Fig. 5. Saliency-Guided Training strategy.

visual importance of various regions. This process is relatively quick and negligible compared to the overall training time. During training, we utilize saliency maps as guidance by assigning higher loss weights to salient regions. Our loss function comprises two components: global loss and salient region loss. The global loss measures the difference between the original frame  $x$  and the reconstructed frame  $y$  by reconstruction loss  $L(\cdot)$ . The salient region loss employs saliency maps  $S$  to mask the original  $x$  and reconstructed frames  $y$ , computing reconstruction loss  $L(\cdot)$  only within the masked salient regions. The reconstruction loss  $L(\cdot)$  can be composed of a combination of Mean Squared Error (MSE), L1 loss and Structural Similarity Index Measure (SSIM). By combining the global loss and salient region loss through weighted summation, we use the weight parameter  $\lambda$  to control the reconstruction priority of salient regions.

$$\begin{aligned}
 S &= \text{SaliencyDetection}(x) \\
 x_s &= \text{mask}(x, S), y_s = \text{mask}(y, S) \\
 L_{total} &= L(x, y) + \lambda L(x_s, y_s)
 \end{aligned}
 \tag{6}$$

### 3.4 Model Compression Pipeline

In the task of video compression, to further enhance the rate distortion performance, additional model compression are required after the video regression training. To further reduce size, post-processing techniques include pruning, quantization, and entropy coding.

**Pruning.** For the trained model, we perform unstructured pruning, followed by fine-tuning to recover the model’s representational capacity. Specifically, we refer to the importance score assessment of method [14],  $score(\theta) = \frac{|\theta|}{\sqrt{P}}$ . Where  $|\theta|$  is the absolute value of the parameter  $\theta$ , and  $P$  is the total number of parameters in the layer to which  $\theta$  belongs. This is based on the intuition that wider layers have greater redundancy, and pruning these layers has less impact compared to

narrower layers. Weights with importance score below a certain threshold are set to zero. The pruning formula is as follows:

$$\theta_i = \begin{cases} 0 & \text{if } score(\theta_i) < score(\theta_q) \\ \theta_i & \text{otherwise} \end{cases} \quad (7)$$

where  $score(\theta_q)$  is the  $q$  percentile importance score for all parameters.

**Quantizing.** Weights quantizing significantly reduces model size, with research indicating that quantizing model parameters to 8 bits doesn't notably degrade performance. Methods [13, 14] introducing quantization-aware training (QAT) [28] into the process have shown promising results. Following QAT fine-tuning, the model has an acceptable descent of reconstruction performance even with 6-bit quantization and demonstrates improved Ratio-Distortion performance in video compression tasks.

**Entropy Coding.** After quantization, the model undergoes Huffman coding [29], further reducing the space by approximately 10%.

## 4 Experiments

### 4.1 Datasets and Implementation Settings

**Datasets.** We conduct our experiments on the Bunny [30] and the UVG [31] dataset. The Bunny video consists of 132 frames at a resolution of  $1280 \times 720$ . The UVG dataset contains seven videos, totaling 3900 frames at a resolution of  $1920 \times 1080$ . UVG dataset encompasses a variety of video scenarios, typically serving as the primary metric for video representation and compression.

**Settings.** We construct multi-scale temporal-spatial feature grids at four different temporal and spatial resolutions:  $t \times 32 \times 18 \times c$ ,  $\frac{t}{2} \times 32 \times 18 \times 2c$ ,  $t \times 16 \times 9 \times c$ , and  $\frac{t}{2} \times 16 \times 9 \times 2c$ . The temporal resolution  $t$  and the number of channels  $c$  are set according to the required model parameters. We employed 4 blocks with upsampling strides of [5, 2, 2, 2] for the Bunny video and [5, 3, 2, 2] for the UVG dataset. Each block's channel is half that of the preceding block, with the initial block's channels determined by the total model parameters. Typically, we allocate 15% of the total parameters to the embedding layer, with the remaining parameters assigned to the decoder. We train three model sizes (S, M, L) for both the Bunny and UVG datasets and compared them with other methods. We use Mean Squared Error (MSE) as the reconstruction loss to guide the training and set the weight of salient region loss  $\lambda$  as 0.1. The training is performed using the patch-wise training method [14], with a batch size of 144. We utilize the Adam [32] optimizer and applies cosine learning rate decay with 10% warm-up epochs and a maximum learning rate of  $2 \times 10^{-3}$ . We conduct all experiments on an RTX 3090 GPU using the PyTorch framework.



## 4.2 Video Representation

**Table 1.** Video reconstruction on bunny.

Size	NeRV	ENeRV	HNeRV	HiNeRV	SNeRV
0.5M	25.77	27.07	31.98	34.51	<b>35.14</b>
1.5M	29.20	31.01	35.57	38.48	<b>39.09</b>
3M	32.67	35.41	37.43	41.04	<b>41.32</b>
avg.	29.21	31.16	34.99	38.01	<b>38.51</b>

We compare our method with existing NeRV methods [7, 8, 11, 14], using the peak signal-to-noise ratio (PSNR) to evaluate reconstruction quality. The comparisons are made under models with the same or similar parameter counts, categorized into small (0.5M), medium (1.5M), and large (3M) sizes, all trained for the same number of 300 epochs. For all NeRV methods, we maintain the same structure and training settings as described in the original papers [7, 8, 11, 14], adjusting only the network width to match the total number of parameters. From the results on bunny and UVG datasets as shown in Table 1 and Table 2, it is evident that our method outperforms the existing ones. Specifically, our approach achieves PSNR values of 30.19, 33.92, and 35.57 for the S, M, and L

**Table 2.** Video reconstruction on UVG.

Model	Size	beauty	bosph	bee	jockey	ready	shake	yach	avg.
NeRV	0.5M	30.53	28.90	32.05	26.48	20.71	28.41	24.89	27.42
ENeRV	0.56M	31.16	29.68	36.10	25.84	20.56	30.99	25.30	28.51
HNeRV	0.53M	31.69	30.49	36.79	26.83	21.02	32.44	25.94	29.31
HiNeRV	0.53M	32.45	32.92	37.22	28.22	24.53	<b>32.23</b>	27.10	30.67
SNeRV	0.53M	<b>32.67</b>	<b>33.55</b>	<b>37.84</b>	<b>29.94</b>	<b>24.73</b>	32.16	<b>27.44</b>	<b>31.19</b>
NeRV	1.5M	32.00	31.09	36.28	28.95	22.79	31.57	26.35	29.86
ENeRV	1.57M	33.25	31.11	37.68	27.59	22.36	33.37	26.00	30.19
HNeRV	1.54M	33.06	33.06	38.65	29.79	23.66	34.06	27.85	31.44
HiNeRV	1.48M	33.64	36.42	39.35	33.30	28.11	34.46	29.00	33.46
SNeRV	1.49M	<b>33.75</b>	<b>36.84</b>	<b>39.43</b>	<b>34.07</b>	<b>29.04</b>	<b>34.66</b>	<b>29.70</b>	<b>33.92</b>
NeRV	3.31M	32.88	33.22	38.44	31.03	24.73	33.52	27.73	31.65
ENeRV	3.29M	34.06	33.94	38.59	29.52	24.34	35.30	27.74	31.92
HNeRV	3.26M	33.58	34.73	38.96	32.04	25.74	34.57	29.26	32.69
HiNeRV	3.17M	34.08	38.68	<b>39.71</b>	36.10	31.53	35.85	30.95	35.27
SNeRV	3.19M	<b>34.14</b>	<b>38.81</b>	39.70	<b>36.46</b>	<b>32.39</b>	<b>35.94</b>	<b>31.54</b>	<b>35.57</b>

models, respectively, on the UVG dataset. Compared to the previous state-of-the-art model, HiNeRV [14], our method shows improvements of 0.52, 0.46, and 0.3, respectively.

The improvements stem from two main factors. Firstly, our structural enhancements, such as multi-scale positional encoding, provide richer initial feature embeddings, and the use of branching structures enhances the network’s fitting capacity. Secondly, the saliency-guided training method effectively prioritizes the reconstruction quality of regions of interest (ROI), resulting in more efficient parameter allocation. This is particularly evident in models with fewer parameters. This aligns with intuitive understanding, as saliency-guided training achieves an effect similar to bit rate allocation. In traditional video encoding, when encoding resources are abundant, bit rate allocation has a limited impact on encoding efficiency. However, under low bit rate conditions, optimizing bit rate allocation significantly enhances encoding efficiency and image quality. Due to resource scarcity, effective bit rate allocation better utilizes limited bandwidth, thereby improving compression performance. Similarly, models with fewer parameters have limited representational capacity. By using saliency-guided training to prioritize the quality of ROI, more effective parameter allocation is achieved, leading to performance improvement.

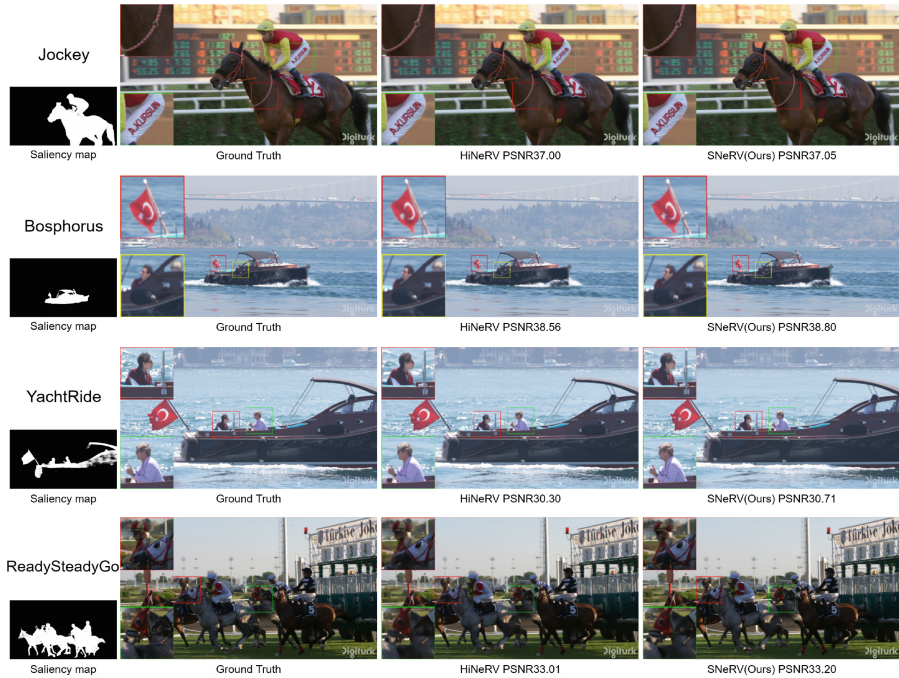


Fig. 6. Visualization of the reconstructed frame.

Figure 6 shows a visual comparison between our proposed method and HiNeRV [14]. Our method, guided by saliency, achieves significant improvements. In Bosphorus, the flag pattern reconstructed by HiNeRV [14] appears noticeably blurry, whereas our method achieves a clearer reconstruction, which is also reflected in the objective metric, with a PSNR difference of 0.24. In jokey, although the PSNR metrics are almost identical, our method excels at emphasizing ROI, resulting in better detail reconstruction in areas like the horse’s reins and the wrinkles on the clothes, thereby providing better visual quality. In YachtRide, our method also achieves finer restoration of details such as facial features and clothing wrinkles. In videos of intense sports like ReadySteadyGo, the difference is even more pronounced.

### 4.3 Video Compression

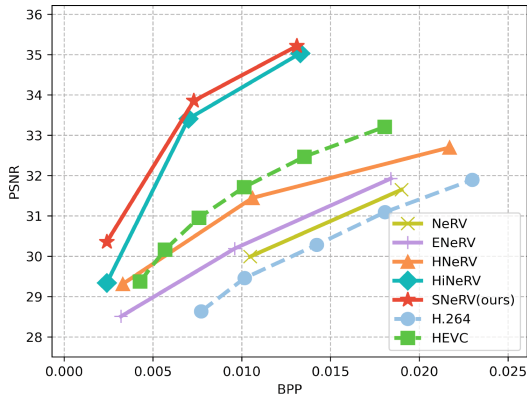


Fig. 7. Video compression results on the UVG.

To evaluate video compression performance, we conduct a thorough analysis centered on two pivotal metrics: the Peak Signal-to-Noise Ratio (PSNR), indicating video reconstruction quality, and the Bits Per Pixel (BPP), assessing compression efficiency. For SNeRV, we adopt the compression pipeline described before. Following the completion of video representation training, we conduct a 30-epoch pruning finetune process, selectively trimming 15% of the parameters for both models. Subsequently, we embark on a 30-epoch QAT training, integrating quantization noise. Finally, we quantized them to 6 bits and employed arithmetic entropy coding for lossless compression to obtain bitstreams. For other NeRV methods [7, 8, 11, 14], we performed pruning, quantization, and entropy coding according to the methods described in their original papers. In addition, we also compare our method with standard video encoders H.264 [19] and HEVC [20], and draw Ratio-Distortion performance graphs as shown in Fig. 7. The results show that our method outperforms HEVC and exhibits higher reconstruction quality than other INR-based methods at the same bit rate.

#### 4.4 Ablation Study

To validate the contributions of various components in SNeRV, we conduct ablation experiments. Starting from the baseline model HiNeRV [14], we gradually add our proposed components and evaluate video representation on the UVG dataset. For all experiments, we follow the setup described in Sect. 4.1 and trained the 3M model. The results are shown in Table 3.

**Multi-scale Temporal-Spatial Feature Grid:** We first replace the embedding layer of the baseline model with our proposed multi-scale temporal-spatial feature grid to verify its contribution.

**SNeRV Block:** Building on the previous step, we replace the basic blocks with our SNeRV blocks.

**Saliency-Guided Training:** Finally, we introduce saliency maps to guide the training process. While this step showed limited improvement in the objective metric PSNR, it significantly enhances visual quality in salient regions by focusing more on the ROI during training. As illustrated in Fig. 6, the detail textures in the salient areas are noticeably better.

**Table 3.** Ablation studies of SNeRV on UVG.

Model	Grid	Block	Train	Size	beauty	bosph	bee	jockey	ready	shake	yach	avg.
HiNeRV				3.17M	34.08	38.68	39.71	36.10	31.53	35.85	30.95	35.27
+Grid	✓			3.19M	34.09	38.72	39.71	36.32	32.21	35.75	31.44	35.46
+Block	✓	✓		3.19M	34.14	38.72	39.70	36.44	32.24	35.92	31.60	35.54
SNeRV	✓	✓	✓	3.19M	34.14	38.81	39.70	36.46	32.39	35.94	31.54	35.57

## 5 Conclusion

In this paper, we propose a NeRV method that considers subjective quality, Saliency-based Neural Representation for Videos (SNeRV). On one hand, we enhance the network’s representation capability by designing multi-scale temporal-spatial feature grids for the embedding layer and SNeRV Block for the decoder. This achieves superior performance in both objective and subjective quality for video representation and compression tasks compared to existing approaches. On the other hand, considering the varying sensitivity of the human eye to different regions, we introduce a saliency-guided training strategy that prioritizes the reconstruction quality of Regions of Interest (ROI), resulting in better visual effects.

**Acknowledgment.** The research is supported by the Fundamental Research Funds for the Central Universities.

## References

1. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **65**(1), 99–106 (2021)
2. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. *Adv. Neural. Inf. Process. Syst.* **33**, 7462–7473 (2020)
3. Chen, Y., Liu, S., Wang, X.: Learning continuous image representation with local implicit image function. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8628–8638 (2021)
4. Dupont, E., Goliński, A., Alizadeh, M., Teh, Y.W., Doucet, A.: COIN: compression with implicit neural representations. *arXiv preprint arXiv:2103.03123* (2021)
5. Zhang, Y., van Rozendaal, T., Brehmer, J., Nagel, M., Cohen, T.: Implicit neural video compression. *arXiv preprint arXiv:2112.11312* (2021)
6. Rho, D., Cho, J., Ko, J.H., Park, E.: Neural residual flow fields for efficient video representations. In: *Proceedings of the Asian Conference on Computer Vision*, pp. 3447–3463 (2022)
7. Chen, H., He, B., Wang, H., Ren, Y., Lim, S.N., Shrivastava, A.: NeRV: neural representations for videos. *Adv. Neural. Inf. Process. Syst.* **34**, 21557–21568 (2021)
8. Li, Z., Wang, M., Pi, H., Xu, K., Mei, J., Liu, Y.: E-NeRV: expedite neural video representation with disentangled spatial-temporal context. In: Avidan, S., Brostow, G., Cisse, M., Farinella, G.M., Hassner, T. (eds.) *European Conference on Computer Vision*, pp. 267–284. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-19833-5\\_16](https://doi.org/10.1007/978-3-031-19833-5_16)
9. Kim, S., Yu, S., Lee, J., Shin, J.: Scalable neural video representations with learnable positional features. *Adv. Neural. Inf. Process. Syst.* **35**, 12718–12731 (2022)
10. He, B., et al.: Towards scalable neural representation for diverse videos. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6132–6142 (2023)
11. Chen, H., Gwilliam, M., Lim, S.N., Shrivastava, A.: HNeRV: a hybrid neural representation for videos. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10270–10279 (2023)
12. Zhao, Q., Asif, M.S., Ma, Z.: DNeRV: modeling inherent dynamics via difference neural representation for videos. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2031–2040 (2023)
13. Lee, J.C., Rho, D., Ko, J.H., Park, E.: FFNeRV: flow-guided frame-wise neural representations for videos. In: *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 7859–7870 (2023)
14. Kwan, H.M., Gao, G., Zhang, F., Gower, A., Bull, D.: HiNeRV: video compression with hierarchical encoding-based neural representation. In: *Advances in Neural Information Processing Systems*, vol. 36 (2024)
15. Koonce, B., Koonce, B.: Mobilenetv3. *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, pp. 125–144 (2021)
16. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A ConvNet for the 2020s. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986 (2022)
17. Wang, A., Chen, H., Lin, Z., Pu, H., Ding, G.: RepViT: revisiting mobile CNN from ViT perspective. *arXiv preprint arXiv:2307.09283* (2023)

18. Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., Sun, J.: RepVGG: making VGG-style ConvNets great again. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13733–13742 (2021)
19. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. *IEEE Trans. Circ. Syst. Video Technol.* **13**(7), 560–576 (2003)
20. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1649–1668 (2012)
21. Lu, G., Ouyang, W., Xu, D., Zhang, X., Cai, C., Gao, Z.: DVC: an end-to-end deep video compression framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11006–11015 (2019)
22. Li, J., Li, B., Lu, Y.: Deep contextual video compression. *Adv. Neural. Inf. Process. Syst.* **34**, 18114–18125 (2021)
23. Hu, Z., Lu, G., Xu, D.: FVC: a new framework towards deep video compression in feature space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1502–1511 (2021)
24. Ladune, T., Philippe, P., Henry, F., Clare, G., Leguay, T.: COOL-CHIC: coordinate-based low complexity hierarchical image codec. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13515–13522 (2023)
25. Kim, H., Bauer, M., Theis, L., Schwarz, J.R., Dupont, E.: C3: high-performance and low-complexity neural compression from a single image or video. arXiv preprint [arXiv:2312.02753](https://arxiv.org/abs/2312.02753) (2023)
26. Fan, D.P., Wang, W., Cheng, M.M., Shen, J.: Shifting more attention to video salient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8554–8564 (2019)
27. Gu, Y., Wang, L., Wang, Z., Liu, Y., Cheng, M.M., Lu, S.P.: Pyramid constrained self-attention network for fast video salient object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 10869–10876 (2020)
28. Fan, A., et al.: Training with quantization noise for extreme model compression. arXiv preprint [arXiv:2004.07320](https://arxiv.org/abs/2004.07320) (2020)
29. Huffman, D.A.: A method for the construction of minimum-redundancy codes. *Proc. IRE* **40**(9), 1098–1101 (1952)
30. Big buck bunny. <http://bbb3d.renderfarming.net/download.html>
31. Mercat, A., Viitanen, M., Vanne, J.: UVG dataset: 50/120 fps 4k sequences for video codec analysis and development. In: Proceedings of the 11th ACM Multimedia Systems Conference, pp. 297–302 (2020)
32. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)



# HNRC: Lightweight Image Compression with Hybrid Neural Representation

Xinyuan Cheng<sup>1</sup>, Dongdong Zhang<sup>1(✉)</sup>, and Xiaolei Zhang<sup>2</sup>

<sup>1</sup> Department of Computer Science and Technology, Tongji University, Shanghai, China

{2331913, ddzhang}@tongji.edu.cn

<sup>2</sup> Department of Geotechnical Engineering, Tongji University, Shanghai, China  
Xiaolei.Zhang@tongji.edu.cn

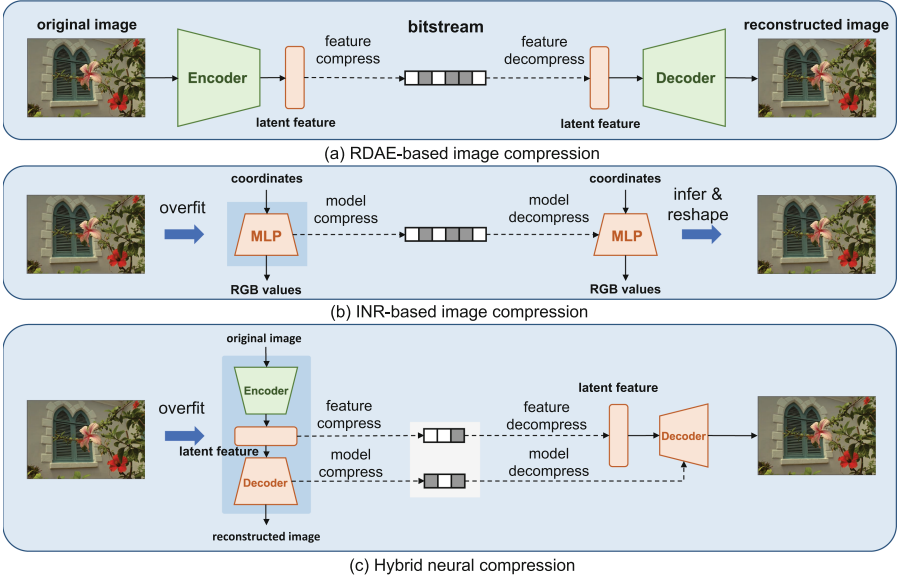
**Abstract.** Recently, image compression methods based on rate distortion autoencoder (RDAE) have achieved advanced performance. However, these methods have high decoding complexity, which limits their application on low-power devices. To address this issue, Implicit Neural Representations (INR) represents images as neural networks that map coordinates to signal values and forms INR-based image compression method. Despite with low decoding complexity, there is a significant performance gap between INR-based approaches and RDAE-based approaches. In this paper, we propose an image compression method with hybrid neural representation (HNRC) to improve compression performance of INR-based approaches while keeping decoding lightweight. Specifically, we design a Groupwise Feature Aggregation module to aggregate feature of different groups, develop a Pointwise Local Modulation module to enhance the representation of local details, and employ a Gaussian Mixture Model to improve the accuracy of rate estimation. Extensive experiments demonstrate that our method achieves an approximate 1.1 dB improvement in terms of PNSR over INR-based approaches on the Kodak dataset while reducing decoding complexity by 88.9%.

**Keywords:** Image compression · Implicit neural representation · Hybrid neural representation

## 1 Introduction

Image compression is an important research task in the field of signal processing for decades. Traditional image compression methods, such as JPEG [1] and JPEG2000 [2], have been widely used in almost all image processing software. With the development of deep learning, image compression methods with rate-distortion autoencoder (RDAE-based) have been widely explored. RDAE-based approaches exploit an end-to-end autoencoder [9] architecture as nonlinear transform to achieve superior performance, as shown in Fig. 1(a). However, the structure of autoencoder is highly complex and computationally demanding, which limits the application on resource-constrained devices.





**Fig. 1.** Different approaches of image compression. We overfit a compact autoencoder with the idea of INR and transmit encoded latent features and decoder parameters as bitstreams

Recently, Implicit Neural Representation (INR) has been proposed as a novel paradigm of data representation. INR aims to construct a continuous function that maps input coordinates to corresponding values. Unlike the traditional discrete grid storage, INR can take advantage of powerful continuous functions to represent complex scenes with compact neural networks, thus it has inherent compression capability. Some studies [3–6] applied INR to image compression and proposed INR-based methods, as shown in Fig. 1(b). INR-based methods utilize a compact neural network to overfit the mapping from coordinate to values for a single image and store image data as model parameters. This type of approach has low decoding complexity and high decoding speed on edge devices. However, it is hard to optimize since changes to individual parameters can have widespread effects on each pixel [7]. Additionally, there is still a performance gap between INR-based methods and RDAE-based methods.

To improve compression performance of INR-based methods, we propose a lightweight image compression approach based on hybrid neural representation. Our general idea is to apply the idea of INR into the autoencoder structure, as shown in Fig. 1(c). We overfit a tiny autoencoder to each image separately, so an image is represented by encoded latent representation and decoder parameters. In RDAE-based methods, the decoder typically consists of multiple transposed convolutions to perform upsampling and complex transforms simultaneously, which requires a substantial number of parameters and computational resources. To address this issue, we factorizes the decoder in RDAE-based method into



groupwise aggregation and pointwise nonlinear transform. Specifically, we design a Groupwise Feature Aggregation (GFA) module to aggregate different grouped features by stacking multiple Implicit Aggregation (IA) blocks in parallel. Subsequently, We develop a Pointwise Local Modulation (PLM) module. Aggregated features are passed through multiple Local Modulation (LM) blocks in series to enhance the representation of local details. Additionally, existing INR-based methods only consider distortion during training, neglecting the impact of rate. We introduce a rate loss of latent features to constrain the rate-distortion balance and employ a context-based Gaussian Mixture entropy model to better estimate the probability distribution of the latent representation. To summarize, our contributions include:

- We propose an image compression method with hybrid neural representation, combining RDAE-based methods and INR-based methods to improve compression performance of INR-based approaches while keeping decoding lightweight.
- We design a Groupwise Feature Aggregation module, which aggregates the latent features of grouped channels effectively. We develop a Pointwise Local Modulation module, which enhances the awareness of local details. We introduce a context-based Gaussian Mixture entropy model to improve the accuracy of rate estimation.
- Experiments have shown that our approach outperforms existing INR-based methods in compression performance while maintaining a low decoding complexity.

## 2 Related Work

### 2.1 RDAE-Based Image Compression

To enhance compression performance, rate-distortion autoencoder (RDAE) image compression was first introduced by [8]. Input images pass through analysis transforms to get latent representations and quantized into discrete values. Finally reconstructed images are obtained through the synthesis transforms. Since quantization, namely rounding to the nearest integer, is not differentiable, it is usually approximated by adding uniform noise [8] or straight-through estimator [10] during training.

The loss function comprises two components: distortion and rate. Distortion measures the reconstruction quality of the decoded image, while rate measures the size of the compressed file. A fully factorized entropy model [8] is used to estimate the rate of quantized latent representation at first. To estimate the rate more efficiently, Ballé et al. [11] propose a hyper-prior model to extract auxiliary information from latent representation, and exploit a univariate Gaussian distribution to estimate the distribution of quantized latent representation from auxiliary information. Subsequent works exploit more complex distributions to model the probability distribution of latent representation, such as mean and scale Gaussian distribution [12], Gaussian Mixture model [13], etc.

However, the synthesis transform usually consists of multiple transposed convolutions to complete upsampling and complex transform simultaneously, which requires a significant number of parameters and computational resources. Consequently, such methods have high decoding complexity, limiting their broader application on low-power devices.

## 2.2 Implicit Neural Representation

Compared to traditional discrete grid representation, Implicit Neural Representation (INR) utilizes multi-layer perceptrons (MLPs) to overfit data signals and construct a continuous function to learn the mapping from input coordinates to corresponding values (such as RGB values, densities, etc.). Due to the spectral bias [14] of neural networks inherently, some approaches have been proposed to improve the representation ability of INR. Tancik et al. [15] propose that Fourier Feature mapping can enable MLPs to learn high-frequency information. Sitzmann et al. [16] propose SIREN network and use periodic activation functions to improve the reconstruction of fine details. As a novel data representation paradigm, INR is widely applied in various tasks including 3D reconstruction [17–19], image super-resolution [20, 21], and data compression [3–6, 22, 23].

## 2.3 Implicit Neural Compression

Most recently, image compression based on INR has become a new paradigm. Dupont et al. [3] propose COIN, encoding an image by overfitting it with a small MLP that mapping pixel locations to RGB values. The weights of this MLP are then transmitted as the code for the image. By this way, image compression problem is transformed into model compression problem. Strumpler et al. [4] employ meta-learning to obtain better network parameter initialization, thereby accelerating the training convergence. SHACIRA [5] reparameterizes learnable feature grids with quantized latent weights and applies entropy regularization in the latent space to achieve high levels of compression across various domains. NIF [6] improves SIREN architecture to accommodate frequency variations in different regions of the image. The model consists of two modules: a Genesis network and a Modulation network. The Genesis network maps coordinates to features through bottleneck layers with sinusoidal activation units and the Modulation network varies the period of the sinusoidal activations.

INR-based methods are characterized by low decoding complexity and fast decoding speed. However, the representation capacity of INR is still limited by the global property, resulting in a performance gap compared to advanced approaches. To improve compression performance of INR-based methods and reduce decoding complexity of RDAE-based methods, we combine the advantages of two approaches and propose an image compression method based on hybrid neural representation.

### 3 Method

The overall architecture of our method is shown in Fig. 2. The input image undergoes an analysis transform to obtain latent features, which are then processed through a factorized synthesis transform to produce the reconstructed image. The factorized synthesis transform consists of a Groupwise Feature Aggregation (GFA) module and a Pointwise Local Modulation (PLM) module. A Gaussian Mixture entropy model is utilized to estimate the rate of latent features. We first detail the architecture of proposed method in Sect. 3.1. Then we introduce Groupwise Feature Aggregation module in Sect. 3.2, Pointwise Local Modulation module in Sect. 3.3, and context-based Gaussian Mixture entropy model in Sect. 3.4, respectively.

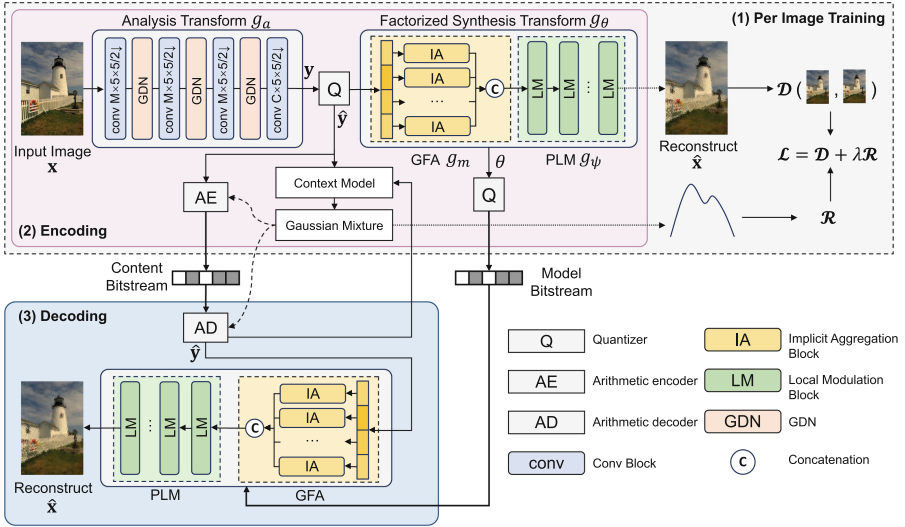


Fig. 2. The overall architecture of proposed method

#### 3.1 Overall Architecture

Define a raw image as  $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ . The input image  $\mathbf{x}$  first passes through an analysis transform  $g_a$  to obtain latent features  $\mathbf{y} \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C}$ :

$$\mathbf{y} = g_a(\mathbf{x}) \quad (1)$$

The structure of analysis transform is same as [11]. The latent  $\mathbf{y}$  is then quantized into a discrete representation  $\hat{\mathbf{y}}$  as:

$$\hat{\mathbf{y}} = Q(\mathbf{y}) \quad (2)$$

In RDAE-based methods, the typical synthesis transform performs upsampling and complex transform simultaneously, which demands substantial computational resources and numerous model parameters. To reduce computational load and the number of transmitted parameters, we replace the original synthesis transform with a factorized form  $g_\theta$ . Specifically, quantized features  $\hat{\mathbf{y}}$  are first divided into groups along the channel dimension. We develop a Groupwise Feature Aggregation (GFA) module  $g_m$  to aggregate latent features of different groups and design a Pointwise Local Modulation (PLM) module  $g_\psi$  to reconstruct the image  $\hat{\mathbf{x}}$  as:

$$\hat{\mathbf{x}} = g_\theta(\hat{\mathbf{y}}) = g_\psi(g_m(\hat{\mathbf{y}})) \quad (3)$$

where  $\theta$  is the total parameters of factorized synthesis transform. During training, we aim to overfit latent  $\hat{\mathbf{y}}$ , factorized synthesis transform  $g_\theta$  for each image  $\mathbf{x}$  by rate-distortion optimization. To avoid ineffectiveness of gradient descent after round operation, we add uniform noise [8]  $\delta \sim \mathcal{U}(-0.5, 0.5)$  to approximate quantization:

$$Q(\mathbf{y}) = \begin{cases} \mathbf{y} + \delta & , \text{if training} \\ \text{round}(\mathbf{y}) & , \text{otherwise} \end{cases} \quad (4)$$

A context-based entropy model is utilized to estimate the probability of latent  $\hat{\mathbf{y}}$ , so the rate of  $\hat{\mathbf{y}}$  can be formulated as:

$$\mathcal{R}(\hat{\mathbf{y}}) = -\log_2 p_\phi(\hat{\mathbf{y}}) = -\log_2 \prod_i p_\phi(\hat{y}_i | \mathbf{ctx}_i) \quad (5)$$

where  $\mathbf{ctx}_i \in \mathbb{R}^S$  represents context pixels of  $\hat{y}_i$  and  $\phi$  is the parameter of entropy model.

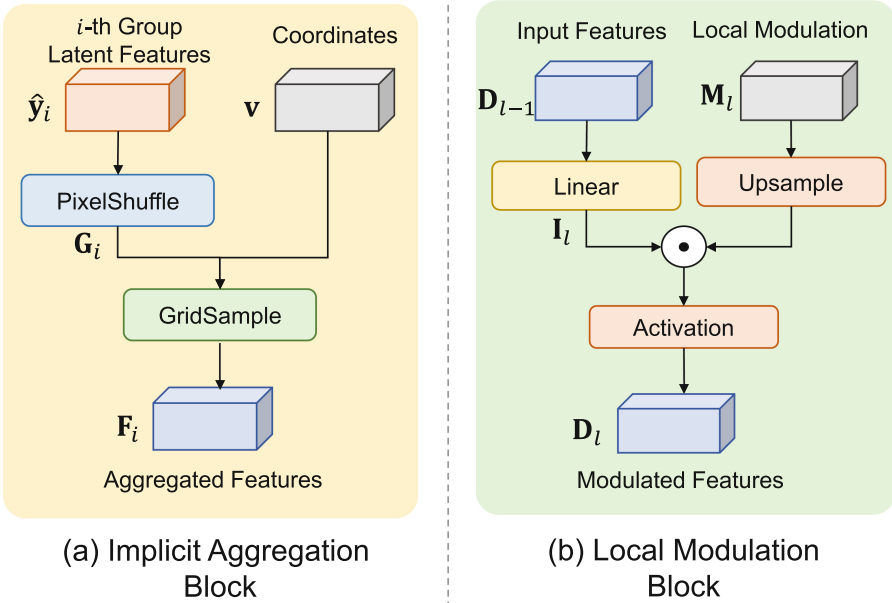
The pipeline of our method includes three steps. First, we overfit parameters of model by rate-distortion optimization for each image. The loss function is defined as:

$$\begin{aligned} \mathcal{L} &= \mathcal{D}(\mathbf{x}, \hat{\mathbf{x}}) + \lambda \mathcal{R}(\hat{\mathbf{y}}) \\ &= \mathcal{D}(\mathbf{x}, g_\psi(g_m(\hat{\mathbf{y}}))) - \lambda \log_2 \prod_i p_\phi(\hat{y}_i | \mathbf{ctx}_i) \end{aligned} \quad (6)$$

where  $\mathcal{D}(\mathbf{x}, \hat{\mathbf{x}})$  represents distortion,  $\mathcal{R}(\hat{\mathbf{y}})$  represents the rate of  $\hat{\mathbf{y}}$ , and  $\lambda$  is the Lagrange multiplier to balance the trade-off between rate and distortion.

In the encoding process, we adopt arithmetic coding to encode the latent  $\hat{\mathbf{y}}$  with estimated probability and use uniform quantization to compress the parameters of the factorized synthesis transform  $\theta$  before transmission.

In the decoding process, the parameters of the factorized synthesis transform  $\theta$  are firstly decoded from the model bitstream. Then, we obtain the quantized latent features  $\hat{\mathbf{y}}$  by decoding the content bitstream with arithmetic decoding. Finally, we perform forward step to get reconstructed image  $\hat{\mathbf{x}}$ . The implementation details of each module are described below.



**Fig. 3.** The structures of each proposed module, include Implicit Aggregation (IA) block (left) and Local Modulation (LM) block (right)

### 3.2 Groupwise Feature Aggregation Module

To aggregate the latent features of different channels effectively, we design a Groupwise Feature Aggregation (GFA) module, including multiple Implicit Aggregation (IA) blocks in parallel. The structure of IA block is shown in Fig. 3(a). First, we divide the quantized latent features  $\hat{\mathbf{y}}$  into  $N$  groups along the channel dimension and  $i$ -th group has  $C_i$  channels. After that,  $i$ -th group of latent feature  $\hat{\mathbf{y}}_i \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_i}$  is input to  $i$ -th IA block. Then the  $i$ -th grouped features  $\mathbf{G}_i \in \mathbb{R}^{h_i \times w_i \times T_i}$  are computed as:

$$\mathbf{G}_i = \text{pixelshuffle}(\hat{\mathbf{y}}_i, 2^{i-1}) \quad (7)$$

where  $i = 1, 2, \dots, N$ , and  $\text{pixelshuffle}(\cdot, 2^{i-1})$  represents the pixel shuffle operation [24] with  $2^{i-1}$  upscale factor. This aims for different groups to focus on features at different scales. Subsequently, grouped features  $\mathbf{G}_i$  are aggregated with pixel coordinates implicitly to obtain the  $i$ -th aggregated feature  $\mathbf{F}_i \in \mathbb{R}^{H \times W \times T_i}$ :

$$\mathbf{F}_i = \text{gridsample}(\mathbf{G}_i, \mathbf{v}) \quad (8)$$

where  $\mathbf{v} \in \mathbb{R}^{H \times W \times 2}$  represents pixel coordinates of the input image. Finally, aggregated features of each group are concatenated to get output features  $\mathbf{F} \in \mathbb{R}^{H \times W \times T}$ :

$$\mathbf{F} = g_m(\hat{\mathbf{y}}) = \text{concat}(\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_N) \quad (9)$$

The GFA module achieves non-parametric grouped feature upsampling and aggregates with coordinates to enhance the continuity of features.

### 3.3 Pointwise Local Modulation Module

To enhance the representation of local details, we design a Pointwise Local Modulation (PLM) module, composed of multiple Local Modulation (LM) blocks connected in series. The structure of LM block is shown in Fig. 3(b). Assume there are  $L$  LM blocks. For the  $l$ -th LM block, the input consists of the modulated feature  $\mathbf{D}_{l-1}$  from the previous block and a learnable local modulation vector  $\mathbf{M}_l$ . The output is the modulated feature  $\mathbf{D}_l$  for the current block. For the first layer,  $\mathbf{D}_0 = \mathbf{F}$ .

Specifically, the features of the previous block are first linearly transformed to get the intermediate features  $\mathbf{I}_l$ :

$$\mathbf{I}_l = \mathbf{W}_l \mathbf{D}_{l-1} + \mathbf{b}_l \quad (10)$$

where  $\mathbf{W}_l \in \mathbb{R}^{d \times d}$  and  $\mathbf{b}_l \in \mathbb{R}^d$  are weight and bias of the  $l$ -th block, and  $l = 1, 2, \dots, L - 1$ . Then, the intermediate feature  $\mathbf{I}_l$  is dot-producted with the upsampled local modulation vector, and the output modulated features  $\mathbf{D}_l$  are computed as:

$$\mathbf{D}_l = \text{ReLU}(\mathbf{I}_l \odot \text{upsample}(\mathbf{M}_l)) \quad (11)$$

where  $\odot$  represents the element-wise product. We use bicubic interpolation for upsampling. Finally, after a series of LM blocks, we use a linear layer to map modulated features to RGB values:

$$\hat{\mathbf{x}} = g_\psi(\mathbf{F}) = \mathbf{W}_L \mathbf{D}_{L-1} + \mathbf{b}_L \quad (12)$$

where  $\mathbf{W}_L \in \mathbb{R}^{d_{\text{out}} \times d}$  and  $\mathbf{b}_L \in \mathbb{R}^{d_{\text{out}}}$ . Through multi-block local modulation, our method can capture details of different regions of the image.

### 3.4 Context-Based Gaussian Mixture Entropy Model

In RDAE-based methods, a hyper-prior model is usually used to improve the estimation of the rate. However, since the parameters of the hyper-prior network also need to be transmitted in our method, adding a hyper-prior model not only occupies a large number of codewords, but also increases the complexity of decoding. Therefore, in our method, we only employ a compact context model to estimate the probability to balance the trade-off between lightweight design and rate estimation efficiency.

In order to improve the accuracy of entropy model for rate estimation, we use the Gaussian Mixture model proposed in [13] as the prior probability distribution:

$$p_\phi(\hat{\mathbf{y}}|\mathbf{ctx}) \sim \sum_{k=1}^K \omega^{(k)} \mathcal{N}(\boldsymbol{\mu}^{(k)}, \boldsymbol{\sigma}^{2(k)}) \quad (13)$$

The probability of latent  $\hat{\mathbf{y}}$  at  $i$ -th position is calculated as:

$$p_\phi(\hat{y}_i|\mathbf{ctx}_i) = \left( \sum_{k=1}^K \omega_i^{(k)} \mathcal{N}\left(\mu_i^{(k)}, \sigma_i^{2(k)}\right) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right) \right)(\hat{y}_i) \quad (14)$$

The values of each parameter of the Gaussian Mixture distribution is computed by a context model  $g_\phi : R^S \rightarrow R^{3K}$  as:

$$[\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i, \boldsymbol{\omega}_i] = g_\phi(\mathbf{ctx}_i) \quad (15)$$

where  $\boldsymbol{\mu}_i \in R^K$ ,  $\boldsymbol{\sigma}_i \in R^K$  and  $\boldsymbol{\omega}_i \in R^K$  represent means, variances and weights of Gaussian Mixture distribution of  $i$ -th position, respectively.

## 4 Experiments

We conduct extensive experiments to evaluate our method. First, we introduce datasets, evaluation metrics and implementation details. Then, we compare the performance and qualitative results with other methods. Next, we analyze complexity of our method compare to other approaches. Finally, we provide an ablation experiment to verify the effectiveness of each module.

### 4.1 Datasets

We evaluate our model on the Kodak [25] and CelebA [26] datasets. The Kodak dataset contains 24 natural images with a resolution of  $512 \times 768$  or  $768 \times 512$ , which is widely used in image compression task. For the CelebA dataset, which contains a large amount of face images with a resolution of  $178 \times 218$ , we evaluate our method following the same setting of previous works [4, 6].

### 4.2 Metrics

We employ different evaluation metrics to analyze the rate and distortion of image compression. We use the Peak Signal-to-Noise Ratio (PSNR) to measure the distortion, which is computed as:

$$\text{PNSR} = -10 \log_{10}(\text{MSE}) \quad (16)$$

where MSE is the mean square error between raw image and reconstructed image, both normalized to the range of  $[0, 1]$ . We use bitrate to measure the rate, which is defined as:

$$\text{bitrate} = \frac{\text{total bits}}{HW} \quad (17)$$

We use Giga Floating Point Operations (GFLOPs) to measure the decoding complexity. GFLOPs calculates the total number of floating point operations during the forward propagation of the model and is generally used to measure the complexity of the model.

### 4.3 Implementation Details

All experiments are performed on a single RTX3090 GPU. The model is implemented with PyTorch. For Groupwise Feature Aggregation module, latent features are divided into  $N = 5$  groups with  $\{1, 4, 16, 64, 256\}$  channels in each group. For Pointwise Local Modulation module, we set  $L = 3$  LM blocks. For each LM block,  $d$  is set to 12. For context-based Gaussian Mixture entropy model, we use a 3-layer MLP with 12 hidden neurons and  $S = 12, K = 3$ . We use the Adam optimizer with an initial learning rate of  $1e-3$ , and use ReduceLROnPlateau learning rate scheduler with the learning rate decay factor set to 0.5.

### 4.4 Rate-Distortion Performance

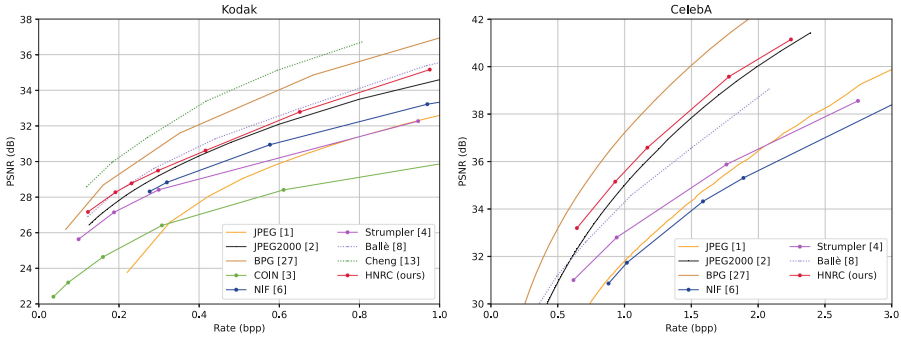
We compare our approach against various compression techniques, including traditional methods, INR-based methods, and RDAE-based methods. Traditional methods include JPEG [1], JPEG2000 [2] and BPG [27]. INR-based methods include COIN [3], NIF [6] and Strmupler [4]. RDAE-based methods include Ballé [8] and Cheng [13].

The rate-distortion curves on Kodak and CelebA dataset are shown in Fig. 4. On the Kodak dataset, our method achieves the best compression performance compared with INR-based approaches. Specifically, at the same bit rate, the PNSR of our method exceeds the baseline NIF [6] by around 1.1 dB. Compared with RDAE-based methods, our method is close to the performance of the Ballé [8] method. It is worth mentioning that the RDAE-based methods have better compression performance at the expense of high decoding complexity. We will compare the decoding complexity of each approach in the next section. On the CelebA dataset, our method also has similar comparison results and outperforms existing INR-based approaches in terms of PSNR. At the similar bit rate, the PSNR of our method exceeds the INR-based Strmupler [4] method by about 2.5dB, and exceeds the RDAE-based Ballé [8] method by about 1.2 dB.

### 4.5 Qualitative Results

The visualization results of each methods are shown in Fig. 5. Due to aggregating with spatial coordinate, the decoded images of our method are smooth and detailed. Although the objective metrics of the RDAE-based methods are higher, our method has comparable results in subjective quality. For example, on the Kodak21 image, our method can better reconstruct the details of the edges of white clouds and lighthouses, while the JPEG2000 method shows more artifacts around edges and Cheng [13] method also loses details inside the clouds.





**Fig. 4.** Rate-distortion curves of our approach and different baselines on Kodak (left) and CelebA (right) datasets in terms of PSNR. It is worth mentioning that Cheng [13] method was not tested on the CelebA dataset due to a mismatch of image resolutions

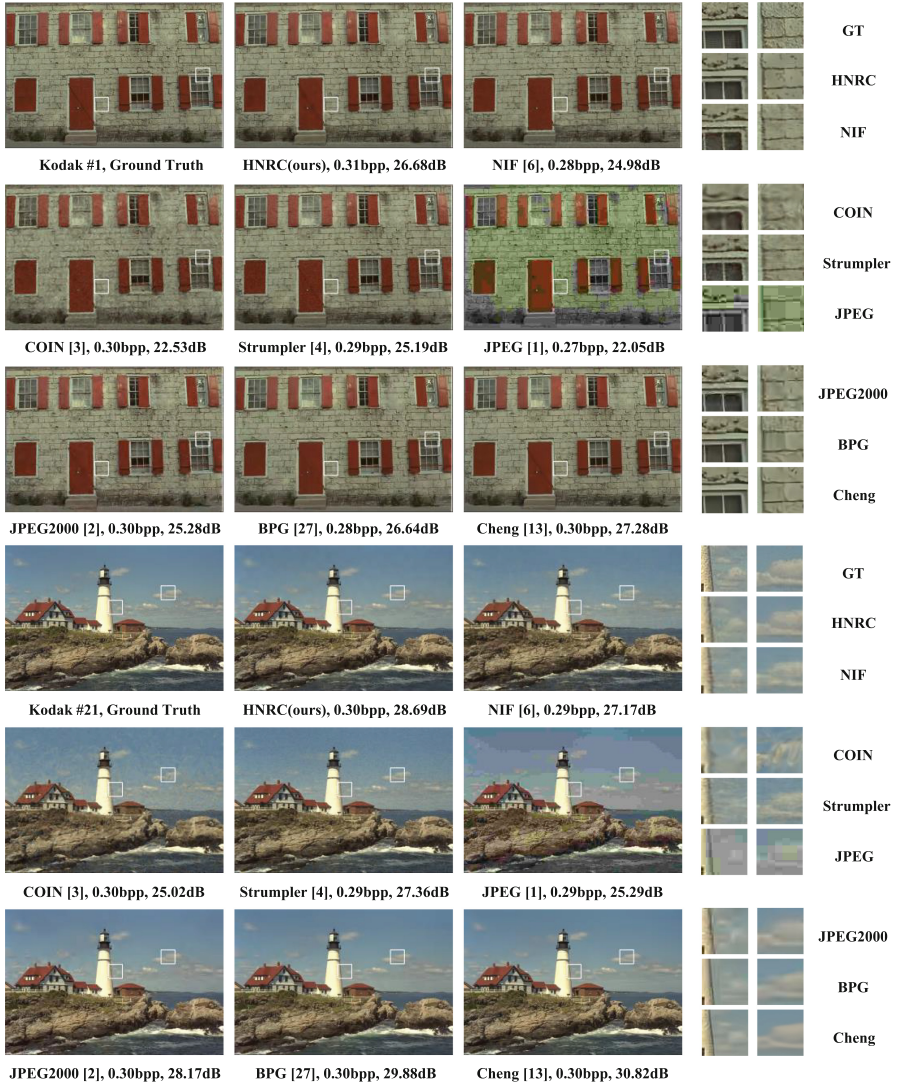
## 4.6 Complexity Analysis

**Decoding Complexity.** In order to verify that our method has lower decoding complexity, we decode images of the Kodak dataset around 0.3bpp on CPU and calculate the floating-point operations during decoding of each method, as shown in Table 1. According to the results, our method demonstrates significantly lower decoding complexity compared to RDAE-based methods, and also shows a notable reduction when compared to the INR-based method. In particular, the floating-point operations during decoding of our method is only 2% of that of Ballé [8] method and 5.4% of that of NIF [6] method. This means that our approach can decode images more efficiently on resource-constrained devices.

**Table 1.** Decoding complexity on CPU of different approaches

Methods	Type	GFLOPs ↓
COIN [3]	INR-based	5.66
NIF [6]	INR-based	11.61
Ballé [8]	RDAE-based	33.23
Cheng [13]	RDAE-based	204.85
HNRC (ours)	Hybrid	<b>0.63</b>

**Encoding Complexity.** RDAE-based methods require to pre-train complex encoder and decoder networks, while INR-based methods do not require pre-training and only need to optimize a compact network for each image individually. Therefore, INR-based methods have higher encoding complexity. For example, it takes around 10 min for INR-based COIN [3] method to optimize a



**Fig. 5.** Visual comparisons of our proposal approach with traditional methods, RDAE-based and INR-based baselines. Bitrate and PSNR for each picture compared to ground truth are reported in this same order in image captions

image from the Kodak dataset. Our method has a similar encoding complexity to existing INR-based methods, taking around 6 min to train each image. Future improvements can employ meta-learning to obtain better network parameter initialization and accelerate convergence speed. Our method is suitable for scenarios where images are pre-encoded on the server and need to quickly decoded on low-power devices.

## 4.7 Ablation Study

We design an ablation experiment to verify the effectiveness of each proposed component. The experimental results in Table 2 show that the Group Feature Aggregation module can aggregate latent features of different groups efficiently and improve the representation ability of the decoder; the Pointwise Local Modulation module can improve awareness of local information, thereby improving the model’s reconstruction of details; compared to univariate Gaussian distribution, Gaussian Mixture model can improve the accuracy of rate estimation, thereby reducing the rate of the latent representation while keeping PSNR almost unchanged.

**Table 2.** Ablation study of proposed method. GFA means Group Feature Aggregation module. PLM denotes Pointwise Local Modulation module and GMM means Gaussian Mixture model

Modules			Bit Rate (bpp) ↓	PSNR (dB) ↑
GFA	PLM	GMM		
✓	✓	✓	0.65	32.78
×	✓	✓	0.66	32.31
✓	×	✓	0.69	32.68
✓	✓	×	0.73	32.80

## 5 Conclusion

We propose HNRC, a lightweight image compression approach based on hybrid neural representation, which combines the advantages of RDAE-based methods and INR-based methods to improve compression performance and reduce the decoding complexity. We develop a Group Feature Aggregation module to perform grouped latent feature upsampling and aggregation, design a Pointwise Local Modulation module to improve the representation of local details, utilize a Gaussian Mixture model to improve the accuracy of rate estimation. Experimental results show that our method outperforms existing INR-based methods in terms of PNSR, and the decoding complexity of our method is only 5.4% of that of INR-based methods.

**Acknowledgments.** This study was supported by the Fundamental Research Funds for the Central Universities.

## References

1. Wallace, G.K.: The JPEG still picture compression standard. *Commun. ACM* **34**(4), 30–44 (1991)
2. Rabbani, M., Joshi, R.: An overview of the JPEG 2000 still image compression standard. *Signal Process. Image Commun.* **17**(1), 3–48 (2002)
3. Dupont, E., Goliński, A., Alizadeh, M., Teh, Y.W., Doucet, A.: COIN: compression with implicit neural representations. arXiv preprint [arXiv:2103.03123](https://arxiv.org/abs/2103.03123) (2021)
4. Strümpfer, Y., Postels, J., Yang, R., Gool, L.V., Tombari, F.: Implicit neural representations for image compression. In: Avidan, S., Brostow, G., Cisse, M., Farinella, G.M., Hassner, T. (eds.) *European Conference on Computer Vision*, pp. 74–91. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-19809-0\\_5](https://doi.org/10.1007/978-3-031-19809-0_5)
5. Girish, S., Shrivastava, A., Gupta, K.: SHACIRA: scalable hash-grid compression for implicit neural representations. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17513–17524 (2023)
6. Catania, L., Allegra, D.: NIF: a fast implicit image compression with bottleneck layers and modulated sinusoidal activations. In: *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 9022–9031 (2023)
7. Lee, J.Y., Wu, Y., Zou, C., Wang, S., Hoiem, D.: QFF: Quantized Fourier Features for neural field representations. arXiv preprint [arXiv:2212.00914](https://arxiv.org/abs/2212.00914) (2022)
8. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: *International Conference on Learning Representations* (2017)
9. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) (2013)
10. Theis, L., Shi, W., Cunningham, A., Huszár, F.: Lossy image compression with compressive autoencoders. In: *International Conference on Learning Representations* (2017)
11. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. In: *International Conference on Learning Representations* (2018)
12. Minnen, D., Ballé, J., Toderici, G.D.: Joint autoregressive and hierarchical priors for learned image compression. In: *Advances in Neural Information Processing Systems*, vol. 31 (2018)
13. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7939–7948 (2020)
14. Rahaman, N., et al.: On the spectral bias of neural networks. In: *International Conference on Machine Learning*, pp. 5301–5310. PMLR (2019)
15. Tancik, M., et al.: Fourier features let networks learn high frequency functions in low dimensional domains. *Adv. Neural. Inf. Process. Syst.* **33**, 7537–7547 (2020)
16. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. *Adv. Neural. Inf. Process. Syst.* **33**, 7462–7473 (2020)
17. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: representing scenes as neural radiance fields for view synthesis. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *ECCV 2020. LNCS*, vol. 12346, pp. 405–421. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58452-8\\_24](https://doi.org/10.1007/978-3-030-58452-8_24)

18. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.* **41**(4), 1–15 (2022)
19. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-NeRF: anti-aliased grid-based neural radiance fields. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19697–19705 (2023)
20. Chen, Y., Liu, S., Wang, X.: Learning continuous image representation with local implicit image function. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8628–8638 (2021)
21. Gao, S., et al.: Implicit diffusion models for continuous super-resolution. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10021–10030 (2023)
22. Chen, H., He, B., Wang, H., Ren, Y., Lim, S.N., Shrivastava, A.: NeRV: neural representations for videos. *Adv. Neural. Inf. Process. Syst.* **34**, 21557–21568 (2021)
23. Chen, H., Gwilliam, M., Lim, S.N., Shrivastava, A.: HNeRV: a hybrid neural representation for videos. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10270–10279 (2023)
24. Shi, W., et al.: 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)
25. Kodak lossless true color image suite (1999). <https://r0k.us/graphics/kodak/>
26. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738 (2015)
27. Bellard, F.: BPG image format (2014). <https://bellard.org/bpg/>

# Author Index

## A

Abbes, Heithem 318  
Afeefa, P. P. 164  
Agarwal, Akshay 197  
Ahsan, Shawly 107  
Akbari, Mohammad 254  
Akiba, Masaki 61  
Arias-Vergara, Tomas 154  
Aysa, Alimjan 13

## B

Benmabrouk, Yasmina 222  
Bhesra, Kirtilekha 197

## C

Cao, Qian 389  
Che, Xin 254  
Cheikh, Imen Ben 318  
Chen, Zi-Rong 183  
Cheng, Xinyuan 404  
Chu, Lingyang 254  
Cruz, Rafael Menelau O. 271

## D

Dai, Weicong 76  
Das, Pranesh 164  
de Moura, Kecia Gomes 271  
Dewan, M. Ali Akber 107  
Diaz, Moises 222  
Djaffal, Souhaila 222  
Djeddi, Chawki 222

## G

Galaida, Aleksandr 154  
Gao, Chen 91  
Gao, Liangcai 1  
Ge, De-Wu 287  
Guan, Wenbo 124  
Guo, Kai 343

## H

Han, Zhiwang 13  
Hasan, Md. Maruf 107  
Hazari, Raju 164  
He, Runlin 208  
Hjaiej, Mohamed 318  
Hoque, Mohammed Moshiul 107  
Hu, Xiaoxu 91

## I

Iwana, Brian Kenji 61

## J

Jia, Zhenhong 208  
Jiang, Nanfeng 29, 45, 303  
Jin, Lianwen 76

## K

Kopparapu, Sunil Kumar 331  
Kulyabin, Mikhail 154

## L

Lan, Haocheng 139  
Lee, Jiseok 61  
Li, Guantin 45  
Li, Guanting 29  
Li, Jian-Min 183  
Li, Mingjun 237  
Li, Peishan 373  
Li, Shaoxin 254  
Li, Xiaoqian 124  
Liao, Wenhui 76  
Lin, Chao-Qun 287  
Lin, Zening 76  
Lin, Zi-Hao 183  
Liu, Zhaoxi 208  
Lu, Jiyu 124

## M

Ma, Guangyi 373  
Ma, Jing 208

Maier, Andreas 154

Man, Wang 45

## O

Ou Yang, Jun Jie 91

Ou, Jie 139

## P

Pan, Song-Liang 303

Panda, Ashish 331

Patil, Hemant A. 356

Purohit, Ravindrakumar M. 356

## Q

Qin, Xugong 91

## S

Sabourin, Robert 271

Sokolov, Gleb 154

Srivastava, Arushi 356

Su, Feng 237

Su, Yan-Fei 287

## T

Tian, Wenhong 139

## U

Ubul, Kurban 13

## W

Wang, Da-Han 29, 45, 183, 287, 303

Wang, Jiapeng 76

Wang, Junfei 373

Wang, Zhaokun 139

Wu, Yun 45

## X

Xiao, Shun-Xin 183

Xie, Xiang 343

Xie, Xiangyu 1

Xiong, Longfei 76

Xu, Shuo 237

Xu, Xuebin 13

## Y

Yadikar, Nurbiya 13

Yang, Fan 45

Yuan, Ziwei 373

Yue, David 254

## Z

Zeng, Gangyan 91

Zhang, Dongdong 389, 404

Zhang, Fengrun 343

Zhang, Mengnan 208

Zhang, Peng 91

Zhang, Xiaolei 389, 404

Zhang, Xu-Yao 29, 45, 183, 287, 303

Zhang, Yong 254

Zhang, Yonghong 373

Zhao, Runbo 91

Zhou, Gang 208

Zhou, Jun 124

Zhou, Xinyue 29, 45

Zhou, Yuxuan 1

Zhu, ShunZhi 29

Zhu, Shunzhi 303

Zhuang, Zeming 237