

Nguyen Thai-Nghe
Thanh-Nghi Do
Salem Benferhat (Eds.)

Communications in Computer and Information Science

2191

Intelligent Systems and Data Science

Second International Conference, ISDS 2024
Nha Trang, Vietnam, November 9–10, 2024
Proceedings, Part II

Part 2

Series Editors

Gang Li , *School of Information Technology, Deakin University, Burwood, VIC, Australia*

Joaquim Filipe , *Polytechnic Institute of Setúbal, Setúbal, Portugal*

Ashish Ghosh , *Indian Statistical Institute, Kolkata, West Bengal, India*

Zhiwei Xu, *Chinese Academy of Sciences, Beijing, China*

Rationale

The CCIS series is devoted to the publication of proceedings of computer science conferences. Its aim is to efficiently disseminate original research results in informatics in printed and electronic form. While the focus is on publication of peer-reviewed full papers presenting mature work, inclusion of reviewed short papers reporting on work in progress is welcome, too. Besides globally relevant meetings with internationally representative program committees guaranteeing a strict peer-reviewing and paper selection process, conferences run by societies or of high regional or national relevance are also considered for publication.

Topics

The topical scope of CCIS spans the entire spectrum of informatics ranging from foundational topics in the theory of computing to information and communications science and technology and a broad variety of interdisciplinary application fields.

Information for Volume Editors and Authors

Publication in CCIS is free of charge. No royalties are paid, however, we offer registered conference participants temporary free access to the online version of the conference proceedings on SpringerLink (<http://link.springer.com>) by means of an http referrer from the conference website and/or a number of complimentary printed copies, as specified in the official acceptance email of the event.

CCIS proceedings can be published in time for distribution at conferences or as post-proceedings, and delivered in the form of printed books and/or electronically as USBs and/or e-content licenses for accessing proceedings at SpringerLink. Furthermore, CCIS proceedings are included in the CCIS electronic book series hosted in the SpringerLink digital library at <http://link.springer.com/bookseries/7899>. Conferences publishing in CCIS are allowed to use Online Conference Service (OCS) for managing the whole proceedings lifecycle (from submission and reviewing to preparing for publication) free of charge.

Publication process

The language of publication is exclusively English. Authors publishing in CCIS have to sign the Springer CCIS copyright transfer form, however, they are free to use their material published in CCIS for substantially changed, more elaborate subsequent publications elsewhere. For the preparation of the camera-ready papers/files, authors have to strictly adhere to the Springer CCIS Authors' Instructions and are strongly encouraged to use the CCIS LaTeX style files or templates.

Abstracting/Indexing

CCIS is abstracted/indexed in DBLP, Google Scholar, EI-Compendex, Mathematical Reviews, SCImago, Scopus. CCIS volumes are also submitted for the inclusion in ISI Proceedings.

How to start

To start the evaluation of your proposal for inclusion in the CCIS series, please send an e-mail to ccis@springer.com.

Nguyen Thai-Nghe · Thanh-Nghi Do ·
Salem Benferhat
Editors

Intelligent Systems and Data Science

Second International Conference, ISDS 2024
Nha Trang, Vietnam, November 9–10, 2024
Proceedings, Part II



Springer

Editors

Nguyen Thai-Nghe  Can Tho University
Can Tho, Vietnam

Thanh-Nghi Do  Can Tho University
Can Tho, Vietnam

Salem Benferhat  University of Artois
Lens, France

ISSN 1865-0929

ISSN 1865-0937 (electronic)

Communications in Computer and Information Science

ISBN 978-981-97-9615-1

ISBN 978-981-97-9616-8 (eBook)

<https://doi.org/10.1007/978-981-97-9616-8>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 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 Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

If disposing of this product, please recycle the paper.

Preface

Our era, marked by an accelerated transition to digital technology, is facing the challenge of managing a huge mass of information from a wide variety of sources and covering many different fields. This heterogeneous data poses major challenges in terms of collection, completion, storage, machine learning, fusion and query answering. To meet these challenges, innovative, interpretable and explainable data science and artificial intelligence technologies need to be developed, enabling these data to be fully and effectively exploited.

Following the success of the first International Conference on Intelligent Systems and Data Science (ISDS) in 2023, these proceedings contain the papers from the second ISDS (ISDS 2024), held at Nha Trang University, Vietnam from November 09–10, 2024. ISDS 2024 provided a dynamic forum in which researchers discussed problems, exchanged results, identified emerging issues and established collaborations in related areas of Intelligent Systems and Data Science.

We received 129 submissions from 9 countries to the main conference and a special session. To handle the review process, we invited 91 expert reviewers. Each paper was reviewed by at least three reviewers. We followed a single-blind process in which the identities of the reviewers were not known to the authors. This year, we used the Easy-Chair conference management service to manage the submission and selection of papers. After a rigorous review process, followed by discussions among the program chairs, 38 papers were accepted as long papers and 10 as short papers, resulting in acceptance rates of 29.46% and 7.75% for long and short papers, respectively. We are honored to have keynote talks by Masayuki Fukuzawa (Kyoto Institute of Technology, Japan) and Yu-Yen Ou (Yuan Ze University, Taiwan).

The conference program included five sessions: Intelligent Systems; Artificial Intelligence in Health Care Analytics; Artificial Intelligence in E-Commerce, Agriculture and Aquaculture; Big Data, IoT and Cloud Computing; and Natural Language Processing.

We wish to thank the other members of the organizing committee, the reviewers and the authors for the immense amount of hard work that has gone into making ISDS 2024 a success. The achievement of the conference was also contributed to by the kind devotion of many sponsors and volunteers.

We hope you enjoyed the conference!

November 2024

Nguyen Thai-Nghe
Thanh-Nghi Do
Salem Benferhat
Tran Ngoc Hai
Nguyen The Han
Nguyen Huu Hoa

Organization

Honorary Chairs

Thanh-Thuy Nguyen

University of Engineering and Technology, VNU
Hanoi, Vietnam

Quoc-Hung Pham

Nha Trang University, Vietnam

Trung-Tinh Tran

Can Tho University, Vietnam

General Chairs

The-Han Nguyen

Nha Trang University, Vietnam

Huu-Hoa Nguyen

Can Tho University, Vietnam

Ngoc-Hai Tran

Can Tho University, Vietnam

Program Chairs

Nguyen Thai-Nghe

Can Tho University, Vietnam

Thanh-Nghi Do

Can Tho University, Vietnam

Salem Benferhat

University of Artois, France

Local Organization Chairs

Nguyen-Khang Pham

Can Tho University, Vietnam

Thi Thu Thuy Pham

Nha Trang University, Vietnam

Publication Chairs

Vu-Thinh Doan

Nha Trang University, Vietnam

Thanh-Dien Tran

Can Tho University, Vietnam

Technical Program Committee

Salem Benferhat	Cril, CNRS UMR8188, Université d'Artois, France
Vo Quoc Bao Bui	Can Tho University, Vietnam
Phan Anh Cang	Vinh Long University of Technology Education, Vietnam
Duc-Hoang Chu	NATIF - National Technology Innovation Fund, Vietnam
Bob Dao	Monash University, Australia
Thanh-Nghi Do	Can Tho University, Vietnam
Thanh-Thai Do	Ho Chi Minh City University of Technology, VNU-HCM, Vietnam
Thanh-Nghi Doan	An Giang University, Vietnam
Van-Hieu Duong	Tien Giang University, Vietnam
Trung-Nghia Duong	German Research Center for Artificial Intelligence (DFKI), Germany
Tan Nghia Duong	Hanoi University of Science and Technology, Vietnam
Vatcharaporn Esichaikul	Asian Institute of Technology, Thailand
Dewan Farid	United International University, Bangladesh
Masayuki Fukuzawa	Kyoto Institute of Technology, Japan
Trong-Minh Hoang	Posts and Telecommunications Institute of Technology, Vietnam
Tomáš Horváth	Eötvös Loránd University, Hungary
Quang Nghi Huynh	Can Tho University, Vietnam
Xuan Hiep Huynh	Can Tho University, Vietnam
Phuoc Hai Huynh	An Giang University, Vietnam
Trung-Hieu Huynh	Industrial University of Ho Chi Minh City, Vietnam
Hoai-Bao Lam	Can Tho University, Vietnam
Khang Lam	Can Tho University, Vietnam
Thi-Bao-Thu Le	Ho Chi Minh City University of Technology, Vietnam
Thi-Phuong Le	EBI School of Industrial Biology, France
Thanh Van Le	Ho Chi Minh City University of Technology, Vietnam
Hong-Trang Le	Ho Chi Minh City University of Technology, VNU-HCM, Vietnam
Danh Luong	Can Tho University, Vietnam
Thanh Ma	Can Tho University, Vietnam
Tan-Ha Mai	Ho Chi Minh City University of Technology, Vietnam

Xuan-Trang Mai	Phenikaa University, Vietnam
Hung Ba Ngo	Can Tho University, Vietnam
Duc-Luu Ngo	Bac Lieu University, Vietnam
Huu-Phat Nguyen	Hanoi University of Science and Technology, Vietnam
Thi-Ai-Thao Nguyen	Ho Chi Minh City University of Technology, Vietnam
Huu-Hoa Nguyen	Can Tho University, Vietnam
Khiem Nguyen	Can Tho University, Vietnam
Van-Hoa Nguyen	An Giang University, Vietnam
Hai Thanh Nguyen	Can Tho University, Vietnam
Bao-An Nguyen	Tra Vinh University, Vietnam
Dinh Hung Nguyen	Nha Trang University, Vietnam
Chanh Nghiem Nguyen	Can Tho University, Vietnam
Vinh Nguyen	Can Tho University, Vietnam
Thanh-Khoa Nguyen	Can Tho University, Vietnam
Huu Van Long Nguyen	Can Tho University, Vietnam
Vu-Lam Nguyen	Kien Giang Community College, Vietnam
Hanh Nguyen	Can Tho University, Vietnam
Chi-Ngon Nguyen	Can Tho University, Vietnam
Khac Cuong Nguyen	Nha Trang University, Vietnam
Manh-Cuong Nguyen	Nha Trang University, Vietnam
Sinh Van Nguyen	International University - Vietnam National University HCMC, Vietnam
Quang-Hung Nguyen	Vietnam National University - Ho Chi Minh City, Vietnam
Trong-Cac Nguyen	Sao Do University, Vietnam
Sichoон Noh	Namseoul University, Korea
Atsushi Nunome	Kyoto Institute of Technology, Japan
Phi Pham	Can Tho University, Vietnam
Truong Hong Ngan Pham	Can Tho University, Vietnam
Hoang-Anh Pham	Ho Chi Minh City University of Technology, Vietnam
Nguyen-Khang Pham	Can Tho University, Vietnam
Van-Nam Pham	Nha Trang University, Vietnam
Thi-Ngoc-Diem Pham	Can Tho University, Vietnam
Thi-Thu-Thuy Pham	Nha Trang University, Vietnam
Thuong-Cang Phan	Can Tho University, Vietnam
Phuong-Lan Phan	Can Tho University, Vietnam
Trong Nhan Phan	Ho Chi Minh City University of Technology, Vietnam
Minh-Tuan Thai	Can Tho University, Vietnam

Nguyen Thai-Nghe	Can Tho University, Vietnam
Minh-Quang Tran	Ho Chi Minh City University of Technology, Vietnam
Thanh-Dien Tran	Can Tho University, Vietnam
Nguyen Minh Thu Tran	Can Tho University, Vietnam
Hoang Viet Tran	Can Tho University, Vietnam
Nguyen Minh Thai Tran	Lincoln University, New Zealand
An C. Tran	Can Tho University, Vietnam
Thien Tran	HCMC University of Foreign Languages and Information Technology, Vietnam
Hoang-Vu Tran	University of Technology and Education – University of Danang, Vietnam
Thi Que Nguyet Tran	Ho Chi Minh City University of Technology, Vietnam
Duc-Tan Tran	Phenikaa University, Vietnam
Quoc Dinh Truong	Can Tho University, Vietnam
Quang Vinh Truong	Ho Chi Minh City University of Technology, Vietnam
Thai Truong	Can Tho University, Vietnam
Quoc-Bao Truong	Can Tho University, Vietnam
Chi Truong	Ho Chi Minh City University of Technology, Vietnam
Phuoc-Hung Vo	Can Tho University, Vietnam
Van-Tai Vo	Tra Vinh University, Vietnam
Thi-Ngoc-Chau Vo	Can Tho University, Vietnam
Duc-Nghia Vu	Ho Chi Minh City University of Technology, Vietnam
	Chung-Ang University, South Korea

Secretaries

Nhut-Khang Lam	Can Tho University, Vietnam
Thi Thanh Van Tran	Nha Trang University, Vietnam

Finance Chairs

Phuong Lan Phan	Can Tho University, Vietnam
Lam Mai Chi Dinh	Can Tho University, Vietnam
Thi Thanh Van Tran	Nha Trang University, Vietnam
Thi Phuong Can	Nha Trang University, Vietnam

Contents – Part II

AI in E-Commerce, Agriculture, and Aquaculture

Experimental Study on Spectrometric Features of Mud Crabs for Automatic Internal Quality Grading	3
<i>Hai-Dang Vo, Nhut-Thanh Tran, and Masayuki Fukuzawa</i>	
Assessing Grain Size Variation Across Rice Panicles Using YOLOv8 and DeepLabv3 Models	15
<i>Van-Hoa Nguyen, Huu-Hiep Nguyen Bui, and Thanh-Phong Le</i>	
BeLightRec: A Lightweight Recommender System Enhanced with BERT	30
<i>Manh Mai Van and Tin T. Tran</i>	
Integrating Kelly Criterion with Technical Indicators for VN30 Stock Market	44
<i>Dao Lan Vy Dinh, Ngoc Hang Tran, Vo Huyen Khanh May, Hung Tran, Tran Duc Minh, Dang Thu Lan, and Van Nhan Vo</i>	

AI in Health Care Analytics

Enhancing the Efficiency of Lung Disease Classification Based on Multi-modal Fusion Model	55
<i>Thi-Diem Truong, Phuoc-Hai Huynh, Van Hoa Nguyen, and Thanh-Nghi Do</i>	
Toward Supporting Breast Cancer Diagnosis Based on Captioning Mammogram and Ultrasound Images	71
<i>Huong Hoang Luong, Hai Thanh Nguyen, and Nguyen Thai-Nghe</i>	
Violence Detection Using Skeleton Data with Graph Convolutional Networks	86
<i>Nha Tran, Hung Nguyen, Dat Ly, and Hien D. Nguyen</i>	
3D Simulation of Brain Tumor from 3D MRI Using Geometric Convolutional Neural Network and Point Clouds	98
<i>Anh-Cang Phan, Khac-Tuong Nguyen, Minh-Phuong Truong, Thi-Hong-Yen Nguyen, and Ngoc-Hoang-Quyen Nguyen</i>	

Deep Reinforcement Active Learning for Stress Recognition	113
<i>Phan Anh Ngoc, Ky Trung Nguyen, Thanh-Tung Tran, Senerath Jayatilake, and Thi Thanh Quynh Nguyen</i>	

Big Data, IoT, and Cloud Computing

A Digital Auto-Plasticity Synapse for All-Digital Resonate-and-Fire Neurons with On-chip STDP Learning	125
<i>Trung-Khanh Le, Trong-Tu Bui, and Duc-Hung Le</i>	
NEURAHOLO: A System for High-Resolution 3D Human Digitization on Holograms	138
<i>Hoang Pham Nguyen, Thai Anh Huynh Ngoc, Luat Le Gia, and Khoi Le Gia</i>	
Face Recognition for Big Data Using Search Engine for Smart System	151
<i>Phat Nguyen Huu, Duong Nguyen Tung, Khanh Nguyen Hoang Nam, and Quang Tran Minh</i>	
Synergistic Mel-Frequency Cepstral Coefficients and Short-Time Fourier Transform for Enhanced Bee States Detection Using Machine Learning	166
<i>Thi-Thu-Hong Phan</i>	
A Real-Time Method for High-Resolution Background Matting	178
<i>Tam Do-Minh, Tan Le-Thanh, My Kieu, Khuong Nguyen-An, Xuan Toan Mai, Hong Tai Tran, and Tuan-Anh Tran</i>	

Intelligent Systems

An Increased Performance of MVDR Beamformer in Diffuse Noise Field	189
<i>Quan Trong The</i>	
Fusing Models for Classifying Intangible Cultural Heritage Images in the Mekong Delta	202
<i>Minh-Tan Tran, The-Phi Pham, Nguyen Thai-Nghe, and Thanh-Nghi Do</i>	
Image Colorization with Dif-EDUNet: A Diffusion-Based Approach	213
<i>Ngoc-Giau Pham, Van-Hieu Duong, Thanh-Hai Le Tong, Hong-Ngoc Tran, and Phuoc-Hung Vo</i>	
Proposing a Solution to Improve Safety for Fiat-Shamir ZKP Scheme on Elliptic Curve	225
<i>Hanh Tran Thi, Nghi Nguyen Van, Minh Nguyen Hieu, Hien Pham Thi, Tu Le Minh, and Thi Tuyet Trinh Nguyen</i>	

Random Forest Model Parameters Optimization	237
<i>Thuy Thi Tran, Nghia Quoc Phan, and Hiep Xuan Huynh</i>	
Natural Language Processing	
Building a Q&A System to Serve Undergraduate Education at Can Tho University	251
<i>Bao-Dang Le Nguyen and Nguyen-Khang Pham</i>	
Automatically Generating a Dataset for Natural Language Inference Systems from a Knowledge Graph	264
<i>Duc Vinh Vo and Phuc Do</i>	
ViFoodNLI: A Dataset for Vietnamese Natural Language Inference in Local Cuisine	277
<i>Long Ngo Hoang Phan and Phuc Do</i>	
VNLegalEase: A Vietnamese Legal Query Chatbot	290
<i>Pham Thi Xuan Hien, Nguyen Thanh Tuong Vy, and Huu-Dung Ngo</i>	
Leveraging NLP for Multilingual Support in Academic Regulations	304
<i>Son-Tin Nguyen, Dinh-Tuan Nguyen, and Thanh-Van Le</i>	
Author Index	313

Contents – Part I

AI in E-Commerce, Agriculture, and Aquaculture

Customer Segmentation and Classification Using K-Modes Clustering with Ensemble Learning	3
<i>Shahriar Rahman Niloy, Toushif Muktashid Hasan, Md. Saiduzzaman Apu, Rakibul Hasan, Kamrul Islam Shahin, Huu-Hoa Nguyen, and Dewan Md. Farid</i>	
Explainable AI for Plant Disease Detection: Assessing Explainability in Classifying Maize Leaves Diseases with Focus Score and Ablation-CAM ...	19
<i>Luyt-Da Quach, Khang Nguyen Quoc, Chi-Ngon Nguyen, and Nguyen Thai-Nghe</i>	
Machine Learning-Based Acoustic System for Maturity Classification of Durian Fruit Before Harvesting	33
<i>Huu-Phuoc Nguyen, Viet-Lam Huynh, Thanh-Phong Duong, Chanh-Nghiem Nguyen, and Nhut-Thanh Tran</i>	
An Embedded System for Eggs Freshness Detection	47
<i>Quoc-Hung Pham, Thanh-Nhan Nguyen, Huy-Hoang Vo, Duy-Khanh Nguyen, Tan-Nhat Pham, and Nhut-Thanh Tran</i>	
Deep Learning for Fashion Consulting	59
<i>Ba Duy Nguyen, Thanh Nhan Dinh, Thi Diem Pham, and Quoc Dinh Truong</i>	

AI in Health Care Analytics

Automatic Segmentation of Masses on Mammograms Using Fuzzy Logic	69
<i>Ho-Dat Tran, Thuong-Cang Phan, Vinh-Phong Nguyen, and Anh-Cang Phan</i>	
EEC-IGE: Diagnosing Eye Diseases with DL-CNN and Integrated Gradients	83
<i>Huong Hoang Luong, Quy Thanh Lu, and Triet Minh Nguyen</i>	
Optimizing Deep Learning for Skin Disease Classification: Leveraging Bayesian Hyperparameter Tuning and Top-K Accuracy Metrics	98
<i>Toan Nguyen, Van H. Ho, and Phuc Do</i>	

Medical Image Segmentation by Improved Nested Unet	114
<i>Song-Toan Tran, Minh-Hai Le, Thai-Son Nguyen, Vinh-Khanh Nghi, and Thanh-Nguyen Nguyen</i>	

Big Data, IoT, and Cloud Computing

Traffic Flow Velocity Estimation from Single Camera Data	129
<i>Quang Tran Minh, Do Thanh Thai, Bui Tien Duc, Nguyen Van Trung, Trong Nhan Phan, Phat Nguyen Huu, Hirokazu Doi, and Fukuzawa Masayuki</i>	
Developing an AI Vision-Based Approach for Extracting Traffic Information from Images	144
<i>Quang Tran Minh, Do Thanh Thai, Bui Tien Duc, Trong Nhan Phan, and Thu Le Thi Bao</i>	

Proposal Feature Fusion Attention Network to Eliminate Fog Effects for Camera	158
<i>Nghia Duong Tan, Thien Pham Ngoc, Khang Nguyen Huu An, Minh Nguyen Nam, Quan Dang Minh, Phat Nguyen Huu, and Quang Tran Minh</i>	

A Multilevel Classification Approach for Chart Identification	173
<i>Xuan Toan Mai, Minh Tuan Kiet La, Hong Tai Tran, and Tuan-Anh Tran</i>	

Building a Wastewater Network Graph from Inspection Videos	188
<i>Minh-Thu Tran-Nguyen, Salem Benferhat, Nanee Chahinian, Carole Delenne, and Thanh-Nghi Do</i>	

Intelligent Systems

Mutual Information-Based Feature Selection for Fault Diagnosis of Induction Motor	205
<i>Ngoc-Tu Nguyen and Thanh-Tam Nguyen</i>	

FLGAN-IDS: Intrusion Detection Using GANs with Federated Learning	216
<i>Pallab Kumar Sarkar, Huu-Hoa Nguyen, and Dewan Md. Farid</i>	

ViTIP: AI-Powered Vietnamese Traditional Instrument Preservation System Using 3D Space	231
<i>Thanh Ma, Hieu Nguyen, Ky Nguyen, Xuan Nguyen, and Thanh-Nghi Do</i>	

Improved Key Player Identification Algorithm in Social Networks Using Parallel Splitting Method	246
<i>Pham Thi Thu Thuy</i>	
Diffusion-Craft Framework for Generating Vietnamese Advertising Banners	254
<i>Duc Minh Nguyen, Sieu Tran, Hao Vo, Thang Cap, Khai Thien Tran, and Tuong Le</i>	
Natural Language Processing	
SBoC: A Segment-Based Bag of Clusters Approach for Document Clustering	265
<i>Quoc-Khang Tran and Nguyen-Khang Pham</i>	
An Enhanced Solution for Multilingual Text-to-MIDI Generation	280
<i>Phi-Hung Ngo, Quoc-Vuong Pham, and Duy-Hoang Tran</i>	
RACOS: AI-Routed Chat-Voice Admission Consulting Support System	295
<i>Thanh Ma, The-Khanh Chau, Phu-An Thai, Tri-Min Tram, Khuong Huynh, and Minh-Thu Tran-Nguyen</i>	
Generating ERD and DDL Scripts from Vietnamese Natural Language Text by Using a Multi-phase	311
<i>Nguyen Dinh Thuan, Nguyen Thi My Tran, and Ton Nu Tu Quyen</i>	
Topic Modelling and Sentiment Analysis of Visitor Experience at Historical Tourism Sites	319
<i>N. M. Ngoc Bui, T. Q. Nhu Nguyen, T. H. Giang Tran, T. Doan Dang, and N. Thang Dang</i>	
Author Index	327

AI in E-Commerce, Agriculture, and Aquaculture



Experimental Study on Spectrometric Features of Mud Crabs for Automatic Internal Quality Grading

Hai-Dang Vo^{1,2}, Nhut-Thanh Tran³, and Masayuki Fukuzawa^{1(✉)}

¹ Graduate School of Science and Technology, Kyoto Institute of Technology, Kyoto 606-8585, Japan

fukuzawa@kit.ac.jp

² Faculty of Multimedia Communications, Can Tho University, Can Tho 94000, Vietnam

³ Faculty of Automation Engineering, Can Tho University, Can Tho 94000, Vietnam

Abstract. The spectrometric features of mud crabs (*Scylla paramamosain*) were analyzed aiming for future automatic grading of their internal qualities such as meat yield and ovarian fullness. Since they are currently evaluated and graded manually, there is a strong demand to automate them based on the objective spectrometric analysis. However, developing a practical spectrometric system and grading model with adequate performance and acceptable manufacturability remains challenging. In particular, the effective spectrometric datasets are essential to develop machine-learning based grading models, which require repetitive prototyping of various spectrometric systems optimized for *in vitro*, semi-*in vivo*, and *in vivo* conditions. In this study, we measured transmission spectra of essential crab components (meat, ovary, liver, and shell) under *in-vitro* condition and analyzed their spectrometric features in detail as the first prototyping stage. The spectral data were acquired with standard and concise spectrometers across various time points from just after killing up to 24 h. Their PCA results with the standard spectrometer in the wavebands from 450 to 850 nm revealed significant differences between the crab components as well as their time-course degradation, demonstrating their effectiveness as essential spectrometric features in machine learning. The results with the concise spectrometer in the wavebands from 410 to 940 nm also revealed component-specific spectra even in three-waveband domains, suggesting the applicability of conventional color CMOS camera for limited-accuracy applications as well as the usefulness of NIR wavebands to increase the performance. These findings are useful in preparing effective spectroscopic datasets for machine-learning based quality grading models, and may strongly assist the development of practical spectrometric systems.

Keywords: Automatic Mud Crab Grading · Spectrometric Features · Ovarian Fullness · Meat Yield

1 Introduction

Mud crab is a seafood with high economic value, comprising four species (*Scylla serrata*, *S. tranquebarica*, *S. paramamosain*, and *S. olivacea*) distributed throughout the world's seas [1]. Due to the increasing demand in domestic and international markets, in addition to catching mud crabs in the natural environment, mud crabs are also raised in different environments such as ponds, mangrove pens, canals, and cellular systems [2]. To reach final consumers, mud crabs are usually harvested by farmers or harvesters and then transferred to intermediate stations such as local collectors, retailers, and exporters [3]. At each station, mud crabs will be evaluated and classified based on several external quality criteria that apply to both male and female crabs, such as weight, size, claw, or missing legs. In addition, several internal quality criteria are also applied specifically to each crab gender. For example, female crabs will be evaluated for ovarian fullness and male crabs will be evaluated for meat yield.

While external quality criteria are easily assessed accurately by visual observation or weight, the internal ones such as ovarian fullness and meat yield are difficult to assess. Currently, these internal quality criteria are indirectly assessed through the firmness when pressing on the crab's shell [4]. For example, to test the meat yield of male crabs, the assessor will press on the middle segment of the bottom carapace. For female crabs, the assessor will press on both sides of the top carapace to assess ovarian fullness. Additionally, the ovarian fullness of female crabs can be evaluated by pointing the crab's mouth towards a strong light source such as sunlight or a light bulb and observing the amount of ovarian fullness from the crab's back [2]. However, these internal quality assessment methods are often performed manually. This leads to the assessment results being influenced by the subjective opinions of the assessors. Therefore, finding solutions to objectively assess the internal quality of mud crabs is necessary.

Machine vision techniques, which combine camera optics and computation including machine-learning (ML) models, are utilized and considered as a useful tool for assessing the external quality of sea-food such as counting, measuring size and color, and classifying gender [5]. For mud crabs, several research groups have also used machine vision for several tasks such as gender identification and classification [6, 7], molting crab detection [8–10], and population counting [11]. In particular, Wang et al. used machine vision, load cell combined with a neural network model to evaluate and classify the quality of river crabs based on gender, fatness, weight, and shell color of crabs [12]. Their classification results achieved an accuracy of 92.7%. Ueki et al. used a conventional camera combined with a deep neural network to classify the gender of horsehair crabs with F1 measure of about 95% [13]. Triyason et al. used machine vision to classify salted crabs based on size and color [14]. These studies showed that machine vision is very suitable for evaluating the external quality of mud crabs. However, to assess the internal quality of seafood products such as freshness, nutritional composition, and chemical residues, spectroscopic or hyperspectral techniques are considered suitable options and have been tested mainly on fish and shrimp [15–22]. Testing of these techniques for assessing the internal quality of other seafood products such as mud crabs is very limited.

Towards automatic internal quality grading of mud crabs, Tran et al. demonstrated a preliminary in-vitro study on spectrometric analysis of mud crabs [23]. The transmission spectra of several essential components of mud crab (meat, ovaries, and shell)

with a concise industrial spectrometer with limited wavebands showed different spectral characteristics in the Near-Infrared (NIR) region such as 680, 760, and 940 nm, and suggested the great potential of spectrometric analysis for the internal quality assessment of mud crabs. A research strategy was also proposed to establish a practical technique of internal quality assessment based on spectrometry combined with ML-models. The strategy included iterative prototyping of various spectrometric systems not only for further in-vitro study as the first stage but also for semi-in vivo and in vivo studies as second and third stages, aiming to obtain high-quality datasets for developing various ML models effectively.

This study aims to examine the spectrometric features of mud crabs under in-vitro condition in detail for a conclusive achievement of the first prototyping stage in our proposed strategy [23]. In addition to the concise spectrometer adopted in the preliminary in-vitro experiment, a standard spectrometer with a sufficient number of wavebands was introduced for analyzing of spectrometric feature in detail. The transmission spectra of essential crab components (meat, ovary, liver, and shell) were measured with the standard spectrometer across various time points from just after killing up to 24 h, and their spectrometric features were examined with Principal Component Analysis (PCA) technique. The transmittance spectra were also measured with the concise spectrometer and examined with the scatter plot to estimate the future applicability of simple regression models.

2 Method

2.1 Sample Preparation

The samples in this study were mature female mud crabs (*Scylla paramamosain*) which was harvested at a crab pond in Bac Lieu province, located in the Mekong Delta region. Approximately 20 commercially-standard individuals were collected. The average weight and volume were 275 g and 259 ml, respectively. The meat content was almost the same, but the ovary and liver contents varied between individuals. The average contents of ovary and liver were 12.3 and 13.2 g, respectively and their deviations were 105% and 37%. As the aim of this experiment was to obtain a typical spectrum of mud crabs, a few individuals with average ovary content were selected for this experiment. The crabs were carefully disassembled and several essential components, such as ovaries, livers, and meats, were extracted without mixing them together. Each of the extracted component was then mixed with distilled water in a 1:1 ratio to prepare a suspension, which was then filled into a glass cuvette with a size of 42.5 × 12.5 × 45 mm. The suspension was stirred before each measurement to ensure that any water-insoluble materials remained dispersed without settling. The shells were not crushed or soaked in water, but were cut to the same dimensions as the cuvette and inserted vertically into the light path of the optical system as a simple thin plate. Figure 1 shows the sample preparation process.

2.2 Optical Setup and Spectrometric Analysis

Figure 2 shows a schematic diagram and a practical implementation of the optical setup in this study. It is a simple transmission-type spectrometric system, but it has been optimized

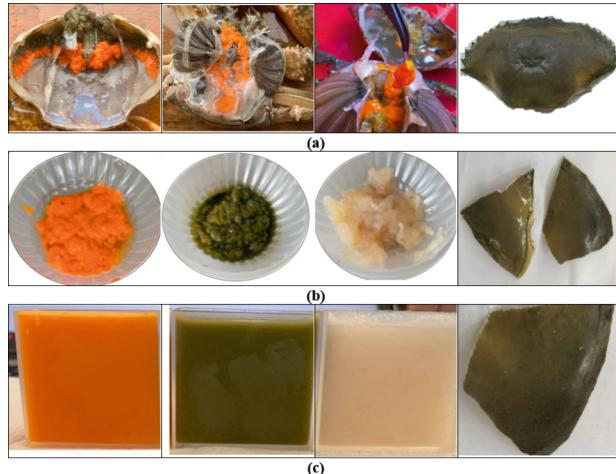


Fig. 1. Sample preparation process of crab components for spectral data acquisition: (a) A disassembled crab portions including the ovary, liver, meat, and shell, (b) Extracted crab components, and (c) Crab component suspensions in cuvettes and a shell sample.

according to the fact that the sample is a suspension, in which light propagates not only through both transmission and scattering. The optical path behind the cuvette was shortened so that the scattered light could be sufficiently introduced into the spectrometer. On the other hand, in order to reduce the effect of scattering of the incident light, the cuvette was illuminated by a collimated light source through the optical path four times as long as the cuvette thickness (12.5 mm). The light source consisted of a halogen bulb (12 VDC, 50 W) with a regulated power source and collimation lens. In preparation for future practical applications, we intentionally selected two industrial spectrometers designed to be embedded into a production system rather than the stand-alone scientific spectrometers to be installed in laboratories. One is a concise type with a limited number of wavebands (AMS AS7265X, 410 to 940 nm, 18 wavebands) that was utilized in our previous study with preliminary in-vitro experiments [23]. The other is a standard type (Hamamatsu C12880MA, 340 to 850 nm, 288 wavebands) that is introduced for the first time in this study. It is designed for embedded applications, but has a sufficient number of wavebands for the spectrometric analysis essential for the full in-vitro experiment in this study. It is well known that the wavebands selected for this study are particularly relevant for analyzing the internal components of mud crabs, such as meat, ovary, and liver. The Visible (VIS) region (450–700 nm) is well-suited for detecting color and surface characteristics, which are key to evaluating the external quality of the crab. Meanwhile, the NIR region (700–940 nm) is known for its ability to penetrate deeper into biological tissues, making it particularly useful for assessing internal qualities such as water content, fat levels, and tissue structure.

The spectral data were acquired several times with the standard-type spectrometer from a certain sample containing a particular crab component at room temperature (about 28°C) at seven different time points: 0, 2, 4, 6, 8, 22, and 24 h from just after killing the crab in order to observe both rapid early changes in quality, as well as later, more

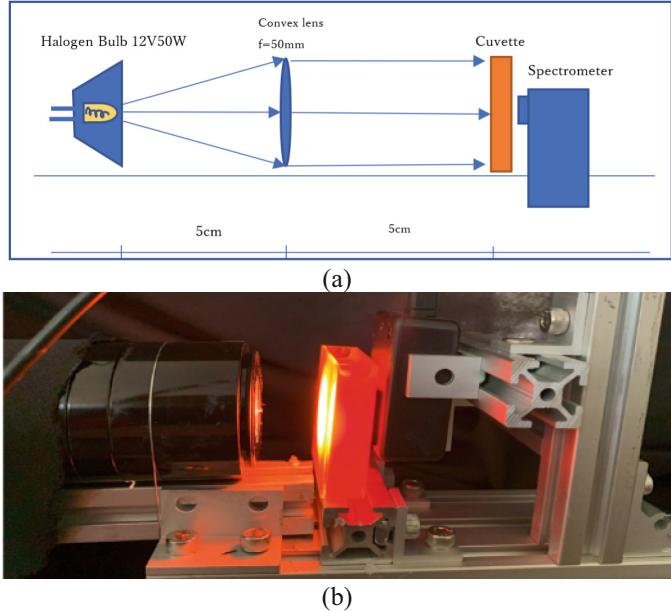


Fig. 2. Overview of the experimental setup for spectral data acquisition: (a) Schematic diagram of the optical setup used for spectral data acquisition. (b) Practical implementation of the spectral data acquisition system.

gradual degradation phases. Each signal acquisition was performed with the minimal duration to avoid warming of the sample and settling of water-insoluble materials. The spectrometer signal $I(\lambda_i)$ of a sample was acquired, and the normalized transmittance $\tau(\lambda_i)$ was obtained by Eq. (1) with the spectrometer signal $I_0(\lambda_i)$ of distilled water acquired in advance,

$$\tau(\lambda_i) = \frac{I(\lambda_i)}{I_0(\lambda_i)}, i = 1, 2, \dots, N \quad (1)$$

where λ_i is the wavelength of i -th waveband and N is the number of the spectrometer wavebands. Consequently, the relative transmittance density $\tau_r(\lambda_i)$ was calculated by Eq. (2) over the focused wavebands from $\lambda_{i_{\min}}$ to $\lambda_{i_{\max}}$,

$$\tau_r(\lambda_i) = \frac{\tau(\lambda_i)}{\sum_{i=i_{\min}}^{i_{\max}} \tau(\lambda_i)}, i = i_{\min}, \dots, i_{\max} \quad (2)$$

where $\lambda_1 \leq \lambda_{i_{\min}}, \lambda_{i_{\min}} < \lambda_{i_{\max}}, \lambda_{i_{\max}} \leq \lambda_N$. In this experiment, we selected the wavebands of $(\lambda_{i_{\min}}, \lambda_{i_{\max}}) = (450 \text{ nm}, 850 \text{ nm})$ including both VIS and NIR regions. Each $\tau_r(\lambda_i)$ was smoothed using a Savitzky-Golay filter with a window size of 25, and then its second derivative was calculated. Finally, Principal Component Analysis (PCA) was applied to the series of smoothed $\tau_r(\lambda_i)$ obtained from four samples with different components at seven different time points, and corresponding three-element PCA vector was obtained to each $\tau_r(\lambda_i)$ as a spectrometric feature.

Another series of normalized transmittance spectra $\tau(\lambda_i)$ was also obtained with the concise spectrometer from the same sample sets and at the same time points as with

the standard spectrometer. Due to the limited number of wavebands, second derivatives were not calculated, nor was PCA analysis applied. Instead, three wavebands of λ_a , λ_b and λ_c with significant differences in $\tau(\lambda_i)$ were selected, and the scatter plot of $\tau(\lambda_a)$, $\tau(\lambda_b)$ and $\tau(\lambda_c)$ was prepared to examine the spectrometric deviation between the components, aiming to estimate the applicability of simple regression models in the future.

3 Experimental Results and Discussion

3.1 Component Dependence and Time-Course Variation

Figure 3 shows typical spectra of relative transmittance density $\tau_r(\lambda_i)$ after smoothing and its second derivative, which were obtained from four crab components (ovary, liver, meat, and shell) just after killing up. The $\tau_r(\lambda_i)$ of ovary and liver revealed component-color dependent spectra, showing high values in the wavebands corresponding to orange (approx. 580–640 nm) and dark green (approx. 490–550 nm), and clearly different spectra in the NIR bands, suggesting its usefulness for their discrimination. In particular, the $\tau_r(\lambda_i)$ of liver exhibited periodic fluctuations not seen in that of ovary, reflecting in a larger amplitude of the second derivative. On the contrary, the meat $\tau_r(\lambda_i)$ showed a relatively flat spectrum compared the other components, reflecting in a smaller amplitude of the second derivative. The saturation trend of the meat $\tau_r(\lambda_i)$ in the NIR wavebands was clearly different from that of ovary and liver, which increased with wavelength. The shell $\tau_r(\lambda_i)$ showed a trend of monotonous increase with wavelength. It was most similar to that of ovary, but the difference was significant in the second derivative. From the above, it was ensured that the $\tau_r(\lambda_i)$ obtained in this experiment showed the component-specific spectra with significant differences.

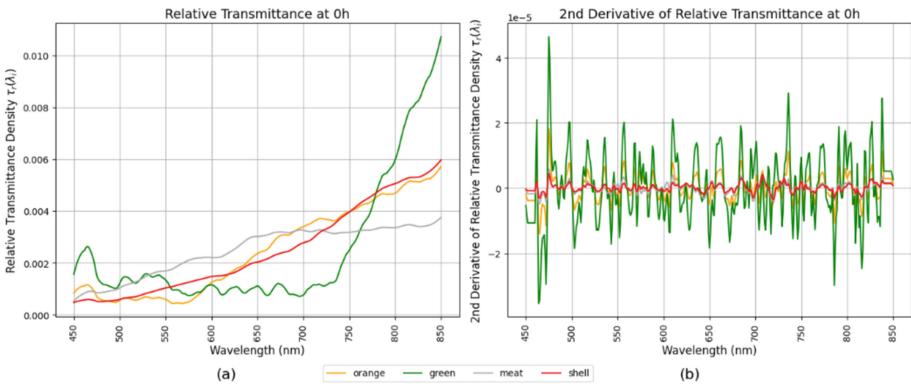


Fig. 3. (a) Typical spectra of relative transmittance density $\tau_r(\lambda_i)$ after smoothing and (b) its second derivative obtained from four crab components at 0 h just after killing up.

Figure 4 shows $\tau_r(\lambda_i)$ of four crab components obtained at seven different time points from 0 to 24 h after killing up. For all components, the $\tau_r(\lambda_i)$ revealed different

spectra over time. In particular, the spectrum-shape were drastically changed over time-course in the liver. It revealed the increase in VIS wavebands and the decrease in NIR wavebands over time. The spectrum-shape of the ovary also changed significantly with the shift of peak wavelength in NIR wavebands. On the other hand, the spectrum-shape change was moderate in the meat and shell, and the peak waveband was not varied significantly. It should be noted again that the optical system was kept constant throughout the entire acquisition process of $I(\lambda_i)$, and normalization and densification processes were designed strictly and performed carefully to obtain $\tau_r(\lambda_i)$ under a common condition. The consistency of the experimental conditions was also verified using the spectrum of distilled water. Therefore, it is obvious that the shape changes in $\tau_r(\lambda_i)$ are due to the sample itself, and it has become clarified that the experiments in this study have sufficient strictness to reflect the time-course variations of crab components.

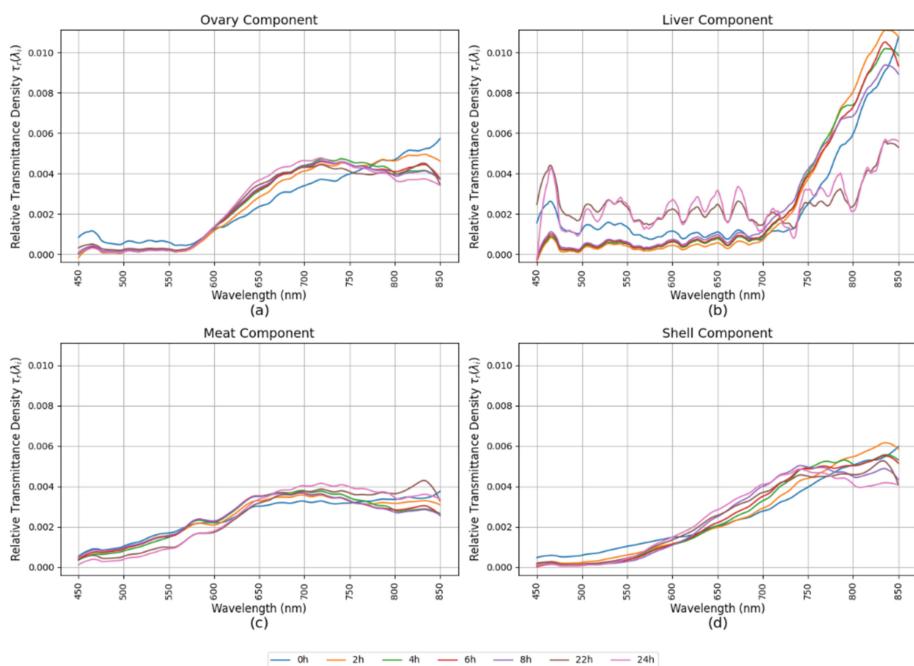


Fig. 4. The relative transmittance densities $\tau_r(\lambda_i)$ of four crab components obtained at seven different time points from 0 to 24 h after killing up.

The observed time-course variations of $\tau_r(\lambda_i)$ may provide valuable insights into the degradation process of mud crabs after killing up. Specifically, the NIR wavebands reflect variations in internal tissue composition, such as moisture loss and changes in fat content, and therefore they are indicative of the crab's freshness. In the early time points (0h to 6h), the $\tau_r(\lambda_i)$ reveal minimal variations, suggesting that the internal qualities of the crabs remain stable. However, by 24 h, the $\tau_r(\lambda_i)$ show significant shifts, particularly in the NIR region, highlighting the degradation of internal tissues and revealing the consistency with decreased freshness and quality.

3.2 Spectrometric Features from PCA Analysis

PCA was performed on the series of $\tau_r(\lambda_i)$ and their second derivatives shown in Fig. 4. Figure 5 shows three-dimensional (3D) scatter plots of three principal components. The PCA results of $\tau_r(\lambda_i)$ revealed that each crab component appeared at a different position of 3D principal component domain, and the appearance position was changed over time even within each crab component. These principal components revealed a good separation between the crab components. On the other hand, the PCA results of the second derivatives showed poorer separation between the crab components compared with that of $\tau_r(\lambda_i)$, while the appearance-position change over time was wider. This fact indicates that the PCA results of the second derivative are more sensitive to time-course variation. From these results, it was found that the principal components obtained from $\tau_r(\lambda_i)$ and its second derivative are promising as a spectrometric feature for discriminating crab components and the time-course variation of each crab component, respectively.

This finding is particularly relevant for the development of automatic grading systems, as it suggests that the quality and freshness of crabs could be assessed non-destructively and repeatedly using a spectrometric system. By combining this time-sensitive spectrometric features with an appropriate ML-model, it would be possible to assess the quality and freshness of crabs at different stages of shipping. It could be highly practical in seafood supply chains, where maintaining the internal quality of crabs is essential for maximizing shelf-life and consumer satisfaction.

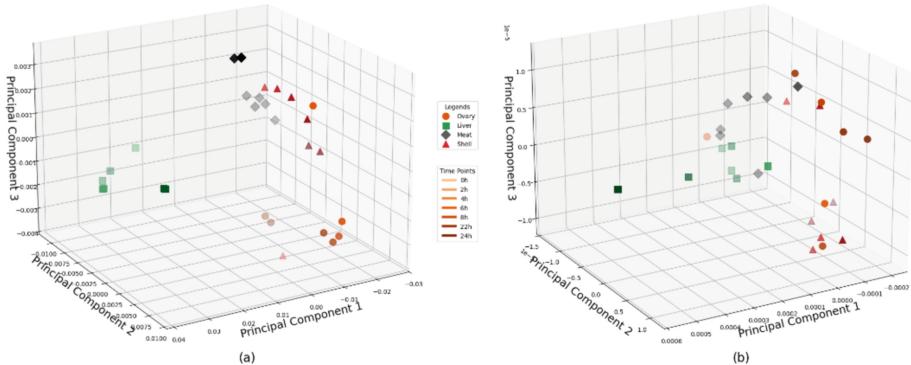


Fig. 5. 3D scatter plots of three principal components: (a) PCA performed on $\tau_r(\lambda_i)$ obtained from four crab components at seven time points, (b) PCA conducted after applying the second derivative to $\tau_r(\lambda_i)$.

3.3 Spectrometric Features Availability and Limited Wavebands

The results of obtaining a series of $\tau(\lambda_i)$ with the concise spectrometer and their 3D scatter-plot observation at specific wavebands are shown in Fig. 6. Figure 6(a) shows $\tau(\lambda_i)$ of 18 wavebands obtained from four crab components just after killing up. As with the standard spectrometer, the component-specific spectra were revealed. Figure 6(b) is a

3D scatter plot of $\tau(460)$, $\tau(535)$, and $\tau(680)$, which correspond to the RGB wavebands of the Bayer filter in conventional color CMOS camera. These spectral components exhibited good separation between crab components. Figure 6(c) is a 3D scatter plot of $\tau(460)$, $\tau(680)$, and $\tau(810)$, which correspond to a NIR waveband in addition to the B and R ones of conventional CMOS camera. It revealed better separation between the crab components than that shown in Fig. 6(b), revealing the improvement by adding NIR waveband.

It is well known that practical analysis of crab components will handle a composite spectrum of multiple components, and therefore a spectrometer with a limited number of wavebands has limitations in advanced applications. However, this experiment suggested the applicability of conventional color CMOS camera in simple applications requiring only limited precision and accuracy. The selected-waveband signals revealed a large spectrometric deviation between the crab components and it implied the applicability of simple regression models such as partial least squares regression. Furthermore, the usefulness of NIR waveband in crab component analysis was strongly suggested.

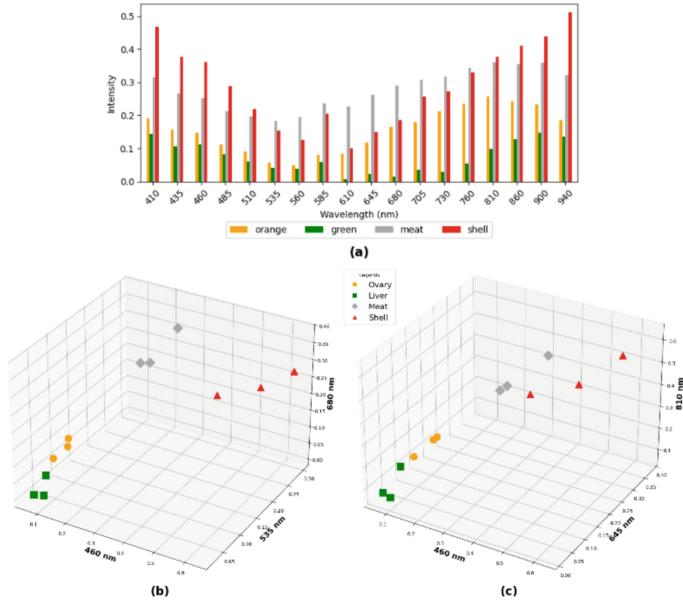


Fig. 6. Experimental results with the concise spectrometer: (a) $\tau(\lambda_i)$ of 18 wavebands obtained from four crab components just after killing up, (b) 3D scatter plot of $\tau(460)$, $\tau(535)$, and $\tau(680)$, and (c) 3D scatter plot of $\tau(460)$, $\tau(680)$, and $\tau(810)$.

4 Conclusion

Aiming for future automatic grading of their internal qualities such as meat yield and ovarian fullness, the spectrometric features of mud crabs (*Scylla paramamosain*) were experimentally studied in detail under in-vitro condition by PCA of the relative transmittance density $\tau_r(\lambda_i)$ using two industrial spectrometers with an optimized optical setup. The series of $\tau_r(\lambda_i)$ obtained with the standard spectrometer from four essential crab components of the ovary, liver, meat, and shell at seven time points from just after killing up to 24 h showed the component-specific spectra with a significant difference as well as the time-course variation. It was found from the PCA results that the principal components obtained from $\tau_r(\lambda_i)$ and its second derivative are promising as a spectrometric feature for discriminating crab components and the time-course variation of each crab component, respectively. These findings suggest the potential of a practical spectrometric system with an appropriate ML model for non-destructive and repetitive assessment of crab quality and freshness. The series of (λ_i) with the concise spectrometer also revealed component-specific spectra even in three-waveband domains, suggesting the applicability of conventional color CMOS camera with an appropriate regression model for simple applications requiring only limited precision and accuracy. The usefulness of NIR wavebands was also suggested to increase the performance.

Since these results provided insights into promising spectroscopic features for assessing crab quality and freshness, this study must be sufficient as a conclusive achievement of the first stage (in-vitro study) of our proposed research strategy [23]. The integration of time-sensitive spectrometric features with ML models holds great potential for non-destructive and repetitive assessment of crab quality and freshness at different stages of shipping in seafood supply chains.

As mentioned in our proposed research strategy, future studies include a semi-in-vivo study to obtain $\tau_r(\lambda_i)$ of each crab ‘portion’ which includes different components and to analyze its correlation with the contained components in the portion based on ML-based techniques. The findings in this study must be very useful in preparing effective spectroscopic datasets for such ML-models, and may strongly assist the development of practical spectrometric systems.

Acknowledgments. This work was supported by JSPS Core-to-Core Program (grant number: JPJSCCB20230005).

References

- Keenan, C.P., Davie, P.J.F., Mann, D.L.: A revision of the genus *Scylla* de Haan, 1833 (Crustacea: Decapoda: Brachyura: Portunidae). *Raffles Bull. Zool.* **46**(1), 217–245 (1998)
- Colin, S., Alessandro, L.: Mud crab aquaculture - A practical manual, FAO (2011). <https://www.fao.org/3/ba0110e/ba0110e.pdf>
- Bhuiyan, M.S., Shamsuzzaman, M.M., Hossain, M.M., Mitu, S.J., Mozumder, M.M.H.: Mud crab (*Scylla serrata* Forsskal 1775) value chain analysis in the Khulna region of Bangladesh. *Aquac. Fish.* **6**(3), 330–336 (2021). <https://doi.org/10.1016/j.aaf.2021.01.004>
- C-AID Consultants. Australian Industry Live Mud Crab Grading Scheme - Version 3 (2016). <https://www.c-aid.com.au/wp-content/uploads/Mud-Crab-Grading-Scheme-V3-2016.pdf>

5. Zion, B.: The use of computer vision technologies in aquaculture – a review. *Comput. Electron. Agric.* **88**, 125–132 (2012). <https://doi.org/10.1016/j.compag.2012.07.010>
6. Cui, Y., Pan, T., Chen, S., Zou, X.: A gender classification method for Chinese mitten crab using deep convolutional neural network. *Multimed. Tools Appl.* **79**(11–12), 7669–7684 (2020). <https://doi.org/10.1007/s11042-019-08355-w>
7. Chen, X., Zhang, Y., Li, D., Duan, Q.: Chinese mitten crab detection and gender classification method based on GMNet-YOLOv4. *Comput. Electron. Agric.* **214**, 108318 (2023). <https://doi.org/10.1016/j.compag.2023.108318>
8. Baharuddin, R.R., Niswar, M., Ilham, A.A., Kashihara, S.: Crab molting identification using machine learning classifiers. In: 2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE) , pp. 295–300. IEEE (2022). <https://doi.org/10.1109/ISMODE53584.2022.9743136>
9. Tang, C., Zhang, G., Hu, H., Wei, P., Duan, Z., Qian, Y.: An improved YOLOv3 algorithm to detect molting in swimming crabs against a complex background. *Aquac. Eng.* **91**, 102115 (2020). <https://doi.org/10.1016/j.aquaeng.2020.102115>
10. Zhang, Z., Liu, F., He, X., Wu, X., Xu, M., Feng, S.: Soft-shell crab detection model based on YOLOF. *Aquac. Int.* (2024). <https://doi.org/10.1007/s10499-024-01426-2>
11. Zakiyabarsi, F., Niswar, M., Zainuddin, Z.: Crab larvae counter using image processing. *EPI Int. J. Eng.* **2**(2), 127–131 (2019). <https://doi.org/10.25042/epi-ije.082019.06>
12. Wang, H., et al.: Quality grading of river crabs based on machine vision and GA-BPNN. *Sensors.* **23**(11), 5317 (2023). <https://doi.org/10.3390/s23115317>
13. Ueki, Y., Toyota, K., Ohira, T., Takeuchi, K., Satake, S.: Gender identification of the horsehair crab, *Erimacrus isenbeckii* (Brandt, 1848), by image recognition with a deep neural network. *Sci. Rep.* **13**(1), 19190 (2023). <https://doi.org/10.1038/s41598-023-46606-x>
14. Triyason, T., Tassanaviboon, A., Puangthamawathanakun, B.: Salted crab grading using computer vision. In: 2023 27th International Computer Science and Engineering Conference (ICSEC) , pp. 310–314. IEEE (2023). <https://doi.org/10.1109/ICSEC59635.2023.10329738>
15. Dixit, Y., Reis, M.M.: Hyperspectral imaging for assessment of total fat in salmon fillets: a comparison between benchtop and snapshot systems. *J. Food Eng.* **336**, 111212 (2023). <https://doi.org/10.1016/j.jfoodeng.2022.111212>
16. He, H.-J., Wu, D., Sun, D.-W.: Nondestructive spectroscopic and imaging techniques for quality evaluation and assessment of fish and fish products. *Crit. Rev. Food Sci. Nutr.* **55**(6), 864–886 (2015). <https://doi.org/10.1080/10408398.2012.746638>
17. Xu, J.-L., Riccioli, C., Sun, D.-W.: Development of an alternative technique for rapid and accurate determination of fish caloric density based on hyperspectral imaging. *J. Food Eng.* **190**, 185–194 (2016). <https://doi.org/10.1016/j.jfoodeng.2016.06.007>
18. Yu, X., Wang, J., Wen, S., Yang, J., Zhang, F.: A deep learning based feature extraction method on hyperspectral images for nondestructive prediction of TVB-N content in Pacific white shrimp (*Litopenaeus vannamei*). *Biosyst. Eng.* **178**, 244–255 (2019). <https://doi.org/10.1016/j.biosystemseng.2018.11.018>
19. Wu, D., Sun, D.-W.: Advanced applications of hyperspectral imaging technology for food quality and safety analysis and assessment: a review — Part II: applications. *Innov. Food Sci. Emerg. Technol.* **19**, 15–28 (2013). <https://doi.org/10.1016/j.ifset.2013.04.016>
20. Rahman, A., Kondo, N., Ogawa, Y., Suzuki, T., Shirataki, Y., Wakita, Y.: Prediction of K value for fish flesh based on ultraviolet-visible spectroscopy of fish eye fluid using partial least squares regression. *Comput. Electron. Agric.* **117**, 149–153 (2015). <https://doi.org/10.1016/j.compag.2015.07.018>
21. Zhang, H., et al.: Non-destructive determination of fat and moisture contents in Salmon (*Salmo salar*) fillets using near-infrared hyperspectral imaging coupled with spectral and textural features. *J. Food Compos. Anal.* **92**, 103567 (2020). <https://doi.org/10.1016/j.jfca.2020.103567>

22. Shao, Y., Shi, Y., Wang, K., Li, F., Zhou, G., Xuan, G.: Detection of small yellow croaker freshness by hyperspectral imaging. *J. Food Compos. Anal.* **115**, 104980 (2023). <https://doi.org/10.1016/j.jfca.2022.104980>
23. Tran, N.-T., Vo, H.-D., Ngo, C.-T., Nguyen, Q.-H., Fukuzawa, M.: Towards automatic internal quality grading of mud crabs: a preliminary study on spectrometric analysis, pp. 3–14 (2024). https://doi.org/10.1007/978-981-99-7666-9_1



Assessing Grain Size Variation Across Rice Panicles Using YOLOv8 and DeepLabv3 Models

Van-Hoa Nguyen^{1,3} , Huu-Hiep Nguyen Bui^{1,3} , and Thanh-Phong Le^{2,3}

¹ Faculty of Information Technology, An Giang University, Long Xuyên 88000, An Giang, Vietnam

{nvhoa, nbhhiep}@agu.edu.vn

² Climate Change Institute, An Giang University, Long Xuyên 88000, An Giang, Vietnam
ltpthong@agu.edu.vn

³ Vietnam National University Ho Chi Minh City, Ho Chi Minh City 70000, Vietnam

Abstract. The Mekong Delta is Vietnam's most crucial rice production region. Rice grain shape quality plays a pivotal role in establishing commercial standards. Grain morphology, characterized by length and width, is one of the key rice quality parameters. This characteristic could be influenced by grain position within the panicle. Numerous studies have employed image processing techniques and machine learning models for rice grain detection, counting, measurement, and classification. However, no prior research has evaluated grain size variation across rice panicles. This study utilizes two machine learning models, YOLOv8 and DeepLabv3, to detect and segment rice grains and brown rice grains (after dehulling) in three panicle sections: top, middle, and bottom. From grain segmentations, grain sizes (length and width) were calculated as rectangles around the convex hull which are the smallest convex boundaries. Experimental results on 11 rice varieties of varied grain lengths, planted in the Vietnam Mekong Delta region, indicate no significant size differences between the sections. YOLOv8 provides better grain detection results compared to DeepLabv3 for images with adjacent grains, while DeepLabv3 gives more accurate grain boundary segmentation results. In addition, we used ANOVA to compare the evaluation results acquired using the two deep learning models to those obtained using the ruler method.

Keywords: Rice panicle · grain size measurement · YOLOv8 · DeepLabv3

1 Introduction

The Mekong Delta is the most important rice production region in Vietnam, significantly contributing to national food security and the second largest rice exporter worldwide. Rice production and export is an extremely complex chain involving breeding, propagation, production, purchasing, and export. Maintaining quality throughout this process is crucial, as deficiencies at any stage can negatively impact the rice's commercial value. Grain size is a key determinant of rice type and plays a pivotal role in establishing commercial standards. However, besides the external factors mentioned above, the internal

factor of the rice variety has not been deeply studied and conclusively determined. This factor is the difference in grain positions on the same panicle of each rice variety regarding the length and width of the rice grains.

There is a conflict between rice export companies and rice suppliers regarding the differences in grain size at different positions on the panicle, leading to a lack of consensus on the acceptable grain length ratio in purchase contracts. Researchers have long asserted that external morphology results from the interaction between plant genes and the environment. In the case of rice, because the flowering process of rice does not occur simultaneously and there are differences between the top and bottom of each panicle, the impact of the environment occurs on the same rice panicle, where grains in favorable environmental and lighting conditions at the top of the panicle develop better external morphology than grains in less favorable positions. The theory of this research is that differences in grain length and width morphology increase the variation in grain length and width within the same panicle in rice variety.

Traditional morphological evaluation of rice grains relies heavily on manual measurement methods. While offering high accuracy, these methods are time-consuming and labor-intensive, limiting their scalability. Computer vision applications in rice grain morphology analysis have become possible due to rapid technological improvement. This technology facilitates automation of image data collection and processing, leading to reduced errors, significant time savings, and improved analysis efficiency. The primary goal of research in this field is to automate the measurement and analysis of the rice grain dimensional parameters, such as length, width, and area. Deep learning, particularly Convolutional Neural Networks (CNNs), plays a crucial role in automating rice grain size analysis. By accurately identifying and segmenting individual grains within images, CNNs enable precise measurement of length, width, and area. This approach significantly improves efficiency and accuracy compared to traditional manual methods.

Furthermore, leveraging advancements in computer vision and deep learning, the application of these technologies not only improves the accuracy and speed of rice grain assessments but also helps establish new standards in quality control. These systems can analyze the size, shape, color, and surface characteristics of rice grains in a detailed and consistent manner, facilitating the development of high-quality rice varieties and enhancing the value of white rice products. Research on rice panicle image analysis involves counting rice panicles from UAV images [1], calculating the spikelet number per rice panicle [2], detecting and counting grains per rice panicle [3–5], and assessing rice maturity [6]. In addition, research related to detecting, counting, measuring and classifying rice grains includes measuring the size of rice grains [7, 8], white rice grains [9], detecting filled/unfilled of rice grains [8, 10], classifying rice grain varieties [9–11, 16], classifying rice grain quality [14, 16, 17].

In this study, we aimed to investigate the variation in rice grain size across different positions within a panicle. We employed YOLOv8 and DeepLabv3 to measure the size of both rice and brown rice grains. This serves as valuable evidence for rice breeders, helping them understand the variation in grain morphology within the same panicle and enabling them to develop improved rice varieties. Additionally, the research provides indirect evidence of the effect of the environment on seed size with export companies and rice suppliers.

2 Related Works

Rice grain size, or length and width, is a characteristic of the rice variety. However, in studying this characteristic, researchers typically focus only on the average value of the rice grains and select them randomly for measurement. The standard evaluation system for rice of IRRI requires a rice sample size of 10 seeds at the mature grain stage to measure grain length and grain width [18]. Bangladeshi scientists also use the method of random seed selection to measure the physical properties of rice grains, including length, width, and thickness, for eight improved rice varieties [19]. Research from the Czech University of Life Sciences Prague employed a Vernier caliper and randomly selected 20 seeds from each cultivar for measurement [20]. To evaluate the physical properties of rice varieties grown in the Mekong Delta, Vietnamese researchers also use a method of random selection of 100 seeds and a caliper with an accuracy of 0.1 cm for measurement [21]. This indicates that scientists have not paid much attention to the variation in grain shape within the same rice variety, particularly the position of the grains on the panicle. Therefore, in case of any queries from rice trading businesses, scientists may not be able to provide explanations. This is a scientific shortcoming and requires further research.

Research on analyzing images of rice panicles focuses on tasks like counting panicles in paddy fields, estimating spikelet numbers per rice panicle, and counting seeds/grains per rice panicle. Zhou et al. used deep learning approaches for image segmentation to automate the counting of rice panicles in UAV images. They utilized a region-based fully convolutional network (R-FCN) to achieve excellent accuracy in panicle identification [1]. Zhao et al. employed an interdisciplinary method that integrated image analysis with a 5-point calibration model to rapidly estimate the spikelet number per rice panicle [2]. Deng et al. counted grains/seeds on a rice panicle using deep learning methods such as Faster R-CNN [3]. Wu et al. used linear regression and deep learning models to count the grain number per rice panicle [4]. Lu et al. proposed a high-throughput, separation-free method for extracting on-panicle rice grain phenotyping traits using visible light scanning imaging and the Faster R-CNN model for grain detection [5]. Wang et al. used the random forest regression algorithm to estimate rice maturity with 22 color features representing the greenness of crop leaves extracted from the rice panicle images [6].

Furthermore, studies on rice grain analysis, such as detection and classification, employ a variety of methodologies, including image processing, feature extraction and classification, and deep learning. Ruslan et al. used LabVIEW to extract the morphological features of four varieties of rice grains cultivated in Malaysia. The morphological features are length, width, aspect ratio and rectangular aspect ratio [7]. Feng et al. used image processing techniques such as backlight photography to capture a grayscale image of a group of rice grains, and then applied a clustering algorithm to distinguish between filled and unfilled rice grains based on their grayscale values [8]. The authors also obtained the length and width of seeds, which are the length and width of optimal enclosing rectangles for seeds. Birla et al. proposed a method for calculating the size of rice grains using image processing. This method detects rice objects and calculates the white rice grain size based on the number of pixels. It also includes steps for detecting chalky and broken rice grains [9].

For the approach based on the extraction feature and classification of rice grains, Ansari et al. proposed an inspection method to classify three varieties of rice seeds by capturing images with an RGB camera. They extracted 20 features and classified these rice varieties using traditional models such as SVM and KNN [11]. Tran Thi Kim Nga et al. have extracted features of 17 varieties of rice grains planted in Vietnam. These features include morphologies, color, and texture that are classified by SVM, combining binary particle swarm optimization and the SVM models [12]. Cinar and Koklu determined features of five rice varieties of the same brand, such as morphology, shape, and color [17]. These features are extracted based on the image processing techniques. According to the authors, the most effective and specific features are roundness, compactness, shape factor, aspect ratio and eccentricity.

For the methods based on deep learning for classifying rice grains, a deep learning model incorporating a transformer encoder and coordinate attention module was developed for detecting, counting, and classifying rice and white rice grains, using VGG16, YOLO, and ResNet50. Tran Thi Kim Nga et al. used VGG16 and ResNet50 models to classify 17 rice grain varieties [13]. The TCLE–YOLOv5 model has been used by Zou for detecting and counting rice grains [15]. For evaluating white rice grain quality, Nguyen Hong Son and Nguyen Thai Nghe used a CNN model to recognize and to classify whole and broken rice grains [16]. Gilanie et al. used convolutional neural networks-based models such as VGG-19, ResNet50, and Inception-v3 to classify seven brown rice varieties cultivated in Pakistan [15].

3 Materials and Methods

3.1 Data Processing

Image Acquisition. For the objective was to detect, count grains, measure grain shape and classify rice grains on panicles, we selected 11 varieties that are 6 rice traditional varieties selected under the project with code C2022–16-09 funded by VNUHCM, two new varieties were bred by Climate Change Institute (F54 and F56) and three rice varieties commonly grown in the Mekong Delta include DT8 (Đài Thom 8), OM18 and OM380. 6 varieties selected from traditional rice varieties including ANP (Nàng Tây Đùm), G3 (Hai Nguyên lứa), G7 (Lùn Cắn), G13 (Nàng Thom), G17 (Tài Nguyên) and G18 (Tráng bà lớn) [22]. In our study, panicles of rice varieties were separated into three parts depending on their length, and each part was labeled top, middle and bottom of panicles as shown in Fig. 1. Rice grains in each part of the panicles were separated from spikelets and scanned together in image at resolution of 1200 dots per inch with an office scanner (Brother, DCP-7060D). Brown rice grains were obtained by dehulling from the rice grains of each panicle section and scanned at the same resolution. The scanner was covered with green paper to avoid light noise caused by reflection and projection [4].

Image Augmentation and Data Labeling. Deep neural networks require a large amount of training data for optimal performance. Data augmentation, a technique that artificially expands datasets by manipulating existing images (e.g., adjusting brightness, adding noise, rotating images), can significantly improve model performance. In this study, we specifically employed multi-angle image rotation to increase the number of

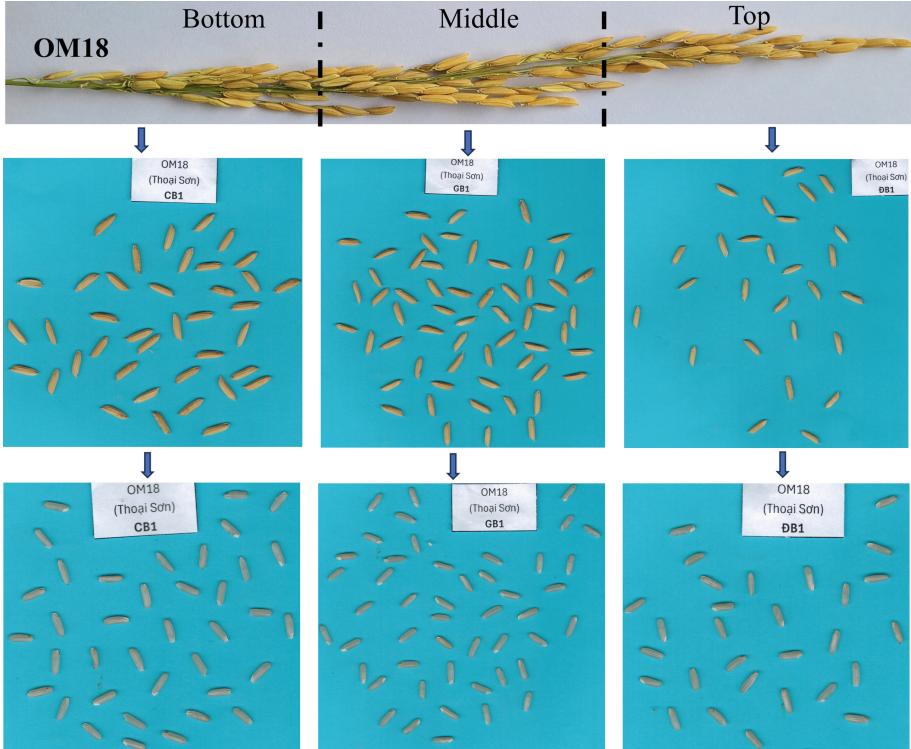


Fig. 1. A division of the OM18 rice panicle into top, middle, and bottom sections, with images showing rice and brown rice grains for each part.

training samples for rice grain detection. The ground truth (GT) images for the object detection model were bounding boxes labeled. To obtain the labeled samples, the images in the dataset were manually tagged using the Anylabeling annotation software [23]. Red boundary boxes were used to designate the areas containing rice and brown rice grains, as seen in Fig. 2. The polygon annotation results were saved in a json file and used to train the model using the training dataset and to calculate the model's performance with the validation and test datasets.

3.2 Deep Learning Models

To automatically classify rice and brown rice grains, many machine learning-based systems begin by extracting features and then training a classifier. This manual method is often time intensive. However, the rapid progress of deep learning models has enabled the automation of the processes of representation and feature learning. In this study, we proposed to use two variants of the CNN model: YOLOv8 and DeepLabv3.

YOLOv8. YOLO is a Deep Convolutional Neural Network (DCNN) model designed for real-time object detection. YOLOv8 (version 8) represents the cutting edge of real-time

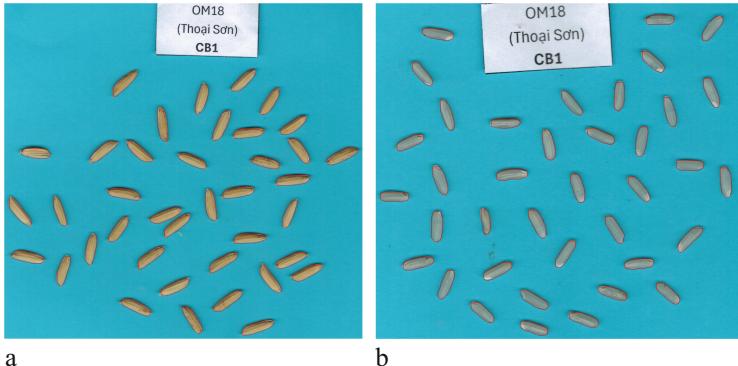


Fig. 2. Sample labeled images: (a) Sample labeled image of rice grains from the middle panicle; (b) Sample labeled image of brown rice grains from the middle panicle.

object detection technology [24]. YOLOv8 integrates advanced deep learning techniques and optimized architectures to enhance both performance and computational efficiency. YOLOv8 architecture is a hybrid approach combining convolutional layers with transformer components. This fusion allows YOLOv8 to effectively capture both local and global features in images. This model employs an adaptive anchor-free mechanism, which eliminates the need for predefined anchor boxes, making it more flexible and capable of detecting objects of various shapes and sizes with higher precision. YOLOv8 introduces a novel spatial pyramid pooling layer that enhances feature extraction by aggregating context information at multiple scales. This layer helps the model to maintain high accuracy even when dealing with objects at different scales and aspect ratios. Furthermore, YOLOv8 incorporates advanced techniques like path aggregation network and feature pyramid network, which improve feature fusion from different layers, enhancing the detection of small and overlapping objects.

DeepLabv3. DeepLabv3 is a Deep Convolutional Neural Network (DCNN) designed for semantic image segmentation, developed by researchers at Google and the University of Oxford [25]. This model addresses the challenge of accurately segmenting objects in complex scenes by leveraging several advanced techniques. It utilizes atrous convolutions to capture features at multiple scales, expanding the receptive field without increasing the number of parameters. Additionally, DeepLabv3 introduces atrous spatial pyramid pooling, which applies atrous convolutions with varying dilation rates in parallel to aggregate multi-scale information. This enhances the model's ability to recognize objects of different sizes and shapes within the same image. Moreover, DeepLabv3 uses powerful backbone networks such as ResNet and Xception. These backbones are augmented with residual connections, which facilitate the training of deep architectures by ensuring efficient gradient flow and mitigating the vanishing gradient problem.

3.3 Experimental Design

To achieve the objective of our study, we proposed the whole process of rice detection, measurement and classification of brown rice grains as shown in Fig. 3. Following

image acquisition, dataset creation, and annotation, we trained two deep learning models: YOLOv8 and DeepLabv3. These models were used to detect and segment rice and brown rice grains. Accurately segmenting grains is crucial for tasks like rice variety analysis, which often require grain counting and size measurement on the panicle. We calculated grain size (length and width) using the bounding polygons generated by the segmentation models. Broken and dead brown rice grains were excluded from size measurements by employing a classification model with three categories: sound, broken, and dead grains. To validate the accuracy of our size measurements, we compared them with measurements obtained using a ruler.

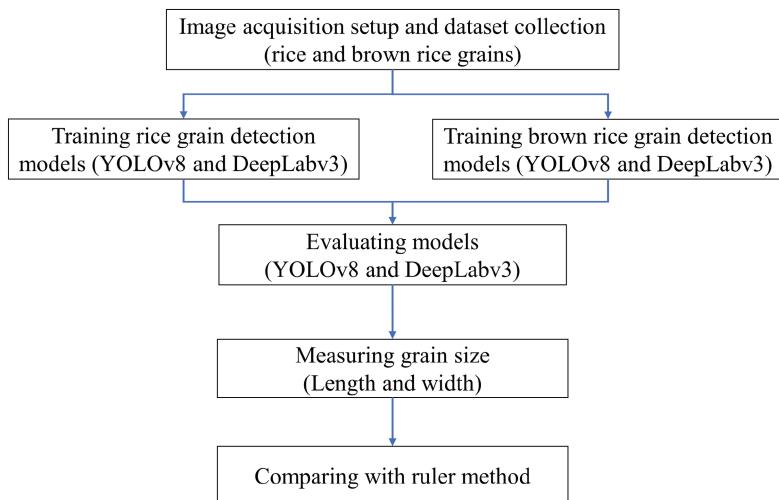


Fig. 3. Research design was proposed in our work

The Backbone of Deep Learning Models. DeepLabv3 focuses on classifying each pixel in an image to a specific class. DeepLabv3 often utilizes atrous convolutions or architectures better suited for capturing spatial context. In this study, we used the ResNet-50 backbone in both DeepLabv3. YOLOv8, on the other hand, focuses on real-time object detection with a custom CSPDarknet53 backbone.

Size Measurement of Rice and Brown Rice Grains. To determine the size of a rice and brown rice grain from the segmentation output of deep learning models, its length and width were measured using the convex hull method. First, the segmentation output was converted into a binary mask to isolate the grain shape, followed by extracting contours to identify the grain boundary. Each contour was then encapsulated by its convex hull, simplifying the shape into its smallest convex boundary. Next, the dimensions (length and width) of the grain were obtained by fitting a rotated rectangle around the convex hull. This rectangle's dimensions were adjusted to ensure the longer dimension was assigned as the width, considering any rotation. Finally, the pixel values along the length and width of the rectangle were converted to millimeters using predefined image resolution values, enabling accurate measurement of the grain's physical dimensions. This method

provides a robust framework for quantifying grain size from segmentation outputs [8, 26, 27].

3.4 Evaluation Metrics

Detecting rice grains involves balancing accuracy and speed. To evaluate model performance, this study uses precision (P), recall (R), Average Precision (AP).

Precision is the ratio of correctly identified positive cases to the total number of cases classified as positive within the entire sample. Accuracy is determined using Formula (1).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (1)$$

Recall represents the proportion of actual positive cases to the predicted positive cases. Recall is calculated using Formula (2)

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

As expressed in Eqs. (1) and (2), true positive (TP) indicates the number of correct predictions as positive samples, and false positive (FP) represents the number of incorrect predictions as positive samples. False negative (FN) represents the number of incorrect predictions as negative samples.

AP (Average Precision) represents the area under the precision–recall curve enclosed by the curve and the coordinate axis. It is calculated using Formula (3):

$$\text{AP} = \int_0^1 (\text{Precision} \times \text{Recall}) dx \quad (3)$$

4 Results

4.1 Experimental Setup

For each of the 11 rice varieties, we selected 10 panicles, resulting in a dataset of 330 rice grain images and 330 brown rice grain images. Table 1 provides a breakdown of grain counts for each variety. The number of rice and brown rice grains in the images ranged from 5 to 65. The total number of rice grains across all images was 10,311. These were distributed within the panicles as follows: 1,930 in the top sections, 4,704 in the middle sections, and 3,677 in the bottom sections. The total number of brown rice grains was slightly higher at 8,471. Their distribution within the panicles was: 1,850 in the top sections, 3,615 in the middle sections, and 3,006 in the bottom sections. The number of rice and brown rice grains differs because some brown rice grains are broken or destroyed during the dehulling process.

The annotation results were stored in files and used for two purposes: training the model with the training dataset and evaluating the model’s performance on the validation

Table 1. The number of rice and brown rice grains for each variety.

Varieties	Rice grains			Brown rice grains		
	Top	Middel	Bottom	Top	Middel	Bottom
F54	111	429	281	115	369	271
F56	136	476	472	136	336	309
ANP	198	376	348	184	349	316
G3	126	425	318	124	240	238
G7	116	417	390	116	301	290
G13	141	351	288	123	266	204
G17	194	352	180	194	289	179
G18	129	318	315	129	192	181
DT8 (TS)	182	495	313	182	390	313
OM18 (TS)	415	634	384	365	452	336
OM380 (TS)	182	431	388	182	431	369
Total	1,930	4,704	3,677	1,850	3,615	3,006

and test datasets. The rice and brown rice grain images were randomly allocated to the training, validation, and test datasets in a 6:2:2 ratio to ensure a representative sample for each stage. Additionally, we applied a multi-angle image rotation technique with a 90*-degree rotation to all images in the training dataset to further augment the data and improve model robustness.

The experiment was carried out using Ubuntu 22. The processor model is Intel(R) Core (TM) i5–13500 @2.5GHz. The GPU model is NVIDIA P104–100. The machine contains 32GB of RAM and a 1TB mechanical hard disk. The programming language used is Python 3.10. The deep learning framework utilized is PyTorch 2.3.1. The GPU acceleration libraries used are CUDA 12.5.

The parameters of two deep learning models are shown in Table 2. Due to the relatively small size of rice grains, we employed a large image size of 1280x1280 pixels for both models to capture sufficient detail. To ensure model stability during training, lower learning rates of 0.01 and 0.0001 were used.

Table 2. The parameters of two models.

Model	Image size	Learning rate	Batch size	Epochs
YOLOv8	1280 × 1280	0.01	4	100
DeepLabv3	1280 × 1280	0.0001	3	100

4.2 Detection Results

Table 3 presents a comparison of YOLOv8 and DeepLabv3 performance in detecting and segmenting rice and brown rice grains. Overall, YOLOv8 demonstrates superior results compared to DeepLabv3, which can be attributed to its specialized CSPDarknet53 backbone optimized for efficient and accurate real-time object detection, including small and overlapping objects.

Table 3. Comparison of evaluation metrics for models.

Model	Rice grains			Brown rice grains		
	P	R	mAP@0.5	P	R	mAP@0.5
YOLOv8	0.999	1.0	0.995	0.997	1.0	0.994
DeepLabv3	0.992	0.885	0.887	0.982	0.923	0.911

On the other hand, DeepLabv3 specializes in semantic segmentation, generating detailed pixel-level masks, while YOLOv8 focuses on object detection using bounding boxes. DeepLabv3, concentrating on pixel-wise information, typically provides more accurate segmentation results for tasks requiring precise object boundaries, such as grain size measurement.

DeepLabv3, specialized for semantic image segmentation, often merges adjacent or overlapping objects into a single segment. This characteristic can negatively impact the accurate detection and segmentation of closely spaced rice grains. Consequently, DeepLabv3 achieved a recall of 0.885 on the rice grain test dataset, likely due to the presence of adjacent rice grains within some images. In contrast, the absence of adjacent grains in the brown rice grain test dataset resulted in higher precision (0.982) and recall (0.923) values for DeepLabv3.

4.3 Measurement Results

Following the detection and segmentation of rice and brown rice grains within the test dataset using both the YOLOv8 and DeepLabv3 models, we calculated the size of the grains for each panicle section. To accurately calculate the size of rice grains, we eliminated groups of adjacent grains that were merged into single segments by DeepLabv3. The mean and standard deviation of these measurements were then determined and are presented in Table 4, 5, 6 and 7.

The results of rice size measurements from grain segmentation using both models (see Tables 4 and 5) revealed no significant differences in rice grain length across panicle sections. However, rice grain width assessments showed consistently larger values for YOLOv8 compared to DeepLabv3 because DeepLabv3 provides more precise object boundaries. In addition, the top section of panicles in long-grain varieties such as F54 and DT8 shows a standard deviation (SD) of approximately 0.50, indicating variability in grain lengths within this section of the panicle. This section of the DT8 showed a 6% coefficient of variation (CV) in grain length, indicating low variability and the

negative kurtosis value of -1.33 suggests a platykurtic distribution, with grain lengths concentrated around the mean and fewer outliers.

Table 4. Rice grain size measurement of 11 varieties by YOLOv8.

Varieties	Top		Middle		Bottom	
	Length ± SD	Width ± SD	Length ± SD	Width ± SD	Length ± SD	Width ± SD
F54	9.82 ± 0.47	2.91 ± 0.38	10.08 ± 01	2.91 ± 0.19	10.31 ± 0.53	3.20 ± 0.30
F56	9.14 ± 0.15	3.16 ± 0.13	9.25 ± 0.38	3.15 ± 0.26	9.25 ± 0.38	3.15 ± 0.26
ANP	7.64 ± 0.50	3.33 ± 0.27	7.87 ± 0.27	3.01 ± 0.23	7.58 ± 0.27	3.53 ± 0.19
G3	9.81 ± 0.11	3.14 ± 0.09	9.80 ± 0.35	3.07 ± 0.27	9.75 ± 0.52	3.29 ± 0.39
G7	8.47 ± 0.14	3.15 ± 0.06	8.42 ± 0.38	3.17 ± 0.17	8.40 ± 0.38	3.31 ± 0.20
G13	8.33 ± 0.16	3.15 ± 0.09	8.69 ± 0.31	3.22 ± 0.17	8.69 ± 0.19	3.37 ± 0.10
G17	7.88 ± 0.14	3.38 ± 0.14	7.98 ± 0.38	3.11 ± 0.11	8.36 ± 0.31	3.53 ± 0.29
G18	9.19 ± 0.11	3.14 ± 0.03	9.10 ± 0.43	3.260.28	9.21 ± 0.37	3.20 ± 0.16
ĐT8	9.03 ± 0.54	3.21 ± 0.23	9.14 ± 0.54	3.21 ± 0.23	9.02 ± 0.47	3.22 ± 0.23
OM18	9.24 ± 0.09	3.10 ± 0.04	9.16 ± 0.41	3.22 ± 0.24	9.23 ± 0.51	3.11 ± 0.16
OM380	9.11 ± 0.12	3.09 ± 0.09	9.16 ± 0.45	3.39 ± 0.25	9.35 ± 0.33	3.09 ± 0.21

Table 5. Rice grain size measurement of 11 varieties by DeepLabv3.

Varieties	Top		Middle		Bottom	
	Length ± SD	Width ± SD	Length ± SD	Width ± SD	Length ± SD	Width ± SD
F54	9.88 ± 0.58	2.71 ± 0.15	9.76 ± 0.37	2.62 ± 0.27	9.92 ± 0.48	2.72 ± 0.14
F56	9.02 ± 0.35	2.70 ± 0.23	8.99 ± 0.37	2.62 ± 0.19	8.88 ± 0.35	2.68 ± 0.15
ANP	7.25 ± 0.30	2.86 ± 0.13	7.59 ± 0.30	2.55 ± 0.16	7.30 ± 0.24	3.11 ± 0.14
G3	9.73 ± 0.34	2.69 ± 0.11	9.55 ± 0.32	2.69 ± 0.11	9.71 ± 0.55	2.67 ± 0.26
G7	8.41 ± 0.39	2.80 ± 0.09	8.46 ± 0.45	2.82 ± 0.08	8.14 ± 0.38	2.78 ± 0.09
G13	8.28 ± 0.26	2.62 ± 0.19	8.99 ± 0.37	2.62 ± 0.19	8.88 ± 0.35	2.68 ± 0.15
G17	7.71 ± 0.31	3.02 ± 0.20	7.65 ± 0.35	3.11 ± 0.13	8.42 ± 0.37	3.16 ± 0.19
G18	9.15 ± 0.51	2.67 ± 0.17	8.71 ± 0.41	2.63 ± 0.15	9.09 ± 0.38	2.68 ± 0.13
ĐT8	8.87 ± 0.61	2.61 ± 0.16	8.76 ± 0.53	2.59 ± 0.10	8.64 ± 0.45	2.59 ± 0.10
OM18	8.89 ± 0.45	2.46 ± 0.13	8.30 ± 0.36	2.49 ± 0.10	9.03 ± 0.46	2.54 ± 0.12
OM380	9.02 ± 0.44	2.61 ± 0.15	8.81 ± 44	2.74 ± 0.14	9.18 ± 0.31	2.58 ± 0.14

As shown in Tables 6 and 7, grain length measurements using YOLOv8 and DeepLabv3 for 11 brown rice varieties showed no significant difference between panicle parts. However, grain widths were consistently larger for YOLOv8 compared to DeepLabv3.

To assess the accuracy of machine learning models in measuring rice grain size, we compared their results with measurements obtained using a traditional ruler method.

Table 6. Brown rice grain size measurement of 11 varieties by YOLOv8.

Varieties	Top		Middle		Bottom	
	Length ± SD	Width ± SD	Length ± SD	Width ± SD	Length ± SD	Width ± SD
F54	7.23 ± 0.34	2.43 ± 0.18	7.29 ± 0.31	2.43 ± 0.18	7.37 ± 0.31	2.38 ± 0.18
F56	7.01 ± 0.27	2.44 ± 0.16	7.10 ± 0.28	2.53 ± 0.14	7.10 ± 0.31	2.63 ± 0.19
ANP	5.80 ± 0.31	2.62 ± 0.19	5.65 ± 0.25	2.70 ± 0.15	6.01 ± 0.19	2.44 ± 0.12
G3	7.25 ± 0.19	2.37 ± 0.17	7.24 ± 0.25	2.38 ± 0.16	7.38 ± 0.28	2.45 ± 0.13
G7	7.02 ± 0.33	2.94 ± 0.23	6.96 ± 0.34	2.75 ± 0.19	6.92 ± 0.27	2.54 ± 0.16
G13	6.44 ± 0.26	2.45 ± 0.13	6.45 ± 0.20	2.50 ± 0.19	6.64 ± 0.20	2.75 ± 0.14
G17	6.21 ± 0.25	2.78 ± 0.16	6.16 ± 0.34	2.83 ± 0.26	6.21 ± 0.24	2.84 ± 0.11
G18	6.91 ± 0.26	2.40 ± 0.12	7.03 ± 0.28	2.65 ± 0.16	7.10 ± 0.30	2.74 ± 0.18
DT8	6.77 ± 0.38	2.36 ± 0.15	6.79 ± 0.34	2.320.16	6.76 ± 0.29	2.37 ± 0.13
OM18	6.86 ± 0.31	2.25 ± 0.08	6.88 ± 0.37	2.31 ± 0.11	6.81 ± 0.38	2.23 ± 0.12
OM380	6.74 ± 0.32	2.39 ± 0.18	6.83 ± 0.37	2.41 ± 0.12	6.75 ± 0.32	2.360.13

Table 7. Brown rice grain size measurement of 11 varieties by DeepLabv3.

Varieties	Top		Middle		Bottom	
	Length ± SD	Width ± SD	Length ± SD	Width ± SD	Length ± SD	Width ± SD
F54	7.26 ± 0.33	2.20 ± 0.19	7.04 ± 0.22	2.20 ± 0.19	7.14 ± 0.30	2.18 ± 0.19
F56	6.80 ± 0.26	2.21 ± 0.14	6.70 ± 0.27	2.21 ± 0.10	6.63 ± 0.30	2.22 ± 0.11
ANP	5.49 ± 0.35	2.45 ± 0.25	5.53 ± 0.25	2.56 ± 0.12	5.71 ± 0.20	2.20 ± 0.08
G3	7.04 ± 0.16	2.15 ± 0.13	6.87 ± 0.40	2.11 ± 0.12	7.07 ± 0.27	2.130.11
G7	6.45 ± 0.25	2.31 ± 0.12	6.50 ± 0.31	2.27 ± 0.15	6.42 ± 0.31	2.27 ± 0.14
G13	6.12 ± 0.41	2.24 ± 0.12	6.20 ± 0.25	2.27 ± 0.10	6.28 ± 0.14	2.33 ± 0.10
G17	5.95 ± 0.26	2.56 ± 0.13	5.86 ± 0.33	2.57 ± 0.24	5.98 ± 0.27	2.18 ± 0.09
G18	6.66 ± 0.26	2.18 ± 0.13	6.56 ± 0.27	2.20 ± 0.10	6.65 ± 0.28	2.27 ± 0.08
DT8	6.51 ± 0.38	2.11 ± 0.10	6.49 ± 0.43	2.08 ± 0.13	6.46 ± 0.36	2.13 ± 0.11
OM18	6.65 ± 0.30	2.03 ± 0.08	6.54 ± 0.36	2.03 ± 0.10	6.52 ± 0.40	1.99 ± 0.10
OM380	6.43 ± 0.33	2.09 ± 0.13	6.48 ± 0.37	2.10 ± 0.10	6.36 ± 0.41	2.15 ± 0.12

To ensure a diverse dataset, we selected three rice varieties with distinct grain lengths: long, medium, and short. For each variety, 50 grains were chosen and measured using two machine learning models and the ruler method. In the ruler method, each grain was measured individually using an electronic Kapusi ruler. Measuring fifty grains and calculating the mean as the final results. The results of this comparison are presented in Table 8 and 9. ANOVA analysis [28] revealed no significant difference ($p < 0.01$) in grain length and width measurements obtained using different methods. These findings support the reliability of using deep learning models for assessing rice grain shape.

Table 8. Length and width of fifty rice grains by three methods.

Varieties	YOLOv8		DeepLabv3		Ruler	
	Length ± SD	Width ± SD	Length ± SD	Width ± SD	Length	Width
Long grain	11.06 ± 0.64	2.70 ± 0.22	11.15 ± 0.6	2.3 ± 0.18	11.60	2.14
Medium grain	9.12 ± 0.35	3.08 ± 0.15	9.06 ± 0.38	2.75 ± 0.10	9.44	2.64
Short grain	6.99 ± 0.34	4.05 ± 0.28	6.92 ± 0.36	3.7 ± 16	7.14	3.60

Table 9. Length and width of fifty brown rice grains by three methods.

Varieties	YOLOv8		DeepLabv3		Ruler	
	Length ± SD	Width ± SD	Length ± SD	Width ± SD	Length	Width
Long grain	8.23 ± 0.52	2.14 ± 0.18	8.18 ± 0.54	1.88 ± 0.18	7.91	1.75
Medium grain	7.04 ± 0.30	2.60 ± 0.14	6.86 ± 0.31	2.32 ± 0.10	6.79	2.23
Short grain	5.44 ± 0.29	3.44 ± 0.34	5.16 ± 0.30	3.13 ± 0.19	4.92	2.98

5 Conclusion

We presented an investigation of grain size variation across rice panicles of 11 varieties with varying grain lengths planted in the Mekong Delta. We employed YOLOv8 and DeepLabv3 to detect and segment rice grains and brown rice grains from three panicle sections (top, middle, and bottom). From grain segmentations, grain sizes (length and width) were calculated as rectangles around the convex hull which are the smallest convex boundaries. Our findings revealed no significant differences in grain size between panicle sections. Additionally, YOLOv8 demonstrated superior performance in detecting closely spaced grains, while DeepLabv3 provided more precise grain boundary segmentation. Furthermore, a comparison using ANOVA showed that both deep learning models yielded comparable results to the traditional ruler technique. In the future, we will evaluate grain sizes on rice panicles under various cultivation models (for example, two and three crops per year with different fertilizer application methods) and explore additional deep learning models.

Acknowledgement. This research is totally funded by Vietnam National University Ho Chi Minh City (VNUHCM) under Grant Number C2022–16-09/HĐ-KHCN, Vietnam.

References

1. Zhou, C., et al.: Automated counting of rice panicle by applying deep learning model to images from unmanned aerial vehicle platform. Sensors **19**(14), 3106 (2019). <https://doi.org/10.3390/s19143106>

2. Zhao, S., Gu, J., Zhao, Y., Hassan, M., Li, Y., Ding, W.: A method for estimating spikelet number per panicle: integrating image analysis and a 5-point calibration model. *Sci. Rep.* **5**(1), 16241 (2015). <https://doi.org/10.1038/srep16241>
3. Deng, R., et al.: Automated counting grains on the rice panicle based on deep learning method. *Sensors* **21**(1), 281 (2021). <https://doi.org/10.3390/s21010281>
4. Wu, W., et al.: Image analysis-based recognition and quantification of grain number per panicle in rice. *Plant Methods* **15**(1), 122 (2019). <https://doi.org/10.1186/s13007-019-0510-0>
5. Lu, Y., Wang, J., Fu, L., Yu, L., Liu, Q.: High-throughput and separating-free phenotyping method for on-panicle rice grains based on deep learning. *Front. Plant Sci.* **14**, 1219584 (2023). <https://doi.org/10.3389/fpls.2023.1219584>
6. Wang, R., Han, F., Wu, W.: Estimation of paddy rice maturity using digital imaging. *Int. J. Food Prop.* **24**(1), 1403–1415 (2021). <https://doi.org/10.1080/10942912.2021.1970581>
7. Ruslan, R., Aznan, A.A., Azizan, F.A., Roslan, N., Zulkifli, N.: Extraction of morphological features of malaysian rice seed varieties using flatbed scanner. *Int. J. Adv. Sci. Eng. Inf. Technol.* **8**(1), 93 (2018). <https://doi.org/10.18517/ijaseit.8.1.2752>
8. Feng, X., et al.: Size measurement and filled/unfilled detection of rice grains using backlight image processing. *Front. Plant Sci.* **14**, 1213486 (2023). <https://doi.org/10.3389/fpls.2023.1213486>
9. Birla, R., Chauhan, A.P.S.: An efficient method for quality analysis of rice using machine vision system. *J. Adv. Inf. Technol.* **6**, 140–145 (2015). <https://doi.org/10.12720/jait.6.3.140-145>
10. Kumar, A., Taparia, M., Madapu, A., Rajalakshmi, P., Marathi, B., Desai, U.B.: Discrimination of filled and unfilled grains of rice panicles using thermal and RGB images. *J. Cereal Sci.* **95**, 103037 (2020). <https://doi.org/10.1016/j.jcs.2020.103037>
11. Ansari, N., Ratri, S.S., Jahan, A., Ashik-E-Rabbani, M., Rahman, A.: Inspection of paddy seed varietal purity using machine vision and multivariate analysis. *J. Agric. Food Res.* **3**, 100109 (2021). <https://doi.org/10.1016/j.jafr.2021.100109>
12. Nga, T.T.K., Pham, T.V., Tam, D.M., Koo, I., Mariano, V.Y., Do-Hong, T.: Combining binary particle swarm optimization with support vector machine for enhancing rice varieties classification accuracy. *IEEE Access* **9**, 66062–66078 (2021). <https://doi.org/10.1109/ACCESS.2021.3076130>
13. Tran-Thi-Kim, N., Pham-Viet, T., Koo, I., Mariano, V., Do-Hong, T.: Enhancing the classification accuracy of rice varieties by using convolutional neural networks. *Int. J. Electr. Electron. Eng. Telecommun.* **12**(2), 150–160 (2023). <https://doi.org/10.18178/ijeetc.12.2.150-160>
14. Gilanie, G., Nasir, N., Bajwa, U.I., Ullah, H.: RiceNet: convolutional neural networks-based model to classify Pakistani grown rice seed types. *Multimedia Syst.* **27**(5), 867–875 (2021). <https://doi.org/10.1007/s00530-021-00760-2>
15. Zou, Y., et al.: Rice grain detection and counting method based on TCLE–YOLO model. *Sensors* **23**(22), 9129 (2023). <https://doi.org/10.3390/s23229129>
16. Nguyen, H.S., Nguyen, T.-N.: Deep learning for rice quality classification. In: 2019 International Conference on Advanced Computing and Applications (ACOMP), pp. 92–96. IEEE, Nha Trang (2019). <https://doi.org/10.1109/ACOMP.2019.00021>
17. Koklu, M., Cinar, I.: Determination of effective and specific physical features of rice varieties by computer vision in exterior quality inspection. *SJAFS* **35**(3), 229–243 (2021). <https://doi.org/10.15316/SJAFS.2021.252>
18. IRRI. Standard evaluation system for rice, 5th ed. Manila (2013)
19. Hoque, S.N., et al.: Grain physical properties analysis of some improved rice varieties. *Asian J. Crop Soil Sci. Plant Nutr.* **6**(2), 242–250 (2022)
20. Nádvorníková, M., Banout, J., Herák, D., Verner, V.: Evaluation of physical properties of rice used in traditional Kyrgyz Cuisine. *Food Sci. Nutr.* **6**(6), 1778–1787 (2018). <https://doi.org/10.1002/fsn3.746>

21. Lai, D.Q., Ngo, T.A., Nguyen, Q.L., Nguyen, H.D., Pham, D.T.: Physical and chemical properties of rice varieties grown in Mekong delta. *Vietnam J. Sci. Technol.* **60**(5), 767–784 (2022). <https://doi.org/10.15625/2525-2518/14447>
22. Le, T.P.: List of flood-tolerant seasonal rice varieties used in nature-based agricultural models (Vietnamese). Accessed 07 Jan 2024. <https://cci.agu.edu.vn/tong-hop-hoat-dong/tin-tuc-su-kien/danh-sach-cac-giong-lua-mua-chiu-ngap-su-dung-cho-mo-hinh-nong-nghiep-thuan-thien/>
23. Wang, W.: Advanced auto labeling solution with added features. Github repository (2023). <https://github.com/CVHub520/X-AnyLabeling>
24. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO (2023). <https://github.com/ultralytics/ultralytics>
25. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII*, pp. 833–851. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_49
26. Yao, Q., Chen, J., Guan, Z., Sun, C., Zhu, Z.: Inspection of rice appearance quality using machine vision. In: 2009 WRI Global Congress on Intelligent Systems, pp. 274–279. IEEE, Xiamen (2009). <https://doi.org/10.1109/GCIS.2009.91>
27. Gresina, F., Farkas, B., Fábián, S.Á., Szalai, Z., Varga, G.: Morphological analysis of mineral grains from different sedimentary environments using automated static image analysis. *Sed. Geol.* **455**, 106479 (2023). <https://doi.org/10.1016/j.sedgeo.2023.106479>
28. Kaufmann, J., Schering, A.: Analysis of variance ANOVA. In: Kenett, R.S., Longford, N.T., Piegorsch, W.W., Ruggeri, F. (eds.) *Wiley StatsRef: Statistics Reference Online*, 1st edn.. Wiley (2014). <https://doi.org/10.1002/9781118445112.stat06938>



BeLightRec: A Lightweight Recommender System Enhanced with BERT

Manh Mai Van¹ and Tin T. Tran²

¹ Faculty of Information Technology, Ton Duc Thang University,
Ho Chi Minh City, Vietnam
maivanmanh@tdtu.edu.vn

² Artificial Intelligence Laboratory, Faculty of Information Technology,
Ton Duc Thang University, Ho Chi Minh City, Vietnam
trantrungtin@tdtu.edu.vn

Abstract. The trend of data mining using deep learning models on graph neural networks has proven effective in identifying object features through signal encoders and decoders, particularly in recommendation systems utilizing collaborative filtering methods. Collaborative filtering exploits similarities between users and items from historical data. However, it overlooks distinctive information, such as item names and descriptions. The semantic data of items should be further mined using models in the natural language processing field. Thus, items can be compared using text classification, similarity assessments, or identifying analogous sentence pairs. This research proposes combining two sources of item similarity signals: one from collaborative filtering and one from the semantic similarity measure between item names and descriptions. These signals are integrated into a graph convolutional neural network to optimize model weights, thereby providing accurate recommendations. Experiments are also designed to evaluate the contribution of each signal group to the recommendation results.

Keywords: Recommender System · Collaborative Filtering · Graph Convolution Network · Natural Language Processing

1 Introduction

Recommender systems play a crucial role in both theoretical research on information retrieval and practical, everyday applications. E-commerce applications utilize recommender models to present users with items that suppliers believe the users will be interested in and likely to purchase [1]. Similarly, library or book-selling applications leverage user interaction histories to curate and recommend book catalogs to users. In the initial phase of research, recommender systems focused on identifying the characteristics of target users, such as age, gender, residential address, and preferences, and then selecting items that matched these characteristics to recommend to users. However, the vast amount of product

information, the rapid emergence of new items, and the challenges in collecting personal information and preferences from users pose significant difficulties. The next phase in recommender systems development addressed these challenges through collaborative filtering techniques, which identify similarities between users based on the set of items they interact with. The more items two users have interacted with in common, the higher the similarity between them. Items that a user has interacted with are then recommended to other users with high similarity to the original user.

In collaborative filtering models, the characteristic information of users and items is excluded, such as book titles, movie genres, or previous user feedback. This can lead to information loss when evaluating the similarity of items. Collaborative filtering models can be implemented on graph neural networks, where users and items are represented by vertices in the graph, and their relationships are explored through high-order propagation. Moreover, this propagation process can be efficiently implemented using prominent machine learning libraries like TensorFlow, Numpy, and Pandas. Based on collaborative filtering, we can delve deeper and exploit the similarity between items to supplement the explored graph. Items similar to those a user has interacted with should be recommended to that user. Items within the same category, or having similar descriptions or names, significantly influence the user.

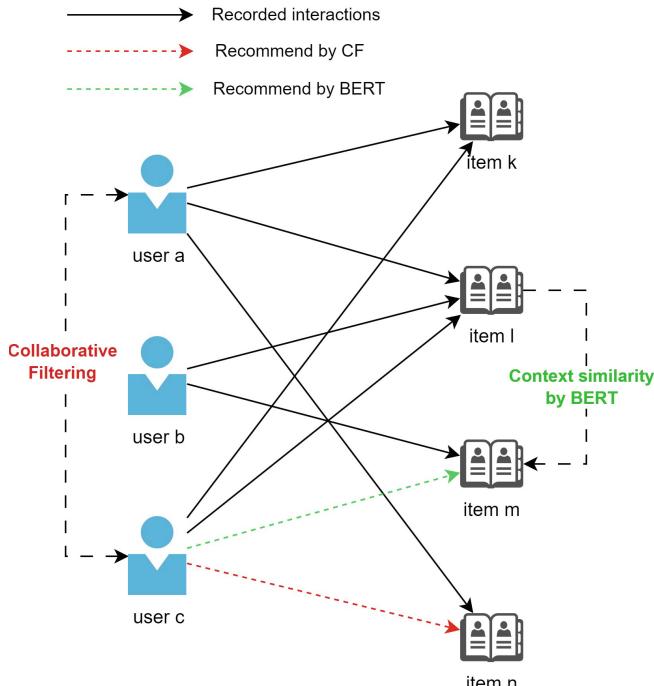


Fig. 1. Recommender system by collaborative filtering

With natural language processing models, evaluating the semantic similarity between two text segments can be conducted using text vectorization techniques and measuring the distance between these vectors in the language representation space. The signal propagation process on the convolutional graph network highlights the interactions between users and items, or between items themselves, through collaborative filtering, emphasizing the distinctive features of the objects. The convergence of feature values is regulated by weight matrices, which are trained with input data sets. The feature vectors of users and items are calculated after several iterations of propagation. In the following sections, we will detail the components of the recommendation system that explores the similarity between users and items, integrating semantic similarity signals between items into the output feature vectors. We summarize the recommendation model as shown in Fig. 1. Finally, we conduct experiments on some of the most recent and real-world data sets to evaluate the proposed model.

2 Literature Review

2.1 Deep Learning in Context-Aware Recommender System

Recent developments in recommender systems, particularly since 2018, have heavily focused on leveraging deep learning and context-aware techniques to enhance personalization and accuracy. Deep learning has revolutionized the field by enabling the modeling of complex user-item interactions, surpassing traditional linear methods. Neural networks, including autoencoders, convolutional neural networks (CNNs), and recurrent neural networks (RNNs), have proven highly effective in capturing latent features and patterns from large, unstructured datasets such as user behavior, text, and multimedia content. This shift has allowed platforms like YouTube and Netflix to substantially enhance the relevance and accuracy of their recommendations. However, these models come with higher computational demands and require large-scale datasets for effective training. Research by Zhang et al. highlights the success of deep learning models in improving prediction accuracy while also acknowledging the challenges of scalability and computational cost in real-world applications [2].

Additionally, context-aware recommender systems (CARS) have gained attention as they integrate contextual factors such as time, location, and user mood to enhance recommendation relevance. Contextual data, combined with deep learning models, allows for more dynamic and situation-specific recommendations, increasing user engagement and satisfaction. For example, a music recommendation system might suggest different songs depending on the time of day or a user's current activity. Publication [3] explored the impact of CARS in improving recommendation quality by adapting to real-time user environments. While these systems offer more personalized experiences, they also introduce complexity in data collection and processing, alongside concerns about user privacy and the explainability of recommendations.

2.2 Graph Convolution Networks

Graph Convolutional Networks (GCNs) are advanced deep learning models designed to process graph-structured data like social networks, transportation networks, and molecular structures. Unlike traditional neural networks that handle grid-like data (such as images or text), GCNs leverage the non-Euclidean structure of graphs to learn node and edge features. GCNs extend the concept of convolution from grid data to graph data, using mathematical transformations to aggregate information from neighboring nodes, as detailed in publication [4]. This occurs through convolutional layers, where each applies a linear transformation to node features and combines them according to the graph structure [5, 6]. This allows GCNs to learn complex node representations and inter-node relationships, enhancing performance in tasks such as node classification, graph classification, and link prediction. Due to their effective handling of graph-structured data and ability to uncover hidden relationships, GCNs have become a powerful and increasingly popular tool in fields such as social network analysis, computational biology, and other artificial intelligence applications.

Furthermore, Light Graph Convolution Networks (LGCN), which are models in studies [7, 8], has shown that removing complex components such as weight matrices and bias vectors will not increase the convergence rate of feature embeddings but also achieve better precision and recall. This is explained because the observed data sets are recorded imploid and the interaction matrix is a sparse matrix because the number of interactions of a user is very small compared to the number of all items.

2.3 Natural Language Processing

BERT (Bidirectional Encoder Representations from Transformers) [9] and TF-IDF (Term Frequency-Inverse Document Frequency) [10] are prominent models in natural language processing, each serving distinct purposes. BERT, developed by Google, is an advanced language model that understands the context of words in sentences using a bidirectional approach, considering both preceding and succeeding words. It is trained on a vast amount of unlabeled text data through tasks like Masked Language Model and Next Sentence Prediction, and can be fine-tuned for various tasks such as text classification, question answering, and named entity recognition, consistently achieving superior performance in real-world applications. On the other hand, TF-IDF is used to assess the importance of a word in a document relative to a corpus of documents. It combines Term Frequency (TF), which measures a word's frequency in a document, and Inverse Document Frequency (IDF), which measures the word's rarity across the corpus. TF-IDF helps identify important words while reducing the impact of common words, making it useful for text search and classification. By combining the strengths of TF-IDF and BERT, applications can achieve a balance between efficiency and accuracy, leveraging TF-IDF's ability to quickly identify significant terms and BERT's deep contextual understanding to deliver high-quality results in various natural language processing tasks.

3 Proposed Model

We propose model BeLightRec which is based on LGCN to propagate collaborative signals and discover the feature vectors of both users and items, and enhanced with semantic similarity measures between item titles and descriptions. The model uses BERT for similarity evaluation. Furthermore, we introduce several metrics to evaluate and compare the proposed models with state-of-the-art models.

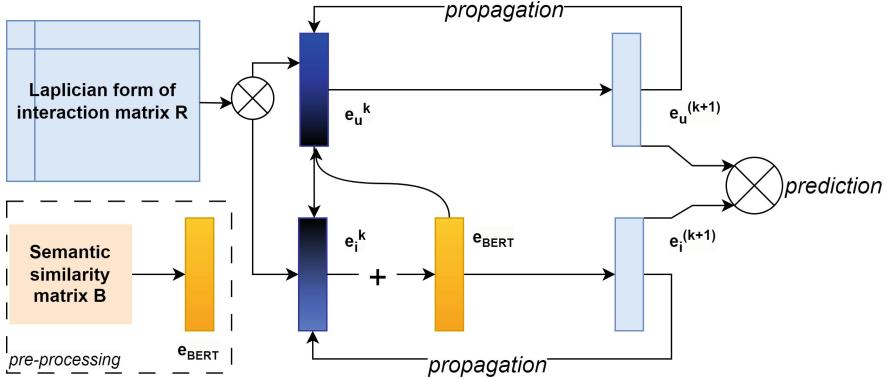


Fig. 2. Overview of the proposed model

We use two matrices as inputs to the LGCN: matrix R contains records interactions between users and items; and matrix B records semantic similarity between item titles and descriptions. All matrix elements are L1-normalized to ensure values lie between 0.0 and 1.0, minimizing the impact of different measurement units. The LGCN will propagate signals and iterate several times before finding the characteristic vector values for users and items. An overview of the model is presented in Fig. 2.

3.1 Data Preprocessing

In recommendation systems, interactions between users and items can be represented in two ways: implicit and explicit. Implicit data store interactions as binary values, while explicit data store user ratings of items during interactions. E-commerce systems often allow users to rate or provide feedback on products they have purchased or viewed. For datasets containing implicit reviews, the matrix representing user-item interactions is very sparse. Furthermore, customers with fewer than k interactions will create noise in the propagation process. In most studies, such users are removed from the dataset using k -core algorithm [6, 12].

The algorithm takes a similarity matrix between users and items as input and outputs a subset of user-item pairs. It begins by calculating a ratio between the number of users and the number of items. The algorithm then examines each item to determine if it has a minimum number of interactions; items that meet this threshold are selected into a new subset. After this filtering step, the algorithm computes how many users should be chosen based on this subset of items. For each user, it identifies the items they have interacted with and calculates how closely this matches the selected subset of items using a similarity measure. The users with the highest similarity scores are selected, and the final output is generated as the set of these top users paired with the filtered items. This output represents the most relevant user-item pairs. We defined this output matrix R , and formed it out in Laplician form to serve as input to the proposed model.

3.2 Semantic Similarity Measurement

Item names and descriptions can be treated as text documents, and BERT's tokenization technique can convert these texts into tensors. These tensors form embeddings for each initial text. The model takes the average of the token embeddings representing words in each text to create a unique vector $\overrightarrow{item_i}$ representation for each $item_i$. Finally, the model uses the cosine similarity measure to calculate the similarity between the embedding vectors of two items. The resulting score reflects the semantic similarity between two items on a 0 to 1 scale, with 1 indicating maximum similarity. By combining BERT and TF-IDF methods, we can achieve a more accurate and comprehensive assessment of the semantic similarity between items, thereby supporting the proposed model effectively. The BERT (or TF-IDF) similarity measure between items i and j is described by Eq. 1 to integrate both BERT and TF-IDF, the values of the similarity measures can be averaged or weighted to reflect the role of each measure.

$$BERT(item_i, item_j) = \cos(\overrightarrow{item_i}, \overrightarrow{item_j}) \quad (1)$$

Similarity matrix B: It's size is $m \times m$ where m is the number of items. Each element $B_{i,j}$ represents the similarity calculated by Eq. 1 based on item names and descriptions. All matrix elements are L1 normalized to ensure each value lies between 0.0 and 1.0, minimizing the impact of measurement units.

3.3 Components of the Proposed Model

Signal Propagation and Embedding Collection: The feature signals of users are collected from the input matrices described in the previous section, then propagated within the model as embeddings e_u^k (representing user features) and e_i^k (representing item features). This process is repeated k times within the LGCN [7]. The embedding e_u^{k+1} at the end of each iteration is used as the input for the next propagation and is calculated by Eq. 2, with N_u^R and N_i^R , are respectively number of neighboring users of u and items of i in the matrix R .

$$e_u^{k+1} = \sum_{i \in N_u^R} \frac{1}{\sqrt{|N_u^R| |N_i^R|}} e_i^k \quad (2)$$

Simultaneously, the item feature embedding e_i^{k+1} is also calculated and propagated as shown in Eq. 3.

$$e_i^{k+1} = \sum_{u \in N_i^R} \frac{1}{\sqrt{|N_i^R| |N_u^R|}} e_u^k + \sum_{b \in N_i^B} \frac{1}{\sqrt{|N_i^B| |N_b^B|}} e_b^k \quad (3)$$

After several iterations of signal propagation, the characteristic vectors of users and items are found by Eq. 4.

$$e_u = \frac{1}{K} \sum_{k=1}^K e_u^k \quad ; \quad e_i = \frac{1}{K} \sum_{k=1}^K e_i^k \quad (4)$$

Prediction and Loss Function: The characteristic vectors for users and items converge after several iterations of propagation, and the prediction score between user u_i and item i_j can be calculated by Eq. 5.

$$\hat{y}_{ui} = e_{u_i}^* {}^\top e_{i_j}^* \quad (5)$$

The Bayesian personalized ranking (BPR) method is the optimal choice for implementing the loss function because it is the most effective ranking method for datasets with implicit feedback [11]. We use two observation sets: Ω_{ui}^+ representing the interacted items and Ω_{ui}^- representing the non-interacted items. The loss function is calculated by Eq. 6.

$$Loss_{bpr} = \sum_{\Omega_{ui}^+} \sum_{\Omega_{uj}^-} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \| \Phi \|_2^2 \quad (6)$$

4 Experiments

4.1 Datasets Description

We obtained the latest datasets and removed users with fewer than 10 interactions to enrich the training set. We split the data into training and testing sets with a ratio of 80% and 20%, respectively. Statistics on the datasets are described in Table 1. In that table, we record the number of users and items, the number of interactions in the survey dataset, density is the ratio of the number of interactions to the product of the number of users with items, and average text count is the average number of characters in the description of the items in the dataset.

- **Amazon-Book dataset** is widely used in the field study of recommendation systems. It is one of the datasets provided by Amazon, including detailed information about book products, user reviews, ratings, and other interactions between users and books.
- **Recipes dataset** is a collection of information about various dishes and their preparation methods. These datasets might also have additional information like recipe names, cuisine types, user ratings, reviews, and images of the completed dishes.

- **Hawaii dataset** is part of a larger collection of location-based data from Google Local, hosted by UCSD. This dataset focuses on providing detailed information about various locations in Hawaii, including restaurants, hotels, and attractions.

Table 1. Statistic of the experiment datasets.

Dataset	#Users	#Items	#Interactions	Density	Ave. text
AmazonBook	27,034	32,157	992,700	1.14e-3	8,599
Recipes	3,969	5,637	188,135	8.41e-3	344
Hawaii	35,604	9,111	1,135,062	3.5e-3	102

4.2 Baseline Models

We use the same datasets and repeat the experiments on all the following baseline models to demonstrate the result:

- **BPR-MF** [11] is a popular method in recommendation systems, combining Matrix Factorization (MF) and Bayesian Personalized Ranking (BPR). MF decomposes the user-item rating matrix into two smaller matrices representing users and items to predict ratings. BPR optimizes personalized ranking by maximizing the probability that an interacted item is ranked higher than a non-interacted one.
- **NGCF** [12] leverages graph structures to capture complex interaction information. Users and items are represented as nodes in a graph, with interactions as connecting edges. NGCF applies graph neural network layers to propagate and aggregate information across layers, enabling the model to learn high-order relation features.
- **LightGCN** [7] simplifies traditional graph neural networks by removing complex components like transformation matrices and non-linear activation functions, focusing instead on the aggregation and propagation of information between nodes.
- **BERT** only uses semantic similarity measure between items and does not have signal propagation loop like other GCN models.
- **BeLightRec** is the LGCN model enhanced with BERT. The semantic similarity signal between items should be collected during the propagation process.
- **BeLightRec+W** is a traditional GCN with weight matrices and bias vector. We implemented this model as a variant of proposed LGCN.

4.3 Experimental Settings and Metrics

Precision, recall, and NDCG@k are key metrics used to evaluate the performance of information retrieval and recommendation systems.

- **Precision** measures the proportion of relevant items retrieved out of the total items retrieved. It reflects the accuracy of the system in returning relevant results. High precision means that the results returned are mostly relevant.
- **Recall** assesses the proportion of relevant items retrieved out of the total relevant items available. It indicates the system’s ability to find all relevant items. High recall means that the system can identify most of the relevant items.
- **NDCG@k** (Normalized Discounted Cumulative Gain at k) is a comprehensive metric that considers the relevance and the position of the retrieved items. It is particularly useful in ranking tasks. NDCG@k evaluates the quality of the top-k results by accounting for the graded relevance of items, giving higher scores to more relevant items appearing higher in the list. We conducted experiments with Top-5 and Top-20 settings in our experiments.

To ensure fair experimental results, we maintain consistent parameters across all models. Specifically, we set the learning rate to 0.001, the L2 normalization coefficient to 1×10^{-5} , and the number of LGCN layers to three, with each layer having an embedding size of 64. We also apply the same early stopping strategy used by NGCF and LightGCN.

4.4 Experiment Results

The overall performance comparison is shown in Table 2 with all models on three datasets. The results showed that BeLightRec consistently outperforms other models across all datasets and metrics. Even though BPR-MF is not a GCN-based models, we still experiment with that model as a classic benchmark. The results progressively improve from BPR-MF to NGCF and LightGCN, aligning with published insights on these models. The LightGCN and BeLightRec models both outperform their corresponding models, NGCF and BeLightRec+W. This demonstrates the effectiveness of LGCN on implicit datasets.

We split the results table into two parts for Top-5 and Top-20. In each part, the AmazonBook, Recipes, and Hawaii datasets are presented respectively. In each dataset, the three metrics recall, precision, and ndcg are applied to each of the mentioned models.

4.5 Ablation Study

Contribution of Each Propagation Signal. During propagation, the model aggregates the similarity signal between items using an interactive filtering method with the semantic similarity signal using the BERT model’s measurement. To show the contribution of each signal, we removed one of the two sources

Table 2. Overall performance comparisons

Dataset	AmazonBook			Recipes			Hawaii		
	recall	precision	ndcg	recall	precision	ndcg	recall	precision	ndcg
<i>Top-5 results</i>									
BPR-MF	0.01965	0.01968	0.02382	0.01490	0.02354	0.02580	0.04656	0.05531	0.06341
NGCF	0.01856	0.01912	0.02144	0.01612	0.02526	0.02631	0.05210	0.06235	0.07136
LightGCN	0.02074	0.02133	0.02374	0.01624	0.02440	0.02625	0.05368	0.06398	0.07343
BeLightRec+W	0.02063	0.02117	0.02375	0.01653	0.02564	0.02749	0.05441	0.06479	0.07431
BERT	0.02228	0.02265	0.02528	0.01534	0.02358	0.02590	0.05340	0.06405	0.07308
BeLightRec	0.02610	0.02653	0.02938	0.01789	0.02681	0.02944	0.05782	0.06896	0.07935
<i>Top-20 results</i>									
BPR-MF	0.56010	0.01492	0.03948	0.04499	0.01848	0.03567	0.11835	0.03540	0.08817
NGCF	0.05502	0.01490	0.03664	0.05090	0.01988	0.03790	0.13709	0.04174	0.10134
LightGCN	0.05851	0.01582	0.03940	0.05186	0.02012	0.03875	0.13741	0.04179	0.10264
BeLightRec+W	0.05886	0.01603	0.03971	0.05180	0.02079	0.03911	0.14026	0.04265	0.10441
BERT	0.06720	0.01811	0.04424	0.04681	0.01898	0.03659	0.13943	0.04252	0.10323
BeLightRec	0.07563	0.02034	0.05020	0.05415	0.02125	0.04148	0.14628	0.04447	0.10991

in turn and compared the increase in precision, recall, and NDCC@5 in the conducted experiment. The LightGCN model does not care about the semantic similarity signal measured by BERT, while the BERT model in our experiment does not repeat the propagation process of GCN.

Based on the experimental results, we depict them in Fig. 3 to show the percentage improvement of each model compared to the classic MF-BPR model. We find that there are 3 comparison scenarios between the two signal sources under discussion as follows

- **CF is better than BERT in dataset Recipes:** In the recipe descriptions, the text describing each item is presented as a list of ingredients with very little specific description of them. Users write about the recipes in terms of how they feel rather than the objective description of the recipe. This makes it difficult for the BERT model to capture semantic similarity signals and contributes almost nothing to the precision metric when applied alone.
- **BERT is better than CF in dataset AmazonBook:** The book titles and their reviews are very detailed and directly related to the book content in this dataset. This has led to the effectiveness of the BERT model in assessing semantic similarity between items, i.e. books.
- **CF and BERT are equal in dataset Hawaii:** This is an experiment where the two sources of signal contribute roughly equally. First, although the item descriptions in the Hawaii dataset are shorter in length than the item descriptions in the Recipes dataset, they focus on describing locations and geographic information. Second, the correlation between users measured based on the locations they visited suggests that users chose items based on category rather than description.

In all three comparison scenarios, when combining both signal sources, the results lead to a sharp increase in the values of precision and recall measures. This has demonstrated the superiority of the GCN model and the addition of the semantic similarity signal source between items in each propagation step.

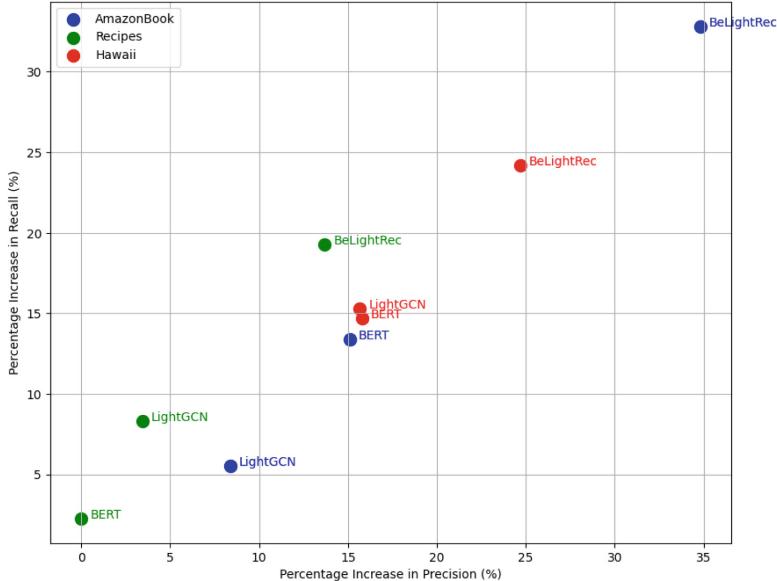


Fig. 3. Percentage increase in Precision vs Recall from BPR-MF

We randomly select a few item descriptions from the three datasets in the experiment, and list them in Table 3, to observe the grammatical and lexical structure of those descriptions. We found that the item descriptions of each dataset were presented as follows.

- **AmazonBook:** The items in this dataset are books, and they have very detailed and clear descriptions. Each description starts with the book title, author name, summary, and the reviewer's sentiment. For that reason, the BERT model does a good job of assessing the semantic similarity between pairs of items, and this contributes to the output of our proposed model.
- **Recipes:** The item descriptions in this dataset start with the dish name, followed by a list of ingredients in an array, and finally the steps to prepare it. Despite having longer text lengths than the item descriptions in the Hawaii dataset, the semantic similarity measure in the Recipes dataset did not perform well when used as a single signal. We also assume that ingredient similarity does not necessarily lead to dish similarity.

Table 3. Text descriptions of items in datasets

Dataset	Item	Item description
AmazonBook	#60	Russian Winter: A Novel Daphne Kalotay (Author) A mysterious jewel holds the key to a life-changing secret, in this breathtaking tale of love and art, betrayal and redemption. When she decides to auction her remarkable jewelry collection, Nina Revskaya, once a great star of the Bolshoi Ballet, believes she has finally drawn a curtain on her past. Instead, the former ballerina finds herself overwhelmed by memories of her homeland and of the events, both glorious and heartbreakingly tragic, that changed the course of her life half a century ago.
AmazonBook	#3171	Blaze: A Novel Richard Bachman: From Publishers Weekly Written circa 1973, this trunk novel, as Bachman's double (aka Stephen King) refers to it in his self-deprecating foreword, lacks the drama and intensity of Carrie and the horror opuses that followed it. Still, this fifth Bachman book (after 1996's The Regulators) shows King fine-tuning his skill at making memorable characters out of simple salt-of-the-earth types. Clayton Blaze Blaisdell has fallen into a life of delinquency ever since his father's brutal abuse rendered him feebled-minded.
Recipes	#2693	15 min shrimp scampi - truly delicious seafood/pasta dish that is nearly impossible to mess up & a great stand by for rushed dinners. ['start salted water boiling for pasta', 'heat olive oil', 'saute garlic & onions over high heat till onions start to turn clear', 'add wine, turn to medium heat & reduce for 5 min add shrimp till warmed through', 'stir in butter till melted', 'turn heat to low / warm', 'salt & pepper to taste', 'cover till pasta is ready', 'serve together']
Recipes	#3644	smoky refried beans - 'leave some whole for texture' "[heat oil in a heavy skillet over medium-low heat', 'add whole garlic cloves and cook , turning once , until browned on both sides , about 5 min', 'smash garlic cloves with a fork just enough to break them open and flatten them a little', 'add pinto beans and liquid and cook until heated through , about 5 min', 'add spices and salt to taste', 'stir well', 'turn the heat down to medium-low and smash beans with a big fork', "you don't have to smash every single bean""
Hawaii	#1821	McDonald's Fast food restaurant Breakfast restaurant Coffee shop Hamburger Restaurant Sandwich shop - Classic, long-running fast-food chain known for its burgers, fries & shakes.
Hawaii	#7843	Quiksilver Clothing store Men's clothing store Skateboard shop Surf shop Women's clothing store

- **Hawaii:** Although the item descriptions in this dataset are the shortest in length, they contain a lot of keywords. These keywords often refer to the classification and properties of the places, i.e., the items. This factor is what makes the semantic similarity measure in the Hawaii dataset not worse than the AmazonBook dataset, even though the description text length of the Hawaii dataset is the shortest among the experimental datasets.

It is clear that the grammatical structure, vocabulary and writing style in the item descriptions of the datasets contribute greatly to the semantic similarity measure measured by BERT.

5 Conclusion

We have presented a machine learning model utilizing graph convolutional networks to explore the similarities between items based on their names and descriptions using metrics from the BERT and TF-IDF language models. Applying natural language processing models to a graph neural network-based data mining model is challenging but has yielded better results by uncovering item information that collaborative filtering methods missed. Our model consists of multiple modules that allow for enhancements or customizations when applied to specific recommendation systems.

The semantic correlation between items measured by BERT model has contributed significantly to the models' metrics. Therefore, further in-depth research on semantics should be conducted in the future. First, the data pre-processing step should normalize text segments and remove meaningless characters. Second, the distance between each pair of vectorized item descriptions needs to be improved in terms of calculation formula. Finally, the models should adjust the influence weight of semantic similarity measure based on the nature of the item descriptions.

References

1. Udokwu, C., Zimmermann, R., Darbanian, F., Obinwanne, T., Brandtner, P.: Design and implementation of a product recommendation system with association and clustering algorithms. *Procedia Comput. Sci.* **219**, 512–520 (2023). <https://doi.org/10.1016/j.procs.2023.01.319>
2. Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: a survey and new perspectives. *ACM Comput. Surv.* **52**, 1–38 (2019). <https://doi.org/10.1145/3285029>
3. Zhang, L., Li, X., Li, W., Zhou, H., Bai, Q.: Context-aware recommendation system using graph-based behaviours analysis. *J. Syst. Sci. Syst. Eng.* **30**, 482–494 (2021). <https://doi.org/10.1007/s11518-021-5499-z>
4. Kipf, T., Welling, M.: Semi-supervised classification with graph convolutional networks (2017). <https://doi.org/10.48550/arXiv.1609.02907>
5. Tran, T., Snasel, V.: improvement graph convolution collaborative filtering with weighted addition input. *Intell. Inf. Database Syst.* 635–647 (2022). https://doi.org/10.1007/978-3-031-21743-2_51

6. Nguyen, L., Tran, T.: CombiGCN: an effective GCN model for recommender system. *Comput. Data Soc. Netw.* 111–119 (2024). https://doi.org/10.1007/978-981-97-0669-3_11
7. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: simplifying and powering graph convolution network for recommendation. In: *Proceedings Of The 43rd International ACM SIGIR Conference On Research And Development In Information Retrieval*, pp. 639–648 (2020). <https://doi.org/10.1145/3397271.3401063>
8. Yu, J., Yin, H., Xia, X., Chen, T., Cui, L., Nguyen, Q.: Are graph augmentations necessary? Simple Graph Contrastive Learning for Recommendation (2022). <https://doi.org/10.1145/3477495.3531937>
9. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding (2019). <https://doi.org/10.48550/arXiv.1810.04805>
10. Bafna, P., Pramod, D., Vaidya, A. Document clustering: TF-IDF approach, March 2016. <https://doi.org/10.1109/ICEEOT.2016.7754750>
11. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: *Proceedings Of The Twenty-Fifth Conference On Uncertainty In Artificial Intelligence*, pp. 452–461 (2009). <https://doi.org/10.48550/arXiv.1205.2618>
12. Wang, X., He, X., Wang, M., Feng, F., Chua, T.: Neural graph collaborative filtering. In: *Proceedings Of The 42nd International ACM SIGIR Conference On Research And Development In Information Retrieval*, July 2019. <https://doi.org/10.1145/3331184.3331267>



Integrating Kelly Criterion with Technical Indicators for VN30 Stock Market

Dao Lan Vy Dinh¹, Ngoc Hang Tran¹, Vo Huyen Khanh May¹,
Hung Tran¹, Tran Duc Minh¹, Dang Thu Lan²,
and Van Nhan Vo³

¹ DATCOM Lab, Faculty of Data Science and Artificial Intelligence,
College of Technology, National Economics University, Hanoi, Vietnam
hung.tran@neu.edu.vn

² Faculty of Business and Economics, Phenikaa University, Yen Nghia, Ha Dong,
Hanoi, Vietnam

³ Faculty of Information Technology, Duy Tan University, Da Nang 550000, Vietnam

Abstract. Recently algorithmic trading has revolutionized financial markets. This study investigates the application of the Kelly criterion and technical indicators pairs (relative strength index RSI, moving average convergence divergence (MACD) on-balance volume (OBV), Bollinger bands (BB)) to optimize trading strategies within the Vietnamese VN30 index. Historical data from 2021–2024 was split into 2 parts, one is to find parameters for Kelly Criteria, another is used to analyze and backtest the proposed models. Results indicate that combining RSI and MACD is effective during downtrends, while RSI with OBV and BB performs better in uptrends. The Kelly Criterion consistently enhances risk management during periods, especially in downtrends. This research fills a gap in the literature and offers a foundation for future studies on algorithmic trading in emerging markets.

Keywords: technical analysis (TA) · technical analysis (TI) · RSI · MACD · BB · OBV · Vietnam's Stock Market

1 Introduction

Algorithmic trading has revolutionized financial markets by leveraging computers to implement high-speed strategies. In stock trading, these strategies use TA based on historical price and volume data to forecast future price movements, in contrast to fundamental analysis (FA), which assesses a stock's intrinsic value. This study explores algorithmic trading within Vietnam's VN30 index, integrating RSI with TI and optimizing capital allocation for risk minimizing through the Kelly criterion.

Previous research shows mixed results on the efficiency of TA. Rosillo *et al.* [11] found that RSI, MACD, Momentum, and Stochastic indicators' performance

varies by company type. Radukić *et al.* [10] studied MACD for long-term trends in Serbia, while Salkar *et al.* [12] (2021) showed RSI and MACD achieved up to 12% returns in intraday trading. Other studies, including Le *et al.* [4] and Phan *et al.* [13], explored various TA strategies for VN30 stocks, finding that sma-optimized portfolios and RSI were efficient in different market conditions. Despite advances, research on portfolio optimization in Vietnam is limited, with studies like Luu, T. Q. [7] and Dao, B. [3] exploring genetic algorithms and the t copula Mean-CVaR framework, respectively, for risk management. However, research on the Kelly Criterion applied in Vietnam's market remains underexplored.

This paper addresses the shortcomings of prior studies by proposing a strategy that combines RSI with other TI to enhance trading in the Vietnamese VN30 stock market. Focusing on intra-day trading, our study uses quantitative time series analysis to maximize profit opportunities and applies the Kelly Criterion for capital allocation and risk management. Tested with real data from Vietnam's stock market, the results demonstrate the profitability of combining RSI with other TI and the efficiency of the risk management approach. To the best of the authors' knowledge, this is the first study exploring this problem. Key contributions include: 1) Investigating RSI combined with MACD, OBV, and BB indicators; 2) Integrating the Kelly criterion with these indicator combinations to propose and backtest a trading algorithm; 3) Showing that RSI and MACD are efficient for stock accumulation during recessions, while RSI with OBV and BB are recommended for uptrend markets.

2 Background

In this section, we introduce TI, the Kelly Criterion, and the dataset used to inform trading decisions, as well as the performance metrics used to assess the proposed strategies.

2.1 Technical Indicators

TIs are crucial in stock trading, providing quantitative analyses to support informed decision-making. They help identify market trends and generate buy or sell signals [5,9]. By incorporating TIs, traders can better predict price movements and enhance strategies, complementing FA for a more comprehensive approach.

More specifically, **RSI** assesses the velocity and magnitude of price movements to identify overbought or oversold conditions [5]; **MACD** is a momentum indicator; it tries to predict stock market trends by a comparison between short and long-term trends [9]; **OBV** is a volume-based indicator that measures cumulative buying and selling pressure by adding volume on up days and subtracting it on down days [9]; **BB** is a volatility indicator consisting of a middle band and two outer bands that are standard deviations away from the middle band [16].

2.2 Kelly Criterion

The Kelly Criterion is a mathematical formula designed to determine the investment fraction associated with the optimal long term expected logarithmic returns and later applied to others gambling game and financial markets [6]. Since then many researchers has studied the use of Kelly criterion on determine the optimal investment ratio [2, 8]. It is formulated as $f = \frac{p(1+b)-1}{b}$ where f is the fraction of capital to be allocated to the investment. Symbols p and b denote the winning rate and the decimal odds, respectively.

2.3 Performance Metrics

In our study, we use four performance metrics, including average return (AR), average profit (AP), average loss (AL), and volatility of rate of return (VL) to examine the efficiency of trading strategies as follows: 1) **AR** is the mean rate of return for 30 stocks in the VN30 Index after backtesting given as $AR = \frac{1}{30} \sum_{i=1}^{30} R_i$ where R_i is the rate of return for stock i . 2) **AP** is the mean rate of return for stocks with positive returns after backtesting expressed as $AP = \frac{1}{n_{\text{positive}}} \sum_{i \in \text{Positive}} R_i$, where n_{positive} is the number of stocks with positive returns, and the summation is over all stocks with positive returns. 3) **AL** is the mean rate of return for stocks with negative returns after backtesting. It is formulated as $AL = \frac{1}{n_{\text{Negative}}} \sum_{i \in \text{Negative}} R_i$ where n_{negative} is the number of stocks with negative returns, and the summation is over all stocks with negative returns. 4) **VL** is the standard deviation of the rate of return for 30 stocks in the VN30 Index after backtesting. It is calculated as $VL = \sqrt{\frac{1}{30} \sum_{i=1}^{30} (R_i - AR)^2}$ where R_i is the rate of return for stock i , and AR is the mean rate of return for all 30 stocks as calculated above.

2.4 Dataset

Data was collected from the VN30 index, comprising the top 30 stocks selected by state securities commission (SSC) based on market capitalization and liquidity. The collection period spans from January 1, 2021, to January 1, 2024, covering around 744 trading days [15]. The dataset includes closing prices, opening prices, and trading volumes of VN30 stocks, gathered using the Vnstock library [14].

3 Trading Strategies

To assess the efficiency of various indicators, RSI was designated as the primary metric. Subsequently, RSI was integrated with additional indicators, including MACD, OBV, and BB. Moreover, Kelly criterion was employed to analyze the influence of capital allocation on investment performance.

3.1 RSI and MACD

This combined RSI-MACD strategy uses RSI (14) and MACD (12, 26, 9) to identify buy and sell signals [9]. When RSI exceeds 30 (oversold) and MACD crosses above the Signal line, a buy signal is generated. Conversely, a sell signal occurs when RSI falls below 70 (overbought) and MACD crosses below the Signal line [1]. This strategy leverages both momentum and trend-following indicators for more reliable trading decisions.

3.2 RSI and OBV Strategy

To make a fair comparison with other strategies, RSI is calculated with the same period, i.e., a period of 14 days as mentioned in Subsect. 3.1. This RSI-OBV strategy combines RSI (14) and OBV slope (5) to identify buy and sell signals. When RSI crosses above 30 (oversold) and OBV slope is positive, a buy signal is generated [9]. Conversely, a sell signal occurs when RSI falls below 70 (overbought) and OBV slope is negative [1]. This strategy uses both momentum and volume indicators to improve trading decisions.

3.3 RSI and BB Strategy

RSI-BB strategy combines RSI (14) and BB (15, 2) to identify buy and sell signals. A buy signal is generated when price is at or below the lower BB and RSI is at or below 30 (oversold). Conversely, a sell signal occurs when price is at or above the upper BB and RSI is at or above 70 (overbought) [1]. This strategy leverages both momentum and volatility indicators to improve trading decisions.

3.4 Trading Algorithm

The trading algorithm scans VN30 stocks during the backtest period (Jan. 1, 2021 to Jan. 2, 2024). For each stock, it calculates indicators and generates buy/sell signals based on defined conditions. Buy signals trigger purchases using the Kelly criterion in Subsect. 2.2. Sell signals result in selling a calculated number of shares. The algorithm updates holdings and revenue accordingly following the $t + 2$ rule. The pseudocode Algorithm 1 outlines this process in detail.

4 Numerical Results

The dataset is split into two periods: estimate parameters period (Jan. 1, 2019 to Jan. 2, 2021) and Jan. 1, 2021 to Jan. 2, 2024. Without loss of generality, the parameters are set up for running the algorithm as follows:

- Total budget for investment: 100 millions VND.
- Decimal odds are derived by calculating the ratio of the mean profit to the mean loss, following an analysis of the three most efficient indicator pairs, which are demonstrated in Table 1.

Algorithm 1. Trading Algorithm

```

1: Purpose: Backtesting the trading strategy efficiency in real-life settings.
2: Inputs: Historical data, initial cash/capital.
3: Outputs: Portfolio value, performance metrics.
4: Definitions:
5: Buy signal: Strategy 3.1, 3.2, 3.3.
6: Sell signal: Strategy 3.1, 3.2, 3.3.
7: Kelly criterion: Calculated as mentioned in Subsection 2.2
8: Settlement: 2-day process of executing a buy or sell order.
9: Pending buys/sells: Orders awaiting execution.
10: for each day in data do
11:   Settle pending buys and sells (if any)
12:   if new year (Jan 2024) then
13:     Calculate portfolio value
14:     continue
15:   end if
16:   if buy signal then
17:     Calculate allocation based on Kelly criterion
18:     Buy shares and deduct the cost from the cash
19:     Schedule settlement for shares
20:   end if
21:   if holding shares then
22:     Determine sell condition
23:     if sell condition met then
24:       Calculate shares to sell and sell shares
25:       Update remaining holdings
26:       Schedule settlement for revenue
27:     end if
28:   end if
29:   Calculate and store portfolio value
30: end for
31: Settle any remaining pending buys/sells
32: Calculate final portfolio value and store it

```

- The results of the winning rate are calculated based on the dataset train_period and they are showed in Table 1.
- Detail implementation for algorithms and trading strategies can be found here [15].

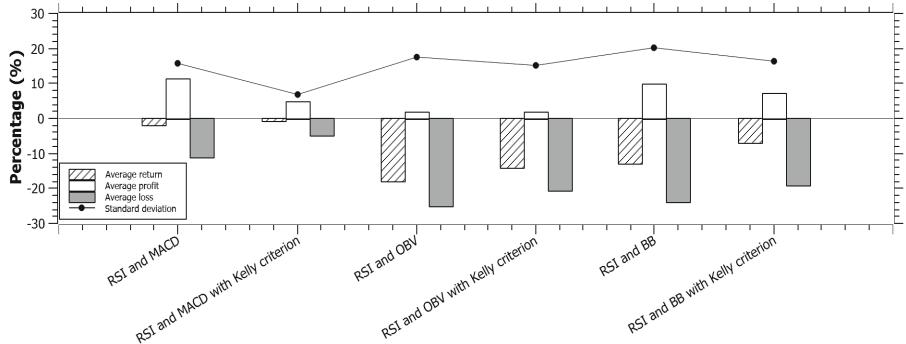
4.1 RSI Combines with Other Indicators in Recession Period (2022)

In Fig. 1, we compare the performance of the algorithm for two cases: not apply Kelly criterion (NAKC) and apply Kelly criterion (AKC), and the data back-test went through a recession period (2022), during which the stock market in Vietnam was in a downtrend and nearly collapsed.

More specifically, considering NAKC cases for challenging periods, the AP of the combination of RSI and MACD is the highest, and the AL is the lowest. This indicates that the combination of RSI and MACD is the most efficient compared to other indicator combinations, suggesting that RSI and MACD should be the first choice in TA during recession time. In contrast, the AP and AL of the

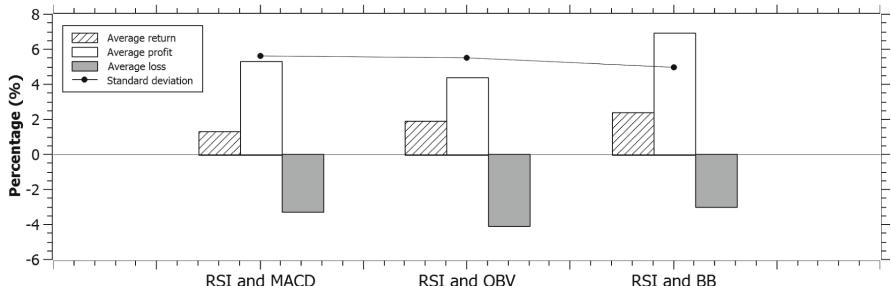
Table 1. Winning rate and decimal odds

Combinations	Winning rate p (%)	Decimal odds b
RSI and MACD	55.21	3.00
RSI and OBV	79.77	2.39
RSI and Bollinger band	54.46	2.31

**Fig. 1.** Different indicator combinations with and without using the Kelly criterion on VN30 in the difficult period 2022.

RSI and OBV combination are the worst, implying that this combination is not effective in TA in downtrends.

Comparing NAKC and AKC in Fig. 1, we find that AP for NAKC consistently outperforms AKC across all indicator combinations, while AL is more pronounced for NAKC. This is because NAKC invests all funds immediately at a buy point, whereas AKC allocates funds incrementally. In a downward market, NAKC suffers greater losses due to its lack of flexibility to buy at lower prices, while AKC's progressive buying results in smaller losses, making its AR superior. Thus, the Kelly criterion effectively protects assets. Notably, Fig. 1 shows that AR with AKC is highest (-0.80) when combining RSI and MACD, indicating this strategy is effective for accumulating stocks during a market downturn.

**Fig. 2.** Different indicator combinations using the Kelly criterion in 2023

4.2 RSI Combines with Other Indicators in Uptrend Period (2023)

In Fig. 2, we present the AR, AP, and AL for different indicator combinations during the uptrend period of 2023. It is noteworthy that the AR for the combinations of RSI and BB, as well as RSI and OBV, outperform the combination of RSI and MACD. This can be attributed to the fact that during an uptrend, the RSI rarely drops below 30 (oversold), resulting in fewer opportunities to accumulate shares. Consequently, the potential for high profits diminishes as stock prices increase. In other words, the application of the Kelly criterion to risk management and profit enhancement proves more effective with the combinations of RSI and BB, and RSI and OBV during an uptrend market phase. This observation is further validated by Fig. 3 where the backtest went through from 2021 to 2023.

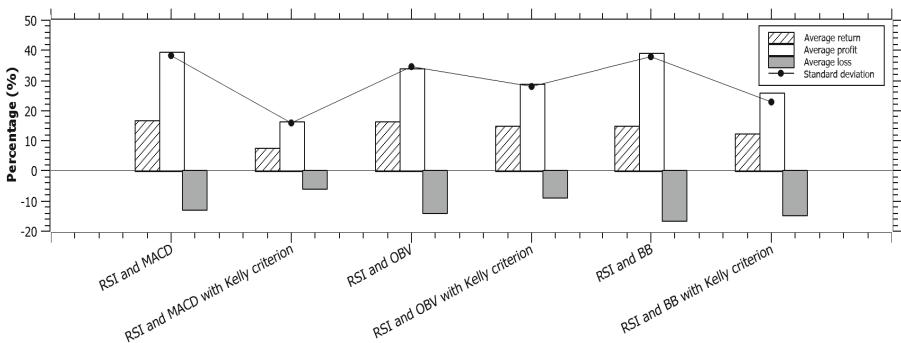


Fig. 3. Different indicator combinations with and without using the Kelly criterion on VN30 from 2021 to 2023

5 Conclusions

Our analysis of RSI combined indicators pairs in VN30 stocks found that RSI-MACD is efficient during recessions, while RSI-OBV and RSI-BB excel in uptrends. However, technical analysis alone may be limiting, considering both technical signals and company fundamentals can potentially achieve better returns and reduced risk since it could provide more comprehensive and profitable trading strategies.

References

1. Bansal, S.: Investigating the efficacy of RSI in the nifty 50 index. Glob. J. Bus. Integral Secur. (2023). <https://www.gbis.ch/index.php/gbis/article/view/159>
2. Chen, L., Sun, L., Chen, C.M., Wu, M.E., Wu, J.M.T.: Stock trading system based on machine learning and Kelly criterion in Internet of Things. Wirel. Commun. Mob. Comput. **2021**(1), 7632052 (2021)

3. Le, A., Dao, B.: Portfolio optimization under mean-CVaR simulation with copulas on the Vietnamese stock exchange. *Investment Manag. Financ. Innov.* **18**(2) (2021)
4. Le, H.A., et al.: Ứng dụng phương pháp học máy trong giao dịch chứng khoán theo chỉ báo bằng ngôn ngữ lập trình python. *Tạp chí Khoa học và kỹ thuật trường Đại học Bình Dương* **7**(1) (2024)
5. Lin, X., Yang, Z., Song, Y.: Intelligent stock trading system based on improved technical analysis and echo state network. *Expert Syst. Appl.* **38**(9), 11347–11354 (2011)
6. Rotando, L.M., Thorp, E.O.: The Kelly criterion and the stock market. *Am. Math. Mon.* **99**(10), 922–931 (1992)
7. Luu, T.Q.: Application of artificial intelligence-genetic algorithms to select stock portfolios in the Asian markets. *Int. J. Adv. Comput. Sci. Appl.* **13**(12) (2022)
8. Mu-en, W., Jia-Hao, S., Gautam, S., Jerry, L.: Informative index for investment based on Kelly criterion. *Enterp. Inf. Syst.* **16**, 1–20 (2021). <https://doi.org/10.1080/17517575.2021.1939425>
9. Nti, I.K., Adekoya, A.F., Weyori, B.A.: A systematic review of fundamental and technical analysis of stock market predictions. *Artif. Intell. Rev.* **53**(4), 3007–3057 (2020)
10. Radukic, S., Radović, M.: Long term trend analysis in the capital market – the case of Serbia. *J. Central Banking Theory Pract.* **3** (2014). <https://doi.org/10.2478/jcbtp-2014-0013>
11. Rosillo, R., Fuente, D., Brugos, J.A.: Technical analysis and the Spanish stock exchange: testing the RSI, MACD, momentum and stochastic rules using Spanish market companies. *Appl. Econ.* **45**, 1541–1550 (2013). <https://doi.org/10.1080/00036846.2011.631894>
12. Salkar, T., Shinde, A., Tamhankar, N., Bhagat, N.: Algorithmic trading using technical indicators. In: 2021 International Conference on Communication information and Computing Technology (ICCICT), pp. 1–6 (2021). <https://doi.org/10.1109/ICCICT50803.2021.9510135>
13. Tam, P.H., Cuong, N.T.: Effectiveness of investment strategies based on technical indicators: evidence from Vietnamese stock markets article info JEL classification keywords. *J. Insurance Financ. Manag.* **3** (2018)
14. Vu, T.: Vnstock library (2024). <https://vnstocks.com/>
15. Vy, D.D.L., Hang, T.N., May, V.H.K.: Source Code and Data. <https://shorturl.at/1ul1E>
16. Williams, O.: Empirical optimization of Bollinger bands for profitability (2006)

AI in Health Care Analytics



Enhancing the Efficiency of Lung Disease Classification Based on Multi-modal Fusion Model

Thi-Diem Truong^{1,2} , Phuoc-Hai Huynh^{1,2} , Van Hoa Nguyen^{1,2} , and Thanh-Nghi Do^{3,4}

¹ Faculty of Information Technology, An Giang University,
Long Xuyên, An Giang, Vietnam

ttdiem,hphai,nvhoa}@agu.edu.vn

² Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Vietnam

³ College of Information Technology, Can Tho University, Cantho 92000, Vietnam

⁴ UMI UMMISCO 209 (IRD/UPMC) Sorbonne University,
Pierre and Marie Curie University, Paris 6, France

dtnghi@cit.ctu.edu.vn

Abstract. Lung diseases affect millions of people worldwide, posing a serious health challenge. Timely and accurate diagnosis is crucial for improving patient outcomes and ensuring effective treatment. In this paper, we propose a multi-modal model that integrates both chest X-ray images and clinical information text to improve the efficiency of lung disease classification. Our proposed model trains a Support Vector Machine (SVM) model on top of the fine-tuned VGG16 model and Extreme Gradient Boosting (XGBoost) model with TF-IDF feature extraction. We started by collecting a new real-world dataset of chest x-ray images and clinical information at General Hospital of An Giang area province. Experimental results on newly collected real dataset show that the VGG16 model, which uses chest X-ray images, achieves an accuracy of 62.66%, indicating limitations when relying solely on image data for lung disease classification. With clinical text data, the SVM and XGBoost models with TF-IDF feature extraction achieves accuracy of 87.40% and 89.26%, respectively. Our multi-modal fusion model achieves the highest accuracy of 89.64%. These results highlight the effectiveness of our multi-modal approach, providing a more accurate and comprehensive diagnosis of lung disease classification.

Keywords: Lung disease classification · multi-modal model · X-ray image · clinical text · support vector machines

1 Introduction

Today, lung diseases are one of the primary global health concerns. Common lung diseases, such as asthma, pneumonia, lung cancer, tuberculosis, and chronic obstructive pulmonary disease (COPD), are widely acknowledged as major

causes of illness and some of the most common causes of death worldwide [23]. According to the World Health Organization (WHO), pneumonia is the leading cause of death in children worldwide, accounting for 14% of all deaths in children under 5 years old, claiming the lives of 740,180 children in 2019 [16]. According to the 2022 Global Asthma Report, asthma is a common chronic disease, estimated to affect 262 million people worldwide and cause more than 1,000 deaths per day [15]. Consequently, precise classification of lung conditions can aid physicians in selecting optimal treatment approaches and enhancing treatment outcomes for patients.

In modern medicine, accurate diagnosis of diseases is crucial for ensuring effective patient treatment. Medical data, including clinical text and medical images, play a vital role in providing detailed information about the patient's condition. Clinical text reports typically contain detailed information about symptoms, medical history, and diagnostic conclusions by physicians, while X-ray images offer a visual depiction of internal body structures, aiding in the identification of abnormalities. Traditionally, classification methods often focus on a single data type such as chest X-ray images or clinical text reports, each providing partial insights into the patient's condition [10]. This limitation restricts the utilization of comprehensive information available for making accurate diagnostic decisions. Therefore, integrating multiple data modalities has shown promise in enhancing diagnostic accuracy by gathering additional information from diverse perspectives. This comprehensive approach allows classifiers to achieve a deeper understanding of disease conditions, effectively addressing the inherent ambiguity and uncertainty of lung diseases. There is increasing interest in harnessing the power of deep learning techniques and language models to develop multi-modal models that can classify diseases using diverse data sources, such as medical images and clinical text.

In this paper, we propose a multi-modal model that integrates chest X-ray images and clinical text data for lung disease classification. Our approach enhances classification performance by training a model SVM on top of a fine-tuned VGG16 model and Extreme Gradient Boosting (XGBoost) with Term Frequency-Inverse Document Frequency (TF-IDF). This method significantly outperforms the classification accuracy achieved using only chest X-ray images or clinical text data independently. Our research contributions focus on two key areas. Firstly, we built a new dataset that integrates chest X-ray images with relevant clinical text data from General Hospital of An Giang area province. This dataset is labeled based on discharge outcomes from electronic medical records (EMRs) and categorized into 12 classes: Normal, COPD, Covid-19, Asthma, Tuberculosis, Pulmonary Oedema, Respiratory Failure, Pleural Effusion, Pneumothorax, Malignant Neoplasm, Pneumonia, and Pulmonary Collapse. Secondly, we propose a multi-modal model that trains SVM on top of a fine-tuned VGG16 model and XGBoost with TF-IDF for lung disease classification. Experimental results show that our proposed multi-modal model achieves the highest accuracy of 89.64%, outperforming the uni-modal approaches, with VGG16 achieving 62.66% and XGBoost achieving 89.26%. This highlights the

combined advantages of integrating both image and text data to enhance the accuracy and effectiveness of lung disease classification.

Our paper includes the following sections: Sect. 1 provides background information, Sect. 2 introduces related research, Sect. 3 describes the methodologies used in this study, experimental evaluation is detailed in Sect. 4, and conclusions along with future directions are presented in Sect. 5.

2 Related Work

Advancements in machine learning techniques have opened up great prospects in the classification of lung diseases, promising substantial improvements in the diagnosis and treatment processes. Researchers and healthcare professionals have been focusing on developing machine learning models to classify various types of diseases based on diverse medical data. Given the diversity of medical data, numerous studies have focused on the analysis and exploitation of medical data based on both imaging and text. Convolutional Neural Networks (CNN) have become the standard for medical image analysis due to their ability to understand complex features from available data, often surpassing human capabilities in many image comprehension tasks [17, 21]. During the Covid-19 outbreak, many research efforts focused on diagnosing Covid-19 through chest X-ray images using deep learning techniques [1, 7, 11]. In parallel, machine learning models have also been widely applied in disease classification based on clinical text descriptions. Supervised machine learning techniques, including SVM [19], Random Forests [2], and decision trees [3], have demonstrated effectiveness in disease classification using textual data [4, 12, 14].

In recent years, there has been increasing interest in using multi-modal approaches to disease classification, leveraging diverse data sources to improve the accuracy of diagnosis and treatment outcomes. Several studies have investigated the integration of machine learning models with both medical imaging data and clinical text information, demonstrating promising outcomes in various healthcare applications [9, 13, 22]. In research [8], Hayat et al. proposes MedFuse, a LSTM-based fusion module capable of handling both uni-modal and multi-modal inputs. MedFuse is employed to predict hospital mortality rates and classify phenotypes using clinical time-series data and chest X-ray images. It has demonstrated enhanced performance compared to alternative complex multi-modal fusion approaches.

3 Methodology

3.1 Model Overview

Our proposed model presents a multi-modal approach that integrates chest X-ray images and clinical information to enhance the efficiency of lung disease classification. The proposed model focuses on training a SVM on top of the VGG16 model for chest X-ray images and the XGBoost model for clinical text

data. This process involves two main branches: the chest X-ray image classification branch and the clinical text classification branch. In the image classification branch, chest X-ray images are preprocessed before being fed into a fine-tuned VGG16 model to output class probabilities. In the clinical text classification branch, clinical texts are preprocessed and features are extracted using TF-IDF, followed by input into the XGBoost model to generate class probabilities. These two class probabilities are combined through a probability fusion step. Finally, the SVM model is trained on top of the VGG16 and XGBoost models for lung disease classification. Figure 1 illustrates the overall architecture of the proposed model.

3.2 Data Collection

In this study, we constructed a new real-world dataset of chest X-ray images and clinical information at General Hospital of An Giang area province. The dataset collected 17,973 samples, each containing a DICOM-format chest X-ray image and clinical information text. Clinical information of lung disease includes details about symptoms, signs, test results, imaging findings, medical history, treatment process, and the patient's response to treatment. This information helps doctors better understand the patient's health condition and make appropriate treatment decisions. The clinical information was collected from EMRs. In the next preprocessing step, we only selected necessary and important information from EMR to fit the lung disease text classification model.

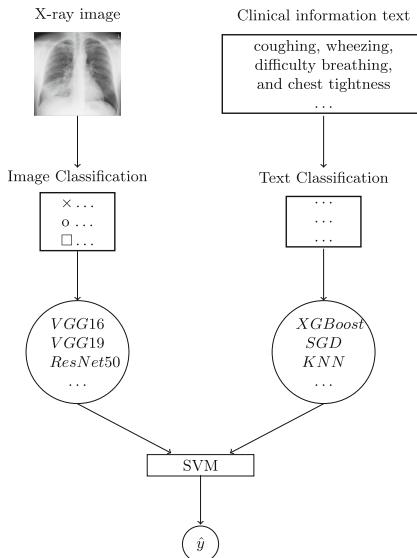


Fig. 1. Training the multi-modal fusion model for lung disease classification

3.3 Data Preprocessing

Chest X-Ray Image Preprocessing: Image preprocessing is an important step in preparing input data for machine learning models. With newly collected real-world datasets, chest X-ray images in DICOM format are converted to JPEG format to reduce file size and ensure compatibility with proposed models. Subsequently, image preprocessing techniques are employed to enhance image quality, simplify computations, and standardize data. Specifically, all images are normalized according to input size requirements corresponding to each network architecture. Furthermore, the training dataset is artificially expanded using data augmentation techniques. This is achieved by applying random image transformations such as rotation, flipping, scaling, and shift. This process enhances the model's generalization ability while reducing overfitting risks.

Clinical Text Preprocessing: Text data preprocessing is performed after data collection to prepare the data before feeding it into machine learning models. This process involves a series of steps to clean and standardize the data, thereby improving the performance of classification models. The raw dataset collected from EMRs at General Hospital of An Giang area province, which is in Vietnamese text and is very complex (with a large amount of information, complexity, and noise). Therefore, we selectively extract relevant information that can be used for the machine learning models in the next step. Additionally, the data is cleaned by correcting errors, supplementing, or removing data records with a lot of missing information. Furthermore, common preprocessing steps for text classification include tokenization, converting all characters in the text to lowercase, removing unnecessary characters, punctuation, numbers, and special characters. The dataset after preprocessing consists of a total of 17,973 records, each record includes 7 attributes: symptoms, general examination, respiratory system, circulatory system, personal medical history, summary of paraclinical results, and medical record summary. After completing preprocessing, the dataset is used to build a lung disease classification model and is ready for the feature extraction step.

3.4 Image Classification Model

In our study, we propose classifying lung diseases based on chest X-ray images using fine-tuned deep learning models. The goal is to use several fine-tuned deep learning models to train on chest X-ray images, classify them, and then select the most accurate model to feed into the proposed multi-modal fusion model. To fine-tune the lung disease classification model on chest X-ray images, we start by selecting pre-trained models such as VGG16, VGG19, InceptionV3, ResNet50, Xception, DenseNet121, MobileNetV2, and EfficientNetB0, which have been trained on a large dataset like ImageNet. We then fine-tune these models on a smaller dataset to improve the accuracy for lung disease classification. The model architecture is modified by removing the final layer of the pre-trained

model, which is typically responsible for classification tasks. Then, add new layers that specifically designed for the new task, such as global average pooling layers, dense layers, and a softmax output layer (a new output layer for classification). Next, freeze the pre-trained layers to prevent their weights from being updated during training. Train the new model on the newly collected chest X-ray dataset using a smaller learning rate to prevent overfitting. Once the new layers are trained, some deeper layers are unfrozen and training continues to fine-tune the model.

Table 1. Fine-tuning configurations for deep learning networks

No	Deep learning networks	Number of fine-tuned last layers
1	VGG16	8
2	VGG19	9
3	InceptionV3	149
4	ResNet50	14
5	Xception	39
6	DenseNet121	19
7	MobileNetV2	76
8	EfficientNetB0	35

During the experiments, we implemented fine-tuning configurations for deep learning networks as shown in Table 1. Based on the experimental results, we analyzed and compared the accuracy of various deep learning models on the current chest X-ray image dataset. We selected the fine-tuned VGG16 model, which achieved the highest accuracy of 62.66% among the experimental models. The fine-tuned VGG16 model will output the probabilities of the lung disease classes. These probabilities will then be used as input for the SVM model to perform classification according to the proposed method.

3.5 Text Classification Model

According to the automatic text classification approach using machine learning models [18], text classification involves two main steps: text data representation and training the classification model. Text data representation, also known as feature extraction from text data, plays a crucial role in transforming unstructured medical information into structured formats suitable for machine learning models. In this study, the input data consists of preprocessed clinical text, which undergoes feature extraction to convert text into numeric vectors. We conducted experiments with several popular feature extraction methods such as TF-IDF, Bag-of-Words (BoW), Word to Vector (Word2Vec), Global Vectors for Word Representation (GloVe), FastText, and Bidirectional Encoder Representations from Transformers (BERT) [6]. Subsequently, we compared and evaluated the accuracy results of these feature extraction methods on a clinical text

dataset. The experimental results showed that the XGBoost model with TF-IDF achieved the highest accuracy of 89.26%. Therefore, we will use the output of the TF-IDF model as input for the next text classification model.

The feature vectors extracted using the TF-IDF method were input into machine learning models for lung disease classification. In our experiments, we trained nine types of machine learning models: Logistic Regression, Gaussian Naive Bayes, K-Nearest Neighbors [5], Multi-Layer Perceptron (MLP), Multinomial Naive Bayes, SVM [20], Random Forest, Stochastic Gradient Descent(SGD), and XGBOOST. From the experimental results, we analyzed, compared, and evaluated the accuracy of these models on the current textual dataset. The results indicated that the XGBOOST model achieved the highest accuracy at 89.26%, surpassing the other models in our experiments. Therefore, we selected the XGBOOST model for lung disease classification on clinical text. The XGBOOST model will provide the probabilities of the lung disease classes, which will then be used as input for the SVM model to perform multi-modal fusion.

3.6 Training Support Vector Machine on Top of VGG16 and XGBoost

The class probabilities obtained from image classification (fine-tuned VGG16 model) and text classification (XGBoost model with TF-IDF feature extraction) are combined into one unified probability vector. This fusion step aims to integrate complementary information from both modalities to enhance diagnostic accuracy.

SVM [20] are among the most popular machine learning techniques for classification and regression tasks. In Vapnik et al. [20], SVM aims to identify the optimal decision boundary to maximize the margin between classes. The SVM was selected due to its ability to effectively manage high-dimensional data and help prevent overfitting. Furthermore, SVM has emerged as a reliable choice across various applications. Therefore, we have opted for the SVM model as our classifier.

In our study, we propose training an SVM model on top of both fine-tuned VGG16 model and an XGBoost model with TF-IDF. By combining the class probabilities of these two models as inputs to the SVM model, we aim to leverage the combined benefits of both image and text information. The final outcome of lung disease classification depends on this integrated approach. This enhanced method aims to achieve more accurate and robust classification of lung diseases.

4 Experiments

4.1 Dataset Description

In the experiment of this paper, we constructed a new dataset of chest X-ray images and clinical information at General Hospital of An Giang area province.

The dataset was extracted from the hospital’s EMRs and preprocessed before being fed into experimental models. The newly collected dataset includes 17,973 samples, each containing a chest X-ray image and corresponding clinical text information, illustrated in Fig. 2.

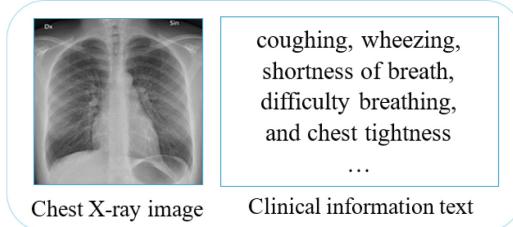


Fig. 2. An example of a data sample

The data labeling was based on patient discharge results from the EMRs. The assigned labels include 12 common lung diseases from the new dataset: Normal, COPD, COVID-19, Asthma, Tuberculosis, Pulmonary Oedema, Respiratory Failure, Pleural Effusion, Pneumothorax, Malignant Neoplasm, Pneumonia, and Pulmonary Collapse. The dataset was randomly divided into a training set consisting of 14,371 samples and a test set consisting of 3,602 samples, as detailed in Table 2.

Table 2. Description of the dataset

No	Label	Train set	Test set
1	Normal	2469	618
2	COPD	490	123
3	Covid-19	2000	501
4	Asthma	153	39
5	Tuberculosis	657	165
6	Pulmonary Oedema	149	38
7	Respiratory Failure	1105	277
8	Pleural Effusion	452	114
9	Pneumothorax	213	54
10	Malignant Neoplasm	128	33
11	Pneumonia	6472	1618
12	Pulmonary Collapse	83	22
	Total	14371	3602

4.2 Image Classification Results

We conducted multiple experiments on lung disease classification based on chest X-ray images, using fine-tuned deep learning models including VGG16, VGG19, InceptionV3, ResNet50, Xception, DenseNet121, MobileNetV2, and EfficientNetB0. The experimental results are illustrated in Fig. 3.

The experimental results highlight significant differences between our model and the remaining experimental models. Fine-tuned models such as VGG16 and VGG19 achieved accuracies of 62.66% and 61.55%, respectively, while EfficientNetB0 also performed fairly well with 61.16%. Other models like ResNet50, DenseNet121, and Xception had accuracies ranging from about 58% to 60%. MobileNetV2 had the lowest accuracy at 51.80%, indicating potential unsuitability for this task.

On the contrary, our proposed model achieved an impressive accuracy of 89.64%, surpassing all uni-modal models used in the experiment for X-ray image classification. The improvement of nearly 27% compared to the best-performing standard model (VGG16) highlights the effectiveness of the enhancements and techniques we implemented. This result demonstrates that our new approach has significantly enhanced classification performance.

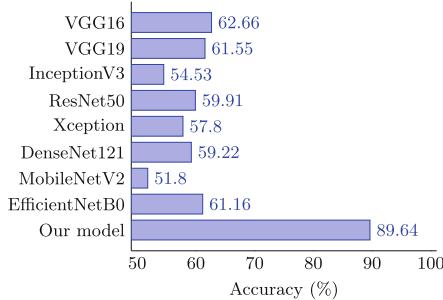


Fig. 3. Image classification results

To determine the next steps in our model development, we analyzed the classification performance of various models, focusing on their accuracies. Based on the experimental results depicted in Fig. 3, VGG16 demonstrates a notable accuracy of 62.66%, outperforming the remaining uni-modal models for image classification. The VGG16 model has demonstrated its effectiveness in capturing and classifying features from chest X-ray images. Therefore, we selected the output of the VGG16 model as the input for the SVM model in the next development stage.

4.3 Text Classification Results

Our study evaluated various machine learning models for lung disease classification using different feature extraction methods from clinical text data. In

clinical text classification, various machine learning models commonly used for experimentation include Logistic Regression, Gaussian Naive Bayes, K-Nearest Neighbors, Multi-layer Perceptron, Multinomial Naive Bayes, Support Vector Machine, Random Forest, Stochastic Gradient Descent, and Extreme Gradient Boosting. Furthermore, the feature extraction methods used in the experiments include BoW, TF-IDF, Word2Vec, GloVe, FastText, and BERT. Figure 4 illustrates the accuracy of various machine learning models using different feature extraction methods for lung disease classification. These results offer important insights into the performance of each model based on the feature extraction methods.

Based on the analysis of machine learning models for lung disease classification from clinical text, several important observations can be made. The Extreme Gradient Boosting (XGBoost) model achieved the highest accuracy across all feature extraction methods, with accuracies of 89.26% (BoW and TF-IDF), 84.62% (Word2Vec), 81.04% (GloVe), 72.13% (FastText), and 81.07% (BERT). This indicates that XGBoost performs well and consistently across various feature extraction approaches. Other models such as Logistic Regression, SGD, MLP and SVM also showed promising results, with accuracies ranging from 72% to over 86% depending on the feature extraction method.

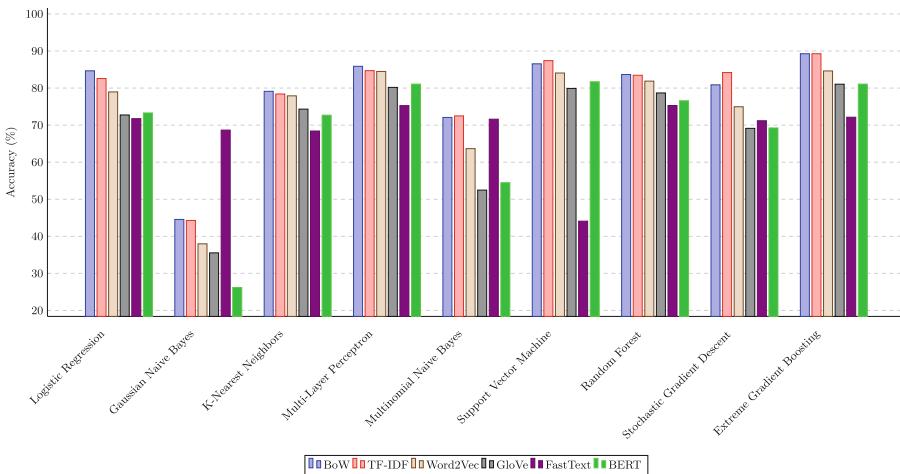


Fig. 4. Compare the accuracy of machine learning models across different feature extraction methods.

Traditional feature extraction methods such as BoW and TF-IDF consistently perform better than word embedding methods such as Word2Vec and GloVe. Gaussian Naive Bayes and Multinomial Naive Bayes tended to exhibit lower accuracy compared to other models, especially when using word embedding methods like Word2Vec and GloVe. However, FastText notably performed

significantly better with the Gaussian Naive Bayes model, while BERT demonstrated substantial improvements over baseline models but showed limitations compared to other feature extraction methods. Overall, XGBoost dominated in accuracy, especially with TF-IDF and BoW. From the experimental results, we chose the XGBoost model with TF-IDF feature extraction because it achieved the highest accuracy of 89.26% compared to the uni-modal models for lung disease classification based on clinical text. Next, we will use the output of the XGBoost model as the input for the SVM model for the next stage. Figure 5 illustrates the accuracy of machine learning models with TF-IDF feature extraction.

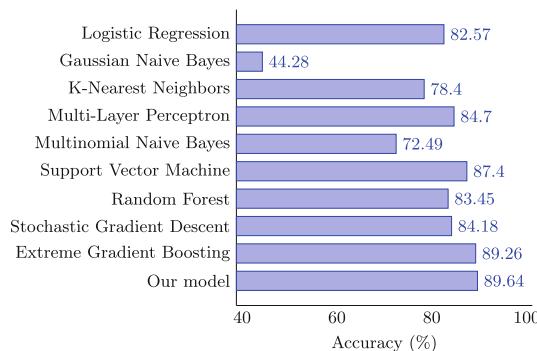


Fig. 5. The accuracy of machine learning models with TF-IDF for clinical text classification

4.4 Experimental Results of Training SVM on Top of VGG16 and XGBoost with TF-IDF

In our study, we explored a multi-modal approach for lung disease classification by combining chest X-ray images and clinical text data. We proposed a multi-modal model by training an SVM on top of the two models, VGG16 and XGBoost with TF-IDF, because these models achieved the highest accuracy in their respective uni-modal classification tasks. Experimental results showed that the accuracy of the VGG16 model for chest X-ray classification reached 62.66%, while the XGBoost model for clinical text classification achieved 89.26%. When combining both methods, the multi-modal model achieved the highest accuracy of 89.64%. These results indicate that integrating information from both data sources (chest X-ray images and clinical text) significantly improves classification effectiveness compared to using uni-modal models. Additionally, to evaluate the effectiveness of the proposed model, we conducted a series of experiments on various fine-tuned deep learning models combined with multiple machine learning models and feature extraction methods. This approach allowed us to compare and select the optimal model for the task of lung disease classification from chest X-ray images and clinical text.

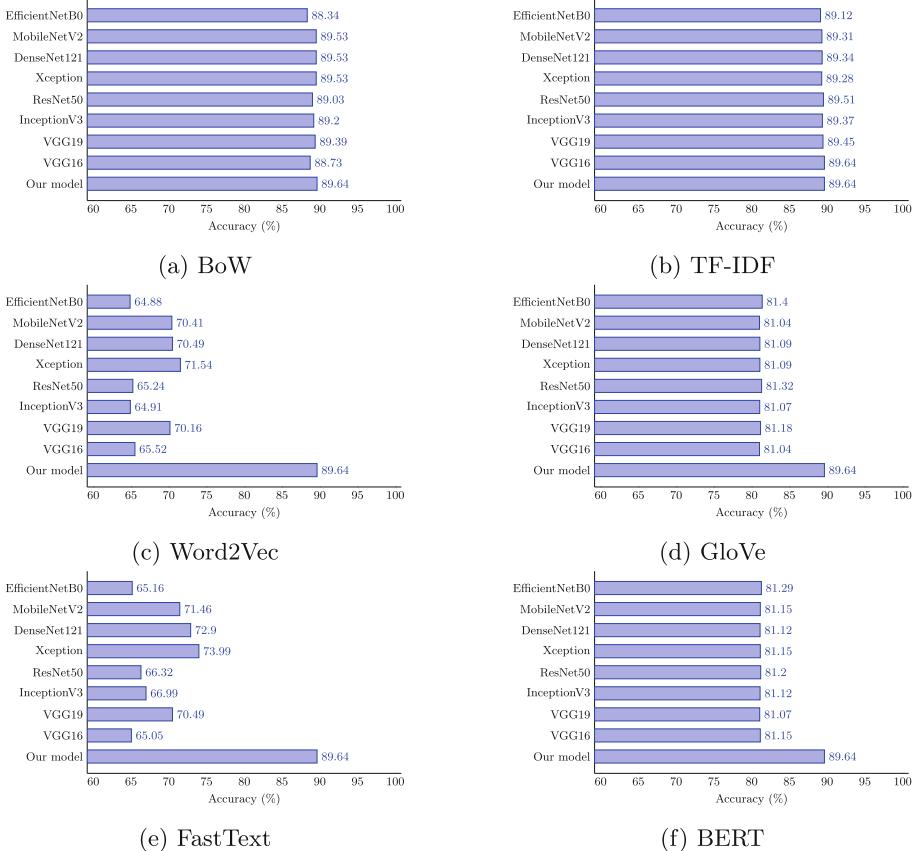


Fig. 6. Compare the accuracy of deep learning models combined with XGBoost according to each feature extraction method

Figure 6 illustrates the comparison of the accuracy of multi-modal models by combining fine-tuned deep learning models for image classification with the XGBoost text classification model using each feature extraction method. Based on the analysis, we observe that the combination of deep learning models with the XGBoost model yields notable results in the task of lung disease classification. Deep learning models such as VGG16, VGG19, InceptionV3, ResNet50, Xception, DenseNet121, MobileNetV2, and EfficientNetB0 all exhibit high accuracy when using the TF-IDF feature extraction method, with most models achieving an accuracy above 89%. Specifically, the VGG19 model achieved an accuracy of 89.45%, InceptionV3 reached 89.37%, and ResNet50 achieved 89.51%, demonstrating their stability and effectiveness when combined with TF-IDF. Most notably, our model achieved the highest accuracy of 89.64% across all feature extraction methods, including BoW, TF-IDF, Word2Vec, GloVe, FastText, and BERT. This highlights the superiority of our model in handling complex

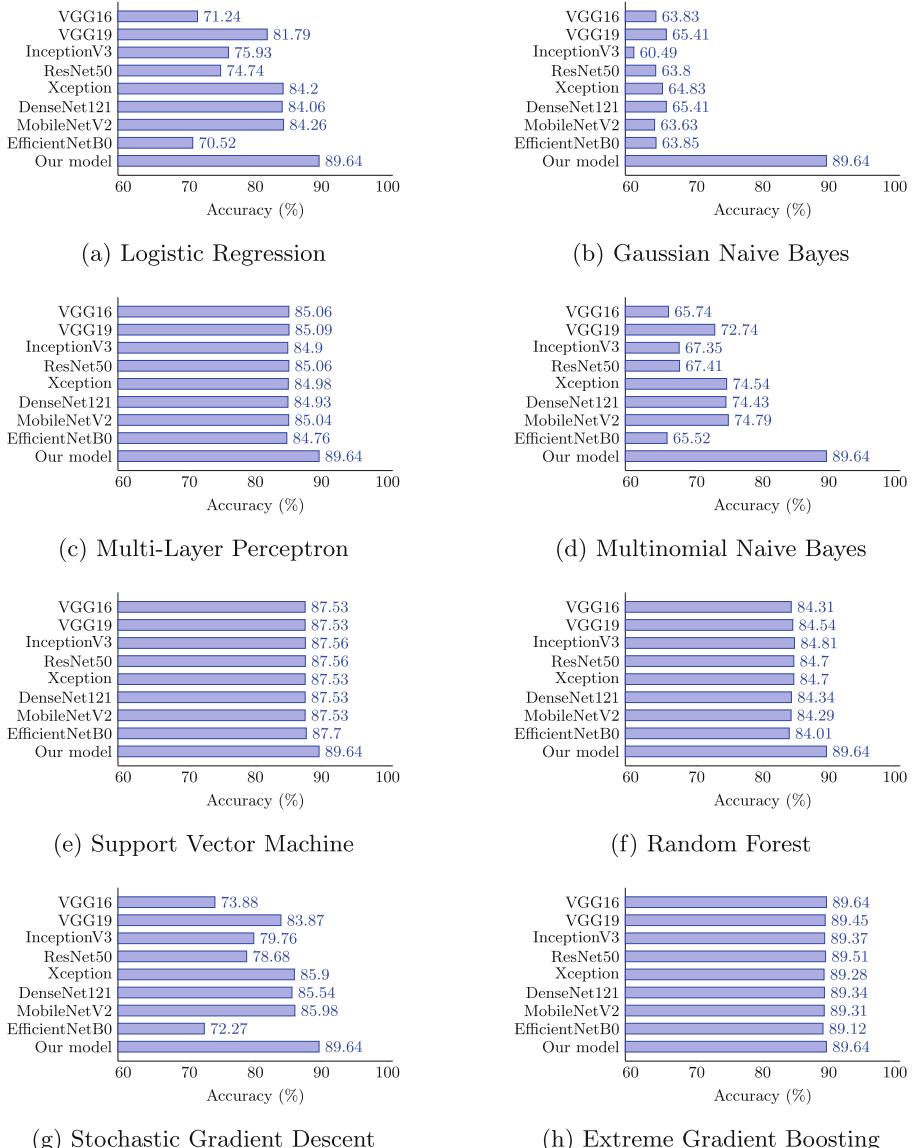


Fig. 7. Comparison of accuracy results for the multi-modal model combining deep learning models with machine learning models using TF-IDF feature extraction

clinical text data. This confirms that the TF-IDF feature extraction method is a robust and effective choice for lung disease classification from clinical text when combined with deep learning models and XGBoost.

In analyzing and comparing the accuracy results of the multi-modal model combining deep learning models with machine learning models using TF-IDF fea-

ture extraction for lung disease classification, several insights emerge. Figure 7 presents a comparison of accuracy results for the multi-modal model combining deep learning models with machine learning models using TF-IDF feature extraction for lung disease classification.

Our proposed multi-modal model, training SVM on top of VGG16 and XGBoost with TF-IDF, consistently achieves the highest accuracy of 89.64% across all classifiers. Notably, while individual deep learning models such as VGG16 and others perform well, their integration into a multi-modal framework with XGBoost significantly enhances classification accuracy, surpassing all unimodal approaches. This comprehensive approach underscores the utility of combining deep learning with traditional machine learning techniques for enhanced medical data classification tasks.

5 Conclusions and Future Work

In this study, we constructed a new dataset of chest X-ray images and clinical information extracted from electronic medical records at General Hospital of An Giang area province. We also proposed training an SVM model on top of VGG16 and an XGBoost model with TF-IDF for lung disease classification. The experimental results demonstrated the effectiveness of the multi-modal approach combining chest X-ray images and clinical text. Specifically, the fine-tuned VGG16 model achieved an accuracy of 62.66% for image classification, while the XGBoost model with TF-IDF achieved an accuracy of 89.26% for clinical text classification. When combining these two methods in a multi-modal model, the accuracy reached 89.64%, higher than each uni-modal model. These results indicate that our approach provides a comprehensive understanding of the patient's condition, enabling quicker and more accurate diagnostic decisions. Additionally, the multi-modal model improves classification performance compared to using only uni-modal models.

In the future, research can expand in several ways: enhancing data by expanding clinical and chest X-ray datasets, optimizing the model, and integrating additional medical data sources. Furthermore, we will also conduct additional experiments with other fusion methods, like early fusion of image and text features, and compare them with late fusion of classification results from the proposed model. These efforts aim to improve machine learning in medical diagnosis, leading to better healthcare and more accurate patient decisions.

Acknowledgment. This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number C2024-16-02.

References

1. Afshar, P., Heidarian, S., Naderkhani, F., Oikonomou, A., Plataniotis, K.N., Mohammadi, A.: Covid-caps: a capsule network-based framework for identification of covid-19 cases from x-ray images. *Pattern Recogn. Lett.* **138**, 638–643 (2020)
2. Breiman, L.: Random forests. *Mach. Learn.* **45**, 5–32 (2001)
3. Breiman, L.: Classification and Regression Trees. Routledge, Milton Park (2017)
4. Buntoro, G.A., Wibawa, A.D., Purnomo, M.H.: Text mining in healthcare for disease classification using machine learning algorithm. In: 2021 International Electronics Symposium (IES), pp. 97–101. IEEE (2021)
5. Cover, T., Hart, P.: Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **13**(1), 21–27 (1967)
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. Do, T.N., Le, V.T., Doan, T.H.: SVM on top of deep networks for covid-19 detection from chest x-ray images. In: JICCE, pp. 219–225 (2022)
8. Hayat, N., Geras, K.J., Shamout, F.E.: Medfuse: Multi-modal fusion with clinical time-series data and chest x-ray images. In: Machine Learning for Healthcare Conference, pp. 479–503. PMLR (2022)
9. Huang, S.C., Pareek, A., Seyyedi, S., Banerjee, I., Lungren, M.P.: Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. *NPJ Digit. Med.* **3**(1), 136 (2020)
10. Huang, S.C., Pareek, A., Zamanian, R., Banerjee, I., Lungren, M.P.: Multimodal fusion with deep neural networks for leveraging CT imaging and electronic health record: a case-study in pulmonary embolism detection. *Sci. Rep.* **10**(1), 22147 (2020)
11. Huynh, P.H., Tran, T.N., et al.: Enhancing covid-19 prediction using transfer learning from chest x-ray images. In: 2021 8th NAFOSTED Conference on Information and Computer Science (NICS), pp. 398–403. IEEE (2021)
12. Khanday, A.M.U.D., Rabani, S.T., Khan, Q.R., Rouf, N., Mohi Ud Din, M.: Machine learning based approaches for detecting covid-19 using clinical text data. *Int. J. Inf. Technol.* **12**, 731–739 (2020)
13. Lee, G., Kang, B., Nho, K., Sohn, K.A., Kim, D.: Mildint: deep learning-based multimodal longitudinal data integration framework. *Front. Genet.* **10**, 617 (2019)
14. Nabilah’Izzaturrahmah, A., Nhita, F., Kurniawan, I.: Implementation of support vector machine on text-based gerd detection by using drug review content. In: 2021 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS), pp. 1–6. IEEE (2021)
15. Organization, W.H.: Global asthma report 2022 (2022). <http://globalasthmareport.org/>, Accessed 2024
16. Organization, W.H.: Pneumonia in children (2022). <https://www.who.int/news-room/fact-sheets/detail/pneumonia>, Accessed 11 Jul 2024
17. Regmi, S., Subedi, A., Bagci, U., Jha, D.: Vision transformer for efficient chest x-ray and gastrointestinal image classification. arXiv preprint [arXiv:2304.11529](https://arxiv.org/abs/2304.11529) (2023)
18. Sebastiani, F.: Machine learning in automated text categorization. *ACM Comput. Surv. (CSUR)* **34**(1), 1–47 (2002)
19. Vapnik, V.N.: An overview of statistical learning theory. *IEEE Trans. Neural Networks* **10**(5), 988–999 (1999)

20. Vapnik, V.: The nature of statistical learning. Theory (1995)
21. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chestx-ray8: hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2097–2106 (2017)
22. Xu, T., Zhang, H., Huang, X., Zhang, S., Metaxas, D.N.: Multimodal deep learning for cervical dysplasia diagnosis. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016, Part II. LNCS, vol. 9901, pp. 115–123. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_14
23. Yadav, P., Menon, N., Ravi, V., Vishvanathan, S.: Lung-gans: unsupervised representation learning for lung disease classification using chest CT and x-ray images. IEEE Trans. Eng. Manage. **70**(8), 2774–2786 (2021)



Toward Supporting Breast Cancer Diagnosis Based on Captioning Mammogram and Ultrasound Images

Huong Hoang Luong^{1,2}, Hai Thanh Nguyen¹, and Nguyen Thai-Nghe^{1(✉)}

¹ Can Tho University, Can Tho, Vietnam

² FPT University, Can Tho, Vietnam

ntnghe@cit.ctu.edu.vn

Abstract. Breast cancer is one of the most dangerous cancers with a high mortality rate, especially for women. Early diagnosis and detection are expected to make the treatment process highly effective. To support doctors and minimize clinical errors, this article proposes a BCICG (Breast cancer image caption generator) approach, which creates breast cancer captioning based on medical images by combining Convolutional Neural Network (CNN) and Transformer architectures. To demonstrate the effectiveness, the proposed approach is then trained on different types of datasets. The results are positive, with the highest BLEU from BLEU-1 to BLEU-4 and Rouge-L measurements being 0.574, 0.521, 0.487, 0.466, and 0.664, respectively. Besides, this study also built a dataset of breast ultrasound images collected in Ca Mau Provincial General Hospital, Vietnam, with descriptions of disease signs based on those images and the doctor's diagnosis results.

Keywords: Breast cancer · Convolutional Neural Network · Transformer · Captioning

1 Introduction

Breast cancer is the most dangerous cancer in women, with 2.3 million cases and 685,000 deaths worldwide in GLOBALCAN 2020 [24]. The incidence has increased significantly from 37.4/100,000 to 47.8/100,000 between 2002–2020 [3, 24]. The 5-year survival rate is over 80% in high-income countries but only 40–66% in low-income countries due to late-stage diagnosis and lack of quality care [6, 7]. Early detection is crucial for effective treatment, even with a potential cure rate of 100%. Regular screening methods include self-examination, clinical assessment, X-ray, and ultrasound. X-ray is effective, and ultrasound aids in dense breast tissue cases [2]. However, in low and middle-income countries, limited access to equipment and skilled specialists hinders effective screening. Therefore, supporting doctors in diagnosing breast cancer through images is essential to reduce errors.

However, image captioning is still a difficult task. First, natural image captioning as in [1, 17] will be utterly different from images in a domain-specific field, especially medical images as in [22]. Next, data in some domain-specific areas is scarce, especially in breast cancer. To create captions for medical images, the researcher must have specific knowledge in that field, an understanding of different types of images for different stages of the disease, and a knowledge of how to describe them or make a diagnosis. Additionally, data shortages are a severe problem. For classification problems described in [20, 21], researchers may only need labeled image data to perform training. To make the caption for medical images, the data must combine images and text to be eligible to train the model. As a result the need for this type of data has caused many difficulties for researchers.

From the above issues, we use the INbreast dataset as described in [11]. This is a mammogram dataset provided with disease descriptions in Portuguese. Next to that we converted the language from Portuguese to Vietnamese and then standardized general doctors' descriptions at Can Tho City General Hospital, Vietnam. Next, [16] collected data on ultrasound images of the breasts of patients at Ca Mau Provincial General Hospital, which we called HiSBreast. This data includes 972 breast ultrasound images (left or right breast ultrasound images or both) of many different patients. Accompanying each ultrasound image are the doctor's diagnosis results, briefly described in Vietnamese. Besides, this article also proposes an approach by combining the CNN model and Transformer architecture to diagnose disease based on different types of breast cancer images. The results obtained are very positive, with the highest BLEU from BLEU-1 to BLEU-4 and Rouge-L measurements being 0.574, 0.521, 0.487, 0.466, and 0.664 respectively.

From the issues presented above, we can see that breast cancer is a dangerous disease, diagnosing and describing the disease based on medical images requires solid professional knowledge. However, research in the field of descriptive biology for breast cancer images is still scarce due to many factors, especially a lack of data. That is why this study was proposed. The main contributions to this article include:

- Because of the problem of the lack of datasets, this study has collected and built a dataset of ultrasound images of breast cancer and disease diagnoses named HiSBreast. The collected dataset includes 972 samples. Each sample will include an ultrasound image, a description of disease signs based on that image, and the doctor's diagnostic results. The image can be an ultrasound image of one breast or both, with a resolution of 320×240 .
- For linguistic synchronization, the study converted the Portuguese language in the INbreast dataset to Vietnamese and standardized the disease descriptions to match Vietnamese and the corresponding image.
- To the author's knowledge, the number of publications on the problem of breast cancer image captioning is still limited, for example in the study of [23]. Therefore, this study has proposed a model that achieves relatively better accuracy than the above study and is presented in Sect. 5.5.

- This research proposed an approach named BCICG (Breast cancer image caption generator) to generate disease captioning based on different types of images of breast cancer by using CNN and Transformer.
- Propose and apply different data preprocessing methods based on the existing datasets to obtain the best possible results.

This study is divided into six sections. The first one is an introduction. Section 2 introduces major related works. Section 3 briefly introduces available datasets, how a new dataset is collected, and the organizational structure of that new data. Then, we present how to pre-process the data and generate a model based on encoder-decoder by combining CNN and Transformer models in Sect. 4. Section 5 presents and analyzes the obtained results. We conclude the study and discuss future work in Sect. 6.

2 Related Work

Following to [23], a novel approach named FCN-MLC-LSTM was proposed for generating captions for ultrasound images of breast cancer. This method integrated a Fully Connected Network (FCN) for encoding and a Long Short-Term Memory Network (LSTM) for decoding, augmented by Multi-label Classification (MLC) for semantic processing. The model, trained on translated reports from the INbreast dataset, achieved significant performance with BLEU scores of 60.7 (BLEU-1) to 23.7 (BLEU-4) and a CIDEr score of 61.7, demonstrating robustness in generating accurate natural language descriptions.

In [8], researchers focused on digital endoscope images in gastrointestinal examinations, employing MobileNetV2 and DenseNet-121 models with Class Activation Maps (CAM) for abnormality detection and medical report generation. Their approach achieved high accuracy levels ranging from 66% to 99%, varying with the specific disease type within the dataset. The study conducted by [5] introduced a model consisting of a visual feature extractor, Transformer-based encoder, and decoder normalized with Memory-driven Conditional Layer Normalization (MCLN). Evaluated on IU X-ray and MIMIC-CXR datasets, this architecture surpassed previous benchmarks, highlighting its effectiveness in encoding and decoding chest X-ray images.

According to [10], researchers proposed a multi-attention hierarchical model designed to map image and textual information into coherent sentence topics, particularly effective for generating detailed captions from chest X-ray datasets. Despite challenges in dataset variability, the method demonstrated efficacy in generating comprehensive textual descriptions. The research presented in [12] introduced a multi-task learning framework incorporating co-attention mechanisms and hierarchical LSTM models for tag prediction and paragraph generation from medical imaging datasets like IU X-ray and PEIR Gross. Their approach outperformed existing methods, achieving superior results across various evaluation metrics.

In [13], the Hybrid Retrieval-Generation Reinforced Agent (HRGR-Agent) was proposed, employing a hierarchical decision-making approach for sentence

generation in medical reports. Trained and validated on CX-CHR and IU X-Ray datasets using reinforcement learning techniques, the model effectively balanced generating accurate and diverse sentences covering diverse medical report contents. Lastly, [14] introduced the Knowledge-driven Encode, Retrieve, Paraphrase (KERP) method, leveraging a Graph Transformer (GTR) to convert image features into structured anomaly graphs. This approach facilitated template retrieval and paraphrasing based on anomaly detection, achieving structured and robust medical report generation with explainable attention mechanisms.

3 Datasets

3.1 INbreast Dataset

The INbreast dataset [11] was collected at Centro Hospitalar de S. João [CHSJ], Breast Centre, Porto, with approval from the Portuguese National Committee of Data Protection and the Hospital’s Ethics Committee. It includes data from 115 cases, comprising 410 full-field digital mammograms. Among these, 90 cases involved female patients with both breasts affected, while 25 cases had one breast removed. The dataset provides highly reliable and accurate information due to its basis in actual cases. Each image is stored in DICOM format, accompanied by manually annotated regions of interest (ROI) and medical reports describing the diseases observed in the images. In addition, the dataset was translated and reviewed by Dr. Quach Quoc Duong, deputy head of the thoracic surgery department at Can Tho City General Hospital, who helps the translation process preserve the semantic integrity of the content.

3.2 HiSBreast Dataset

The HiSBreast dataset [16] was collected using HiS software at Ca Mau Provincial General Hospital, Vietnam, by VNPT Group. It contains breast ultrasound images from inpatients referred for breast ultrasound between 2018 and 2022. The data is stored in JSON file format, with each file covering three months of patient data. The JSON files, named in the format “yearmonth-month-month.json” (e.g., 202210-11-12.json), contain arrays of medical examination information, including:

- **SO_PHIEU_CDHA**: Clinical examination order form code.
- **IMAGE**: Ultrasound images in Base64 format.
- **MO_TA**: Description of the patient’s disease.
- **TEN_CDHA**: Name of the clinical indication type.
- **CHANDOANBENH**: Diagnosis results, including disease codes as per the Vietnamese Ministry of Health list.

Figure 1 and 2 referenced in the original text illustrate a patient’s information structure and an ultrasound image decoded from Base64 format.

```
{
    "SO_PHIEU_CDHA": "CD_45813/2022.1.5_1",
    "HINHANH": "data:image/png;base64,1VBORw0KGgoAAAANSUhEUgAAoAAAAAHgCAYAAAA10dzkAAAAAXNSR0IArs4c6QAAI
    "MO_TA": "U VÚ PHẢI BI RAD 6.<br />\n- HẠCH NÁCH PHẢI.",
    "TEN_CDHA": "Siêu âm tuyến vú hai bên",
    "CHANDOANBENH": "C50-U ác của vú[K vú phả T4N2M0] ",
    "CHANDOAN": " "
},
```

Fig. 1. An example of a structure that describes a patient's information

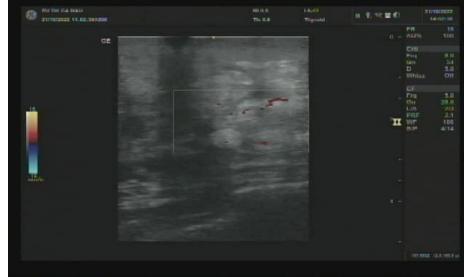


Fig. 2. Ultrasound image of the patient's breast

4 Proposed Approach

This section details the proposed architecture for the model, including techniques for data preprocessing and the components of the captioning generation architecture. Moreover, the entire training and caption generation process is termed BCICG (Breast Cancer Image Caption Generator). Figure 3 illustrates the architecture, which includes:

- (1) Feature Extractor: Utilizes a CNN [18], specifically the ResNet18 [9] model pre-trained on the ImageNet dataset, to extract features from input images. The final convolutional layer creates a feature representation vector for the image.
- (2) Normalizes the input image size and processes caption text by removing punctuation and spaces, separating Vietnamese words, and converting words into vectors.
- (3) Decoder: Employs a Transformer [25] to decode the representation vector from the encoder into a text string. The Transformer predicts each word in the caption based on the relationships between words learned through self-attention.

4.1 Preprocessing

This subsection outlines the various preprocessing techniques applied to the data before it is fed into the model, ensuring consistent and efficient processing for image caption generation.

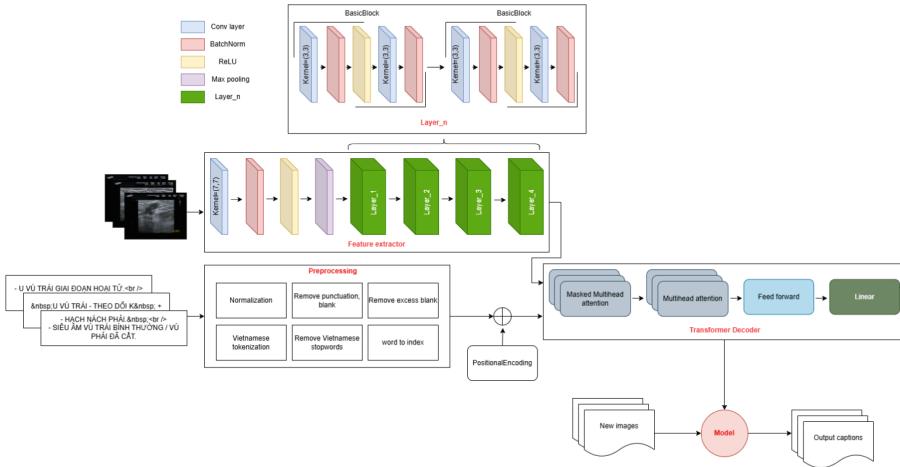


Fig. 3. The proposed BCICG architecture

Text Normalization. Medical reports often have inconsistent descriptions for the same condition due to different doctors and patients. Additionally, some descriptions include information not visible in the images, such as biopsy results. Removing such information is necessary to avoid noise in the caption generation process.

- Origin: Một nốt với biên dạng mô đệm, nằm ở vú phải có đường kính 2 cm, với các dấu hiệu nghi ngờ ác tính. Sinh thiết dưới hướng dẫn siêu âm 4 mảnh để nghiên cứu giải phẫu bệnh học. Hạch nách (-). Những thay đổi trong chẩn đoán mô học của bệnh ác tính - Bi-Rads - 6.
- Pre-processed: Một nốt với biên dạng mô đệm, nằm ở vú phải có đường kính 2 cm, sinh thiết nghi ngờ ác tính - Bi-Rads - 6

Removing Punctuation and Blank Spaces. Punctuation and excessive white spaces in the original captions can introduce noise during training. For example, symbols like “ - Bi-Rads - 6” should be removed to improve the model’s performance.

- Origin: Một nốt với biên dạng mô đệm, nằm ở vú phải có đường kính 2 cm, sinh thiết nghi ngờ ác tính BiRads 6.
- Pre-processed: Một nốt với biên dạng mô đệm nằm ở vú phải có đường kính 2 cm sinh thiết nghi ngờ ác tính BiRads 6

Vietnamese Tokenization. Properly determining word boundaries is crucial in Vietnamese due to its many compound words. Tokenization helps the model understand text semantics better and improves processing efficiency by removing stopwords, which do not carry significant meaning but add noise and reduce

model accuracy.

- Origin: một nốt với biến dạng mô đệm nằm ở vú phải có đường kính 2 cm sinh thiết nghỉ ngò ác tính birads 6.
- Pre-processed: một nốt với biến_dạng mô đệm nằm ở_vú phải có đường_kính 2 cm sinh_thiết_nghi_ngò_ác_tính birads 6.

Word to Index (W2I). This technique converts text into a set of keywords or phrases, simplifying natural language processing by reducing the data size and improving algorithm efficiency. Words not in the dictionary are mapped to a special “UNK” notation. In Eq. 1, let D be a dictionary with N words, D_i be the i^{th} word in D , and UNK (Unknown) be a special notation used when the word is not in the dictionary. The formula to map a word to an index is:

$$W2I(w) = \begin{cases} i & if w \in D \\ UNK & if w \notin D \end{cases} \quad (1)$$

Positional Encoding. Since the Transformer model uses self-attention and does not directly encode token positions, positional encoding [4] is added to help the model understand the context of tokens within a sequence. This technique involves adding sinusoidal patterns to the input embeddings based on their positions.

4.2 Transformer Decoder

The Transformer Decoder is important for transforming input strings into output strings within the Transformer model. Its main components in Algorithm 1:

Algorithm 1. Transformer Decoder Components

1. **Positional Encoding:** Adds positional information to input words.
 2. **Embedding:** Converts input words into vectors.
 3. **Transformer Decoder Layer:**
 - (a) **Self-attention:** Focuses on important parts of the input and output.
 - (b) **Multi-head attention:** Gathers information from multiple perspectives.
 - (c) **Feed-forward network:** Learns complex transformations on vectors.
 4. **Transformer Decoder:** Comprises multiple Transformer Decoder Layers.
 5. **Linear Layer:** Produces the final output.
-

In Eq. 2, the process involves using matrices Q , K , and V of the input data, divided into h parts for attention calculation, combined via a weight matrix W^O in Eq. 3. The formulas for the i -th head and multi-head attention are:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (3)$$

According to Eq. 4, Feed-forwards in the decoder are fully connected layers applied to each token:

$$\text{FF}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \quad (4)$$

In addition to the above components, a Transformer decoder has two critical components: the mask generator and the linear layer. The mask generator prevents previewing words later in the output string and handles padding words. The Linear layer converts the representation vector into probabilities of words in the dictionary. It is responsible for generating the final output sequence.

5 Experiments and Evaluation

In this section, all the models used are fine-tuned to obtain the best results and used to compare the results with each other. Experimental data is divided into two parts, training, and testing, with a ratio of 90%-10%. All experiments used the same set of model-specific hyperparameters except for the baseline model which was run by default and without fine-tuning. Specifically, the hyperparameters used on the MobileNet-LSTM model are as follows: Encoder model with Image feature includes $\text{dropout} = 0.3$, $\text{dense} = 256$, $\text{activation} = \text{ReLU}$, and sequence feature includes $\text{dropout} = 0.2$, $\text{LSTM} = 256$, and $\text{output_dim} = 256$. Decoder model includes $\text{dense} = 521$, $\text{activation} = \text{softmax}$, $\text{epoch} = 100$ and $\text{batch_size} = 32$. Meanwhile, the hyperparameters used on the proposed model are as follows: $\text{batch_size} = 32$, $\text{num_of_heads} = 64$, $\text{num_of_decoder_layers} = 10$, $\text{embedding_size} = 512$, $\text{learning_rate} = 0.0001$.

Before began evaluating the experiments, we determined that the experiments did not overlap in terms of information and results. In Experiment 1, text generation was created based on the INBreast dataset. Next, Experiments 2 and 3 were run into the HiSBreast dataset but in different categories including description and diagnosis. These experiments help us define the performances of the proposed model in different environments.

5.1 Metric

The effectiveness of the model and the quality of automatic description generation are evaluated using BLEU [19] and ROUGE-L [15] metrics. Moreover, both datasets including INBreast [11] and HiSBreast [16] were divided in an 8-1-1 ratio with 80% for training and 10% for validation and testing in each experiment. In Formula 5, BLEU evaluates the quality of automatic translation models by focusing on common words and n-grams between the predicted model and the reference description. BLEU-1 measures the accuracy of single words (unigrams),

BLEU-2 measures consecutive pairs of words (bigrams), BLEU-3 measures consecutive word pairs (trigrams), and BLEU-4 measures 4-grams. The BLEU-1 formula is:

$$BLEU - 1 = \frac{1}{n} \sum_{i=1}^n \frac{C_i}{R_i} \quad (5)$$

$$BLEU - n = \frac{BP(ngram)}{BP(ngram_{ref})} \quad (6)$$

ROUGE-L in Eq. 7 is another metric used to evaluate translation quality, calculated based on common words and n-grams between the predicted model and the reference description. The formula for ROUGE-L is:

$$ROUGE - L = \frac{1}{n} \sum_{i=1}^n \frac{R_L(i)}{R_{L_{ref}}(i)} \quad (7)$$

5.2 Experiment 1: Generate the Breast Cancer Medical Reports

Table 1. Experimental results on the INbreast dataset

Approaches	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU	ROUGE-L
MobileNet-LSTM (Baseline)	0.229	0.241	0.196	0.161	0.161	0.556
MobileNet-LSTM	0.301	0.245	0.211	0.178	0.178	0.579
Ours (Baseline)	0.413	0.353	0.310	0.281	0.281	0.494
Ours	0.574	0.521	0.487	0.466	0.466	0.664

In this section, experiments are conducted on the INbreast dataset with disease descriptions in Vietnamese. Data preprocessing includes normalizing data, removing punctuation and excess blanks, performing Vietnamese tokenization, and removing stopwords and indexed words. In Table 1, the models used for training and generating captions are ResNet18-Transformer and MobileNet-LSTM.

To clarify the experiment results, Fig. 4 compares the input image, ground truth, preprocessed ground truth, and generated captions. Matching content in the preprocessing and generated captions columns are highlighted in the same color, even if positions differ. From Fig. 4, the experiment observes that the generated captions can match the ground truth significantly, as seen in Sample 1 and Sample 2. However, in Sample 3, the prediction result is almost entirely incorrect.

Input Image	Ground truth	Pre-processing	Generated captions
	Không có hình ảnh nốt nào gọi ý ác tính, với hóa mô tuyên vú đáng ngờ hoặc nhưng thay đổi dáng kê khác được xác định ở cả hai bên. Kết quả hình ảnh lành tính BiRads 2.	không có hình ảnh nốt nào gọi ý ác tính với hóa mô tuyên vú đáng ngờ hoặc nhưng thay đổi dáng kê khác được xác định ở cả hai bên kết quả hình ảnh lành tính birads 2	không có hình ảnh nốt nào gọi ý ác tính với hóa vi mô đáng ngờ hoặc nhưng thay đổi dáng kê khác được xác định ở cả hai bên kết quả hình ảnh lành tính birads 2
	nhữ ảnh tuyên vú ghi nhận ít nhu mô vú, đa phần là mô mờ, sự thoát triễn mờ tuyên vú ở cả hai bên. Không có hình ảnh nốt riêng lẻ nào gọi ý bệnh ác tính, với hóa vi mô đáng ngờ hoặc những thay đổi dáng kê khác ở cả hai bên. Kết quả hình ảnh lành tính BiRads 2.	nhữ ảnh tuyên vú ghi nhận ít nhu mô vú đa phần là mờ sự thoát triễn mờ tuyên vú ở cả hai bên không có hình ảnh nốt riêng lẻ nào gọi ý bệnh ác tính với hóa vi mô đáng ng ngờ hoặc nhưng thay đổi đáng kê khác ở cả hai bên kết quá hình_ảnh_lành_tinh birads 2	nhữ ảnh tuyên vú ghi nhận ít nhu mô vú trái không có hình ảnh nốt mà gọi ý ác tính với hóa vi mô đáng ngờ hoặc những thay đổi dáng kê khác ở cả hai bên kết quá hình_ảnh_lành_tinh birads 2
	không có hình ảnh nốt nào gọi ý ác tính, với hóa vi mô đáng ngờ hoặc những thay đổi dáng kê khác được xác định ở cả hai bên. Kết quả hình ảnh lành tính BiRads 2.	không có hình ảnh nốt nào gọi ý ác tính với hóa vi mô đáng ng hoặc nhưng thay đổi đáng kê khác được xác định ở cả hai bên kết quá hình_ảnh_lành_tinh birads 2	một nốt nằm ở vú trái có đường kính 25 mm với những dấu hiệu đáng ng là ác tính sinh thiết nghi ng ác tính birads 5

Fig. 4. Compare ground-truth, pre-processed ground-truth, and generated captions

5.3 Experiment 2: Generate Breast Cancer Disease Description

In Table 2, the HisBreast dataset is used. The training data includes patients' breast ultrasound images and corresponding doctors' descriptions. Before training, the text data is preprocessed by removing excess blanks, and stopwords, performing Vietnamese tokenization and removing HTML tags/entities like `
` and ` `. The evaluation models used are the same as in the previous experiment.

Table 2. Experimental results on the HisBreast dataset for Descriptions

Approaches	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU	ROUGE-L
MobileNet-LSTM (Baseline)	0.260	0.194	0.124	0.093	0.093	0.488
MobileNet-LSTM	0.338	0.259	0.168	0.122	0.122	0.488
Ours (Baseline)	0.311	0.243	0.164	0.106	0.106	0.471
Ours	0.483	0.411	0.291	0.234	0.234	0.587

The results from Table 2 and Fig. 5 show that the results in this experiment are lower than those from experiment 1. This can be attributed to factors such as the quality of the dataset, noise in ultrasound images, and inconsistency in describing the same disease sign. While the generated descriptions are relatively accurate, the order of descriptions can vary, as seen in sample 3 of Figure 5. Nevertheless, the proposed model still provides relatively positive results.

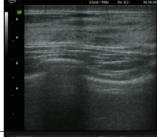
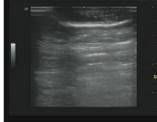
Input Image	Ground truth	Pre-processing	Generated captions
	BUỐU VÚ TRÁI BIRADS 4.	búrou vú trai birads 4	búrou vú trai birads 4
	SIÊU ÂM TUYẾN VÚ TRÁI HIỆN CHUA GHI NHẬN BẤT THƯỜNG / VÚ PHAI ĐÃ PHẪU THUẬT.	siêu âm tuyến vú trái hiện chưa ghi nhận bất thường vú phai đã phẫu thuật	siêu âm tuyến vú phải hiện chưa ghi nhận bất thường hiện không thấy tôn thường dưới vết mổ cũ birads 6 vú trái
	- HẠCH NÁCH TRÁI KHÔNG ĐIỂN HÌNH / VÚ TRÁI ĐÃ PHẪU THUẬT.&br/> - SIÊU ÂM TUYẾN VÚ PHAI HIỆN CHUA GHI NHẬN BẤT THƯỜNG.	hạch nách trái không điểm hình vú trái đã phẫu thuật siêu âm tuyến vú phải hiện chưa ghi nhận bất thường	siêu âm tuyến vú phải hiện chưa ghi nhận bất thường vú trái đã phẫu thuật

Fig. 5. Compare ground-truth, pre-processed ground-truth, and generated captions

Table 3. Experimental results on the HisBreast dataset for Diagnosis

Approaches	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU	ROUGE-L
MobileNet-LSTM (Baseline)	0.085	0.044	0.002	0.001	0.001	0.452
MobileNet-LSTM	0.118	0.081	0.014	0.008	0.008	0.518
Ours (Baseline)	0.116	0.087	0.065	0.010	0.010	0.368
Ours	0.426	0.354	0.272	0.211	0.211	0.599

5.4 Experiment 3: Generate Breast Cancer Disease Diagnosis

In this experiment, the HisBreast dataset is used. Table 3 shows the comparison results between models. Moreover, the evaluation models used are the same as in the previous experiment. To clarify the results of this experiment, Fig. 6 compares the input image, ground truth, preprocessed ground truth, and generated diagnosis. In the HisBreast dataset, breast ultrasound images are inconsistent; some show one breast while others show both, often due to prior mastectomies. Additionally, inconsistencies in patient diagnoses contribute to lower accuracy in diagnosis generation.

5.5 Comparison with State-of-the-Art

Because of the differences in data structures, evaluation matrices, and experimental languages. Table 4 just compares the proposed model with a similar study using an FCN-MLC-LSTM model on the INbreast dataset without other methods in the related works section. While the proposed method yields lower results for BLEU-1, it outperforms the other model in BLEU-2, BLEU-3, and BLEU-4, showing the effectiveness of combining CNN and Transformer. Additionally, the proposed model is evaluated using the ROUGE-L measure for accuracy, whereas

Input Image	Ground truth	Pre-processing	Generated captions
	C50-U ác của vú[trái], ct2NxM0]	u_ ác của vú trái ct2nxm0	u_ ác của vú trái
	L02-Áp xe da, nhọt, nhọt cum[TD: áp xe vú trái]	áp xe da nhọt nhọt cum td áp xe vú trái	áp xe da nhọt nhọt cum vú trái
	C50-U ác của vú[K vú trái]	u_ ác của vú k vú trái	u_ ác của vú

Fig. 6. Compare ground-truth, pre-processed ground-truth and generated captions

the other study used the CIDEr measure to focus on diversity and information in the generated captions.

Table 4. Comapre with the previous study.

Reference	Architecture	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU	ROUGE-L	CIDEr
Ref. [23]	FCN-MLC-LSTM	0.607	0.412	0.326	0.237	0.237	-	0.617
Ours	ResNet18-Transformer	0.574	0.521	0.487	0.466	0.466	0.664	-

5.6 Evaluation

The research conducted experiments on two datasets: standardized mammograms and non-standardized ultrasound images collected directly from a hospital. The goal was to evaluate the generalization of the proposed model. In experiment Sect. 5.2, the consistent and expert-standardized descriptions of mammograms led to higher results (Fig. 7). In experiments Sects. 5.3 and 5.4, the results were lower due to the lower quality and noise in the ultrasound images, inconsistent text descriptions and diagnoses, and the presence of non-observable information in the ultrasound dataset. Despite these challenges, the proposed model demonstrated generalizability in generating captions from both mammograms and ultrasound images, and it can generate various types of descriptions and diagnoses for breast cancer images.

5.7 Limitation and Future Works

The limitations of the proposed BCICG approach have several key areas for improvement. One critical issue is the potential for generating inaccurate or ambiguous captions, which should run more improvement to increase accuracy. Additionally, the current model lacks multilingual processing capabilities,

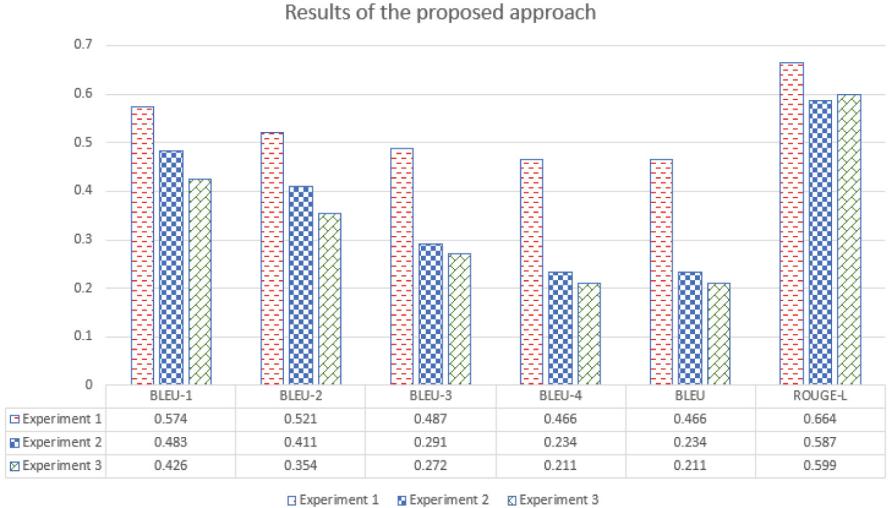


Fig. 7. Results of the proposed approach with various experiments

restricting it to a single language, which limits its adaptability for international applications. Future work should prioritize enhancing the model's ability to accurately capture the nuances of medical language, potentially through the integration of more robust natural language processing techniques, such as fine-tuning with domain-specific datasets and leveraging advanced pre-trained models. Furthermore, developing a multilingual captioning system that can seamlessly translate between Vietnamese and English would greatly enhance its utility in global healthcare contexts. Lastly, incorporating a feedback mechanism from medical professionals during the training process could help reduce errors and increase clinical reliability.

6 Conclusion

Breast cancer is a serious disease that can spread quickly and cause high mortality if not treated promptly. Advancements in treatment are improving survival rates and quality of life. Early detection through regular checkups, breast self-exams, and a healthy lifestyle is crucial. Besides, regular mammography or ultrasound exams help detect breast cancer. Furthermore, an AI model can assist by detecting subtle signs from images, aiding in accurate and timely diagnoses, and helping doctors choose the best treatment options. This study proposes an encoder-decoder model based on CNN and Transformer to generate disease captions from images. Evaluated on mammography and ultrasound datasets, the model shows high effectiveness. Future research will standardize datasets and apply advanced models to enhance accuracy and benefit the medical field.

Acknowledgements. Luong Hoang Huong was funded by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2023.TS.049. Thanks to Ca Mau Provincial General Hospital and Ms. Luong Thi Thu Huong - VNPT Ca Mau in Vietnam for agreeing to and exporting data on HiSBreak. In addition, we would like to thank Dr. Quach Quoc Duong - Can Tho City General Hospital in Vietnam, who assisted in checking and standardizing descriptions in the INbreast dataset.

Conflict of Interest Statement. Luong Hoang Huong has received research grants from the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2023.TS.049. Hai Thanh Nguyen declares that he has no conflict of interest. Nguyen Thai-Nghe declares that he has no conflict of interest.

Ethical Approval. This article does not contain any studies with human participants or animals performed by any of the authors.

Availability of Data, Code, and Material. INbreast dataset for this study is published on the repository link at (<https://www.kaggle.com/datasets/ramanathansp20/INbreast-dataset>) and the HiSBreast dataset is published on the repository link at (<https://doi.org/10.17632/5c723rpwz2.1>).

References

1. Beddiar, D., Oussalah, M., Seppänen, T.: Automatic captioning for medical imaging (MIC): a rapid review of literature. *Artif. Intell. Rev.* (2023)
2. Berg, W.A., et al.: Combined screening with ultrasound and mammography vs mammography alone in women at elevated risk of breast cancer. *JAMA* (2008)
3. Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R.L., Torre, L.A., Jemal, A.: Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: Cancer J. Clin.* **68**(6), 394–424 (2018). <https://doi.org/10.3322/caac.21492>
4. Chen, P.C., Tsai, H., Bhojanapalli, S., Chung, H.W., Chang, Y.W., Ferng, C.S.: A simple and effective positional encoding for transformers (2021)
5. Chen, Z., Song, Y., Chang, T.H., Wan, X.: Generating radiology reports via memory-driven transformer, pp. 1439–1449 (2020). <https://doi.org/10.18653/v1/2020.emnlp-main.112>. <https://aclanthology.org/2020.emnlp-main.112>
6. DeSantis, C.E., et al.: Breast cancer statistics, 2019. *CA: Cancer J. Clin.* **69**(6), 438–451 (2019). <https://doi.org/10.3322/caac.21583>
7. Jedy-Agba, E., et al.: Stage at diagnosis of breast cancer in Sub-Saharan Africa: a systematic review and meta-analysis. *Lancet Glob. Health* (2016)
8. Harzig, P., Einfalt, M., Lienhart, R.: Automatic disease detection and report generation for gastrointestinal tract examination, pp. 2573–2577 (2019). <https://doi.org/10.1145/3343031.3356066>
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015)
10. Huang, X., Yan, F., Xu, W., Li, M.: Multi-attention and incorporating background information model for chest X-ray image report generation. *IEEE Access* **7**, 154808–154817 (2019). <https://doi.org/10.1109/ACCESS.2019.2947134>

11. Moreira, I.C., et al.: Inbreast: toward a full-field digital mammographic database. *Acad. Radiol.* (2011)
12. Jing, B., Xie, P., Xing, E.: On the automatic generation of medical imaging reports, pp. 2577–2586 (2018). <https://doi.org/10.18653/v1/P18-1240>. <https://aclanthology.org/P18-1240>
13. Li, C.Y., Liang, X., Hu, Z., Xing, E.P.: Hybrid retrieval-generation reinforced agent for medical image report generation (2018)
14. Li, C.Y., Liang, X., Hu, Z., Xing, E.P.: Knowledge-driven encode, retrieve, paraphrase for medical image report generation (2019)
15. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries, pp. 74–81 (2004). <https://aclanthology.org/W04-1013>
16. Luong, H.H., Nguyen Thanh, H., Nguyen, T.N., Luong Thi Thu, H.: Hisbreast. Mendeley Data (2024)
17. Messina, P., et al.: A survey on deep learning and explainability for automatic report generation from medical images (2022)
18. O'Shea, K., Nash, R.: An introduction to convolutional neural networks (2015)
19. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation, pp. 311–318 (2002). <https://doi.org/10.3115/1073083.1073135>. <https://aclanthology.org/P02-1040>
20. Peng, L., Qiang, B., Wu, J.: A survey: image classification models based on convolutional neural networks, pp. 291–298 (2022). <https://doi.org/10.1109/ICCRD54409.2022.9730565>
21. Plested, J., Gedeon, T.: Deep transfer learning for image classification: a survey (2022)
22. Selivanov, A., Rogov, O.Y., Chesakov, D., Shelmanov, A., Fedulova, I., Dylov, D.V.: Medical image captioning via generative pretrained transformers (2022)
23. Sun, L., Wang, W., Li, J., Lin, J.: Study on medical image report generation based on improved encoding-decoding method. In: Huang, D.-S., Bevilacqua, V., Premaratne, P. (eds.) ICIC 2019. LNCS, vol. 11643, pp. 686–696. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26763-6_66
24. Sung, H., et al.: Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: Cancer J. Clin.* **71**(3), 209–249 (2021). <https://doi.org/10.3322/caac.21660>
25. Vaswani, A., et al.: Attention is all you need (2023)



Violence Detection Using Skeleton Data with Graph Convolutional Networks

Nha Tran^{1,2,3}, Hung Nguyen^{3(✉)}, Dat Ly³, and Hien D. Nguyen^{1,2,4}

¹ University of Information Technology, Ho Chi Minh City, Vietnam

² Vietnam National University, Ho Chi Minh, Vietnam

³ Ho Chi Minh University of Education, 280 An Duong Vuong Street, Ward 4,
District 5, Ho Chi Minh, Vietnam
hungnv@hcmue.edu.vn

⁴ Computer Science Department, New Mexico State University, Las Cruces, USA

Abstract. Nowadays, surveillance cameras play a crucial role in maintaining security and order in public areas. To optimize their effectiveness, it is essential to have automated tools capable of detecting abnormal behavior in real-time. Previous methods often faced challenges with low-resolution video inputs and changing lighting conditions, which affect the accuracy of violence detection. Previous methods focused on accuracy and relied on a large number of model parameters. Additionally, the graph-based approach has not been extensively applied to violence detection. In this paper, we propose an effective method to detect violent behavior by leveraging the power of Graph Convolutional Networks (GCNs) from vertex and edge feature sets as inputs to the model. The proposed models, named GCNOnlyEdge and GCNVertexEdge, are lightweight models that learn features from skeleton data as a graph. Therefore, the model is effective in identifying violent actions by focusing on key joints and their interactions. We experimented with these models on datasets such as HockeyFights, RWF-2000, and Violence in Movies. Our experimental results show that the GCNVertexEdge model has the best accuracy on experimental datasets, particularly on the Movie dataset. The model provides a robust solution to enhance public safety through improved surveillance systems.

Keywords: Violence Detection · Skeleton · Graph Convolution Network

1 Introduction

The detection of violent behavior in video footage has emerged as a critical area of research, driven by the growing demand for automated surveillance systems to enhance public safety. Traditional approaches, such as the use of Support Vector Machines (SVMs) with feature descriptors like the Histogram of Optical Flow Magnitude and Orientation (HOMO), have laid the groundwork for this field [1]. These methods leverage the motion of pixels between video frames to capture significant changes in scenes, proving effective in initial studies on datasets like Violent Flow and HockeyFights. Zhang et al. [2] extended the descriptor by adding temporal features to capture local information for violence detection on the Violent Flow and HockeyFights datasets.

Deep learning techniques have shown significant success in many video action recognition applications by learning spatiotemporal features directly from raw video data. Several algorithms used in previous studies to detect and classify violent school behaviors include using the CNN – BiLSTM model on the HockeyFights, Movies, and Violent Flows datasets. Pin Wang et al. [3] detected school violence through facial recognition using two models: CNN model and a CNN model based on SPP (Spatial Pyramid Pooling), a technique to handle inputs of different sizes and generate fixed-size features. Experiments were conducted on the Violent Crow and HockeyFights datasets.

The use of human skeleton data has emerged as a promising alternative. Skeleton data offers several advantages over traditional video-based approaches. It reduces the dependency on lighting and color conditions, as it abstracts human movement into joint coordinates, allowing the model to focus on the structural and dynamic aspects of behavior. By concentrating on the skeletal structure, the models can filter out background noise and irrelevant details, leading to more accurate detection of violent actions. Furthermore, skeleton data typically requires less storage and computational power, making it more suitable for real-time applications. In recent years, the use of human skeletons for violence detection has become popular, particularly through the application of Graph Neural Networks (GNNs) due to the structural similarity between human skeletons and graphs. While GNNs have shown promising results in action recognition, their application in violence detection is still in its early stages. This paper proposes a novel method for detecting and recognizing violent behavior using GCNs, leveraging vertex and edge features from skeleton data.

2 Related Work

In recent years, advancements in machine learning and deep learning have led to the application of these techniques in violence detection through video [4]. Soliman et al. [5] have shown promising results using models like VGG 16 and LSTM for spatiotemporal feature extraction, achieving high accuracy. Similarly, Singh et al. [6] used CNNs and RNNs to detect high-motion activities, yielding an accuracy of around 97.23%. However, these methods often involve large model parameters, making them computationally intensive and challenging to deploy in real-time applications. Additionally, they rely on traditional video features, which can be affected by factors like video quality and context, limiting their effectiveness in accurately detecting violent behaviors. This highlights the need to explore alternative data sources, such as skeleton data, which more accurately represent human movements and postures.

Detecting violent situations in videos using human skeleton data has shown superior results. Narynov et al. [7] proposed a method that first extracts human poses from video frames and then uses deep learning models to classify violent behavior, achieving 97% accuracy. Su et al. [8] introduced the Skeleton Point Interaction Learning (SPIL) strategy, which combines local and global focus on 3D skeleton points, improving accuracy in violence detection to over 98% on several datasets. However, skeleton-based methods can suffer from reduced accuracy due to video resolution and poor lighting conditions.

LSTM can face limitations when processing data with complex spatial structures like the human skeleton, as it only utilizes sequential information without considering

the relationships between data points. This leads to missing important spatial structure information, affecting the accuracy of violence detection. In recent years, the use of human skeletons for violence detection has become popular. Notably, Omarov et al. [9] applied the PoseNET model to extract 17 key points on the human skeleton in each frame with high accuracy on a self-built dataset. GNNs have emerged as a promising method for action recognition in video data due to the similarity between the human skeleton and a graph. Consequently, GNNs are widely applied in action recognition tasks in videos, with many studies yielding impressive results [10].

Therefore, the use of graphs for action recognition has also garnered attention from researchers. Yan et al. [11] proposed a Graph Convolutional Network (GCN) model that integrates spatial and temporal information from skeleton frames, achieving high performance on popular action datasets Kinetics and NTU-RGBD. Similarly, Shi et al. [12] developed a two-stream GCN model, using both spatial and temporal information along with a self-attention layer to enhance action recognition accuracy. The authors demonstrated that simultaneously using temporal and spatial information helps the model capture the dynamic changes of actions more effectively.

3 Methods

3.1 Skeleton Feature Extraction

For skeleton data, each joint of a human skeleton in a frame is represented as a node in the graph, and the edges connecting the nodes represent the physical connections between joints or the motion over time. We apply matrices to represent the vertex and edge features of the graph. The use of this feature representation method is inspired by the study [13]. A graph G is defined by a set of vertices V , and a set of edges E as shown in Eq. (1).

$$G = (V, E) \quad (1)$$

where:

- $V = \{v_1, v_2, \dots, v_n\}$ is the set of vertices.
- $E = \{(v_i, v_j) | v_i, v_j \in V\}$ is the set of edges connecting the vertices.

Vertex Feature Matrix. There are 12 rows, each representing a vertex as a keypoint (joint) on the skeleton. Each row consists of 6 features: the coordinates x , y , and z , and the angles between the vector and the x , y , z axes, denoted as θ_X , θ_Y , and θ_Z , shown in the Eqs. (2), (3) and (4). Here, we normalize the coordinates x , y , and z of each keypoint in the range $[0, 1]$.

$$\theta_X = \cos^{-1} \left(\frac{x}{\sqrt{x^2 + y^2 + z^2}} \right) \quad (2)$$

$$\theta_Y = \cos^{-1} \left(\frac{y}{\sqrt{x^2 + y^2 + z^2}} \right) \quad (3)$$

$$\theta_Z = \cos^{-1} \left(\frac{z}{\sqrt{x^2 + y^2 + z^2}} \right) \quad (4)$$

Edge Feature Matrix. There are 12 rows, each row represents an edge (connection between two keypoints) and 8 columns represent the parameters of the distance and the angles between vectors related to the x , y , z axes, and in 3-dimensional space, as shown in Fig. 1. The parameters are calculated by the following formulas: $\Delta x_{ij} = x_j - x_i$, $\Delta y_{ij} = y_j - y_i$, and $\Delta z_{ij} = z_j - z_i$ denote the distances between vectors i and j along the x -axis, y -axis, and z -axis, respectively. Additionally, $\Delta \theta_{xij} = \theta_{xj} - \theta_{xi}$, $\Delta \theta_{yij} = \theta_{yj} - \theta_{yi}$, and $\Delta \theta_{zij} = \theta_{zj} - \theta_{zi}$ indicate the angular differences between vectors i and j along the x -axis, y -axis, and z -axis. $d(i, j)$ describes Euclidean distance between the two vectors, while θ_{xyz} represents the angle between the vectors in 3-dimensional space. These parameters are crucial for analyzing the spatial relationships and orientations of keypoints as described in Eqs. (5) and (6).

$$d(i, j) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} \quad (5)$$

$$\theta_{xyz} = \cos^{-1} \left(\frac{x_j \times x_i + y_j \times y_i + z_j \times z_i}{\sqrt{(z_i^2 + y_i^2 + z_i^2) \times (x_j^2 + y_j^2 + z_j^2)}} \right) \quad (6)$$

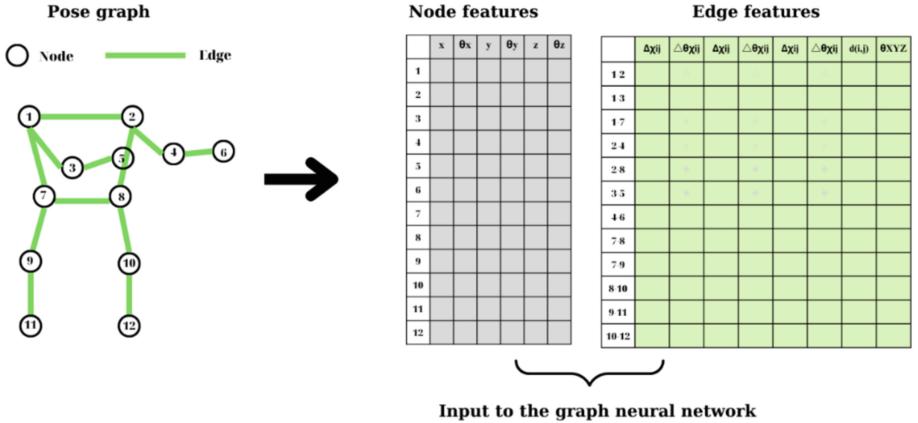


Fig. 1. Vertex and edge feature representation

3.2 GCNonlyEdge Model

The GCNonlyEdge model only uses vertex features and the adjacency matrix, where $\text{edge} = 1$ if there is an edge between two vertices, and it only uses a single skeleton. The vertex features are updated according to the formula provided in Eqs. (7).

$$x'_i = \theta^T \sum_{j \in \mathcal{N}(v) \cup \{i\}} \frac{e_{j,i}}{\sqrt{d_j d_i}} x_j \quad (7)$$

where x'_i is the output feature vector of vertex i after applying the transformation, and θ^T is the parameter of the model learned during training. The calculation $\sum_{j \in \mathcal{N}(v) \cup \{i\}} \frac{e_{j,i}}{\sqrt{d_j d_i}} x_j$ is the aggregation of the neighboring vertices of i , including itself. $\frac{e_{j,i}}{\sqrt{d_j d_i}} x_j$ is the weight for the features of vertex j , where $d_j d_i$ are the degrees of vertices i and j , respectively.

The model takes a vertex feature matrix as input, where each vertex is represented by a 6-dimensional vector (x , y , z coordinates and angles with respect to the coordinate axes). The model includes a GCNConv layer that performs the first convolutional transformation on the graph's vertices. A second GCNConv layer follows with a second convolutional transformation, while maintaining the feature size at 6 dimensions. The global features of the graph are then flattened, transforming them into a 1-dimensional vector for easier processing in subsequent layers. Normalization is applied to the flattened features, improving the stability and performance of the model. Finally, a Sigmoid layer applies the Sigmoid function to convert the predicted output into probabilities for each class. The GCNonEdge model has two main weaknesses: the edge feature weight is always 1, and the model overly relies on vertex degree, which can lead to misidentifying important vertices, as shown in Fig. 2.

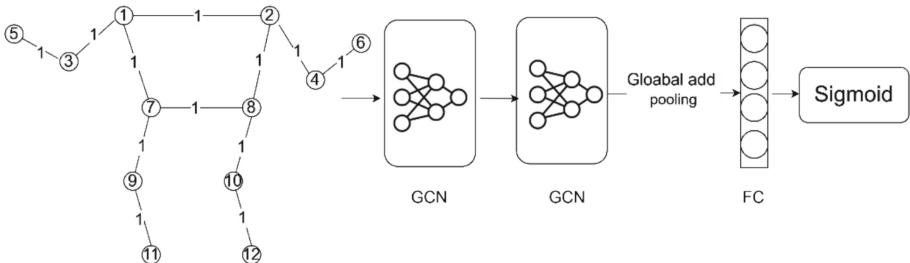


Fig. 2. The GCNonEdge model

3.3 GCNVertexEdge Model

The GCNVertexEdge model addresses the weaknesses of GCNonEdge by incorporating edge features. Instead of only using vertex features and the adjacency matrix with edge weights always equal to 1, GCNVertexEdge computes new features from the features of the two connected nodes. These include the distance and angles between the vertices connected by the edge along the x , y , z axes, the Euclidean distance, and the angle between the two lines connecting the vertices linked by the edge. The formula for updating the vertex features, as shown in Eq. (8)

$$x'_i = \theta x_i + \sum_{j \in \mathcal{N}(i)} x_j h_\theta(e_{i,j}) \quad (8)$$

where x'_i is the output feature vector of vertex i after applying the transformation, $e_{i,j}$ is the feature vector of the edge connecting vertex i and vertex j . The nonlinear function h_θ , with parameter θ , used to compute the new feature value of vertex i based on the

neighboring vertices and the edges connected to vertex i . Here, θ is the parameter of the machine learning model, updated through the training process.

Figure 3 illustrates the GCNVertexEdge model, where the feature of vertex 1 is updated by summing the features of the surrounding vertices (3, 7, 2) and vertex 1. The features of the surrounding vertices are normalized by taking the edge features through a Fully Connected layer, while the updated vertex feature is not normalized. The main difference from GCNonlyEdge is that GCNVertexEdge does not use the vertex degree, and each edge has 8 different features, addressing the limitation of having edge weights always equal to 1 in GCNonlyEdge.

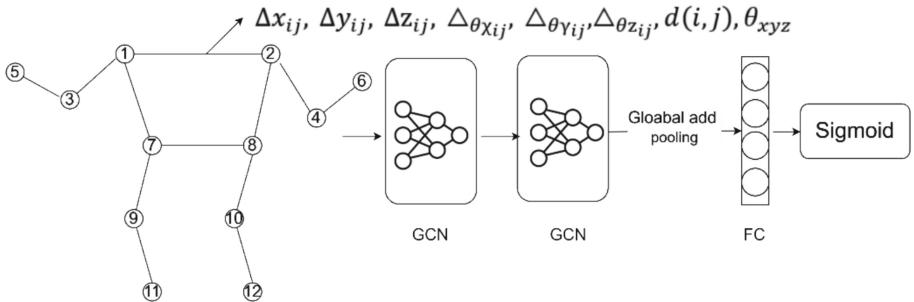


Fig. 3. GCNVertexEdge Model

4 Experiments and Results

4.1 Dataset

In this study, we use videos as input for the model, with data taken from the datasets HockeyFights, RWF-2000, and Violence in Movies. The HockeyFights dataset [14] was introduced in 2011 and contains videos from hockey games, focusing on fighting incidents. It includes a total of 1000 videos, with 500 videos depicting violent behavior and 500 videos without violent behavior. This dataset is particularly useful for evaluating models in sports scenarios where physical confrontations frequently occur.

The Violence in Movies dataset [15], also introduced in 2011, comprises 200 video clips taken from various movies. This dataset is balanced with 100 violent videos and 100 non-violent videos, providing a challenging environment for models to distinguish between staged violence in films and non-violent scenes.

The RWF-2000 dataset [16], released in 2019, is one of the most comprehensive datasets for detecting violent behavior in videos. It includes 2000 video clips recorded from real-life scenarios, primarily from YouTube. This dataset is evenly divided with 1000 violent videos and 1000 non-violent videos, making it ideal for training and evaluating models for practical applications such as surveillance and public safety.

As shown in Fig. 4, the input data consists of videos, which are processed using YOLOv8 to identify the humans appearing in them. Keypoints are then extracted using

MediaPipe Pose [17] based on each bounding box identified by YOLOv8 [18]. Mediapipe Pose will detect the human skeleton, and we select only 12 key points out of the 33 detected points for use in the deep learning model, as shown in Fig. 5. The number of human skeletons appearing in the videos from the three datasets is summarized in Table 1.



Fig. 4. Keypoint extraction process in video

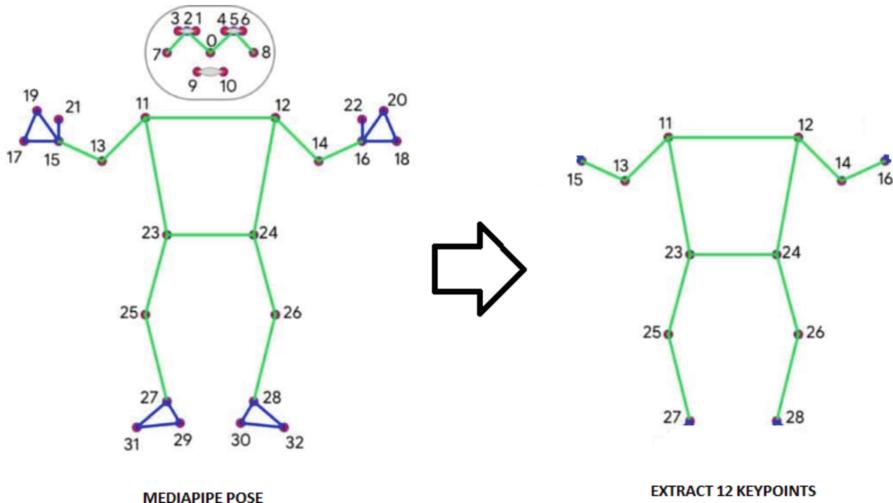


Fig. 5. Keypoints extracted from Mediapipe pose

Table 1. Number of Skeletons extracted.

Data	Hockey Fight	Hockey non Fight	RWF2000 Fight	RWF2000 non Fight	Movie Fight	Movie non Fight
Number of Skeletons	4700	2924	14043	13625	299	510

The datasets used in this study are pre-labeled, so we use the video labels to assign labels to the skeletons extracted from the videos. Specifically, for each skeleton identified by Mediapipe Pose, we only use 12 key points, and each of these points will have x , y , z values. These values are stored in a.csv file. The file structure is described in Fig. 6,

where the data of a skeleton is stored in a single row. The first column is the skeleton index, and the data in the subsequent columns are the x , y , z values of vertex 1. Similarly, columns 3, 4, 5 are the x , y , z values of vertex 2. This pattern continues until columns 33, 34, 35, which are the values for vertex 12.

	0	1	2	3	4	5	6	7	8	9	10	11
0	0.695855	0.160407	-0.29209	0.299698	0.165434	-0.27603	0.755816	0.258796	-0.35287	0.282644	0.276913	-0.27758
1	0.780404	0.251214	-1.12132	0.222852	0.251252	-1.01559	0.844211	0.392287	-1.17678	0.224037	0.408396	-0.72891
11	12	13	14	15	16	17	18	19	20	21	22	23
-0.27758	0.716436	0.300929	-0.6323	0.235064	0.365264	-0.51712	0.620297	0.326196	0.016407	0.401959	0.328912	-0.01627
-0.72891	0.773432	0.459721	-1.8827	0.224184	0.543044	-1.049	0.667982	0.492484	-0.02794	0.369439	0.491236	0.027783
24	25	26	27	28	29	30	31	32	33	34	35	
0.71122	0.380978	-0.57862	0.431786	0.404238	-0.42695	0.629715	0.478625	-0.03469	0.338645	0.487103	0.057231	
0.605826	0.684259	-0.33266	0.425373	0.690106	-0.14015	0.625321	0.833661	0.696484	0.448364	0.82955	0.821231	

Fig. 6. Illustration of.csv file structure

For the model's input data, we divided the data into three parts: 60% for the training set, 20% for the validation set, and 20% for the test set. The data in the training and validation sets are used to train the model. The data in the test set is used to evaluate the model after training.

4.2 Experimental Setup

In the experiments, we used Google Colaboratory (Colab). The hardware specifications include an Intel Xeon processor with two cores @ 2.30 GHz and 13 GB RAM for the CPU, a Tesla K80 with 12 GB GDDR5 VRAM and an Intel Xeon processor with two cores @ 2.20 GHz and 13 GB RAM for the GPU, and a cloud TPU with 180 teraflops of computation, an Intel Xeon processor with two cores @ 2.30 GHz and 13 GB RAM for the TPU. In this study, we used Colaboratory Pro to achieve higher performance, providing approximately 25.46 GB of RAM and 166.83 GB of GPU memory, doubling the performance of standard Colab.

The hyperparameters for our experiments were set as follows: batch size of 128, 100 epochs, and a learning rate of 0.01. The models were implemented using the Python programming language along with major libraries such as OpenCV for image and video processing, PyTorch for neural network implementation.

4.3 Results

In this study, we compare the performance of different models: GCNOnlyEdge, and GCNVertexEdge, on three datasets: HockeyFights, RWF2000, and Movie. The experimental results presented in Table 2, and Table 3, show significant differences in performance among the models.

The GCNOnlyEdge model (Table 2) demonstrates relatively good classification capabilities with the datasets, but still has limitations. Specifically, on the HockeyFights dataset, the model achieves an accuracy of 64.70%, with a precision of 68.05%, recall of 82.14%, and F1-score of 74.27%. For the RWF2000 dataset, the model shows lower

accuracy, achieving only 57.58%, with a precision of 63.12%, recall of 38.73%, and F1-score of 47.82%. On the Movie dataset, the results are slightly better with an accuracy of 68.89%, precision of 64.50%, recall of 57.82%, and F1-score of 60.51%.

Table 2. Results of the GCNOnlyEdge Model.

Dataset	Accuracy	Precision	Recall	F1 - score
HockeyFights	64.70%	68.05%	82.14%	74.27%
RWF2000	57.58%	63.12%	38.73%	47.82%
Movie	68.89%	64.50%	57.82%	60.51%

Table 3. Results of the GCNVertexEgde model.

Dataset	Accuracy	Precision	Recall	F1 - score
HockeyFights	71.39%	76.56%	77.56%	76.97%
RWF2000	62.23%	65.68%	50.43%	56.89%
Movie	77.30%	64.58%	75.00%	69.05%

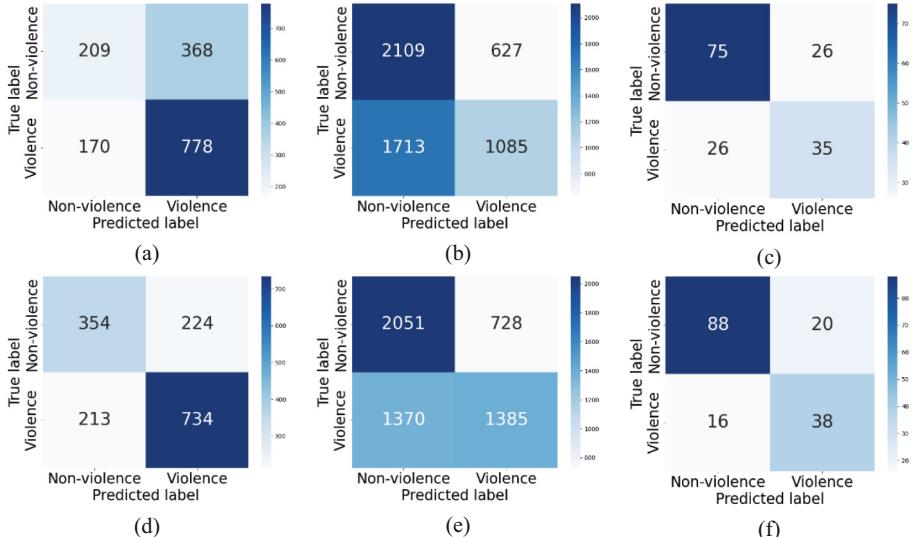


Fig. 7. Confusion matrix: (a) GCNOnlyEdge on HockeyFights, (b) GCNOnlyEdge on RWF2000, (c) GCNOnlyEdge on Movie; (d) GCNVertexEdge on HockeyFights, (e) GCNVertexEdge on RWF2000, (f) GCNVertexEdge on Movie

When using the GCNVertexEdge model (Table 3), we observe a significant improvement in performance. On the HockeyFights dataset, the accuracy increases to 71.39%,

with a precision of 76.56%, recall of 77.56%, and F1-score of 76.97%. On the RWF2000 dataset, the accuracy also improves to 62.23%, with a precision of 65.68%, recall of 50.43%, and F1-score of 56.89%. On the Movie dataset, the model achieves an accuracy of 77.30%, precision of 64.58%, recall of 75.00%, and F1-score of 69.05%. This improvement indicates that using both edge and vertex features in GCNVertexEdge helps the model to learn better and classify more accurately. The differences in performance among the models can be explained by how they process and utilize information from skeleton data. The GCNOnlyEdge model only uses vertex features, while the GCNVertexEdge model adds edge features, which helps improve the learning of relationships between vertices.

Besides, by examining the confusion matrices in Fig. 7, we have gained significant insights into the performance of the models. Among them, the RWF2000 dataset is currently showing poor performance on both models. The models mainly struggle with classifying the “Violence” label as “Non-Violence”. For the HockeyFights dataset, both models are effective at detecting violence but perform poorly in recognizing non-violent situations. In contrast, the model trained on the Movie dataset shows the opposite performance compared to the model trained on HockeyFights.

Table 4. Comparison table of the proposed method with state-of-the-art models

Methods	Hockey Fights	RWF2000	Movies	Parameters
Sudhakaran [19]	97.1%	77%	100%	9.76 M
Guillermo [20]	94.50%	90.25%	98.50%	62583
Deniz [21]	90.10%	–	98.90%	–
Gracia [22]	82.40%	–	97.8%	–
Santos [23]	–	84.75%	–	–
Kang [24]	99%	89.25%	100%	5.29 M
Ours (GCNonlyEdge)	64.70%	57.58%	68.89%	110
Ours (GCNVertexEdge)	71.39%	65.68%	77.30%	475

In Table 4, we compare the performance of the proposed model with other related works in the field of violence detection. The model by Sudhakaran et al. [19] achieved the highest performance on the HockeyFights dataset (97.1%) and Movies (100%), but only 77% on RWF2000. Although it has high accuracy, this model uses up to 9.76 million parameters, requiring significant computational resources. Guillermo et al. [20] also achieved good results, with performance of 94.5% on HockeyFights, 90.25% on RWF2000, and 98.5% on Movies, but still required 62583 parameters. Meanwhile, the models by Deniz et al. [21] and Gracia et al. [22] had average performance, with 90.1% and 82.4% on HockeyFights and good performance on Movies; however, the number of parameters for these models was not provided. Kang et al. [24] achieved very high performance (99% on HockeyFights and 100% on Movies), but their model also required up to 5.29 million parameters.

In contrast, our GCNOnlyEdge model, although having lower performance (64.7% on HockeyFights, 57.58% on RWF2000, and 68.89% on Movies), only uses 110 parameters, showing that it is a very lightweight model. Additionally, the improved model, GCN-VertexEdge, increased performance to 71.39% on HockeyFights, 65.68% on RWF2000, and 77.3% on Movies, while using only 475 parameters, indicating that the model has improved both in terms of performance and compactness.

5 Conclusion

In this paper, we proposed and evaluated lightweight models based on GCNs and skeleton data: GCNOnlyEdge and GCNVertexEdge, using the HockeyFights, RWF2000, and Movie datasets. The experiments revealed significant differences in performance. The GCNOnlyEdge model, while demonstrating reasonable classification ability, was limited by its reliance solely on vertex features. The GCNVertexEdge model, however, showed improved performance by incorporating edge features, highlighting the importance of leveraging both vertex and edge information for better detection accuracy. Despite their simplicity, these models are computationally efficient, making them suitable for deployment in real-time violence detection applications.

In future work, we plan to apply these models to other datasets, further optimize their computational efficiency, and explore their potential in real-time surveillance systems. Our research underscores the potential of using advanced yet lightweight deep learning techniques to enhance public safety and improve the effectiveness of automated surveillance systems.

References

1. Mahmoodi, J., Salajeghe, A.: A classification method based on optical flow for violence detection. *Expert Syst. Appl.* **127**, 121–127 (2019). <https://doi.org/10.1016/j.eswa.2019.02.032>
2. Zhang, T., Jia, W., He, X., Yang, J.: Discriminative dictionary learning with motion Weber local descriptor for violence detection. *IEEE Trans. Circuits Syst. Video Technol.* **27**(3), 696–709 (2017). <https://doi.org/10.1109/tcsvt.2016.2589858>
3. Wang, P., Wang, P., Fan, E.: Violence detection and face recognition based on deep learning. *Pattern Recogn. Lett.* **142**, 20–24 (2021). <https://doi.org/10.1016/j.patrec.2020.11.018>
4. Ullah, F.U., Obaidat, M.S., Ullah, A., Muhammad, K., Hijji, M., Baik, S.W.: A comprehensive review on vision-based violence detection in surveillance videos. *ACM Comput. Surv.* **55**(10), 1–44 (2023). <https://doi.org/10.1145/3561971>
5. Soliman, M.M., Kamal, M.H., El-Massih Nashed, M.A., Mostafa, Y.M., Chawky, B.S., Khattab, D.: Violence recognition from videos using deep learning techniques. In: Proceedings of the 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 1–6 (2019). <https://doi.org/10.1109/icicis46948.2019.9014714>
6. Singh, V., Singh, S., Gupta, P.: Real-time anomaly recognition through CCTV using neural networks. *Procedia Comput. Sci.* **173**, 254–263 (2020)
7. Narynov, S., Zhumanov, Z., Gumar, A., Khassanova, M., Omarov, B.: Physical violence detection in video streaming using partitioned skeleton analysis. In: Proceedings of the 2021 21st International Conference on Control, Automation and Systems (ICCAS), pp. 1–6 (2021). <https://doi.org/10.23919/iccas52745.2021.9649827>

8. Su, Y., Lin, G., Wu, Q.: Improving Video Violence Recognition with Human Interaction Learning on 3D Skeleton Point Clouds (2023). arXiv preprint [arXiv:2308.13866](https://arxiv.org/abs/2308.13866)
9. Omarov, B., Narynov, S., Zhumanov, Z., Gumar, A., Khassanova, M.: A skeleton-based approach for campus violence detection. *Comput. Mater. Continua* **72**(1), 315–331 (2022). <https://doi.org/10.32604/cmc.2022.024566>
10. Ahmad, T., Jin, L., Zhang, X., Lai, S., Tang, G., Lin, L.: Graph convolutional neural network for human action recognition: a comprehensive survey. *IEEE Trans. Artif. Intell.* **2**(2), 128–145 (2021). <https://doi.org/10.1109/TAI.2021.3076974>
11. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1 (2018)
12. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12026–12035 (2019)
13. Azzam, R., Kong, F., Taha, T., Zweiri, Y.: Pose-graph neural network classifier for global optimality prediction in 2D SLAM. *IEEE Access* **9**, 80466 (2021). <https://doi.org/10.1109/ACCESS.2021.3084599>
14. Nievas, E.B., Suarez, O.D., Garcia, G.B., Sukthankar, R.: Hockey fight detection dataset. In: Computer Analysis of Images and Patterns, Seville, Spain, pp. 332–339 (2011). Springer
15. Nievas, E.B., Suarez, O.D., Garcia, G.B., Sukthankar, R.: Movies fight detection dataset. In: Computer Analysis of Images and Patterns, pp. 332–339 (2011)
16. Cheng, M., Cai, K., Li, M.: RWF-2000: An open large scale video database for violence detection. In: Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), pp. 4183–4190 (2021)
17. MediaPipe Pose: Retrieved July 1, 2024, from <https://google.github.io/mediapipe/solutions/pose.html>
18. Ultralytics YOLOv8 Docs: Retrieved July 1, 2024, from <https://docs.ultralytics.com/>
19. Sudhakaran, S., Lanz, O.: Learning to detect violent videos using convolutional long short-term memory. In: Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), pp. 1–6 (2017). IEEE
20. Garcia-Cobo, G., SanMiguel, J.C.: Human skeletons and change detection for efficient violence detection in surveillance videos. *Comput. Vis. Image Underst.* **233**, 103739 (2023)
21. Deniz, O., Serrano, I., Bueno, G., Kim, T.-K.: Fast violence detection in video. In: Proceedings of the 2014 International Conference on Computer Vision Theory and Applications (VISAPP), vol. 2, pp. 478–485 (2014). IEEE
22. Gracia, I.S., Suarez, O.D., Garcia, G.B., Kim, T.-K.: Fast fight detection. *PLoS ONE* **10**(4), e0120448 (2015)
23. Santos, F., Durães, D., Marcondes, F.S., Lange, S., Machado, J., Novais, P.: Efficient violence detection using transfer learning. In: Proceedings of the International Conference on Practical Applications of Agents and Multi-Agent Systems, pp. 65–75 (2021). Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-85710-3_6
24. Kang, M.S., Park, R.H., Park, H.M.: Efficient spatio-temporal modeling methods for real-time violence recognition. *IEEE Access* **9**, 76270–76285 (2021)



3D Simulation of Brain Tumor from 3D MRI Using Geometric Convolutional Neural Network and Point Clouds

Anh-Cang Phan^(✉) , Khac-Tuong Nguyen, Minh-Phuong Truong, Thi-Hong-Yen Nguyen, and Ngoc-Hoang-Quyen Nguyen

Vinh Long University of Technology Education, Vinh Long, Vietnam
`{cangpa,tuongnk,phuongtm,yennth,quyennnh}@vluite.edu.vn`

Abstract. Brain tumor is a disorder caused by the growth of abnormal brain cells. Brain tumor is a major risk for the patient's survival rate and quality of life since it can cause severe impairment of organ function and even death. In the study, the research team presented an approach by utilizing methods to the modeling of brain tumor by replacing 3D with 2D convolutions based on tumor segmentation on 2D images using the Geometric convolutional neural network (gCNN) and Point Cloud. This method not only enables to conduct an accurate reconstruction and location, but also aids in monitoring changes in tumor size and growth rate over time. A result of the study indicates that the implement of gCNN and Point Cloud significantly improves the accuracy and efficiency in brain tumor segmentation and monitoring, thus supporting medical practitioners in the diagnosis and treatment planning process. This approach holds immeasurable promise in the medical field, aiming to enhance the quality of care and treatment for patients with brain tumors.

Keywords: Brain tumor · gCNN · MRI · 3D-Unet

1 Introduction

1.1 Problem Statement

Brain tumors [1] are estimated to be the tenth leading cause of cancer-death in 2023 for both men and women in each age group. Approximately 308,102 new cases were diagnosed globally with primary brain or spinal cord tumor in 2020 [2]. According to the National Brain Tumor Society [3], statistics indicates that around 1 million Americans are living with a diagnosis primary brain tumors, with about 94,390 individuals unexpected in 2023. In this year, the average survival rate for patients is about 35.7%, with 18,990 Americans is expected that have been lost due to a malignant brain tumor [3]. The growth rate of brain tumors appears to vary based on several factors including tumor type, location, biological characteristics, and health situation [4,5]. Malignant tumors

typically grow faster and require urgent treatment, whereas benign tumors may grow slower and can be monitored by periodically [4,5].

Magnetic Resonance Imaging (MRI) [6] is a technique using a magnetic field, computer-generated radio waves to create detailed images of the organs and tissues in your body, such as the brain and spinal cord. MRI provides parametric imaging with contrast-enhanced and has the advantage of being able to visualize anatomy in different planes of MRI brain image. MRI making it more effective at pinpointing the precise location of lesions compared to crucial neuroanatomical structures. This is critical for optimal surgical and radiotherapy planning. However, MRI has limitations such as the inability to detect small calcifications and the current inability to assess changes in the blood-brain barrier [7]. Therefore, modeling brain tumors on MRI images aids in monitoring changes in tumor size and growth rate over time, advancing medical care and treatment for brain tumor patients.

1.2 Related Research

Recent studies have explored Deep Learning Approaches to detect brain tumor and related issues. For reference to specific studies carried out in the past of the collaborative author including Phan Thuong Cang, Phan Anh Cang, Nguyen Khac Tuong, Tran Ho Dat [8] The group author presented methods to detect and segment brain tumors from 3D MRI images by using 3D Unet technique with A-VGG16-UNet model received the highest accuracy of 99%. The research by corresponding authors Phan Anh Cang, Tran Ho Dat, Phan Thuong Cang [9] that proposed an effective method to detect cerebral hemorrhage on 3D CT Scans Scanning using a deep neural network which achieving an average accuracy of 92.5%. Research of the group Dina Mohammed Sherif El-Torky, Maryam Nabil Al-Berry, Mohammed Abdel-Megeed Salem, Mohamed Ismail Roushdy [10] conducted a survey on 3D visualization of brain tumors using MRI images, presenting a two-step process: tumor segmentation and 3D modeling. Haewon Byeon et al. [11] introduced a new method for brain tumor segmentation using a smart cascading U-Net model with Dynamic Convolution Self-Attention Network. In order to reconstruct more detailed spatial information on brain tumours, the principal design is a two-stage cascade of 3DU-Net. Andrés Serna and Flavio Prieto's research [12] focused on 3D brain tumor modeling using endoscopic neuromonitoring and neural networks, demonstrating the advantage of neural network models in encoding tumor morphology without prior knowledge. It can be developed in two stages: offline and online adapting training. Experimental tests were performed by using virtualized phantom brain tumors. Soumya S Pillai; Rajesh Kannan Megalingam [13] Detection and 3D Modeling of Brain Tumor using Machine learning and Conformal Geometric Algebra for brain tumor detection and 3D modeling using MRI images, highlighting the effectiveness of 3D modeling as an aid in diagnosis and treatment planning.

2 Background

2.1 Brain Tumor

Common warning signs of brain tumors include headaches, nausea, vomiting, eye problems, seizures, hearing problems, speech problems, and cognitive changes [14, 15]. Waking up frequently with a headache can be the first sign of a brain tumor. Persistent nausea and vomiting without reasons could increased intracranial pressure (ICP). Vision problems, such as blurred or double vision, loss of peripheral vision, or seeing flashing lights or colors often occur when brain tumor exerts enough pressure on the optic nerve. Seizures, especially new-onset seizures in adults, though, the cause is unknown, which accompanied by headaches is the significance of the warning signs. Additionally, tumors can cause numbness, weakness, or lack of coordination in the arms, legs, or one side of the face or body may indicate that the tumor is pressing on motor control areas. Speech problems such as stuttering or difficulty finding words, occur when the tumor impacts language abilities by affecting specific language processing areas. Finally, cognitive difficulties such as confusion, memory loss, and difficulty concentrating, can significantly affect the patient's quality of life. Recognizing and addressing potential symptoms promptly is crucial for diagnosing and treating brain tumors [14, 15].

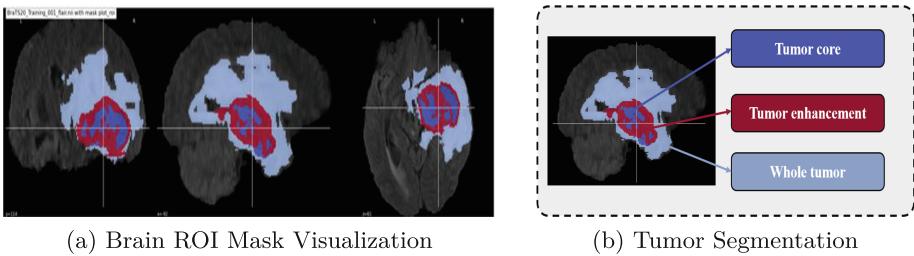


Fig. 1. Brain Tumor Regions

Figure 1.a and 1.b illustrates the three regions of a brain tumor. Firstly, the central core region of the tumor, where the tumor is the most active and typically the focal point of new tumor cell growth, increases the risk of severe health issues and the potential spread of the disease. Secondly, the surrounding tumor region, which mostly contains malignant tumor cells, significantly impacts brain function. Lastly, the peritumoral edema region represents tumor growth and exerts pressure on surrounding structures, causing damage and inflammatory responses. This emphasize the importance of the effective diagnostic and treatment methods to address the severity of brain tumors.

2.2 Network Models

3D Modeling. [16] is the process of creating three-dimensional representations of an object or a surface. Transforming 2D images into 3D typically involves reconstructing depth information from 2D images. Depth information allows for determining the position and distance of objects within the image relative to the observer. Achieving this requires using image processing methods and algorithms, including segmentation, classification, hierarchy, and matching, to analyze and understand the distinctive features of the image.

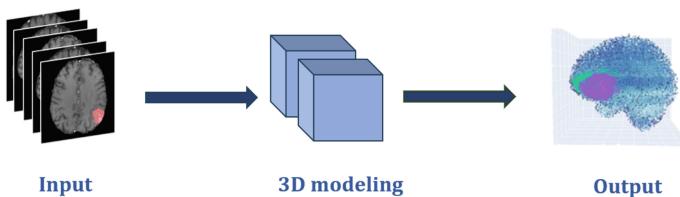


Fig. 2. Converting MRI Slices to 3D Models

Figure 2 illustrates the conversion process from MRI slice images to 3D models. The authors employed Geometric Convolutional Neural Networks (gCNN) and Point Cloud based on segmented datasets to construct 3D brain tumor models.

3D U-Net. 3D U-Net [16] is a deep neural network architecture used in medical image processing, particularly for segmentation tasks. This architecture is an extension of the U-Net but is designed to handle 3D data instead of the 2D data processed by the original U-Net. The 3D U-Net architecture consists of two main parts: an encoding part and a decoding part. This work investigates 3D encoding-decoding architectures trained using patch-based techniques to reduce memory consumption and mitigate the effects of data imbalance. Various trained models are then combined to create an ensemble, leveraging each model's strengths to enhance performance. 3D U-Net has been employed in numerous medical applications, such as medical image segmentation, cancer detection, disease classification, and predicting patient treatment outcomes.

Geometric Convolutional Neural Network (gCNN). [17] Geometric Convolutional Neural Network (gCNN) is a neural network that utilizes convolutional and pooling layers based on surfaces. The functions of these layers are similar to traditional CNNs (Krizhevsky et al., 2012; Chatfield et al., 2014; LeCun et al., 2015), but they process surface data. The architecture includes an input data layer, convolutional layers with data refresh, batch normalization layers (Ioffe and Szegedy, 2015), linear units activated by the ReLU function (Glorot et al.,

2011), pooling layers, and a fully connected softmax output layer. It includes the conceptual architecture of gCNN and the implemented architecture of gCNN. In the conceptual architecture, when data on the cortical surface enters the convolutional layer with batch normalization and ReLU activation units, feature maps (corresponding to the number of filters at each convolutional layer) are created. The size of the nodes decreases from N_1 to N_L as data passes through the pooling layers. gCNN ends with a multi-layer classifier. In the implemented architecture, input data with $40,962 \text{ nodes} \times 25 \text{ sample points per filter} \times 2 \text{ hemispheres}$ are convolved with 36 filters, then reduced to $42 \text{ nodes} \times 36 \text{ filtered outputs}$ (i.e., features) after five convolution and pooling steps. As the data passes through the layers, the number of features increases, but the node size decreases. Finally, the convolved-pooled data enters a fully connected multi-layer perceptron network, comprising a hidden layer with 50 nodes and a softmax output layer with two nodes.

Point Clouds. A point cloud [18] is a collection of points identified in 3D space, with each point containing information about its coordinates (x, y, z) and possibly additional data such as color, intensity, or other attributes. Point clouds have become one of the most critical data formats for representing three-dimensional spaces, especially with the development of data acquisition devices like LiDAR (Light Detection and Ranging), laser scanners, and 3D imaging systems. These devices can capture millions of data points in a short period, creating detailed maps of the surrounding environment. Notable methods in this field include PointNet and PointNet++, deep neural networks capable of learning directly from point clouds without prior transformation. These methods use point aggregation functions to gather information from individual points and create features representing the entire point cloud. This approach enhances accuracy in tasks such as classification, segmentation, and 3D object detection. Figure 3 demonstrates the use of Point Cloud to build a 3D model of the human brain.

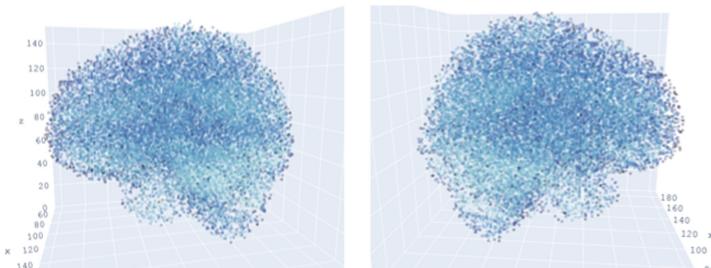


Fig. 3. Using Point Cloud Methods to Construct a 3D Human Brain Model

2.3 Evaluation Metrics

In the segmentation problems, the Dice score is used to evaluate the similarity between a predicted segmentation mask and the ground truth segmentation mask. The Dice score is calculated as in Eq. 1, where A is the predicted segmentation and B is the ground truth.

$$Dice = \frac{2|A \cap B|}{|A| + |B|} \quad (1)$$

In object detection and classification problems, especially multi-class classification problems, it is necessary to choose appropriate methods for evaluation and comparison. The evaluation metrics used in this study include Accuracy (Eq. 2), Precision (Eq. 3), Sensitivity (Eq. 4), and Loss (Eq. 5), where TP is true positive, TN is true negative, FP is false positive, FN is false negative, K is the number of classes, y is the actual value, and \hat{y} is the predicted value.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Specificity = \frac{TP}{TP + FN} \quad (4)$$

$$Loss = \sum_{i=1}^K y_i \log(\hat{y}_i) \quad (5)$$

3 Proposed Method

Figure 4 illustrates the proposed method for segmentation and 3D modeling of brain tumors on 3D MRI images. The model consists of three main stages: training, testing, and 3D brain tumor modeling. During the training and testing stages, a 3D-Unet network with different backbones is used to segment brain tumors. Based on the segmentation results, we proceed to 3D brain tumor modeling using a gCNN network and the point cloud method.

3.1 Training Phase

Data Preprocessing Involves Filtering and Enhancing. Preprocessing and data preparation are foundational steps to ensure the accuracy and efficiency of the machine learning model. This process includes removing unnecessary signals and creating clear contrasts between brain structures, aiding in the precise identification and differentiation of lesions and healthy tissues. Following this, the cortical region segmentation stage is conducted. The cortical region contains a wealth of critical information for the analysis and diagnosis

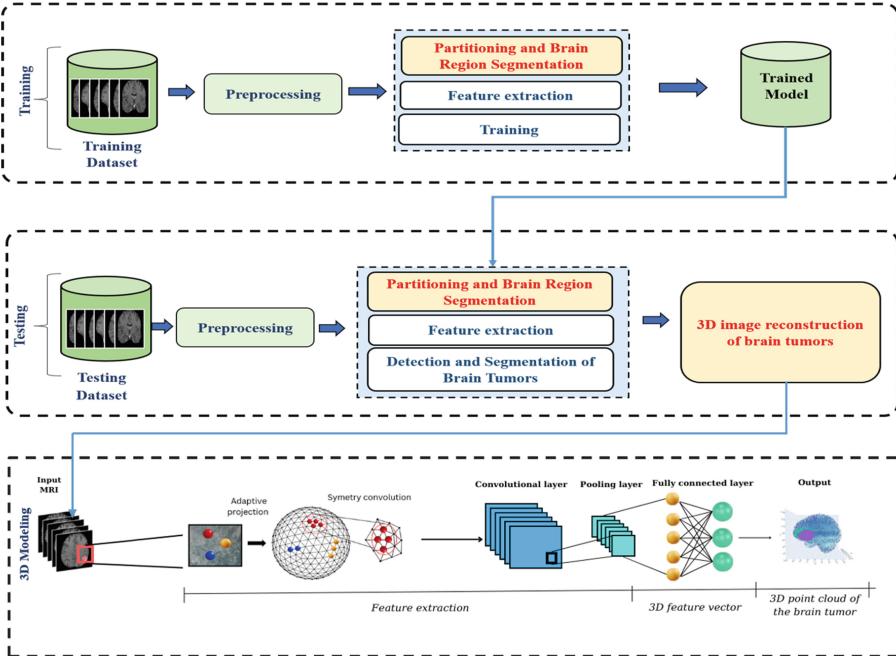


Fig. 4. Proposed 3D Brain Tumor Modeling Method

of brain-related pathologies. Segmenting the cortical region allows the model to focus on important areas, removing noise and irrelevant regions, thereby improving the accuracy of detection and classification. Figure 5 shows the 55th slice images of a patient, divided into five types: FLAIR, T1, T1ce, T2 and Mask. Since the dataset includes various MRI sequences (T1, T1CE, T2, FLAIR), we normalized the image intensities across sequences and treated each as a separate channel, allowing the model to utilize information from each type. We also applied data augmentation and sequence-specific preprocessing to enhance image quality. Finally, cross-validation across all sequences ensured the model's stability and accuracy when working with different MRI types.

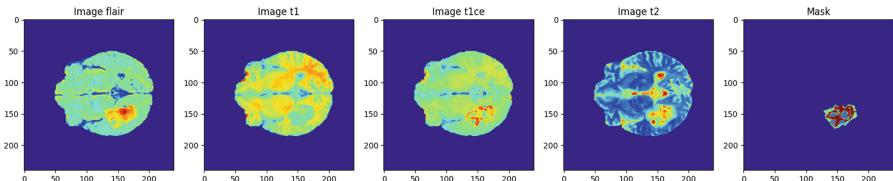


Fig. 5. Input Data

Feature Extraction and Training. After cortical region segmentation, the next step is feature extraction and training. Our solution involves developing deep learning networks for the detection and segmentation of brain hemorrhages. The 3D U-Net model is enhanced by incorporating different backbones, such as MobileNet-V2 and DenseNet-121. These improvements not only accelerate the training process but also minimize computational resource requirements and memory usage.

3.2 Testing Phase

In this stage, test data is prepared from an independent dataset that does not overlap with the training and validation data. Similar to the training phase, the test data also undergoes preprocessing, cortical region segmentation, and feature extraction. This process includes removing unnecessary signals, segmenting the cortical region to focus on critical areas, and using techniques such as image rotation and flipping to enhance data diversity. After preprocessing and feature extraction, the test data is fed into the trained model for prediction and classification. The model is evaluated using key performance metrics such as accuracy, sensitivity, specificity, and F1 score. These metrics evaluate the model's capability to precisely predict and accurately classify new brain images. Finally, the results from the model are used in the 3D modeling stage. This step creates detailed and intuitive 3D images, aiding in diagnosis and treatment planning.

3.3 Study 3D Modeling of Brain Tumor

So in this paper, we propose an advanced approach using a Geometric Convolutional Neural Network (gCNN) and point cloud methods to convert 2D MRI slices into detailed 3D models. Interpolation and surface reconstruction techniques build the 3D model from segmented 2D images.

The process begins with segmented MRI images, passing through gCNN's feature extraction layers to obtain characteristic information, and finally constructing tumors in 3D form. Convolutional layers in the gCNN apply filters to extract spatial features from the segmented images, while pooling layers reduce feature size, enhancing efficiency and minimizing overfitting. This process creates a 3D feature vector representing critical tumor information.

The fully connected layer uses this 3D feature vector to perform final calculations and predictions, identifying tumor regions. The gCNN output includes detailed segments of the brain tumor, converted into a 3D model using the point cloud method. This 3D point cloud visually represents the tumor's shape and size.

The model is adjusted based on results to ensure high accuracy, providing a valuable tool for brain tumor diagnosis and treatment. High accuracy in tumor segmentation and 3D modeling helps doctors differentiate various brain tissue regions, allowing for effective treatment decisions. The combination of gCNN's learning and feature extraction capabilities with the point cloud method has

created a powerful tool, extending the health span and opening new research directions.

4 Experiments

4.1 Dataset

We utilized the BraTS2020 and BraTS2021 datasets [19, 20] obtained from Kaggle, each patient has approximately 155 2D slices for each 3D MRI volume which are used in brain tumor segmentation research. The BraTS (Brain Tumor Segmentation) datasets contain 3D MRI images from multiple patients with brain tumors. Figure 6 illustrates the five types of images in the dataset, including flair, t1, t1ce, t2, and mask, for the same patient at slices 50 and 60. FLAIR (Fluid Attenuated Inversion Recovery - T2 fluid suppression): provides good contrast between the tumor and surrounding edema, eliminating signals from water and fluid in the brain T1 (native T1): offers information about tissue structure and the distribution of gray and white matter in the brain., T1ce (contrast-enhanced T1): uses a contrast agent during MRI to enhance contrast between the tumor and surrounding areas., T2 (T2 with inverse signal of T1): enhances signal quality of brain structures and differentiates tumor regions from normal regions. Mask: accurately identifies brain tumor regions, facilitating precise brain tumor modeling.

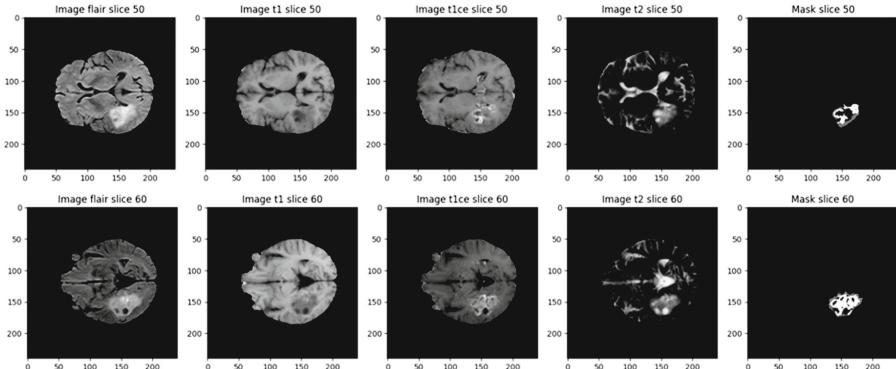


Fig. 6. MRI Images of a Patient in the Experimental Dataset

4.2 Scenarios and Training Results

Table 1 presents the scenarios corresponding to the 3D U-Net models with a learning rate of 0.001, 100 epochs, and 4 classes. We conducted experiments with the BraTS 2020 and BraTS 2021 datasets.

Table 1. Proposed scenarios and training parameters

Scenario	Backbone	Network	Learning Rate	Epochs	Num classes
1	VGG16	3D U-Net	0.001	100	4
2	MobileNet-V2	3D U-Net	0.001	100	4
4	DenseNet-121	3D U-Net	0.001	100	4

Figure 7 presents the accuracy metrics of the scenarios during training. Specifically, Scenario 1 has training accuracy and validation accuracy values of 0.99–0.99, Scenario 2 has 0.98–0.94, and Scenario 3 has 0.99–0.97. In Scenarios 2 and 3, the validation accuracy values are lower than the training accuracy values, whereas in Scenario 1, the validation accuracy is approximately equal to the training accuracy. The training loss and validation loss values for Scenario 1 are 0.02–0.02, for Scenario 2 are 0.03–0.18, and for Scenario 3 are 0.02–0.14. In Scenarios 2 and 3, the validation loss values are higher than the training loss values, while in Scenario 1, the validation loss is approximately equal to the training loss. Observing these results, Scenario 1 shows better performance compared to the other scenarios.

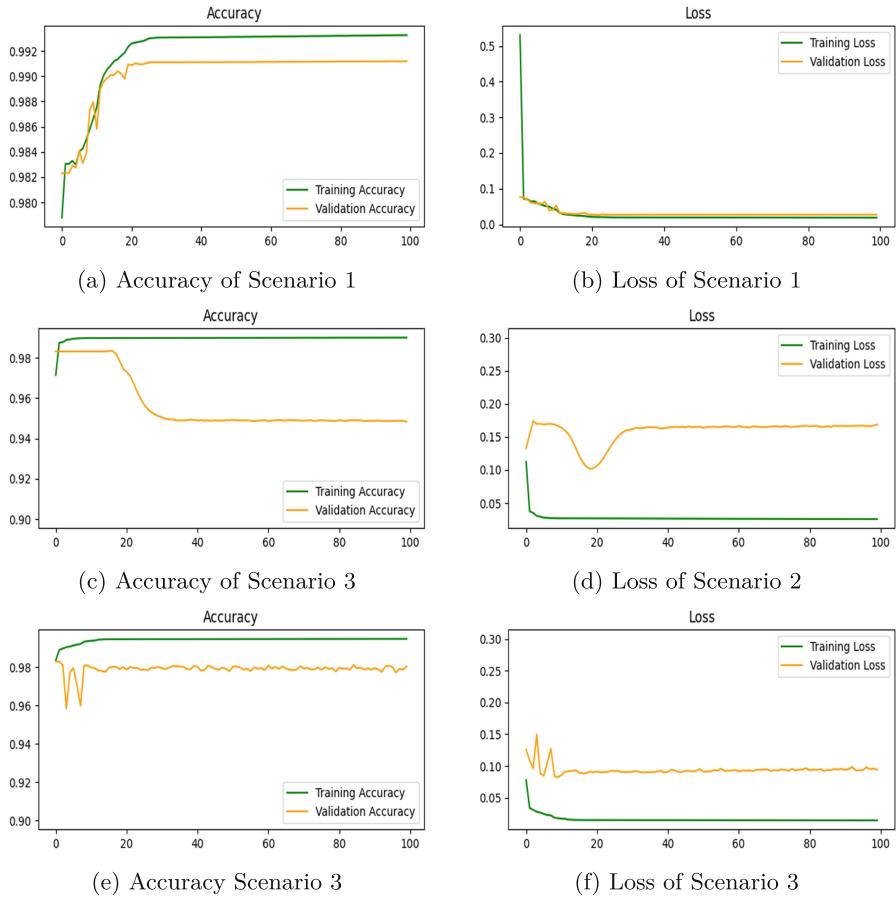
Table 2. Training Results

Scenario	Accuracy	Loss	Dice	Precision	Sensitivity	Training Time
1	0.99	0.02	0.58	0.99	0.99	255.43 min
2	0.98	0.03	0.46	0.96	0.94	245.67 min
3	0.99	0.02	0.58	0.98	0.97	251.67 min

The comparison results of the 3D U-Net model evaluation combined with the backbone of 3D U-Net in Table 2 show that Scenario 1 has the best evaluation results with an accuracy of 0.99, a loss of 0.02, a dice score of 0.58, a precision of 0.99, and a sensitivity of 0.99. Meanwhile, the other scenarios have lower evaluation results. Scenario 2 has an accuracy of 0.98, a loss of 0.03, a dice score of 0.46, a precision of 0.96, and a sensitivity of 0.94. Scenario 3 has an accuracy of 0.99, a loss of 0.02, a dice score of 0.58, a precision of 0.98, and a sensitivity of 0.97. Regarding training time, Scenario 2 has the fastest training time at 245.67 min, followed by Scenarios 3 and 1 at 249.49 min, 251.67 min, and 255.43 min, respectively. From the results in the summary table, it can be seen that Scenario 1 gives the best results compared to the other scenarios.

4.3 Testing Results

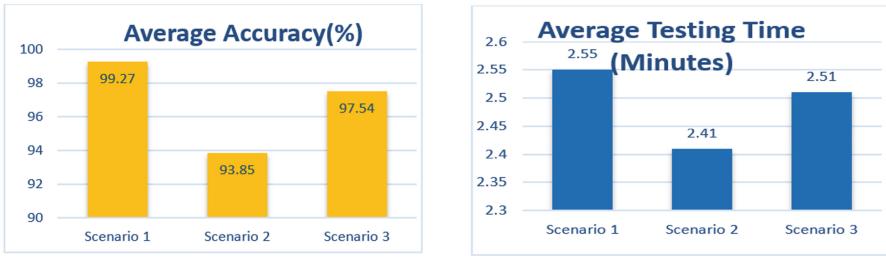
Figure 8 presents the average accuracy on the test dataset and the average prediction time for each scenario. Regarding testing accuracy (Fig. 8.a), the proposed scenarios have accuracies of 99.27%, 93.85%, and 97.54% for scenarios 1,

**Fig. 7.** Evaluation measure

2 and 3, respectively. Regarding average prediction time (Fig. 8.b), the proposed scenarios have times of 2.55 min, 2.41 min, and 2.51 min for scenarios 1, 2, and 3. From these results, it can be seen that scenario 1 provides the best performance in terms of both accuracy and time compared to the other scenarios.

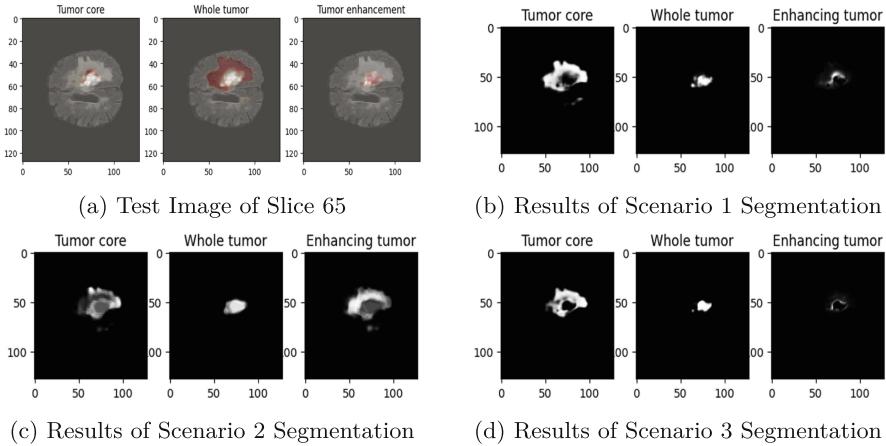
Based on the segmentation results in Fig. 9 of the same patient, Scenario 1 demonstrates the best segmentation performance among the tumor regions. Based on these results, we will utilize the segmentation results from Scenario 1 for the 3D modeling phase of the brain tumor.

Geometric Convolutional Neural Network (gCNN) and Point Cloud have yielded impressive results in analyzing and 3D modeling brain tumors from MRI data. With the ability to learn and extract features from Euclidean space data in medical images, gCNN can understand the spatial relationships among data points. In the case of brain tumors, gCNN can accurately detect and segment



(a) Test accuracy for the different scenarios

(b) Average forecast time of scenarios

Fig. 8. Training Time for Each Scenario**Fig. 9.** Test Results of Patient 125 at Slice 65

tumor regions, helping to identify and distinguish between different tissue types and tumor structures. Simultaneously, the point cloud method accurately constructs 3D models from the data, facilitating detailed observation and analysis of the tumors' shape and size. These results provide crucial information for diagnosing and planning treatment for patients, opening doors for advanced imaging analysis methods in medical and biological research.

Figure 10 illustrates the testing results of the 3D brain tumor modeling of a patient using a point cloud. Figure 10.a shows different views of the tumor displaying all segments, Figure 10.b shows the normal brain region without the tumor, Figure 10.c shows the central core of the tumor, Figure 10.d shows the surrounding area of the tumor core, and Fig. 10.e shows the peritumoral edema region that has the potential to expand and develop.

Based on the results in Fig. 10 and Table 3, which show the analysis of normal brain MRI and tumor-related regions, detailed data include 17259 points (72.86%) corresponding to a volume of 17.26 cm³ of the normal brain MRI. The

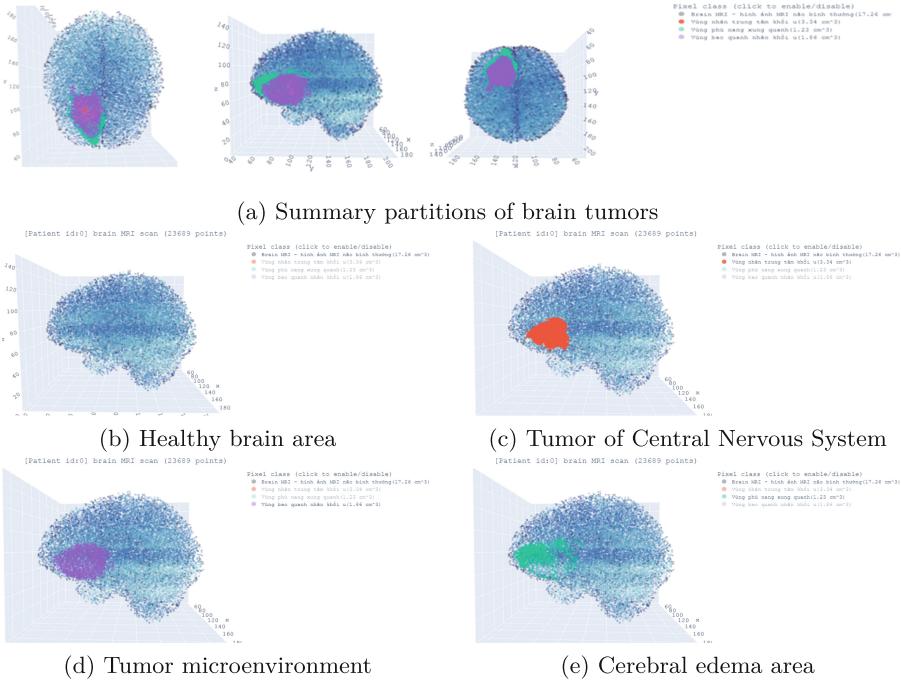


Fig. 10. An illustrative example for 3D Modeling in Brain Tumors

Table 3. Data on test data partitions of a patient

No.	Partitions of brain tumors	Points	Volumetric (cm ³)	Percentage (%)
1	Healthy brain area (10.b)	17259	17.26	72.86
2	CNS Tumor (10.c) - Tumor core	3343	3.34	14.11
3	Tumor enhancement (10.d)	1859	1.86	7.85
4	Cerebral Edema (10.e) - Whole Tumor	1228	1.23	5.18

tumor core region was identified with 3343 points (14.11%), with a volume of 3.34 cm³. The surrounding tumor region was identified with 1859 points (7.85%), with a volume of 1.86 cm³. The peritumoral edema region was identified with 1228 points (5.18%), with a volume of 1.23 cm³. After conducting the experiments, we used the 3D Slicer tool to convert MRI slices of a patient into 3D model.

5 Conclusion

This research proposed an advanced and effective method for modeling brain tumors from 2D MRI images into 3D structures using Geometric Convolutional

Neural Network (gCNN) and point cloud methods. Our method opens new opportunities for monitoring and evaluating tumor progression over time, providing practical benefits for diagnosing and planning treatment. This supports doctors in making more accurate and effective medical decisions. Additionally, applying gCNN and point cloud methods extends beyond medicine, with significant potential in other fields such as engineering, geography, and entertainment. Due to its ability to process and analyze complex geometric data, this approach can bring breakthroughs in 3D technology research and application. Based on the results presented in our paper, we have introduced a novel method for brain tumor detection using the gCNN network. This method enables visualization and precise localization of tumor regions, allowing for accurate determination of their positions and volumes.

References

1. Siegel Mph, R.L., et al.: Cancer statistics. CA Cancer J. Clin. **73**(1), 17–48 (2023). <https://doi.org/10.3322/CAAC.21763>
2. Brain Tumor: Statistics | Cancer.Net. Accessed 30 May 2024. <https://www.cancer.net/cancer-types/brain-tumor/statistics>
3. Brain Tumor Facts. Accessed 30 May 2024. <https://braintumor.org/brain-tumors/about-brain-tumors/brain-tumor-facts/>
4. Ferté, C., et al.: Tumor growth rate (TGR) is an early indicator of anti-tumor drug activity in phase I clinical trials. Clin. Cancer Res. **20**(1), 246 (2014). <https://doi.org/10.1158/1078-0432.CCR-13-2098>
5. He, L.N., et al.: Pre-Treatment tumor growth rate predicts clinical outcomes of patients with advanced non-small cell lung cancer undergoing anti-PD-1/PD-L1 therapy. Front. Oncol. **10**, 621329 (2021). [https://doi.org/10.3389/FONC.2020.621329/BIBTEX](https://doi.org/10.3389/FONC.2020.621329)
6. Brain Tumor Diagnosis MRI, Imaging | Moffitt. Accessed 31 May 2024. <https://www.moffitt.org/cancers/brain-tumor/diagnosis/mri/>
7. Maravilla, K.R., Crysyp Sory, W.: Magnetic resonance imaging of brain tumors. Semin. Neurol. **6**(1), 33–42 (1986). <https://doi.org/10.1055/S-2008-1041445>
8. Phan, T.C., Phan, A.C., Nguyen, K.T., Tran, H.D.: Detection and segmentation of brain tumors on 3D MR images using 3D U-net. In: Dang, T.K., Küng, J., Chung, T.M. (eds.) Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications, FDSE 2023, CCIS, vol. 1925, pp. 528–541. Springer, Singapore (2023). https://doi.org/10.1007/978-981-99-8296-7_38
9. Phan, A.-C., Tran, H.-D., Phan, T.-C.: Efficient brain hemorrhage detection on 3D CT scans with deep neural network. In: Dang, T.K., Küng, J., Chung, T.M., Takizawa, M. (eds.) FDSE 2021. LNCS, vol. 13076, pp. 81–96. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91387-8_6
10. El-Torky, D.M.S., Al-Berry, M.N., Salem, M.A.-M., Roushdy, M.I.: 3D visualization of brain tumors using MR images: a survey. Curr. Med .Imaging Rev **15**(4), 353–361 (2019). <https://doi.org/10.2174/1573405614666180111142055>
11. Byeon, H., et al.: Brain tumor segmentation using neuro-technology enabled intelligence-cascaded U-Net model. Front Comput. Neurosci. **18**, 1391025 (2024). [https://doi.org/10.3389/FNCOM.2024.1391025/BIBTEX](https://doi.org/10.3389/FNCOM.2024.1391025)

12. Serna, A., Prieto, F., Titular, P.: Hacia el modelado 3d de tumores cerebrales mediante endoneurosonografía y redes neuronales, Revista Ingenierías Universidad de Medellín, vol. 16, no. 30, pp. 129–148, May 2017. <https://doi.org/10.22395/RIUM.V16N30A7>
13. Pillai, S.S., Megalingam, R.K.: Detection and 3d modeling of brain tumor using machine learning and conformal geometric algebra. In: Proceedings of the 2020 IEEE International Conference on Communication and Signal Processing, ICCSP 2020, pp. 257–261, July 2020. <https://doi.org/10.1109/ICCP48568.2020.9182225>
14. Brain Tumour Symptoms | Brain Tumour Research. Accessed 31 May 2024. <https://braintumourresearch.org/pages/information-brain-tumour-symptoms>
15. Brain Tumor Symptoms. Accessed 31 May 2024
16. Griffey, J.: Chapter 2: The Types of 3-D Printing, Library Technology Reports (2014)
17. Seong, S.B., Pae, C., Park, H.J.: Geometric convolutional neural network for analyzing surface-based neuroimaging data. Front. Neuroinform. **12**, 318212 (2018). <https://doi.org/10.3389/FNINF.2018.00042/BIBTEX>
18. Point cloud and the produced 3D model | Download Scientific Diagram. Accessed 04 Jun 2024
19. Bakas, S., et al.: Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Sci. Data **4**, 1–13 (2017). <https://doi.org/10.1038/SDATA.2017.117>
20. Menze, B.H., et al.: The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Trans. Med. Imaging **34**(10), 1993–2024 (2015). <https://doi.org/10.1109/TMI.2014.2377694>



Deep Reinforcement Active Learning for Stress Recognition

Phan Anh Ngoc¹, Ky Trung Nguyen^{1,2(✉)}, Thanh-Tung Tran^{1,2},
Senerath Jayatilake³, and Thi Thanh Quynh Nguyen⁴

¹ School of Computer Science and Engineering, International University,
Ho Chi Minh City, Vietnam
ntky@hcmiu.edu.vn

² Vietnam National University, Ho Chi Minh City, Vietnam

³ School of Computing and Creative Technologies University of the West of England,
Bristol, UK

⁴ The University of Danang - University of Science and Technology, Danang,
Vietnam

Abstract. Stress has become a significantly common mental condition in the current society. It is important for people to be aware of this condition effectively when it occurs. Due to the current improvements in sensor technology, have significantly improved the accuracy and efficiency of collecting data related to human physiological biomarkers. This study focuses on employing Deep learning techniques to accurately determine mental stress states of mental stress based on the WESAD public dataset, a collection of physiological data to detect stress and affect the state. The study has demonstrated the effectiveness of using Long Short-Term Memory (LSTM) and proposed a novel framework namely, Deep Reinforcement Active Learning to enhance the accuracy of LSTM for stress classification. The proposed framework has achieved 93% accuracy, which resulted 4.91% improvement from the original study. This improvement in accuracy was achieved only by employing respiration data, which clearly demonstrates the potential of the proposed framework.

Keywords: Affective Computing · Physiological Data · Long Short-term Memory · Active Learning · Reinforcement Learning

1 Introduction

Stress is essentially a fight and flight response triggered by the Sympathetic Nervous System (SNS) in response to events perceived as harmful, threatening, and frightening to an individual. This response is beneficial as it enables people to quickly react to emergencies, thereby offering protection. However, chronic stress can lead to a host of detrimental physical and mental health issues, including cardiovascular diseases, anxiety disorders, depression, and impaired immune

function. Therefore, it is crucial to detect stress immediately to have appropriate measures for maintaining well-being and reducing stress levels. Currently, there are number of methods used in measuring and observing stress levels. When stress occurs, it triggers psychological, physiological, and behavioral symptoms. Psychological symptoms may include emotional and mental responses such as anxiety, depression, and difficulty in concentration, etc. This can be diagnosed by answering self-report questionnaires of psychologists [1]. Recognising stress via behavioural symptoms do not require any specialised equipment but rather focuses on observing daily activities and interactions. In addition to the behavioural cues, stress can also be automatically identified in real-time by monitoring bodily functions (biomarkers), such as respiration rate (RESP), heart rate (HR), blood volume pulse (BVP), and skin temperature (TEMP). These biomarkers could be easily captured by using physiological sensors embedded in smart devices. Users can access detailed information about their bodily function indices and stress levels in real-time through applications on smart wearable devices such as Empatica E4 or RespiBAN as introduced in [2]. This method has become increasingly popular as smartwatches and wearable devices have become more affordable for a wider audience. Most algorithms used for building models to recognize stress from these biomarkers are based on traditional machine learning (ML), such as Decision Tree (DT), K-Nearest Neighbors (kNN), Linear Discriminant Analysis (LDA), and Random Forest (RF) in [2–4]. However, these ML approaches often depend on manually engineered features or hand-craft features, necessitating a model design that can automate feature extraction to minimize errors caused by manual intervention and reduce the need for expert involvement. Therefore, in this study, authors perform the LSTM approach, which is capable of autonomously learning to identify relevant features from raw physiological data during the training and testing process. Furthermore, the LSTM model and Deep Learning in general, the model can predict better when the model is provided with more training data. Hence, this study has employed Active Learning [5] to augment data for training and gradually fine-tune the LSTM model's parameters, thereby making better predictions. In addition, Reinforcement Learning enhances the accuracy of LSTM performance as described in [6]. Therefore, in this paper, authors present a novel approach to integrating Active and Reinforcement Learning into the LSTM model that will be detailed in Sect. 3. Section 2 presents related work, and Sect. 4 demonstrates our experiments on the WESAD public dataset. Section 5 presents the conclusion of this study.

2 Related Work

The earlier research on stress detection often uses a fewer number of physiological data and most of them used ML to train models. Sano et al. [4] collected both physiological accelerometer data (ACC), skin conductance (SC), and behavioural data in five days from 18 subjects by mobile phone and wearable sensors. They applied a Support Vector Machine (SVM) with different kernels (Linear and

Radial Basis Function), and SVM in conjunction with principal component analysis (PCA), and kNN, PCA + kNN. With hand-craft features are extracted from the physiological and behavioural data, the best model gives the highest result with an accuracy of around 87.5%. In recent years, the field of affective computing (AC) has seen a significant shift from traditional ML models to deep learning paradigms. While ML models have demonstrated their effectiveness in various applications, their reliance on manual feature engineering poses significant challenges. Experts are required to identify, select, and transform the most relevant features from the data to ensure model efficiency [7]. Moreover, building upon the advancements highlighted in [8], which demonstrate the potential of end-to-end deep learning models to streamline affect recognition by eliminating the need for manual feature engineering, our work takes a focused approach to leverage LSTM networks. The selection of LSTM is predicated on its demonstrated superiority in handling sequential data. Moreover [6] has proved that reinforcement learning with Long Short-term memory (RL-LSTM) is a promising approach to solving non-Markovian RL tasks with long-term dependencies. By seeing the advantages and opportunities RL and LSTM have brought, the authors propose a Deep Reinforcement Active Learning to improve the accuracy of stress detection using the WESAD dataset.

3 Deep Reinforcement Active Learning Framework for Stress Recognition

3.1 Proposed Framework

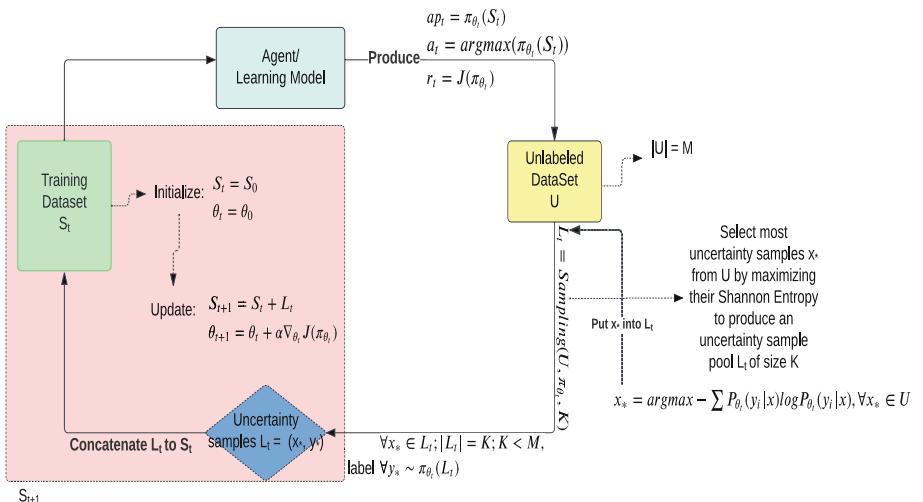


Fig. 1. Deep Reinforcement Active Learning Framework

Figure 1 illustrates our general framework as a control decision loop through many time step $t \in [0, T]$ to generate a trajectory $\tau = \{(S_t, a_t, r_t)\}$ which contains a sequence of rewards r_t along with states S_t and actions a_t . Indeed, when the framework starts the loop with the first time step $t = 0$, the agent or the learning model takes the input as a segmented signal $x = \{x_1, \dots, x_N\}$ with length N and its associated class label $y = \{y_1, \dots, y_N\}$ known as the initial training data set or state $S_t = S_0$ with initial weights $\theta_t = \theta_0$. With this general framework, the learning model can be any ML model (e.g., Recurrent Neural Network, LSTM, etc.). In this work, the learning model used is LSTM to learn and classify the stress and non-stress labels from the training dataset, thereby LSTM model produces action probabilities $ap_t = \pi_{\theta_t}(S_t)$, an action $a_t = argmax(\pi_{\theta_t}(S_t))$, and return a reward $r_t = J(\pi_{\theta_t})$, where π_{θ_t} is the policy is parametrized by learnable parameters θ , ap_t is the output probability, a_t is predicted labels, and reward r_t is the accuracy of corrected label that LSTM model classified in S_t ; when $t = 0$ the $ap_0 = \pi_{\theta_0}(S_0)$, $a_0 = argmax(\pi_{\theta_0}(S_0))$, and $r_0 = J(\pi_{\theta_0})$. This reward r_t can be used as an objective to be maximized for this framework using the following Eq. (1).

$$\max_{\theta} J(\pi_{\theta_t}) = \mathbf{E}_{\tau \sim \pi_{\theta_t}} [\sum_{t=0}^T \gamma^t r_t] \quad (1)$$

where the γ^t is the discounted sum of rewards from time step t to the end of the trajectory, the gradient ascent was performed to maximize the objective and update the policy parameters θ_{t+1} as following Eq. (2).

$$\theta_{t+1} = \theta_t + \alpha \nabla_{\theta_t} J(\pi_{\theta_t}) \quad (2)$$

where the α is the learning rate, which manages the parameter update's size, and the term $\nabla_{\theta_t} J(\pi_{\theta_t})$ is known as the policy gradient and computed as follows.

$$\nabla_{\theta_t} J(\pi_{\theta_t}) = \mathbf{E}_{\tau \sim \pi_{\theta_t}} [\sum_{t=0}^T \gamma^t r_t \nabla_{\theta_t} \pi_{\theta_t}(a_t | S_t)] \quad (3)$$

After the LSTM model is trained on S_t , its model can be used to perform label prediction on the test dataset namely unlabeled dataset U , where U contains only segmented signal of length M without any associated labels. At this testing phase, active learning [5] is used in the framework to create an uncertainty sample pool L_t of size K , where samples of pool L_t are selected from U by computing its largest entropy. To do this, the entropy of all samples in U is calculated by the output probability $\pi_{\theta_t}(U)$ using Shannon Entropy [9] presented in the following Eq. (4):

$$x_* = arg \max_x - \sum_i P_{\theta_t}(y_i | x) \log P_{\theta_t}(y_i | x) \quad (4)$$

After the entropy of all samples in U is calculated, the K samples with the highest entropy are selected to form the set L_t . At this point, the K samples

of pool L_t can also obtain their predicted labels $y_* \in Y$, which are computed by Monte Carlo sampling on the output probability as follows $y_* \sim \pi_{\theta_t}(L_t)$. Then, the uncertainty sample pool L_t was concatenated to training dataset S_t to produce an updated state $S_{t+1} = S_t + L_t$.

With these updated state S_{t+1} and policy parameters θ_{t+1} , the control loop repeats the next iteration again and again until reaching a maximum time step $t = T$.

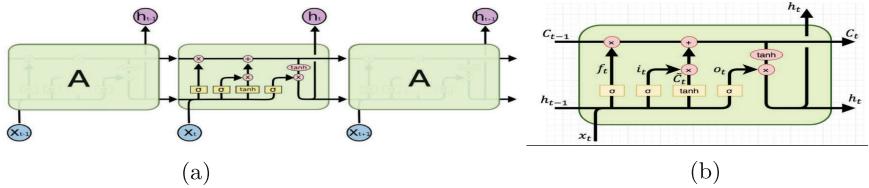


Fig. 2. (a) LSTM Model (b) LSTM Cell

3.2 Learning Model: LSTM for Stress Recognition

This section presents the LSTM model which is used as the learning model in our general framework. LSTM was introduced in [10] and is a variant of Recurrent Neural Network (RNN) to deal with the vanishing gradient problem of RNN, which uses a forget gate in each cell to prevent the vanishing gradient. Figure 2a presents an architecture of common LSTM composed of many different cells, where the input of each cell is a signal $x_{t_{se}}$ was segmented between start time t_s and end time t_e from raw signal x , thus the length of input segmented signal was produced is N or $t_{se} \in [1, N]$. The key idea of LSTM is to transport these segmented signals through the cells. Therefore, the function of LSTM removes or adds the new input signal to the cell as shown in Fig. 2b using the forget gate was defined as following Eq. (5):

$$f_{t_{se}} = \text{sigmoid}(\theta_{xf}x_{t_{se}} + \theta_{hf}h_{t_{se}-1} + b_f) \quad (5)$$

and it outputs a number between 0 (being forgotten) and 1 (being kept) for each number in the cell state C_{t-1} . The next step is to decide what new information is going to be stored in the cell state using the input gate layer. This can be done with the following Eqs. (6) and (7):

$$i_{t_{se}} = \text{sigmoid}(\theta_{xi}x_{t_{se}} + \theta_{hi}h_{t_{se}-1} + b_i) \quad (6)$$

$$\tilde{C}_{t_{se}} = \tanh(\theta_{xg}x_{t_{se}} + \theta_{hg}h_{t_{se}-1} + b_g) \quad (7)$$

In the next step, these two values will be combined to create an update to the cell state using the following Eq. (8):

$$C_{t_{se}} = f_{t_{se}} * C_{t_{se}-1} + i_{t_{se}} * \tilde{C}_{t_{se}} \quad (8)$$

Afterward, the output gate decides the part of the cell needed as the output in the form of a new hidden state using the following Eqs. (9) and (10) :

$$o_{t_{se}} = \text{sigmoid}(\theta_{xo}x_{t_{se}} + \theta_{ho}h_{t_{se}-1} + b_o) \quad (9)$$

$$h_{t_{se}} = o_{t_{se}} * \tanh(C_{t_{se}}) \quad (10)$$

where $\theta_{xf}, \theta_{xi}, \theta_{xg}, \theta_{xo} \in \mathbf{R}^{n_I \times n_H}$ and $\theta_{hf}, \theta_{hi}, \theta_{hg}, \theta_{ho} \in \mathbf{R}^{n_H \times n_H}$ are weighted matrix parameters. The dimension n_I is the size of the input vector, and n_H is the size of the hidden vector. b_f, b_i, b_g, b_o are bias vectors, $*$ is the element-wise multiplication. $f_{t_{se}}, i_{t_{se}}, C_{t_{se}}, \tilde{C}_{t_{se}}, o_{t_{se}}$ are the forget gate, input gate, cell state, new cell state, and output gate, respectively. When each input vector feeds into the cell of LSTM, an output of hidden states $h_{t_{se}} = \text{LSTM_Cell}(x_{t_{se}}, h_{t_{se}-1})$ is computed which fully connected to an output layer with activation softmax to produce the probability of all possible labels Y . Because our task is binary classification, thus the dimension $n_Y = 2$ is the size of the labels. Next, predict labels of segmented signal $x_{t_{se}}$ at each time-sliced window t_{se} by maximizing the conditional probability $P(y_{t_{se}}|x_{t_{se}}) = \frac{\exp(\theta_o h_{t_{se}} + b_Y)}{\sum_l (\theta_o h_l + b_Y)}$, where $\theta_o \in \mathbf{R}^{n_H \times n_Y}$ is the weight matrix parameter at output layer and $b_Y \in \mathbf{R}^{1 \times n_Y}$ is the weighted class label vector or biased vector which is computed using samples and their labels in training dataset as following Eq. (11):

$$b_Y[y_i] = \frac{\# \text{ of sample has label } y_i}{\# \text{ of all samples}}, \quad \text{where } y_i \in [1, Y] \quad (11)$$

4 Experiments

4.1 Dataset and Preprocess

WESAD is a publicly accessible multimodal dataset for wearable stress and emotion detection [2]. The 15 subjects were asked to follow the guide to study three different affective states: baseline (sitting or standing to read a neutral magazine), amusement (watching funny videos), and stress (doing tasks to stimulate stress based on the Trier Social Stress Test). In this study, authors experimented with respiration only data, which was collected from RespiBAN. The data goes through two segment processes: resample and segment function. The resample method reduces the size of the dataset and noise which downsamples the respiration data rate from 700 Hz to 4 Hz. The down-sampled data is then loaded and stored into a Data Frame structure for the next segmentation. In this study, the data is segmented every 10 s (40 values) with step size equal to the fixed-size window to make sure no overlap happens between segments. For each segment, a single label is determined, by taking the mode of the labels within the window.

Table 1. LSTM result

Person	Precision	Recall	F1	Accuracy
S2	0.93	0.91	0.92	0.93
S3	0.95	0.94	0.94	0.95
S4	0.86	0.76	0.78	0.84
S5	0.99	0.98	0.99	0.99
S6	0.90	0.90	0.9	0.92
S7	0.89	0.85	0.87	0.90
S8	0.92	0.91	0.91	0.93
S9	0.92	0.92	0.92	0.94
S10	0.95	0.91	0.93	0.94
S11	0.88	0.90	0.89	0.91
S13	0.78	0.81	0.79	0.82
S14	0.90	0.93	0.91	0.92
S15	0.93	0.91	0.92	0.93
S16	0.98	0.95	0.96	0.97
S17	0.95	0.95	0.95	0.96
Mean	0.92	0.90	0.91	0.92

Table 2. LSTM-based RAL result

Person	Precision	Recall	F1	Accuracy
S2	0.93	0.91	0.92	0.93
S3	0.97	0.93	0.95	0.96
S4	0.84	0.80	0.81	0.85
S5	0.99	0.99	0.99	0.99
S6	0.90	0.90	0.9	0.92
S7	0.89	0.85	0.87	0.90
S8	0.92	0.91	0.91	0.93
S9	0.94	0.92	0.93	0.94
S10	0.95	0.91	0.93	0.94
S11	0.90	0.92	0.90	0.92
S13	0.79	0.83	0.80	0.82
S14	0.91	0.94	0.92	0.93
S15	0.93	0.91	0.92	0.93
S16	0.98	0.95	0.96	0.97
S17	0.95	0.95	0.95	0.96
Mean	0.92	0.91	0.91	0.93

Authors use the same labelling method for stress and non-stress as described in the original paper [2], which is to combine the two states *amusement* and *baseline* into the *non-stress* class and keep the *stress* labels intact. Afterward, the data has been split into 14 subjects out of 15 subjects to train, the remaining one has been used for the testing process is known as Leave-One-Subject-Out cross-validation.

4.2 Experimental Results

This section presents the two results of the baseline with the LSTM model and the LSTM with Reinforcement Active Learning as follows.

Baseline Result with LSTM: LSTM has been trained as presented in Sect. 3.2 with a learning rate of 0.001, and 128 hidden units using Adam optimizer, dropout 0.5, a random seed equal to 1 to obtain reproducible results, a batch size of 64, and the binary-entropy loss for a binary-class label classification task. Table 1 detailed the best performance result of our LSTM model for 15 subjects with training epochs from 1 to 16.

LSTM with Reinforcement Active Learning Result: This experiment has used the LSTM model with the parameters as described above, $\alpha = 0.1 \times 10^{-5}$ and $\gamma = 0.99$, and loop with $T = 5$ iterations to reinforce the LSTM model. In each iteration, K has been chosen from 1 to 5 to query the most uncertain

samples and their labels as mentioned in Sect. 3.1. Table 2 details the result of our proposal framework of reinforcement active learning (RAL) using the maximize Shannon Entropy strategy. From this result, in comparison with LSTM performance, it is evident that the performance of our proposal approach can enhance the accuracy and recall in several subjects (shown in bold).

5 Conclusion

In this study, the authors propose a framework known as the Deep Reinforcement Active Learning framework that is capable of segmenting time-series sensor signals into smaller, manageable segments in contrast to the traditional method of manually identifying and extracting the signal. The result demonstrates that the proposed framework could enhance LSTM, achieving an accuracy rate of 93%, higher than the original performance of the LSTM model by 1%. In addition, a benefit of using this framework is that not needing to collect additional data, which is known to be a very time-consuming and expensive task, to enhance the accuracy of the model.

Acknowledgment. This research is funded by the International University, VNU-HCM under grant number T2022-01-IT. This research is also supported by the central Interdisciplinary Laboratory in Electronics and Information Technology “AI and Cooperation Robot”, International University - Vietnam National University of Ho Chi Minh City. We would like to thank for supporting machines in experiments.

References

1. Can, Y.S., Arnrich, B., Ersoy, C.: Stress detection in daily life scenarios using smart phones and wearable sensors: a survey. *J. Biomed. Inf.* **92**, 103139 (2019)
2. Schmidt, P., Reiss, A., Dürichen, R., Marberger, C., Laerhoven, K.V.: Introducing wesad, a multimodal dataset for wearable stress and affect detection. In: Proceedings of the 2018 on International Conference on Multimodal Interaction, Boulder, CO, USA, 16–20 October 2018
3. Garg, P., Santhosh, J., Dengel, A., Ishimaru, S.: Stress detection by machine learning and wearable sensors. In: 26th International Conference on Intelligent User Interfaces (2021)
4. Sano, A., Picard, R.W.: Stress recognition using wearable sensors and mobile phones. In: 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013, Geneva, Switzerland, 2–5 September 2013
5. Settles, B.: Active learning literature survey. Technical Report (2010)
6. Bakker, B.: Reinforcement learning with long short-term memory. In: Advances in Neural Information Processing Systems 14, NIPS 2001 (2001)
7. Elhajj, F.A., Deriche, M., Khalid, N.: Heartbeat classification of arrhythmia using hybrid features extraction techniques. In: 20th International Multi-Conference on Systems, Signals & Devices, SSD 2023 (2023)
8. Dziezyc, M., Gjoreski, M., Kazienko, P., Saganowski, S., Gams, M.: Can we ditch feature engineering? End-to-end deep learning for affect recognition from physiological sensor data. *Sensors* **20**(22), 6535 (2020)

9. Shannon, C.E.: A mathematical theory of communication. *Bell Syst. Tech. J.* **27**(3), 379–423 (1948)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)

Big Data, IoT, and Cloud Computing



A Digital Auto-Plasticity Synapse for All-Digital Resonate-and-Fire Neurons with On-chip STDP Learning

Trung-Khanh Le^{ID}, Trong-Tu Bui^{ID}, and Duc-Hung Le^(✉)^{ID}

Faculty of Electronics and Telecommunications, The University of Science, Vietnam National University, Ho Chi Minh City, Vietnam
1dhung@hcmus.edu.vn
<http://www.fetel.hcmus.edu.vn/>

Abstract. Synapses are important components in a spiking neural network (SNN), which is an advanced and bio-physically plausible neural network. The Spike Timing-Dependent Plasticity (STDP) learning rule, whose target object is the synapse's plasticity, is a common training method for SNN. Most of the realization of synapses in hardware is usually defined in the amplitude of action potentials to work with leaky integrate-and-fire (LIF) neurons, which mainly focus on the voltage of signals and lack of neural dynamic properties. The all-digital resonate-and-fire (ADRAF) neuron was an alternative spiking neuron model that solved the weak features of LIF neurons. The existing methods of using synapses in digital hardware are not suitable for this new model due to the absence of dynamic characteristics, especially in on-chip learning. This study proposed a new definition of synaptic plasticity based on spike-time transmitting. A new learning method based on STDP was developed to apply to the new plasticity. As a result, a digital synapse that can modify its plasticity automatically depending on the activity of the spikes coming from the pre-synapse neurons was designed and simulated to prove the new learning method. The design was implemented on a 28-nm FPGA device without any multipliers or finite-state machines.

Keywords: synapse · resonate-and-fire · neuron · learning · on-chip · plasticity · STDP

1 Introduction

Synapses in Spiking Neural Networks (SNN) are the key components beside spiking neurons. Their plasticities are the target parameters of the most learning and training methods. The most popular method is the Spike Timing-Dependent Plasticity (STDP) learning rule [1]. In the rule, synaptic plasticity indicates the connection strength between two neurons. Then, the rule suggested a method to modify synaptic plasticity according to the timing between the pre- and post-spikes. STDP rule was considered to be an implementation of the Hebb learning

rule [2]. It allows to turn the rule into hardware much easier than the traditional back-propagation method, which is the standard training method in multi-layer perceptron (MLP). There were some publications of on-chip STDP learning such as [3] in 2009 used floating-gate devices, [4] in 2015 and [5] in 2023 implemented 6T SRAM cells and its peripherals as synapses, [6] in 2018 trained the memristor-based synapses, [7] in 2022 and [8] in 2023 designed with analog CMOS circuits. Most the designs were based on special devices or analog circuits. Implementing it on digital devices is usually under the types of memory arrays that store the pre-defined or random plasticity values requires, such as [9] in 2020 proposed an event-driven system, [10] in 2021 realized on-chip STDP units with Block RAMs of an FPGA, and [11] in 2022 used SRAM with a simplified linear STDP learning rule. Synaptic devices in these papers were not separate devices. They also require a lot of hardware resources to control the value selection and hold history spikes [12]. Furthermore, the target neurons of those synapses were integrate-and-fire (IF) and leaky integrate-and-fire (LIF) neurons that are popular spiking neuron models but lack neural dynamic properties. These types of neuron models primarily care about the level of input signals. As a consequence, synapses typically ignore the impact of spike latency and primarily rely on the amplitude of action potentials [13], especially in digital designs.

In recent years, an all-digital resonate-and-fire (ADRAF) neuron model was presented in [14]. The operation of the new model depends not only on the quantity of input spikes but also on the spike-time information. To fire an action potential, the ADRAF neuron compares the interval of the input spikes with its eigenperiod. The amplitude of input spikes does not have any effect because of the digital system. Therefore, the amplitude-based synapse and synaptic plasticity are not compatible with this kind of neuron model. It requires time-based definitions for synapse and plasticity. The concept of time-based spike propagation was discussed in [13] in 2006 and [15] in 2008, then turned into digital hardware as conduction delay of axons by [16] in 2012. In this study, we proposed a new definition of synaptic plasticity to work with ADRAF neurons. By combining this definition with the STDP rule, we developed a new rule to teach a time-based synapse on a chip. As a result, a digital design of an auto-plasticity synapse was synthesized. To present the research more deeply, we organized the content into the following sections: Sect. 1 is the introduction, Sect. 2.2 discusses the new definition of synaptic plasticity, Sect. 3 presents a digital auto-plasticity synapse and its implementation on a 28-nm FPGA device, and the final Sect. 4 is our conclusion.

2 Synaptic strength and plasticity

2.1 All-digital resonate-and-fire neuron model

Resonate-and-fire (RAF) neuron is not a new model, it was first presented in 2001 by Izhikevich [17]. In this model, a neuron will generate a spike in two cases:

- The duration between two input spikes matches the oscillation of the membrane's voltage. It is defined as a resonant event.
- The duration of the input spikes is too short and within the duty cycle of the above oscillation. It is defined as a coincidental event.

These features were considered the upgraded functions of the IF and LIF neurons. They help to explain the frequency preference of biology neurons. Although resonance is an important characteristic of a neuron, the model was almost forgotten in digital hardware realization until 2023. The publication [14] in 2023 turned the model into hardware by proposing a digital model and a simple general digital circuit without using complex designs (multipliers, floating-point units,...). The primary distinction between the original RAF model and the ADRAF model lies in the form of the subthreshold oscillation, which in the ADRAF model takes the form of a pulse train instead of a damped sinusoidal wave. Its operation is briefly shown in Fig. 1.

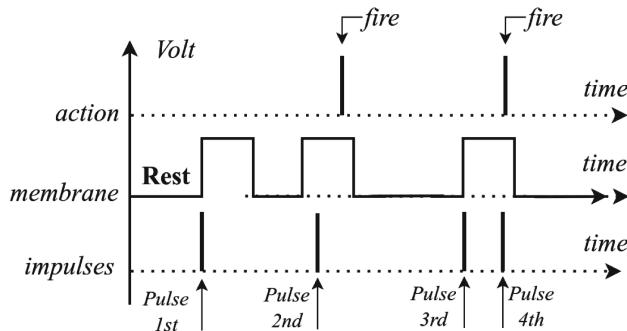


Fig. 1. Working principle of the ADRAF neuron model.

2.2 Amplitude-Based Synaptic Plasticity

According to the Hodgkin classification [18], there are neurons (class 1) that fire spikes depending on the strength of the input signal; there is another type (class 2) that fires spikes depending on the interval of the input signal, regardless of the signal strength. IF and LIF neurons usually work as class 1 neurons. Their action potentials depend on the amplitude of the input spikes. Therefore, synaptic plasticity for this class is defined as the change in amplitude from pre-spike to post-spike. This amplitude modification is named synaptic weight and also mentioned as synaptic strength. In digital systems, it is usually described as follows:

$$i_{post} = w \times i_{pre} \quad (1)$$

Where i_{pre} and i_{post} are pre- and post-synapse spikes, w is synaptic weight. The higher the w is, the stronger the synaptic connection is. Equation (1) has two popular models:

- Current-based model: synaptic weight converts input spikes in the form of voltage into the form of current.
- Conductance-based model: the model is the same as the current-based model, but it also counts the balance potential of a synapse.

However, (1) still lacks spike-time information between the first and the second spike on the same synapse. For example, how can a LIF neuron recognize fast spikes whose intervals are different but cause the same average level after synapses?

2.3 Time-Based Synaptic Plasticity

While class 1 neurons are sensitive to spike amplitude, class 2 neurons are sensitive to spike interval. An ADRAF neuron, which fires an action potential dependent on spike interval, primarily belongs to class 2 neuron. If the neuron has many dendrites on its input side, it can fire an action potential easily because of a coincidental event. Therefore, each dendrite of the neuron should have a synapse. However, in both coincidence and resonance, the neuron measures the time between a pair of spikes. In these cases, amplitude-based synapses have no effect. Thus, a synapse for this kind of neuron should modify the transmission time of a spike instead of amplitude. The time relation between the pre-spikes and post-spikes over a synapse can be described as follows:

$$t_{postspike} = t_{prespike} + \Delta t \quad (2)$$

With $t_{postspike}$ is the occurring moment of the spike after a synapse from $t_{prespike}$, which is the moment of the incoming spike to the synapse, and Δt is the delay time due to connection strength:

- If the $\Delta t \mapsto 0$, $t_{postspike} \mapsto t_{prespike}$. It means the neurons are totally connected.
- If the $\Delta t \mapsto \infty$, $t_{postspike} \mapsto \infty$. It means the neurons are totally unconnected.

So, let define synaptic plasticity $w_{synapse} = \Delta t$, the strength of a synapse can be calculated as follows:

$$s = \left| \frac{1}{1 + w_{synapse}} \right| \quad (3)$$

Where s is the synapse's strength, its value should be in the range of $(0, 1]$. When $w_{synapse} = 0$, $s = 1$ indicates the fully connection between two neurons. When $w_{synapse} \mapsto \infty$, the strength $s \mapsto 0$ means the neural link does not exist. So, varying the value of $w_{synapse}$ in the range of $[0, \infty]$ modifies the link of neurons. Figure 2 visualizes (3) to demonstrate the relationship more clearly.

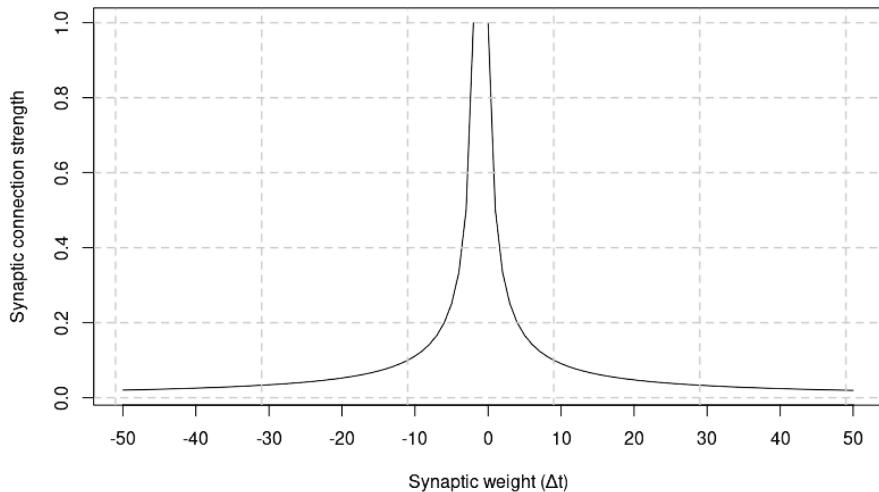


Fig. 2. Variation of synaptic connection strength according to spike-time delay.

According to the curves in Fig. 2, the variation of this time-based synaptic plasticity has an opposite direction from the amplitude-based one. The lower the $w_{synapse}$ is, the stronger the synaptic connection is.

3 Digital Auto-Plasticity Synapse

3.1 Digital Synapse

In Sect. 2.3, we proposed that a synapse for ADRAF neurons could play a role of time delay. To have the delay function, we implemented a counter with comparison to design the synapse. The general design of the synapse device is shown in Fig. 3.

The digital synapse in Fig. 3 has an n-bit Binary Counter to work as a time delay. The time unit of the delay comes from the ‘High speed clock’. Pre-spikes come into the ‘iSig’ to activate the ‘enCount’. This allows the Binary Counter to postpone sending out spikes to the ‘oSig’. The delay time is modified by the value in the ‘n-bit plasticity’. During the duration of time counting, any new pre-spikes do not have any effect on the operation of the synapse.

When the value of the counter reaches the plasticity’s value, the ‘match’ signal activates the D flip-flops ‘Q1’ to send out a high state to the ‘oSig’. The ‘Q2’ plays the role of converting that state to spike by resetting the whole device. The multiplexer ‘u10’ will bypass the delay if the value of the plasticity is 0. The neural connection is strongest in this case.

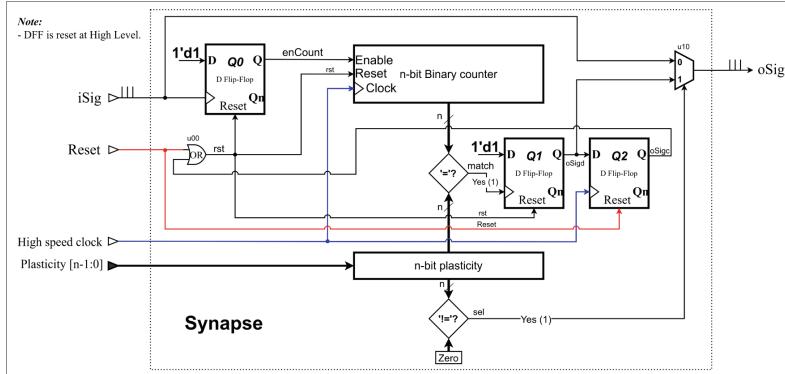


Fig. 3. General design of the digital synapse.

3.2 Auto-Plasticity Controller

According to the guidance mechanisms [19], the neural links are formed when there are neural activities. These activities may be caused by the actions of pre-synapse neurons. By combining with the STDP learning rule, we suggest a Spike-Count-and-Lock (SCL) rule as follows:

1. If there are many pre-spikes, the synapse becomes stronger.
2. If there are both pre-spikes and post-spikes, the synaptic plasticity falls into a locked state.

From the proposed SCL rule, an ‘Auto Plasticity’ controller was added in parallel with the suggested digital synapse as in Fig. 4.

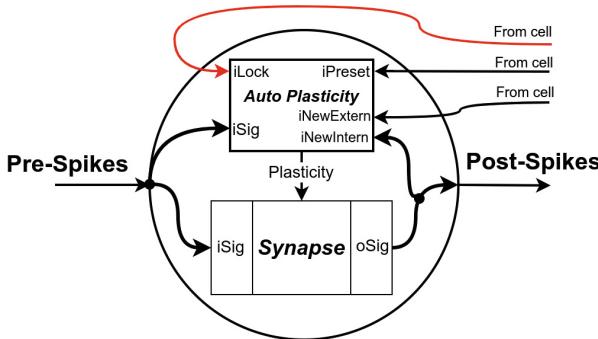


Fig. 4. General structure of the Digital Auto-Plasticity Synapse.

The ‘Auto Plasticity’ device in Fig. 4 uses the signals ‘iSig’ and ‘iLock’ to update the plasticity value using the SCL rule:

- Spike Counting:* if there is a pre-spike on the ‘iSig’, the device decreases the plasticity value to strengthen the synapse strength. The more pre-spikes that come to the synapse, the more quickly the synapse generates post-spikes. This mechanism is similar to the growing of axons in a brain [19].
- Locking:* the ‘iLock’ signal is the spike taken from the action potential of a post-synapse nerve cell. If there are action potentials on the ‘iLock’, the plasticity does not change its value to hold the current strength of the neural link. We can conclude that the synapse’s strength is locked. This mechanism is similar to the STDP rule.

To clear the captured spikes after a training interval, the controller needs two signals, ‘iNewExtern’ and ‘iNewIntern’, to be aware of that. An implementation of the SCL rule into the Auto-Plasticity device is shown in Fig. 5.

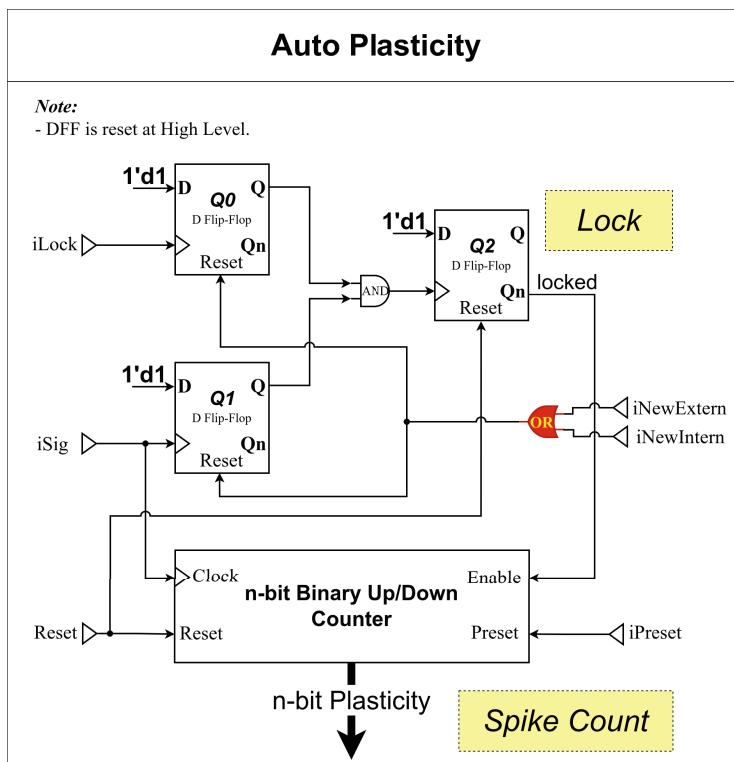


Fig. 5. A digital Auto-Plasticity (DAP) device.

The n-bit Binary Up/Down Counter in the implementation decreases its value on each edge of the ‘iSig’. The edge is stored temporarily in the D flip-flop ‘Q1’. The ‘Q0’ captures the response from a post-synapse neuron cell. The AND

gate checks the existence of the pre-spike and post-spike to activate the ‘locked’ signal, which enables or disables the counter.

The ‘iNewExtern’ signal comes from the post-synapse neuron. It indicates that the neuron returns to its resting state, and its membrane’s subthreshold oscillation stops. The ‘iNewIntern’ signal comes from the output spike of the synapse to avoid the case that the post-synapse neuron is dead or inactive. If there is any signal on the two signals, ‘Q0’ and ‘Q1’ should be cleared to be ready for the new activation from the pre-synapse neuron. The ‘iPreset’ signal sets the initial value for the counter. By combining the digital synapse device with the auto-plasticity device in the above structure, we have a DAP synapse that can do on-chip STDP learning.

3.3 Resource Utilization

To evaluate the resource utilization, we synthesized one 16-bit DAP synapse on a 28-nm FPGA Cyclone V device. The Register-transfer level (RTL) of the ‘Auto-Plasticity’ device is exposed in Fig. 6. The RTL design on a FPGA requires 1-bit adders to build binary counters. This issue can be solved in VLSI design by using JK flip-flop. The synthesis results are shown in Table 1.

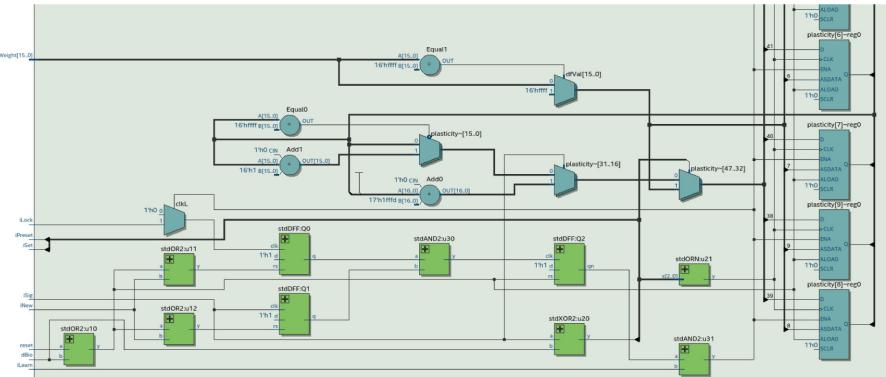


Fig. 6. RTL design of Auto-Plasticity devices.

Table 1 shows that one 16-bit DAP synapses require 0.17% of total adaptive logic modules (ALMs) and 0.02% of total registers. No multiplier is required because synaptic plasticity does not multiply with input spikes as amplitude-based synapses. To evaluate the predictability of resource utilization, we increased the number of synthesized synapses to 36. We selected this number to fully utilize all of our development kit’s input-output pins, and the results are shown in Table 2.

Table 1. Summary of the resource utilization of one 16-bit DAP synapse on the DE10-nano development kit.

Resource	Available	Used	Utilization
Logic (in ALMs)	41,910	73	0.17%
Registers	166,036	38	0.02%
Block Memory bits	5,662,720	0	0%
DSP blocks	112	0	0%
18 x 18 Multiplier	224	0	0%
PLL	6	0	0%
DLL	4	0	0%

Target device is Cyclone V 5CSEBA6U23I7. (110K logic elements).

Table 2 shows that 36 16-bit DAP synapses require 5.5% of total ALMs and 0.83% of total registers. That means the increase in resources is almost linear. So, with the capability of the Cyclone V 5CSEBA6U23I7, we can synthesize about 648 16-bit DAP synapses to implement a visual recognition system. For example, a face recognition system typically employs a 24x24 pixel bounding box, resulting in 576 input synapses and the remaining number for internal layers.

Table 2. Summary of the resource utilization of 36 16-bit DAP synapses on the DE10-nano development kit.

Resources	Total	Used	Utilization
Logic Modules	41,910	2287	5.5%
Registers	166,036	1385	0.83%
Memory bits	5,662,720	0	0%
DSP blocks	112	0	0%
Multipliers	224	0	0%
PLLs	6	0	0%
DLLs	4	0	0%

Target device is Cyclone V 5CSEBA6U23I7. (110K logic elements).

Table 3 expresses how our SCL rule and the designed DAP synapses are much simpler than in the other publications. We did not need to use floating-point units, multipliers, or FSMs to turn the STDP learning rule into on-chip learning. Thanks to the locking mechanism, the designed DAP synapse with the SCL rule requires two buffers to work as counters, while the other digital methods require at least three buffers: weight values in a memory array, spike history, and address buffer.

Table 3. Comparison of the resource type complexity for on-chip synapse training.

Resource type	This study 2024 SCL	This study 2015 STDP	[4] 2015 STDP	[9] 2020 STDP	[10] 2021 STDP	[11] 2022 STDP
Counters/shifters	☒	☒	☒	☒	☒	☒
Multipliers	□	☒	□	☒	□	□
Floating-point units	□	□	□	□	□	□
Finite state machine (FSM)	□	□	☒	☒	☒	☒

3.4 Experiments

To verify the proposed operation, two designed 16-bit DAP synapses were connected to a 16-bit ADRAF neuron and simulated in Verilog. Figures 7 and 8 demonstrated the simulation results.

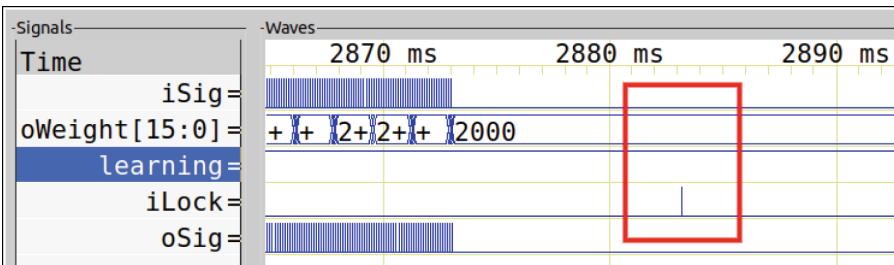


Fig. 7. An unlocked synapse due to the mismatched time between pre-spikes and action potential from its post-synapse neuron.

Figure 7 shows the spikes and learning state when a synapse is still unlocked. The initial plasticity of the synapse is 2000. Before 2880 ms, there were many pre-spikes sent to the ‘iSig’ of the synapse. The synaptic weight, the ‘oWeight’, decreased continuously. However, the post-synapse neuron did not reply to these spikes, so there was no lock event on the ‘learning’ signal, which turns to low if the synapse is locked. After 2880 ms, an action potential returned from the post-synapse neuron to the ‘iLock’ path, but there was no pre-spike. The synapse understood that the reply was not for it. Therefore, the synapse was still unlocked.

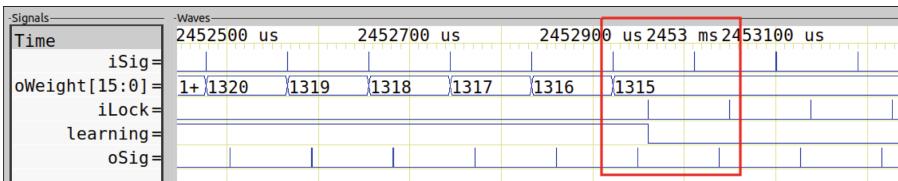


Fig. 8. A DAP synapse was locked by the SCL rule.

The operation of the locked synapse is illustrated in Fig. 8. It depicts two stages of the SCL rule:

- Before the moment of 2453 ms: the signal ‘iSig’ received spikes from the pre-synapse neuron. Each post-spike on the signal ‘oSig’ was postponed from each spike on the ‘iSig’ due to the 16-bit plasticity ‘oWeight’. The ‘learning’ signal was in the *High* state to indicate that the synapse was not locked. There was no action potential on the ‘iLock’ signal. The plasticity value continuously decreased from *2000* to *1315* on every pre-spike to the ‘iSig’ path.
- After the 2453 ms moment: action potentials returned from the post-synapse neuron to the ‘iLock’. Because of this event, the synapse locked its strength, and then the plasticity ‘oWeight’ was fixed at the value *1315*. Through this synapse, the time delay between the pre-spike and post-spike was constant.

So, Figs. 7 and 8 show that the DAP synapses and SCL rule worked as expected.

4 Conclusion

According to this research, a new synaptic plasticity was proposed to define a type of synapse that is compatible with ADRAF neurons. Based on these definitions, we constructed a fully digital design and optimally synthesized 16-bit DAP synapses on a 28-nm FPGA. The auto-plasticity function is the most advantageous feature of the design. On-chip learning or training can be implemented effectively because of this property. With the help of DAP synapses, SNNs based on ADRAF neurons can be trained on-chip with the STDP learning method. Beside that, these SNNs can also run the parallel processes of learning and perceiving. Furthermore, the digital design of the DAP synapse allows for the realization of online on-chip learning SNNs into standard digital IC design flows without the need for mixed-signal designs or special materials. With all those benefits, intelligent systems will be integrated into small chips easily.

References

1. Song, S., Miller, K.D., Abbott, L.F.: Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nat. Neurosci.* **3**(9), 919–926 (2000). <https://doi.org/10.1038/78829>
2. Caporale, N., Dan, Y.: Spike timing-dependent plasticity: a Hebbian learning rule. *Annu. Rev. Neurosci.* **31**(1), 25–46 (2008). <https://doi.org/10.1146/annurev.neuro.31.060407.125639>
3. Pankaala, M., Laiho, M., Hasler, P.: Compact floating-gate learning array with STDP. In: International Joint Conference on Neural Networks, pp. 2409–2415 (2009). <https://doi.org/10.1109/IJCNN.2009.5178879>
4. Seo, J., Seok, M.: Digital CMOS neuromorphic processor design featuring unsupervised online learning. In: 2015 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), Daejeon, Korea (South), pp. 49–51 (2015). <https://doi.org/10.1109/VLSI-SoC.2015.7314390>
5. Liu, S., Wang, J.J., Zhou, J.T., Hu, S.G., Yu, Q., Chen, T.P., Liu, Y.: An area-and energy-efficient spiking neural network with spike-time-dependent plasticity realized with SRAM processing-in-memory macro and on-chip unsupervised learning. *IEEE Trans. Biomed. Circuits Syst.* **17**(1), 92–104 (2023). <https://doi.org/10.1109/TBCAS.2023.3242413>
6. Shukla, A., Ganguly, U.: An on-chip trainable and the clock-less spiking neural network with 1R Memristive synapses. *IEEE Trans. Biomed. Circuits Syst.* **12**(4), 884–893 (2018). <https://doi.org/10.1109/TBCAS.2018.2831618>
7. Joo, B., Han, J.-W., Kong, B.-S.: Energy- and area-efficient CMOS synapse and neuron for spiking neural networks with STDP learning. *IEEE Trans. Circuits Syst. I Regul. Pap.* **69**(9), 3632–3642 (2022). <https://doi.org/10.1109/TCSI.2022.3178989>
8. Rubino, A., Cartiglia, M., Payvand, M., Indiveri, G.: Neuromorphic analog circuits for robust on-chip always-on learning in spiking neural networks. In: 2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS), pp. 1–5 (2023). <https://doi.org/10.1109/AICAS57966.2023.10168620>
9. Asgari, H., Maybodi, B.M.-N., Kreiser, R., Sandamirskaya, Y.: Digital multiplier-less spiking neural network architecture of reinforcement learning in a context-dependent task. *IEEE J. Emerg. Sel. Topics Circuits Syst.* **10**(4), 498–511 (2020). <https://doi.org/10.1109/JETCAS.2020.3031040>
10. Guo, W., Yantir, H.E., Fouda, M.E., Eltawil, A.M., Salama, K.N.: Toward the optimal design and FPGA implementation of spiking neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(8), 3988–4002 (2022). <https://doi.org/10.1109/TNNLS.2021.3055421>
11. Sun, C., et al.: An energy efficient STDP-based SNN architecture with on-chip learning. *IEEE Trans. Circuits Syst. I Regul. Pap.* **69**(12), 5147–5158 (2022). <https://doi.org/10.1109/TCSI.2022.3204645>
12. Valencia, D., Alimohammad, A.: A generalized hardware architecture for real-time spiking neural networks. *Neural Comput. Appl.* **35**(24), 17821–17835 (2023). <https://doi.org/10.1007/s00521-023-08650-6>
13. Izhikevich, E.M.: Polychronization: computation with spikes. *Neural Comput.* **18**(2), 245–282 (2006). <https://doi.org/10.1162/089976606775093882>
14. Le, T.-K., Bui, T.-T., Le, D.-H.: Modeling and designing of an all-digital resonate-and-fire neuron circuit. *IEEE Access* **11**, 62318–62336 (2023). <https://doi.org/10.1109/ACCESS.2023.3287994>

15. Wang, S.S.-H., et al.: Functional trade-offs in white matter axonal scaling. *J. Neurosci.* **28**(15), 4047–4056 (2008). <https://doi.org/10.1523/JNEUROSCI.5559-05.2008>
16. Belhadj, B., Joubert, A., Temam, O., Heliot, R.: Configurable conduction delay circuits for high spiking rates. In: IEEE International Symposium on Circuits and Systems, pp. 2091–2094 (2012). <https://doi.org/10.1109/ISCAS.2012.6271696>
17. Izhikevich, Eugene M.: Resonate-and-fire neurons. *Neural Netw.* **14**(6–7), 883–894 (2001). [https://doi.org/10.1016/s0893-6080\(01\)00078-8](https://doi.org/10.1016/s0893-6080(01)00078-8)
18. Hodgkin, A.L.: The local electric changes associated with repetitive action in a non medullated axon. *J. Physiol.* **107**(2), 165–181 (1948). <https://doi.org/10.1113/jphysiol.1948.sp004260>
19. Kandel, E.R., Schwartz, J.H., Jessell, T.M., Siegelbaum, S.A., Hudspeth, A.J.: The growth and guidance of axons. In: Principles of Neural Science, ch. 54, 5th edn. McGraw Hill Professional, New York, pp. 1218–1220 (2013)



NEURAHOLO: A System for High-Resolution 3D Human Digitization on Holograms

Hoang Pham Nguyen^(✉), Thai Anh Huynh Ngoc^(✉), Luat Le Gia, and Khoi Le Gia

College of Information and Communication Technology- Can Tho University, Can Tho, Vietnam
pnhoang@ctu.edu.vn

Abstract. Amid the growing prominence of Holographic Projection technology in Vietnam's technology trend, we introduce NEURAHOLO: an innovative system designed to convert 2D images into 3D models for holographic displays. The system compares two advanced image processing algorithms for background removal U²-net and Mediapipe Selfie Segmentation followed by PIFuHD for 3D model generation, and a 3D Led Holodisplay for visualization. Our evaluation on the Supervisely Person Dataset with over 1000 images resulted in an IoU score of 0.9309 for U²-net, outperforming Mediapipe Selfie Segmentation, which scored 0.8467. The Chamfer Distance was used to measure the accuracy of the 3D models, with an average error of just 0.986, thereby corroborating the system's adeptness at precise 3D model re-creation.

Keywords: 3D human digitization · background removal · U²-net

1 Introduction

The digital transformation era has witnessed a surge in demand for immersive and impactful visualization solutions, extending beyond traditional media and advertising to encompass television and online content broadcasting. In this landscape, Augmented Reality (AR) and Hologram projection technologies have emerged as frontrunners, demonstrating immense potential across diverse research fields and practical applications. AR integrates digital information into the real environment, fostering powerful and intuitive interactive experiences for users [1]. Holograms, on the other hand, enable the creation of lifelike 3D imagery viewable from multiple angles without the need for specialized glasses [2]. Due to the advantage of displaying 3D content simultaneously for multiple users without requiring additional personal devices, Holographic projection is preferred over VR glasses in community-oriented applications. Consequently, the research and development of hologram display content and new hologram techniques are trends.

Benton and Bove, in their seminal work “Holographic Imaging” (2008), described 21 types of holograms, delving into the fundamental principles of Holographic imaging, image creation techniques, and their practical applications [2]. Among these classifications, hologram models capable of changing display content through Phase conjugation and Real image projection or displaying 3D images via Holographic Television are

considered highly applicable due to their ability to create diverse content changes or integrate interactive content controls.

Over the past three years, Hologram technologies have undergone significant advancements in display technology, incorporating a diverse array of 3D projection techniques. These innovations have paved the way for novel methodologies in education, healthcare, and entertainment, exemplified by virtual teaching models and 3D medical imaging [3, 4].

The utilization of 3D avatars on Hologram platforms has gained paramount importance in the current context. 3D avatars deliver vivid and realistic imagery, facilitating remote communication and interaction, particularly amidst the global pandemic. A. Ghosh emphasized the potential of holography to generate detailed 3D medical images, empowering doctors and medical personnel with clearer insights into the internal structures of the human body [4]. Aligning with this development trend, Holograms have also found widespread application in the entertainment industry. Liang vividly described how online concerts could leverage Holograms to create an immersive experience for audiences by showcasing artists as lifelike 3D images [5]. In the realm of commerce, Holograms are employed to craft captivating and engaging 3D advertisements, effectively capturing consumer attention [6].

Despite the remarkable advancements and promising applications, Hologram-based technologies still face certain challenges that hinder their widespread adoption. A primary concern is the high cost associated with deploying Hologram systems. Sullivan's study delved into the financial implications of Hologram technology, highlighting that the implementation of current Hologram systems necessitates complex hardware and software, driving up investment and maintenance expenses [6]. Additionally, the visibility and interactivity of 3D images within Hologram systems require further refinement to ensure realism and enhance user engagement [7, 8]. Another critical factor demanding attention and optimization is the response time of Hologram systems to minimize disruptions and ensure seamless user experiences. Kim et al. conducted research and proposed methods to improve the response time of a Hologram system displaying a remotely controlled 3D character named Holobot, significantly enhancing user experiences by notably reducing system response time [9].

To effectively address the design requirements of a 3D avatar Hologram system, the system must incorporate the following fundamental components:

- *Holographic Display*: The core element of the system, responsible for generating 3D Holographic imagery. Holographic display types such as Spatial Light Modulators (SLMs) and surface meta-holograms are commonly employed to achieve high efficiency and resolution.
- *Capture Devices*: These devices meticulously gather the requisite 3D data for Hologram image generation. They may encompass multiple cameras, depth sensors, or specialized 3D scanning devices to capture the physical characteristics of objects from diverse angles.
- *Computational Units*: High-performance computers meticulously process the acquired data to generate the Hologram. These units execute intricate algorithms for real-time holography, ensuring smooth motion and accurate depth cues.

- *Telepresence and Control Interfaces*: For interactive applications, such as telepresence robots, the system incorporates control interfaces like exoskeletons for hand and arm movements, foot devices for mobility control, and Head-Mounted Displays (HMDs) to deliver immersive visual and auditory feedback.
- *Reaction Systems*: To augment interactivity, Hologram devices must provide users with diverse feedback modalities, ranging from visual and auditory feedback to simulated tactile sensations, enhancing the realism of the experience.

These components synergistically collaborate to create a complete system capable of generating and displaying realistic and interactive 3D holographic avatars. As advancements continue, the integration and optimization of these technologies will further enhance their applications across various fields, ensuring a more immersive and engaging user experience.

The objective of this research is to introduce and evaluate NEURAHOLO, a novel system designed for high-resolution 3D human digitization on holograms. The paper is organized as follows: Sect. 2 presents the system model and its components, detailing the background removal techniques and 3D model rendering process. Section 3 describes the experimental setup and results, including the performance evaluation of the background removal methods and the accuracy of the 3D human digitization. Section 4 discusses the findings, challenges, and future work in the integration of artificial intelligence with holographic projection systems.

2 Research Content

2.1 System Model

In this paper, we propose a novel system, NeuraHolo, for reconstructing and visualizing 3D human models from a single image or a video sequence. The system consists of two main stages: background removal and 3D model rendering. For the background removal stage, we employ a deep learning-based approach to accurately separate the foreground (human subject) from the background. The extracted foreground image is then fed into a 3D reconstruction model, PIFuHD [15], to generate a high-fidelity 3D model of the human subject. The rendered 3D model is subsequently displayed on a holographic device to enhance the user's visual experience. Figure 1 illustrates the overall pipeline of NeuraHolo.

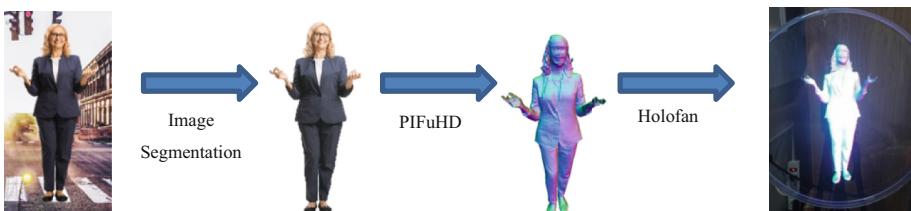


Fig. 1. Operation diagram of Neuraholo system

We evaluated two background removal methods using the MediaPipe Selfie Segmentation model [12] and the U²-Net model [13].

MediaPipe Selfie Segmentation is a symmetrical network architecture with global average pooling applied to both the encoder and decoder blocks, optimized for CPU execution [12]. The model is based on MobileNetV3 with FCN structures and is fine-tuned using Neural Architecture Search (NAS) to achieve optimal performance while minimizing resource consumption. The model takes an input tensor with dimensions $256 \times 256 \times 3$ (length, width, and color channels) and outputs a tensor with dimensions $256 \times 256 \times 1$ representing the segmentation mask. The input image is automatically resized to match the tensor before being fed into the corresponding MobileNetV3 model (Fig. 2).

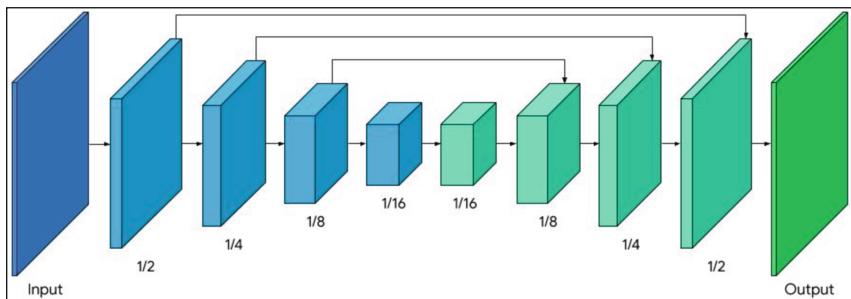


Fig. 2. Mediapipe Selfie Segmentation Model Architecture

In addition to the MediaPipe Selfie Segmentation model, we also evaluated another background removal method using the U²-Net model [13]. The U²-Net architecture is based on a U-Net structure with an attention mechanism and residual connections. The model takes an input tensor with dimensions $256 \times 256 \times 3$ and outputs a tensor with dimensions $256 \times 256 \times 1$ representing the segmentation mask. The input image is automatically resized to match the tensor before being fed into the U²-Net model. The U²-Net architecture consists of three main components represent in the Fig. 3:

- *Encoder*: The encoder consists of six stages, each corresponding to a different size of the RSU (Residual SU) blocks. The encoder's task is to extract features at multiple spatial resolutions from the input image.
- *Decoder*: The decoder consists of five stages with a symmetrical structure similar to the encoder. The decoder's task is to concatenate features extracted from the previous stage and its corresponding encoder stage.
- *Feature Map Fusion Module*: The feature map fusion module is responsible for generating saliency probability maps. These maps represent the probability of each pixel belonging to the foreground object in the image.

The U²-Net design allows for a deep architecture with rich multi-scale features and relatively low computational and memory costs. Additionally, since the U²-Net architecture is built on RSU blocks that do not use any pre-trained backbone transferred from

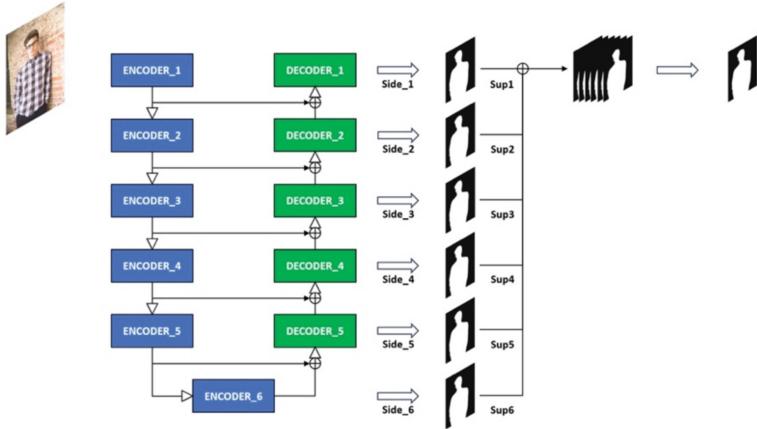


Fig. 3. U²-net Model Architecture

image classification, it is flexible and easily adaptable to different working environments with negligible performance loss.

2.2 3D Human Digitization

The goal of this work is to reconstruct a detailed 3D model from a single image with a human subject. The model should achieve high fidelity in areas such as fingers, facial features, and clothing folds. Previous methods have often failed to fully utilize the information in high-resolution images, focusing instead on global inference for mapping between the 2D image and the 3D model. This results in 3D models with insufficient detail. In this study, we propose a method that utilizes local features in high-resolution images to reconstruct a 3D model with high detail. Our method is an improved version of the recently introduced Pixel-Aligned Implicit Function (PIFu) method [14]. This method aligns and represents pixel points on the 3D model in a seamless manner based on coarse inferences and learned image features from the high-resolution input image in a principled manner. Each level is gradually integrated with the missing information from the previous coarse levels. Then, the geometric information of the front surface is synthesized at the highest level. Once the front surface information is available, the next step is to reconstruct the back surface, which is the missing information in the image. This issue is addressed by utilizing an image-to-image translation network to construct the back surface. Our multi-level pixel-based inference and alignment method eliminates ambiguity and significantly improves the quality of the reconstruction with more consistent detail between visible and occluded regions.

The Pixel-Aligned Implicit Function (PIFu) model aims to reconstruct a 3D object from a single 2D image. To achieve this, it utilizes an implicit function $f(X)$ to predict the binary occupancy value for any 3D position within the continuous camera space. The model is trained in an end-to-end manner using a neural network architecture.

The function $f(X)$ takes a 3D position $X = (x, y, z)$ as input and outputs a binary occupancy value $P(X) \in \{0, 1\}$. This value represents whether the corresponding 3D point is inside ($P(X) = 1$) or outside ($P(X) = 0$) the object.

The Pixel-Aligned Implicit Function (PIFu) model is trained in an end-to-end manner using a neural network architecture to model the function $f(X)$ that predicts the binary occupancy value for any 3D position within the continuous camera space. The training process involves extracting features from the input image and the corresponding 2D projected position, and then estimating the occupancy value for the given 3D query point:

$$f(X, I) = g(\Phi(x, I), Z) \quad (1)$$

where $Z = Z(x)$ is the depth along the projection ray determined by the projection position. The function $g()$ will focus on the depth Z to distinguish the occupancy of 3D points along the projection ray. Function Φ uses convolutional neural network (CNN) architecture to extract 2D image features and function $g()$ uses multilayer Perception architecture - MLP (Multilayer Perceptron).

The original Pixel-Aligned Implicit Function (PIFu) model has limitations in terms of representation expressiveness due to the constraints of feature resolution. Its performance is optimal for input image resolutions of 512×512 and 128×128 , but it struggles with higher resolutions due to hardware memory limitations. To address this issue, we use PIFuHD (Multi-Level Pixel-Aligned Implicit Function), an enhanced version of PIFu that can effectively utilize 1024×1024 input images to generate more detailed and realistic 3D reconstructions.

The PIFuHD architecture consists of two levels of PIFu modules, as illustrated in Fig. 4:

- **Coarse Level:** This level employs the original PIFu model, taking a 512×512 input image and extracting backbone features with a resolution of 128×128 .
- **Fine Level:** This level focuses on refining and adding detailed features such as facial features, fingers, and clothing folds. It utilizes a 1024×1024 input image and generates backbone features with a resolution of 512×512 . Additionally, it incorporates embedded features extracted from the coarse level instead of using absolute depth values.

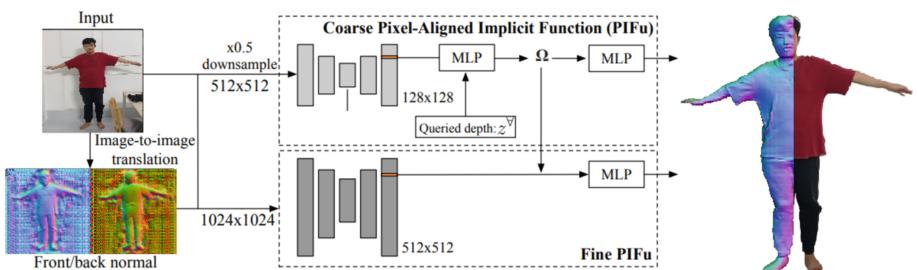


Fig. 4. PIFuHD Model Architecture

In addition to the standard PIFu module, the coarse level in PIFuHD also utilizes the predicted front/back mask to refine the occupancy estimation. This is achieved through the following formula:

$$f^L(\mathbf{X}) = g^L(\Phi^L(x_L, I_L, F_L, B_L), Z) \quad (2)$$

where I_L is the lower resolution input, F_L and B_L are the predicted normalized maps at the same resolution. $x_L \in \mathbb{R}^2$ is the projected 2D position of \mathbf{X} in the image space of I_L .

Fine level processing:

$$f^H(\mathbf{X}) = g^H(\Phi^H(x_H, I_H, F_H, B_H), \Omega(X)) \quad (3)$$

In which I_L, F_L, B_L are the input images, front and back normal map at 1024×1024 resolution. The 2D projection location at high resolution is denote by $x_H \in \mathbb{R}^2$, in this case $x_H = 2x_L$. The function Φ^H encodes images features from high-resolution input and has a similar structure to the low-resolution feature extractor Φ^L .

The main different is that Φ^H does not cover the entire images. However, thanks to the fully convolutional architecture, the network can be trained with a random sliding window and still infer at the original image resolution (i.e. 1024×1024). Finally, $\Omega(X)$ is a 3D embedding extracted from the coarse level network, where we take the output features feom an intermediate layer of g^L .

The fine level processing in PIFuHD plays a crucial role in enabling the model to handle high-resolution input images and generate detailed 3D reconstructions. By utilizing features from the coarse level, processing image patches, and avoiding global depth normalization, PIFuHD overcomes memory limitations and achieves improved reconstruction quality and detail. This makes PIFuHD a valuable tool for creating high-fidelity 3D reconstructions from high-resolution images.

2.3 Backface Prediction

Predicting the accurate shape of the back of a person is an ill-posed problem because it is not directly observed in the image. Therefore, the back must be fully inferred by the MLP prediction network, and due to the ambiguous and multimodal nature of this problem, the 3D reconstruction tends to be smooth and lacking in features.

Instead, if part of this inference problem is shifted to the feature extraction stage, the network can generate a sharper reconstruction shape. To achieve this, we predict normal maps (normal maps) instead of 3D geometry in the image space and provide these normal maps as features for the pixel-aligned prediction (PIFu) modules. Then, the 3D reconstruction is guided by these maps to infer a specific 3D geometry, making it easier for MLPs to generate more details. We predict back and front normals in image space using the pix2pixHD network, which maps from RGB color to normal maps. We found that this produces reasonable outputs for unseen backfaces for limited problem domains such as the subject's back clothing.

2.4 Holographic Projection

After completing the 3D model reconstruction using the PIFuHD model, we extract a front-facing view image for display on the hologram device. Specifically, in this study, we use Holofan (3D Led Fan Hologram).

The working principle of fan holograms is based on the persistence of vision (POV) system. The characteristic of the human eye is that it cannot perceive motion that occurs in less than 50 ms. Therefore, if a point in space flashes and repeats at a frequency of 20 Hz (20 times per second), the human eye will not see this flashing. Based on the above POV principle, LED strips are attached to the blades of the hologram screen and will flash when rotating at high speed to create a 3D floating image effect in the air.

The input data is an image with a width of w pixels and a height of h pixels. The hologram display device will display the image as a circle with radius L_{max} , where L_{max} is the number of LEDs on a fan blade. This circle will be divided into n angular regions called “time slots” because the angle of each region corresponds exactly to the rotation time from the reference point to that region. The angle between two adjacent time slots α will be calculated using the formula $\alpha = (2 * \pi)/n$. In the radial direction, each time slot will be divided into L_{max} subregions corresponding to the L_{max} LEDs. Finally, the input pixel matrix will be converted to points on the corresponding time slots using the following formula:

$$P_{x,i} = L_j * \cos(\alpha_i) + c_x \quad (4)$$

$$P_{y,i} = L_j * \sin(\alpha_i) + c_y \quad (5)$$

$$V_{i,j} = A[I_x + w * (h - 1 - I_y)] \quad (6)$$

$$I_x = \frac{P_{x,i} * w}{2 * L_{max}} \quad (7)$$

$$I_y = \frac{P_{y,i} * h}{2 * L_{max}} \quad (8)$$

where, P_i is the pixel coordinate of the slot i , $I(x,y)$ is the coordinate of the pixel on the input image calculated by P_i , $V(i,j)$ is the display value of the slot i at the position of the LED j . The process of converting pixels on the input image matrix to coordinates of points to display is shown in Fig. 5.

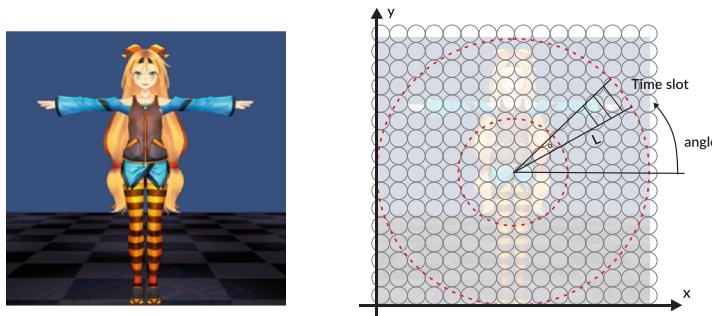


Fig. 5. The process of converting from pixel matrix to display points on time slot

Using the method described above, we have successfully converted an input image in pixel matrix format to a format suitable for the holofan device. The results will be presented in the next section.

3 Experimental Results

Our experiments were carried out on a personal computer system equipped with an AMD Ryzen R5-4600H CPU, an NVIDIA GTX 1650Ti GPU, 8 GB DDR4 RAM, CUDA 10.8, Python 3.10.8 and Windows 11 operating system.

3.1 Evaluation of Background Removal Stage

We evaluated the performance of the background matting process on the Supervisely Person Dataset, which consists of 2,667 images of people and corresponding ground truth masks. We calculated the Intersection over Union (IoU) metric between the prediction masks of the U²-Net and Mediapipe Selfie Segmentation models and the ground truth masks for each image. We then calculated the average IoU for both methods. Figure 6 shows some sample results of this testing process. After testing, we found that the average IoU for the U²-Net model was 0.9309, which is higher than the average IoU for the Mediapipe Selfie Segmentation model of 0.8467. Therefore, we will choose to use the U²-Net model in our NeuraHolo system.



Fig. 6. Qualitative comparision of the Mediapipe Selfie Segmentation and U²-net.

3.2 Evaluation of 3D Human Digitization

To measure the accuracy of 3D models, we will use Chamfer distance as it is a standard method in 3D model reconstruction and computer vision evaluation. This method allows for fair comparisons between different models, providing a quantitative assessment of performance. Chamfer distance will evaluate the similarity between the reconstructed

surface and the ground truth surface by calculating the average distance between points in one set to the nearest point in the other set. In addition, Chamfer distance is relatively simple and efficient to compute, making it particularly suitable for data structures such as K-D trees. Chamfer distance is also symmetric when calculating the distance from both directions from points in the reconstructed surface to points in the ground truth surface and vice versa. This can detect errors in both directions and help provide a better overall accuracy assessment.

To calculate the Chamfer distance between two sets of points (point clouds), we need to know the average distance between points in one set to the nearest point in the other set, and vice versa. The calculation process is as follows:

- *Point Set Definition:* Let P and Q represent two point sets corresponding to the reconstructed model from PIFuHD and the ground truth model, respectively.
- *Nearest Point Distance Calculation:*
 - For each point p in set P , find the corresponding point q in set Q and calculate the distance $d(p,q)$.
 - For each point q in set Q , find the corresponding point p in set P and calculate the distance $d(q,p)$.
- *Chamfer Distance Calculation:* The Chamfer distance is the average of these nearest point distances. Specifically, it is represented by the following formula:

$$CD(P, Q) = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \|p - q\|^2 + \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P} \|q - p\|^2 \quad (9)$$

where:

$CD(P,Q)$ is Chamfer distance of two set P, Q .

$\|p - q\|$ is the Euclidean distance between p and q .

- *Average Value Calculation:* After obtaining the Chamfer distance between two sets, we repeat the above steps multiple times and take the average of the results.

We evaluated the performance of our 3D reconstruction pipeline using 20 3D models from the Render People dataset. The Chamfer distance was used to calculate the point-to-surface distance. We compared the 3D reconstruction results for two cases: images with and without background matting. 50,000 random points were sampled on the surface using a combination of uniform sampling and importance sampling. The sampling and Chamfer distance calculation were performed 1000 times to obtain an average Chamfer distance metric of 0.986 for the background matted dataset and 1.105 for the non-background matted dataset. The units for both the point-to-surface distance and the Chamfer distance were centimeters (cm). The background matted dataset achieved better results because it can handle images with complex backgrounds or watermarks. An illustrative example of this error is shown in Fig. 7 below.

The Neuraholo system performed well in most cases where the input image contained complete information about the subject. However, the system still has limitations in cases where the subjects are holding objects or overlaps. Some results of the stages in the pipeline are shown in Fig. 8 below.

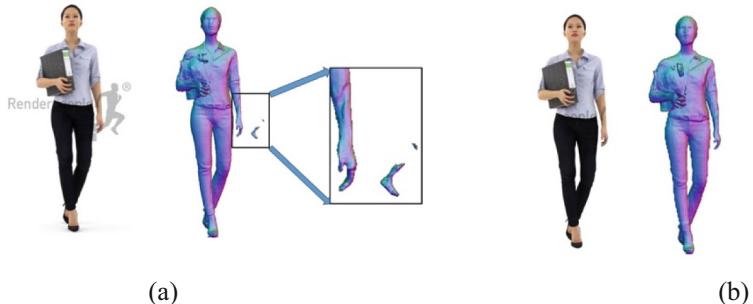


Fig. 7. Comparision of result in two case: (a) 3D reconstruction artifacts caused by watermarking interference, (b) Successful 3D reconstruction using background removal

Input image	Remove Background	3D Model	Holographic projection

Fig. 8. Some samples results of NeuraHolo system

4 Discussion and Future Work

In this study, we have developed an application that takes a single 2D image of a person and reconstructs a 3D model of that person, which can then be displayed on a holographic projection device. The reconstructed 3D human models achieve a relatively good level of detail from a single 2D input image. The average processing time of the system is relatively fast. However, the system is not yet able to achieve real-time processing due to hardware limitations of the processing device and the holographic display device. Future work could explore triangle-based hole filling to address incomplete 3D reconstructions. We propose a hybrid approach combining Ear Clipping and Constrained Delaunay Triangulation to ensure structural integrity..The next phase of this research will focus on evaluating the 3D reconstruction performance of the proposed method compared to alternative approaches, including STAR (A Sparse Trained Articulated Human Body Regressor), VIBE (Video Inference for Body Pose and Shape Estimation), and FLAME (Faces Learned with an Articulated Model and Expressions). The integration of artificial intelligence with holographic projection systems holds great potential for innovation in various fields such as education, training, medicine, simulation design, and more. While there are still many challenges to be addressed, artificial intelligence and holographic displays promise to revolutionize the way we interact with information and the world around us.

References

1. Azuma, R.T.: A survey of augmented reality. *Presence Teleoper. Virtual Environ.* **6**(4), 355–385 (1997)
2. Benton, S.A., Bove, V.M., Jr.: *Holographic Imaging*. John Wiley & Sons (2008)
3. Furht, B.: *Handbook of Augmented Reality*. Springer Science & Business Media, New York (2011). <https://doi.org/10.1007/978-1-4614-0064-6>
4. Ghosh, A.: Significance of holographic technology in modern world. In: National Conference on Computational Technologies, vol. 5, pp. 1–2. University of North Bengal, Siliguri, India (2017)
5. Liang: The application of the holographic laser projection in the entertaining performance. In: 2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE), pp. 1–2. IEEE (2016)
6. Tang, T., Zhang, H.: An interactive holographic multimedia technology and its application in the preservation and dissemination of intangible cultural heritage. *Int. J. Digit. Multimed. Broadcast.* **2023**, 1–13 (2023)
7. Chu, D., Park, J.-H., Ferraro, P., Cheng, C.-J., Stoykova, E., Banerjee, P.: Digital holography and 3D imaging: introduction to the joint feature issue in Applied Optics and Journal of the Optical Society of America A. *Appl. Opt.* **62**, DH1 (2023)
8. Torres-Leal, F., Moreno-Rodriguez, H., Rubio-Perez, J., Hernandez-Aranda, R., Rosales-Guzmán, C., Perez-Garcia, B.: Low-cost printed holography for the generation of structured light. *Appl. Opt.* **62**, 7104 (2023)
9. Kim, J., et al.: Holobot: hologram based extended reality telepresence robot. In: 2023 ACM/IEEE International Conference on Human-Robot Interaction, pp. 60–64 (2023)
10. Maimone, A., Georgiou, A., Kollin, J.S.: Holographic near-eye displays for virtual and augmented reality. *ACM Trans. Graph. (Tog)* **36**(4), 1–16 (2017)

11. Bimber, O., Raskar, R.: Spatial Augmented Reality: Merging Real and Virtual Worlds. Available: <https://pages.cs.wisc.edu/~dyer/cs534/papers/SAR.pdf>
12. T. Hou, S. Pisarchyk and K. Raveendran. MediaPipe Selfie Segmentation (2021). [Online]. Available: https://developers.google.com/mediapipe/solutions/vision/image_segmenter
13. Qin, X., et al.: U2-Net: going deeper with nested U-structure for salient object detection. *Pattern Recogn.* **106**, 107404 (2020)
14. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: ICCV (2019)
15. Saito, S., et al.: Pifuhd: multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 84–93 (2020)



Face Recognition for Big Data Using Search Engine for Smart System

Phat Nguyen Huu¹, Duong Nguyen Tung¹, Khanh Nguyen Hoang Nam²,
and Quang Tran Minh^{3,4(✉)}

¹ School of Electronic and Electrical Engineering, Hanoi University of Science and Technology (HUST), Hanoi, Vietnam

phat.nguyenhuu@hust.edu.vn, duong.nt203818@sis.hust.edu.vn

² British International School, Ho Chi Minh City, Vietnam

terrynguyen965@gmail.com

³ Department of Information Systems, Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet, District 10, Ho Chi Minh City, Vietnam

quangtran@hcmut.edu.vn

⁴ Vietnam National University Ho Chi Minh City (VNU-HCM), Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

Abstract. In Vietnam, research and design activities related to recognition systems are making initial strides within universities. Previous scientific research and projects predominantly focused on areas like fingerprint recognition to support security systems. However, topics related to human-computer interaction through facial recognition have yet to be fully developed. This paper aims to elucidate various aspects of the SSD model, DeepFace, IrisNet, and FAISS, encompassing their operational principles and IoT applications in security technology through facial recognition for smart door locks. We perform to evaluate several famous models with existing algorithms. The results indicate that the system achieved nearly 93% accuracy for single faces with an execution time of 0.25 s for the Iris model.

Keywords: Big Data · smart system · face recognition · image processing · IrisNet model

1 Introduction

Today, humans have always aspired to create devices that can think and act like them. To develop robots with capabilities for thinking and independent functioning closely resembling humans, computer vision is a crucial and indispensable element. Compared to the eyes of humans, computer vision enables robots to observe the surrounding world, facilitating their external interactions [1, 2]. Humans have engineered sensors and image processors with capabilities similar to computer vision. Advanced optical lenses have endowed modern cameras with

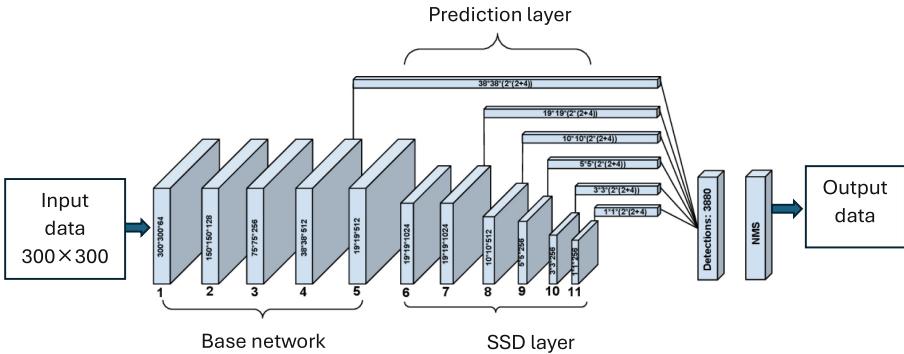


Fig. 1. SSD network in face detection architecture [4].

astonishing precision and sensitivity. These cameras can capture thousands of images per second and achieve remote recognition with remarkable accuracy [3].

In Vietnam, research and design activities related to recognition systems are making initial strides within universities. Previous scientific research and projects predominantly focused on areas like fingerprint recognition to support security systems. However, topics related to human-computer interaction through facial recognition have yet to be fully developed.

Recognizing this as a novel and highly applicable direction, the paper concentrates on amalgamating knowledge of computer vision and digital image processing. The aim is to construct a system where humans can control objects through facial recognition, harnessing the potential of this emerging field.

An advanced form of image synthesis based on artificial intelligence algorithms, concerns about identity verification, misinformation, and privacy infringement have intensified with the rise of deepfake technology [5-7]. Deepfake techniques involve synthesizing realistic-looking images, videos, or audio recordings where individuals appear. This technology poses significant challenges in the realm of computer vision and digital image processing.

Incorporating both facial recognition and Iris recognition capabilities into the smart lock system not only enhances functionality but also strengthens resilience against emerging threats such as deep fake manipulation. By harnessing the power of advanced biometric identification methods, the project aims to establish robust and secure mechanisms for human-computer interaction and object control, paving the way for innovative applications in various domains, including security, access control, and user authentication.

The rest of the paper is presented as follows. In Sect. 2, we introduce and give the related work. Section 3 introduces the models and analyzes them. In Sect. 4, we present and evaluate the effectiveness of the proposed model, respectively. Finally, we give a conclusion in Sect. 5.

2 Related work

The present work is also purely data-driven in the sense that it learns representations from raw pixel space from the faces [8,9]. Instead of the hand-crafted features, we utilize a large set of labeled faces to incorporate the necessary amounts of robustness to such changes as pose, lighting conditions, etc. In this paper, we will discuss two types of the deep network structure used in computer vision. Both are convolution neural networks, The first network is known as convolutional neural network [10,11]- FCN while the second one is convolution neural network (CNN). The first one is based on the Zeiler and Fergus model that uses multiple layers of convolutions, nonlinear activation, local response normalization, and max-pooling. Based on previous work, we place 1X1Xd convolutional layers. The second architecture is derived from the Inception model presented by Szegedy et al. where the Inception model came out tops in the ImageNet 2014 competition. This architecture adopts mixed layers of several parallel convolution and pooling layers and the number of parameters is reduced times, and the computation cost is cut off one-fifth without lessening the accuracy.

Although there is a wealth of research work done on face verification and recognition, only the most relevant work done in the last few years has been discussed in this paper. The authors [9,12] have employed deep networks accompanied by PCA for feature extraction and SVM for classification. For face recognition, Zhenyao et al extended the database into frontal space using a deep network, and then for classification used CNNs then PCA followed by the ensemble of SVMs in case of verification. They used a multi-stage system in which faces are aligned to a 3D model and a multi-class network is used to recognize faces in thousands of people. They also tried using a Siamese network that improves the parameters of L1-distance of face features. The best results, 97.35%, were obtained from a combination of three networks with different alignments and color channels which were trained with a non-linear SVM. Another work used a loss function similar to that used in [13] where they used it to rank images for semantic similarity.

3 Proposal model

3.1 SSD network

The single-shot multi-box detector (SSD) is a prominent object detection architecture known for its real-time capabilities and high accuracy. Developed to efficiently detect objects of varying sizes within images, SSD combines the strengths of both convolutional neural networks (CNNs) and anchor-based object detection frameworks [14–16].

SSD uses a single network to predict object classes and bounding box offsets simultaneously across multiple scales within the image. This is achieved through a series of convolutional layers with different spatial resolutions. Each of these layers is responsible for detecting objects of specific sizes as shown in Fig. 1.

The SSD architecture consists of the following key components:

1. **Base convolutional network:** A pre-trained CNN such as VGG, ResNet, or MobileNet is employed to extract features from the input image. These features are then used for subsequent detection tasks.
2. **Multi-scale feature maps:** SSD introduces a set of auxiliary convolutional layers on top of the base network. These layers progressively decrease in spatial resolution while increasing in receptive field size. This allows the network to detect objects with varying sizes. Each layer is associated with a particular scale of objects, enabling the detection of small, medium, and large objects simultaneously.
3. **Anchor boxes:** A predefined set of anchor boxes is generated for each spatial location in the multi-scale feature maps. These anchor boxes have different aspect ratios and scales to cover a wide range of object shapes and sizes. SSD predicts the offsets among these anchor boxes and the ground-truth bounding boxes of the objects.
4. **Object class prediction:** At each spatial location, the network performs object class prediction. This involves using a series of convolutional filters to estimate the probability distribution over different object classes. These predictions are made for each anchor box.
5. **Bounding box offset prediction:** Alongside class predictions, SSD also predicts the offsets needed to adjust the anchor boxes and make them match the ground-truth bounding boxes more accurately. This ensures the precise localization of objects.
6. **Loss function:** SSD employs a composite loss function that combines the classification loss (using softmax) and the localization loss (using smooth L_1 loss). The loss is calculated for both object presence and absence, as well as for the bounding box offsets.

The localization loss for a single bounding box prediction can be defined using the smooth L_1 loss as

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (1)$$

Where x is the difference between the predicted and the ground-truth box offset.

The overall loss function for SSD involves the sum of classification and localization losses across all anchor boxes and classes. The network is trained using backpropagation and optimization techniques similar to SGD or Adam.

3.2 DeepFace

Deepface is a sophisticated facial recognition system that operates on the principles of deep learning and convolutional neural networks (CNNs) [17, 18]. Deepface has garnered significant attention for its remarkable accuracy in facial recognition tasks that is developed by Facebook's AI Research (FAIR) division [9].

DeepFace employs a hierarchical architecture composed of multiple layers of interconnected neurons, mimicking the human brain's neural connections. The

initial layers capture low-level features such as edges and textures while deeper layers progressively assemble these features into more complex facial attributes such as eyes, nose, and mouth. This hierarchical arrangement enables the model to learn intricate representations of faces in a data-driven manner.

One of the pivotal aspects of DeepFace is the use of CNNs. These networks are well-suited for image analysis tasks due to their inherent ability to retain spatial relationships in data through the application of convolutional filters. DeepFace's CNN architecture enables it to automatically extract relevant features from facial images, facilitating robust identification and verification even in the presence of variations in pose, lighting conditions, and expressions.

Training deepface necessitates an extensive dataset of labeled facial images. The researchers at FAIR leveraged a large-scale dataset containing millions of images of individuals, allowing the model to learn a diverse range of facial characteristics. Furthermore, a triplet loss function is employed during training to ensure that the neural network learns to discriminate between positive and negative pairs of faces, effectively enhancing its ability to distinguish among different individuals.

Deepface's evaluation process entails measuring the similarity between two facial images. By projecting the faces into a shared feature space, the Euclidean distance between their respective feature vectors is computed. If the distance is below a certain threshold, the faces are considered a match. This process allows Deepface to accurately identify individuals and verify their identities with remarkable precision.

3.3 FAISS

Facebook AI similarity search (FAISS) is an efficient and scalable library for similarity search and clustering of high-dimensional data. It was developed by Facebook AI research and is designed to handle large-scale datasets with millions or even billions of data points. The underlying theoretical foundation of FAISS revolves around the utilization of various indexing techniques, distance metrics, and optimization strategies to accelerate nearest-neighbor search operations in high-dimensional spaces [19, 20].

FAISS leverages techniques from the field of information retrieval, such as inverted indices and quantization, to efficiently index and search through large datasets. The library provides multiple indexing structures, including the product quantization (PQ) index, the inverted file with vocabulary (IVF) index, and the hierarchical navigable small world (HNSW) index, among others. These indexing methods help to reduce the search space and prune candidates quickly, resulting in significantly faster search times.

In terms of distance metrics, FAISS supports a wide range of commonly used distance functions, including Euclidean, Manhattan, and cosine distances, as well as more specialized ones like Jaccard and Hamming distances. This flexibility allows FAISS to accommodate various types of data and similarity measures.

To optimize performance, FAISS employs techniques like single instruction multiple data (SIMD), multi-threading, and GPU acceleration. This enables effi-

cient parallel computation and leverages hardware resources effectively, leading to improved query throughput and reduced search latency.

3.4 IrisNet

IrisNet serves as a sophisticated recognition system, adept at identifying individuals based on the unique patterns present. Developed to handle vast datasets and ensure efficient authentication processes, IrisNet operates through a series of well-defined stages, employing cutting-edge algorithms and techniques [22–24].

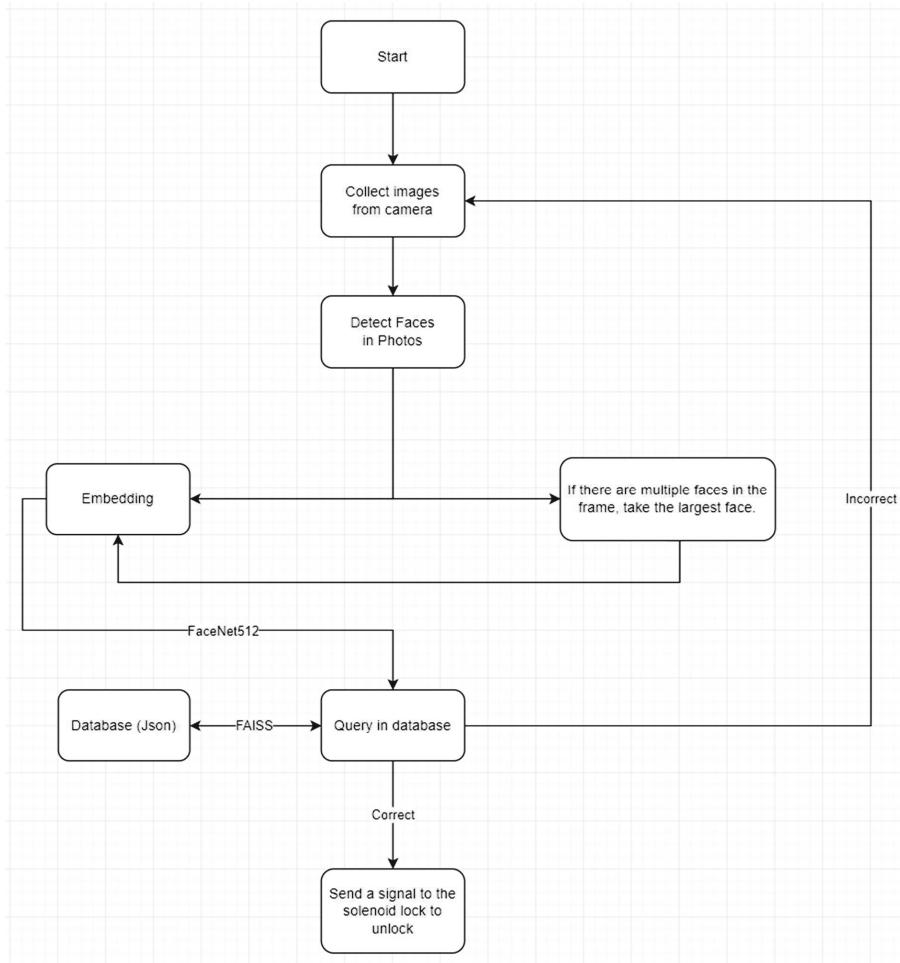


Fig. 2. Block diagram of system.

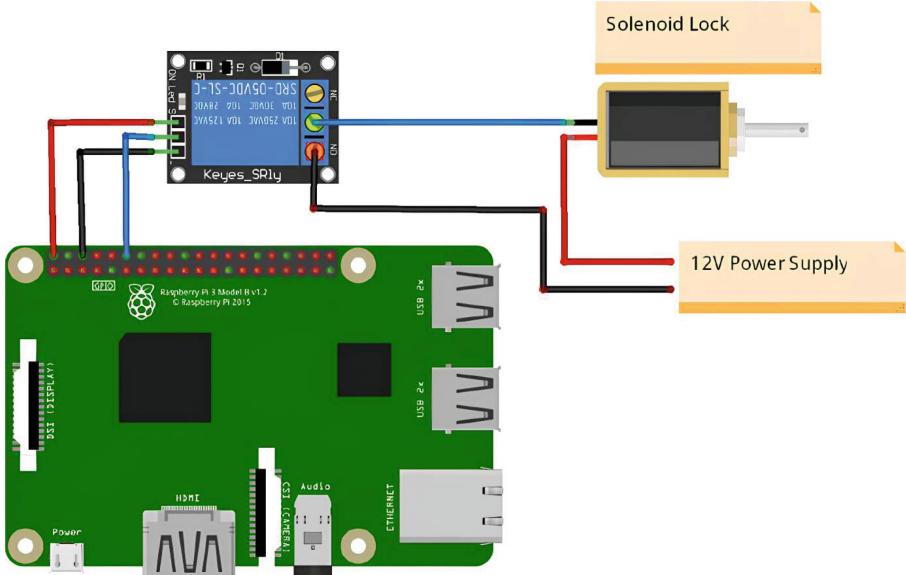


Fig. 3. Schematic diagram of system.

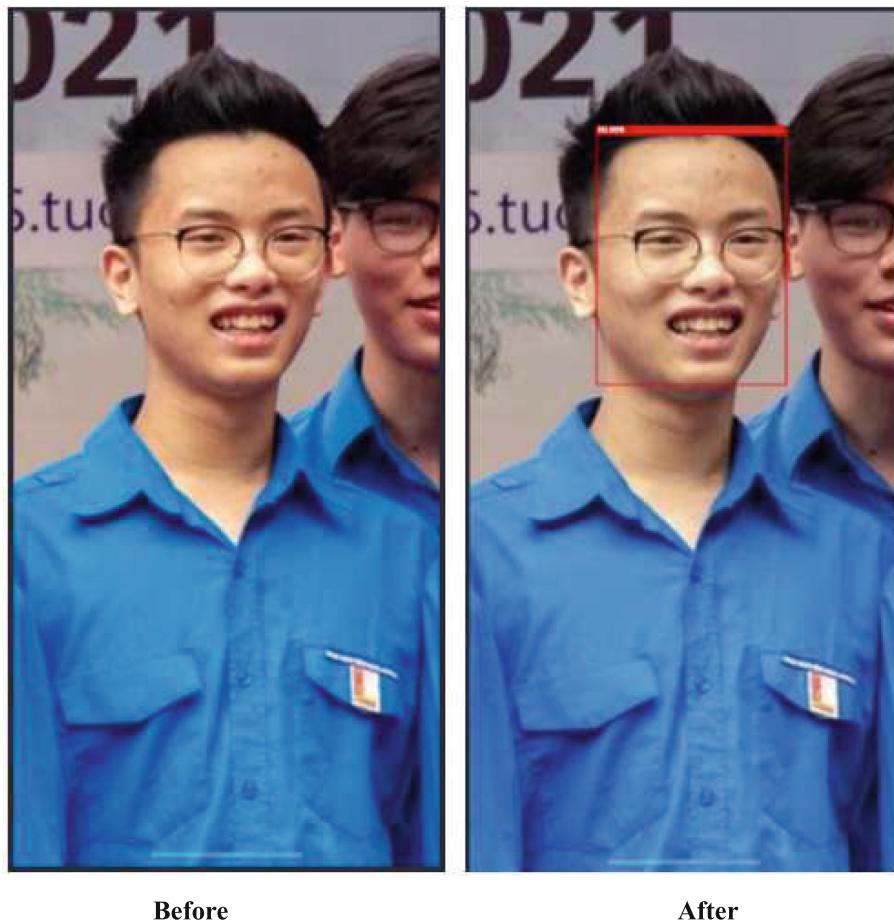


Fig. 4. Several examples of Wider face dataset [21].

Data Acquisition: IrisNet begins by capturing high-resolution images using specialized scanners or cameras. This initial step is crucial for obtaining clear and detailed representations of the Iris patterns.

Image Preprocessing: Once the Iris images are acquired, it applies preprocessing techniques to enhance quality and normalize variations caused by factors like lighting conditions and distortion. This step ensures optimal data quality for subsequent analysis.

Feature Extraction: Iris recognition hinges on the extraction of distinct patterns. Sophisticated algorithms analyze the Iris images to identify and extract unique features such as crypts and furrows. These extracted features form the foundation for creating a template that represents the individual characteristics.



Before

After

Fig. 5. Single face recognition results.



Before

After

Fig. 6. Multi face recognition results.

Template Creation: Based on the extracted features, IrisNet generates a template for each Iris image. These templates serve as compact representations of the iris patterns and are used for comparison during the authentication process.

Database Matching: During authentication, IrisNet compares the template derived from the input image with templates stored in its database. This comparison utilizes advanced pattern-matching algorithms to determine the level of similarity between templates.

Decision Making: IrisNet decides the identity of the individual based on the comparison results. If the similarity exceeds a predetermined threshold, the individual is authenticated and granted access. Otherwise, the authentication attempt may be rejected.

Authentication: Successful authentication grants the individual access to the system or application. In cases of authentication failure, appropriate actions may be taken to maintain security protocols.

In essence, IrisNet leverages state-of-the-art technology to provide accurate and reliable recognition capabilities. By analyzing the unique features, IrisNet ensures secure authentication processes across various domains, including physical access control and digital identity management.

3.5 Deepfake

DeepFake detection is a crucial area of research and development aimed at identifying manipulated media generated by DeepFake technology. Its operation involves delving into several key components and methodologies [25–27]:

Data Collection: Deepfake detection begins with the collection of diverse datasets containing both authentic and manipulated media samples. These datasets serve as the foundation for training machine learning models to discern patterns indicative of deepfake manipulation.

Feature Extraction: plays a pivotal role in deepfake detection, involving the extraction of discriminative features from media content. Various techniques, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are employed to capture intricate visual and temporal patterns present in both authentic and deepfake media.

Model Training: Deepfake detection models are trained using supervised learning techniques where they learn to differentiate between authentic and manipulated media samples based on extracted features. Training involves optimizing model parameters to minimize classification errors and enhance detection accuracy.

Algorithm Development: Researchers develop advanced algorithms tailored to detect subtle artifacts and inconsistencies characteristic of deepfake manipulation. These algorithms leverage techniques such as image forensics, temporal analysis, and anomaly detection to identify anomalies indicative of synthetic media.

Validation and Evaluation: Deepfake detection models undergo rigorous validation and evaluation processes to assess their performance and robustness.

Evaluation metrics, including precision, recall, and F1-score, are utilized to quantify detection accuracy and reliability across diverse datasets and scenarios.

Deployment and Integration: Effective deep fake detection solutions are deployed and integrated into various platforms and applications to combat the proliferation of manipulated media. Integration may involve incorporating detection algorithms into social media platforms, video hosting sites, and digital forensics tools to mitigate the spread of disinformation and malicious content.

In summary, Deepfake detection operates through a comprehensive framework encompassing data collection, feature extraction, model training, algorithm development, validation, and deployment. By leveraging advanced machine learning and computer vision techniques, deepfake detection endeavors to safeguard the integrity of digital media and mitigate the detrimental effects of synthetic manipulation.

4 Experimentation

4.1 Setup

The block diagram of system implementation is set up as shown in Fig. 2.

The proposed system implementation circuit is shown in Fig. 3. In Fig. 3, we use Raspberry Pi3 with a camera and an automatic unlocking and unlocking system based on the received face.

In this paper, we utilize the WIDER FACE dataset for training [21]. THE WIDER FACE is a face detection benchmark dataset. We chose 32,203 images and labeled 393,703 faces with a high degree of variability in scale, pose, and occlusion as depicted in the sample images. WIDER FACE dataset is organized based on 61 event classes. We randomly select 40%, 10%, and 50% for training, validation, and testing for each event class. We adopt the same evaluation metric employed in the PASCAL VOC dataset [28, 29]. Similar to MALF and Caltech datasets [30, 31], we do not release bounding box ground truth for the test images. Users are required to submit final prediction files, which shall proceed to evaluation. Figure 4 shows several examples of the data to perform the system.

4.2 Result

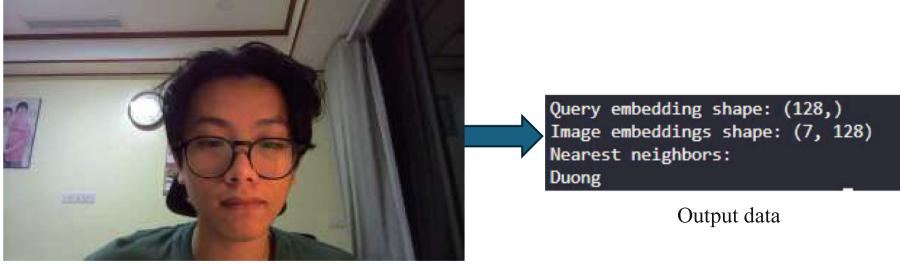
We experiment with two cases for Internet of Things (IoT) applications. In the first case, there is only one single face. In the second case, there are many faces in the frame. The results in Figs. 5 and 6 show that the system recognizes the input objects well. In both Figs. 5 and 6, we see that the system recognizes the face whether standing alone or among many people. It not only recognizes but also eliminates faces that are not the target to be recognized.

We employ the facenet model within deepface for conducting embedding. The results are as shown in Fig. 7. In Fig. 7, to implement the algorithm embedding in Jeston nano hardware, we will have to convert it into a matrix. To reduce the

**Fig. 7.** FaceNet model within DeepFace results.

amount of computation we only spread the result after the decimal point to 8 digits.

We perform recognition with the Iris model. The results are as shown in Figs. 8, 9, and Table 1. In Figs. 8 and 9, we see that the system has correctly recognized the subject based on the eye region. Due to the optimized recognition region, the recognition time is very fast reduced to 0.25 s. This time shows that the algorithm is capable of real-time applications with high accuracy. In Table 1, we see that the Iris model, although using simple hardware, has the same execution time as existing models.



Input data

Fig. 8. DeepFace recognition result.**Table 1.** Comparing the time processing with other models.

Method	Model	Time processing (second)
[32]	Deep Learning on Embedded GPU System	0.23
[33]	Artificial Intelligence	1.4
Our proposal	Using Iris	0.25

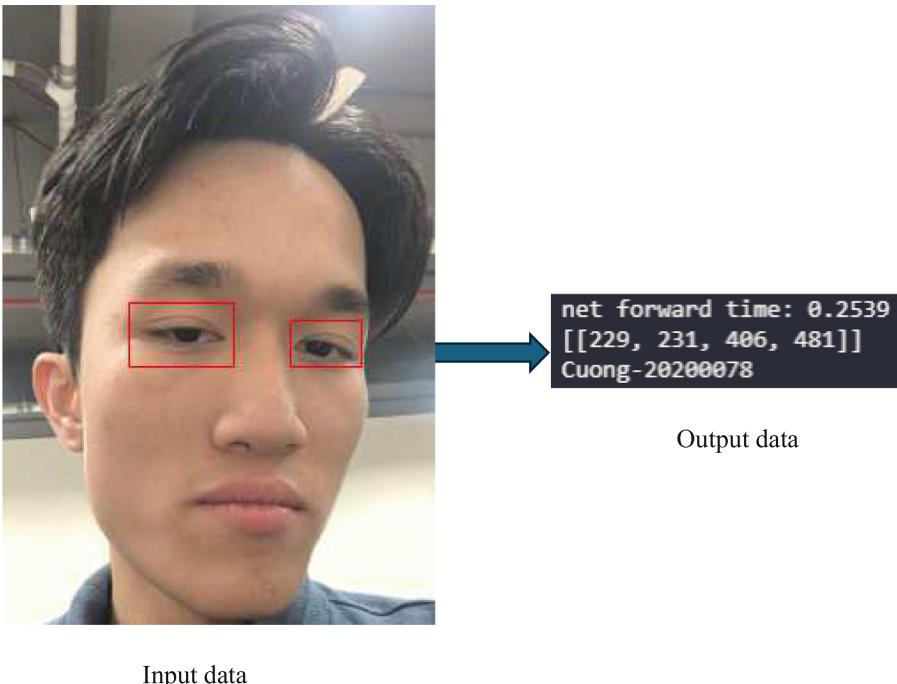


Fig. 9. Iris recognition result.

We also compare the accuracy of the proposed model with existing models. The results are shown in Table 2. In this table, we see that the proposed model outperforms the existing models in terms of accuracy up to 93%. This proves that the Iris model is the best model for real-time recognition applications.

Table 2. Comparing accuracy with other models.

Method	Model	Accuracy (%)
[34]	Daugman's Iris	87.82
[35]	Machine learning	50.10
Our proposal	Using Iris	93.03

5 Conclusion

The paper proposes to apply machine learning models to face recognition for smart applications. The results indicate that the system achieved nearly 93.03% accuracy for single faces with an execution time of 0.25 s for the Iris model. The

models are feasible when applied in intelligent systems. To further enhance the accuracy and security of this product, we will research the integration of an Iris detection system for potential enhancements.

Acknowledgment. We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, and HUST for supporting this study.

References

1. Wang, K.-J., Zheng, C.Y., Mao, Z.-H.: Human-centered, ergonomic wearable device with computer vision augmented intelligence for VR multimodal human-smart home object interaction. In: 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 767–768 (2019)
2. Yang, S., Luo, P., Loy, C.-C., Tang, X.: From facial parts responses to face detection: a deep learning approach. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ser. ICCV ‘15, pp. 3676–3684. IEEE Computer Society, USA (2015)
3. Manakitsa, N., Maraslidis, G.S., Moysis, L., Fragulis, G.F.: A review of machine learning and deep learning for object detection, semantic segmentation, and human action recognition in machine and robotic vision. *Technologies* **12**(2), 15 (2024)
4. C. Thuis, “Ssd-face : Single shot multibox detector for small faces,” in *Computer Science*, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:52830663>
5. Pan, D., Sun, L., Wang, R., Zhang, X., Sinnott, R.O.: Deepfake detection through deep learning. In: 2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT), pp. 134–143 (2020)
6. Swathi, P., Sk, S.: Deepfake creation and detection: a survey. In: 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 584–588 (2021)
7. M. S. Rana, B. Murali, and A. H. Sung, “Deepfake detection using machine learning algorithms,” in *2021 10th International Congress on Advanced Applied Informatics (IIAI-AAI)*, 2021, pp. 458–463
8. Y. Sun, X. Wang, and X. Tang, “Deeply learned face representations are sparse, selective, and robust,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun 2015, pp. 2892–2900
9. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: “Deepface: Closing the gap to human-level performance in face verification,” in. IEEE Conference on Computer Vision and Pattern Recognition **2014**, 1701–1708 (2014)
10. LeCun, Y., et al.: Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1**(4), 541–551 (1989)
11. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986)
12. Zhu, Z., Luo, P., Wang, X., Tang, X.: Recover Canonical-View Faces in the Wild with Deep Neural Networks (2014). [Online]. Available: <https://arxiv.org/abs/1404.3543>
13. Wang, J., et al.: Learning fine-grained image similarity with deep ranking. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, ser. CVPR ‘14, pp. 1386–1393. IEEE Computer Society, USA (2014)

14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision - ECCV 2016, pp. 21–37. Springer International Publishing, Cham (2016)
15. Ye, B., Shi, Y., Li, H., Li, L., Tong, S.: Face ssd: a real-time face detector based on ssd. In: 2021 40th Chinese Control Conference (CCC), pp. 8445–8450 (2021)
16. Ranjana, P., Ramesh, K.: Face mask detection using single shot multibox detector and mobile net. In: 2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing and Communication Engineering (ICATIECE), pp. 1–4 (2022)
17. Masi, I., Wu, Y., Hassner, T., Natarajan, P.: Deep face recognition: a survey. In: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 471–478 (2018)
18. Wang, M., Deng, W.: Deep face recognition: a survey. Neurocomputing **429**, 215–244 (2021)
19. George, G., Rajan, R.: A faiss-based search for story generation. In: 2022 IEEE 19th India Council International Conference (INDICON), pp. 1–6 (2022)
20. Johnson, J., Douze, M., Jegou, H.: Billion-scale similarity search with GPUS. IEEE Trans. Big Data **7**(03), 535–547 (2021)
21. Yang, S., Luo, P., Loy, C.C., Tang, X.: Wider face: a face detection benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2016, pp. 5525–5533 (2016)
22. Kaur, N., Juneja, M.: A review on iris recognition. Recent Adv. Eng. Comput. Sci. (RAECS) **2014**, 1–5 (2014)
23. Alhamdi, N.A., Aldeebri, M.B., Alkout, M.H.: Iris recognition using artificial neural network. In: 2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MISTA), pp. 295–298 (2022)
24. Nazmdeh, V., Mortazavi, S., Tajeddin, D., Nazmdeh, H., Asem, M.M.: Iris recognition; from classic to modern approaches. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0981–0988 (2019)
25. Hu, P., Ramanan, D.: Finding tiny faces. CoRR (2016). [Online]. Available: <http://arxiv.org/abs/1612.04402>
26. M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, “Deepfake detection: A systematic literature review,” *IEEE Access*, vol. 10, pp. 25 494–25 513, 2022
27. Mary, A., Edison, A.: Deep fake detection using deep learning techniques: a literature review. In: 2023 International Conference on Control, Communication and Computing (ICCC), pp. 1–6 (2023)
28. Vicente, S., Carreira, J., Agapito, L., Batista, J.: Reconstructing pascal VOC. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2014, pp. 41–48 (2014)
29. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vision **111**(1), 98–136 (2015)
30. Griffin, G., Holub, A., Perona, P.: Caltech 256. CaltechDATA (2022)
31. Caltech-256 object category dataset. CalTech Report, pp. 1–20 (2014)
32. Saypadith, S., Aramvith, S.: Real-time multiple face recognition using deep learning on embedded GPU system. In: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), vol. 2018, pp. 1318–1324 (2018)

33. Garcia, R.A.C., Lacayanga, R.P., Cruz, F.R.G.: Application of artificial intelligence in adaptive face recognition system. In: 2021 IEEE 11th International Conference on System Engineering and Technology (ICSET), pp. 263–268 (2021)
34. Pillai, J.K., Puertas, M., Chellappa, R.: Cross-sensor iris recognition through kernel learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(1), 73–85 (2014)
35. Nie, L., Kumar, A., Zhan, S.: Periocular recognition using unsupervised convolutional RBM feature learning. In: 2014 22nd International Conference on Pattern Recognition, pp. 399–404 (2014)



Synergistic Mel-Frequency Cepstral Coefficients and Short-Time Fourier Transform for Enhanced Bee States Detection Using Machine Learning

Thi-Thu-Hong Phan^(✉) 

Artificial Intelligence Department, FPT University, Danang, Vietnam
hongptt11@fe.edu.vn

Abstract. This study investigates the potential of sound analysis to detect bee states within beehives, a critical challenge for beekeepers. We propose a novel approach combining Mel-Frequency Cepstral Coefficients (MFCC) and Short-Time Fourier Transform (STFT), effectively creating more informative representative data for the classification of bee states. Experimentation on a real dataset demonstrates that the Random Forest classifier utilizing synergy features extracted from MFCC and STFT outperforms models relying solely on MFCC or STFT features, significantly improving classification accuracy (up to 87.2%). This integrated approach offers advantages in capturing both spectral detail (MFCC) and temporal information (STFT) potentially leading to improved classification accuracy for bee states detection. The findings contribute valuable insights for developing robust bee colony health monitoring systems.

Keywords: Bee sound · NoQueen · Bee states · STFT · MFCC · Machine learning methods · Combined features

1 Introduction

Honey bees are essential pollinators, playing a critical role in maintaining ecosystem balance and enhancing agricultural yield. Their adept foraging supports the reproduction of plants, including key crops, and their hive products, such as honey and beeswax, cater to various human requirements. Furthermore, honey bee populations serve as indicators of overall environmental health, emphasizing their importance and the necessity for conservation measures.

Maintaining healthy bee colonies is crucial for beekeepers. They face constant threats like swarming, Varroa mites, and critically, missing queens. Traditionally, beekeepers rely on regular hive inspections for monitoring their hives. This involves physically opening the hive to visually inspect the colony's health, confirm the queen's presence, and assess her overall well-being, etc. However, these inspections are time-consuming, potentially disruptive to bee behavior and harmful to the queen. This requires a non-invasive approach to monitoring beehives and facilitating interventions. To address these limitations, many studies

have been exploring advanced technology-based methods [3,8]. The Internet of Things (IoT) and artificial intelligence play a crucial role in this new approach. Sensors-equipped beehives can continuously collect data on various environmental factors like temperature and humidity, and most importantly, these systems can capture bee sounds for further analysis.

Sound analysis, combined with machine learning algorithms, holds immense promise for beehive monitoring. Research has shown that this approach is effective in detecting bee states in bee colonies, with various algorithms being employed using sound as input data. For instance, Ruvinka et al. (2021) [13] achieved 92% accuracy using LSTM networks with Mel-frequency cepstral coefficients (MFCC) to detect the queenless in the beehive. Similarly, Nolasco et al. (2019) explored both Support Vector Machines (SVM) and Convolutional Neural Networks (CNN) with MFCC, achieving high accuracy (up to 94% with the addition of Hilbert Huang Transform (HHT)). In [9], Phan et al. investigated new MFCC and hyper-parameters tuning techniques for enhanced performance of bee sound recognition. Truong et al. [14] developed an approach based on deep learning models to identify bee buzzing sounds from other sounds like noise or cricket chirping. Barbisan and Riente [1] achieved high accuracy (up to 98.8%) using both Neural Networks (NN) and Support Vector Machines (SVM) with 20 MFCC features. Further studies explored alternative methods for queen presence detection, Fourer et al. employed STFT with Convolutional Neural Networks (CNN), achieving 96% accuracy. Similarly, Ho et al. (2023) utilized MFCC for feature extraction on a real dataset, reaching a peak accuracy of 91.75%.

In [12], Rustam et al. focused on classifying bee sounds into three categories: Bee, NoBee, and NoQueen. To achieve this goal, the authors employed a variety of feature selection techniques and machine learning algorithms. They achieved slightly higher accuracy with K-Nearest Neighbors (KNN) at 0.83 compared to 0.82 with Random Forest (RF). However, the performance of these models is still suboptimal on their dataset. This raises questions about their inefficiency and potential ways to improve their effectiveness. This challenge motivates us to find answers. Specifically, we propose a novel approach that combines Mel-Frequency Cepstral Coefficients (MFCC) and Short-Time Fourier Transform (STFT), effectively creating more informative representative data for detecting bee states in beehives. Experimental results show that these techniques can significantly improve the accuracy of machine learning methods in recognizing bee states within beehives.

The rest of the paper is structured as follows. Section 2 provides a concise description of the methods employed in the study. Section 3 presents the experiments conducted, the obtained results, and relevant discussions. Section 4 offers concluding remarks summarizing the key findings and potential future directions.

2 Methodology

This paper proposes a methodology for classifying various beehive states through audio analysis as Fig. 1. We leverage a combination of STFT and MFCC for

feature extraction to enhance the performance of ML methods for this detection task.

The process commences with gathering audio samples from beehives. These samples encompass a diverse range of bee activities, aiming to capture audio signatures associated with three distinct states: Bee presence, No bee presence, and No queen presence.

The collected audio samples are then processed to extract features suitable for bee states classification. This work employs two distinct techniques: STFT and MFCC. Each method offers valuable insights into the audio data - STFT revealing the time-frequency distribution and MFCC capturing the spectral envelope relevant to human hearing. Following extraction, the STFT and MFCC features are combined, creating a comprehensive representation of the audio data.

These combined features are subsequently fed into machine learning models for bee states classification. To ensure a comprehensive exploration of the data and identify the most suitable approach, we investigate six different machine learning algorithms: Random Forest (RF), Extra Trees (ET), eXtreme Gradient Boosting (XGBoost), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Logistic Regression (LR). Each algorithm brings its strengths allowing for a robust analysis of the beehive audio data. This exploration holds the potential to achieve the most accurate bee states classification results.

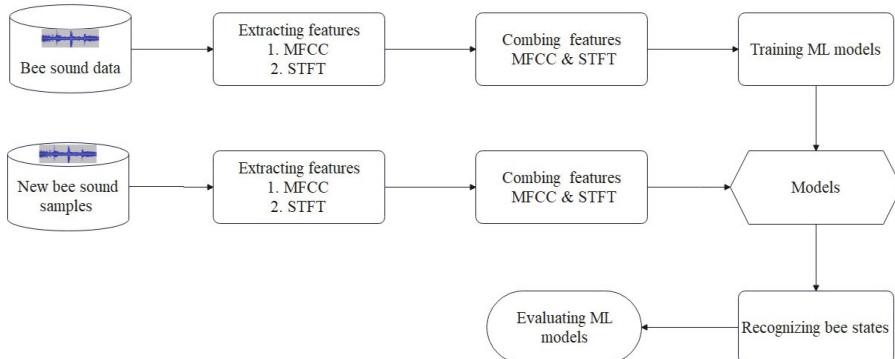


Fig. 1. Overview of proposed approach for recognizing bee states in beehives

2.1 Feature Extraction Methods

a) Short-Time Fourier Transform (STFT)

STFT is a fundamental method in signal processing, utilized across fields like audio analysis, speech recognition, and biomedical engineering [6]. It offers a time-frequency representation, enabling the examination of changing spectral characteristics. Unlike standard Fourier analysis, which is suited for stationary signals, STFT performs localized Fourier transforms over small, overlapping

windows, making it effective for dynamic signals. Figure 2 describes the steps involved in calculating STFT:

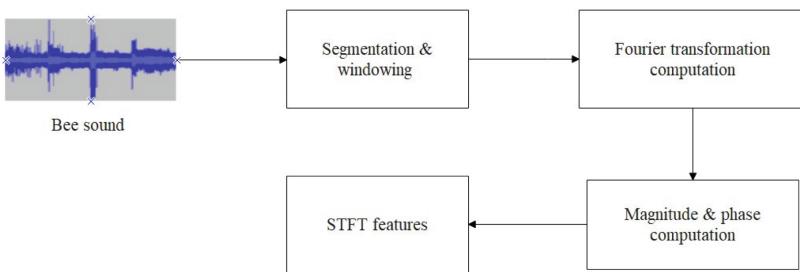


Fig. 2. Schema of STFT method

Divide the Signal into Short Time Segments: The signal is divided into short time segments, called frames, with a fixed length. The frame length determines the resolution of the STFT in time. A shorter frame length will provide better time resolution but lower frequency resolution, while a longer frame length will provide better frequency resolution but lower time resolution.

Apply a Window Function to Each Segment: A window function is applied to each segment to reduce artifacts caused by the abrupt truncation of the signal. Common window functions include Hannning, Hamming, and Gaussian windows. The choice of window function can affect the shape of the STFT peaks.

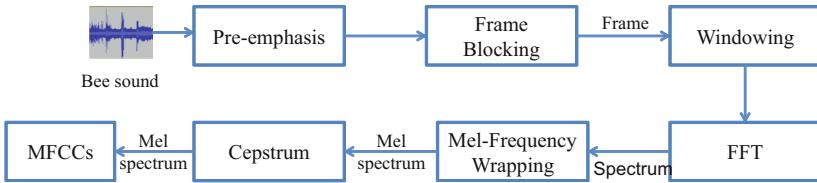
Compute the Fourier Transform of Each Segment: The Fourier transform is computed for each segment to obtain its frequency spectrum. The Fourier transform decomposes the signal into its constituent sinusoidal components, revealing the signal's frequency content within that segment.

Combine the Results: Each segment's magnitude and phase of the Fourier transform are combined to form the STFT. The magnitude represents the strength of each frequency component, while the phase represents the time delay of each frequency component.

Display the STFT: The STFT can be visualized as a time-frequency spectrogram, representing the magnitude by color intensity. The spectrogram shows how the frequency content of the signal changes over time.

b) Mel Frequency Cepstral Coefficients (MFCC)

MFCC is a powerful method for extracting features from audio signals. This technique involves dividing the frequency band into sub-bands on the MEL scale and then applying the Discrete Cosine Transform (DCT) to extract Cepstral Coefficients. The key steps of MFCC, as shown in Fig. 3, include pre-emphasis, framing, windowing, applying the Discrete/Fast Fourier Transform (DFT/FFT), Mel-frequency warping, and cepstrum calculation (inverse DCT).

**Fig. 3.** Schema of MFCC algorithm

Pre-emphasis: The initial stage of MFCC, pre-emphasis, amplifies energy in higher frequencies [10]. This involves passing the signal through a first-order high-pass filter to reduce noise during sound capture.

Frame Blocking and Windowing: MFCC operates on short, stationary intervals of audio data by dividing the signal into overlapping frames. Each frame contains multiple audio samples with some overlap between consecutive frames. Hanning or Hamming windows are commonly applied [11] to enhance harmonics, smooth edges, and reduce edge effects during DFT/FFT computation.

Fast Fourier Transform (FFT) Applying: In this step, each windowed frame is transformed into a magnitude spectrum using FFT, which quickly computes the Discrete Fourier Transform (DFT). This process converts each frame from the time domain to the frequency domain.

Mel-Frequency Wrapping: The Mel spectrum is obtained by applying a Mel-filter bank to the power spectrum (derived from the signal's FFT). These Mel filters mimic human hearing, capturing the energy within specific frequency bands relevant to our perception of sound.

Cepstrum: Mel-scale power spectrum is then converted to the time domain using Discrete Cosine Transform (DCT) to obtain Mel-Frequency Cepstral Coefficients (MFCC).

In this study, we employed three different feature extraction parameters to extract MFCC features from the sound samples: 20 features align with the approach adopted by [1], and 40 and 80 features correspond to the feature extraction method used by [9].

c) Combining MFCC and STFT Features

This study proposes a novel approach to enhance feature representation for bee states classification in beehives. We leverage the complementary strengths of two feature extraction techniques: Mel-Frequency Cepstral Coefficients and Short-Time Fourier Transform. MFCC excels at capturing the perceptual characteristics of sound, mimicking human hearing. This makes it particularly effective for representing signals with prominent pitch content, like bee vocalizations. Additionally, MFCC offers dimensionality reduction, reducing computational complexity. On the other hand, STFT provides a detailed time-frequency representation of the signal. This allows it to capture subtle variations in frequency and transient events that might be crucial for distinguishing bee states in beehives. This characteristic makes STFT well-suited for analyzing non-stationary signals like beehive sounds, which often exhibit complex frequency patterns.

By strategically combining these features, we aim to create a more informative representation of the beehive audio data. This enriched representation is structured as:

$$stft_1, stft_2, \dots, stft_{80}, mfcc_1, mfcc_2, \dots, mfcc_{40}$$

The new features incorporate both the perceptual and temporal aspects of the sounds, potentially leading to improved classification accuracy in identifying different bee states within colonies.

2.2 Machine Learning Models

K-Nearest Neighbors (KNN) is a popular supervised learning algorithm used for both classification and regression tasks [5]. It assigns a label to a new sample based on the labels of its K closest neighbors in the training set. The class label is typically determined by a majority vote among these neighbors, with closer neighbors potentially having more influence. KNN uses a distance metric, usually Euclidean distance, to measure proximity between data points. This method can easily adapt to new data without retraining the model. However, KNN performs poorly in high-dimensional spaces and is sensitive to noise and missing data in the training set.

Support Vector Machines (SVM) is a powerful supervised learning algorithm used for classification and regression tasks [7]. It aims to find the optimal hyperplane in a high-dimensional space that separates different classes with the maximum margin. This hyperplane is determined by support vectors, which are the critical data points that maximize class separation. SVM can handle linearly inseparable data using the kernel trick, mapping input features into higher-dimensional spaces for non-linear classification. In this study, we use the radial basis function (RBF) kernel exclusively, due to its demonstrated effectiveness in various scholarly investigations and publications.

Logistic Regression (LR) is a widely used method primarily for classification tasks. It works best when the relationship between the independent variables and the dependent variable (often binary) can be modeled linearly. LR is particularly useful for estimating the probability of an event occurring. The coefficients produced by LR indicate the extent to which each independent variable influences the likelihood of a specific outcome. Its simplicity and effectiveness make it a popular choice for binary classification problems where understanding variable influences is essential.

Random Forest (RF) is a powerful machine learning algorithm that combines multiple decision trees to improve accuracy and prevent overfitting [2]. Each tree in the forest is created using a random subset of the training data (bootstrapping), promoting diversity among the trees. The trees independently classify or predict outcomes by finding optimal splitting points, often using Gini impurity as a metric. For regression tasks, the predictions of all trees are averaged, while for classification tasks, the final prediction is determined by majority vote.

Extra Trees (ET) or Extremely Randomized Trees, is an ensemble learning algorithm similar to Random Forest (RF) but with some key differences. ET uses the entire original sample, which helps reduce bias. Additionally, ET introduces randomness by selecting split points randomly rather than computing the local optimum using metrics like Gini impurity or entropy. This random selection of split points increases diversity and reduces correlation among the trees, enhancing the algorithm's effectiveness. ET is known for its speed and ability to mitigate overfitting, making it well-suited for large datasets with many features.

XGBoost (XGB) or Extreme Gradient Boosting, is a highly efficient implementation of gradient-boosted decision trees designed for speed and performance [4]. It uses parallel boosting trees to smooth training loss and apply regularization, combining the strengths of multiple base trees for optimal results. The algorithm corrects previous mistakes, learning iteratively to improve performance. To reduce overfitting and accelerate training, XGBoost incorporates randomization techniques, such as selecting subsamples for tree construction and choosing features at various levels. It also employs percentiles to test a subset of candidate splits, significantly speeding up the process while maintaining accuracy, and making it effective for various data science problems.

3 Experiments

3.1 Data Description

To evaluate the proposed approach, we use the dataset in the previous study [12]. This dataset, which comprises 13,792 samples, offers valuable insights into beehive activity through sound recordings, categorized into three classes:

Bee: Sounds generated by normal bee activity within the hive.

NoBee: represents ambient sound, indicating moments when external noise is present. When the queen is old, diseased, or deceased, workers may begin rearing new queens within 12–24 h, leading to a decline in colony activity and sound levels. In such cases, only background or ambient noise is detected, suggesting potential colony collapse or rearing of a new queen.

NoQueen: Sounds associated with queenless hives, potentially signifying colony collapse (due to worker inactivity), queen rearing (leading to decreased activity), or other anomalies.

We divide this dataset into training and testing sets with an 80:20 ratio (Table 1) to conduct and assess experiments.

Table 1. Sample distribution for Bee, NoBee, and NoQueen categories

Class	# Sample	Train (80%)	Test (20%)
Bee	5473	4378	1095
NoBee	3458	2766	692
NoQueen	4861	3889	972

3.2 Results and discussion

The Performance of ML Methods Using Individual Features

a) MFCC Features

Table 2 presents the performance of different machine learning models in identifying bee states (Bee, NoBee, and NoQueen) using MFCC with varying extraction features: 20 features, 40 features, and 80 features. The models considered include KNN, SVM, LR, RF, ET, and XGB. The table demonstrates the impact of the number of MFCC features on the performance of the machine learning models. In general, a model trained with more relevant features tends to achieve higher accuracy, but adding too many or irrelevant features can lead to overfitting and reduced performance. This suggests that extracting more detailed spectral information from the sound recordings can enhance the ability to distinguish between different bee states. RF indicates the most substantial improvement in accuracy with an increasing number of features, reaching a peak of 83.74% with 40 features. This suggests that RF is highly sensitive to feature selection and may benefit from optimization between 20 and 40 features. ET exhibits a similar trend to RF, with a peak accuracy of 84.05% at 40 features and a slight decrease with 80 features.

In [12], Rustam et al. employed a combination of feature selection techniques (PCA, Chi-squared, and SVD) applied to a large set of 1740 MFCC features, achieving the highest accuracy of 83% for bee detection using RF and KNN classifiers. In our study, we utilize a simpler approach that direct extraction of a smaller set of MFCC features (40 features) can achieve a comparable or even better accuracy with 84.05%.

b) STFT Features

Table 3 presents the test accuracy achieved by different machine learning models in identifying bee states using Short-Time Fourier Transform (STFT) features extracted with two different feature sets: 80 features and 257 features. In general, all models except SVM show a slight increase in accuracy with more features (from 80 to 257 features). This suggests that capturing a wider range of frequency information can enhance the ability to distinguish between different bee states. KNN exhibits the most significant improvement (1.83%), followed by RF (0.56%) and ET (0.31%). LR also shows a noticeable improvement (2.88%). Consistent with the previous analysis using MFCC features, XGBoost again achieves the highest accuracy across both feature counts (85.40% with 80 features and 85.72% with 257 features). This reinforces its robustness and effectiveness in bee states identification.

Table 2. Performance of ML methods for identifying bee states using three different MFCC feature sets (%)

Model	20 features	40 features	80 features
KNN	79.46	81.49	82.41
SVM	66.05	68.63	69.53
LR	69.45	75.33	75.64
RF	81.73	83.74	83.54
ET	82.21	84.05	83.18
XGB	80.96	82.58	83.33

When comparing the performance of ensemble ML models (RF, ET, and XGB) using STFT features and MFCC features, it is evident that using STFT features, even with 80 features, they outperform the best performance achieved using MFCC features (84.05%). This indicates that STFT features are more effective in capturing the relevant information for bee states identification compared to MFCC features in this specific case.

Table 3. Performance of ML methods for identifying bee states using two STFT features set (%)

Model	80 features	257 features
KNN	81.66	83.49
SVM	75.01	75.91
LR	68.41	71.29
RF	84.75	85.31
ET	84.10	84.41
XGB	85.40	85.72

The Performance of ML Methods Using Combined Features

The previous analyses have demonstrated the effectiveness of both MFCC and STFT features in bee states identification. MFCC features excel in capturing the perceptual characteristics of sound, while STFT features provide a detailed representation of the frequency spectrum. To harness the strengths of both MFCC and STFT features, we propose a feature fusion approach that combines the extracted features from both methods. We opt to combine MFCC with 80 STFT features after evaluating the trade-off between performance and computational efficiency. While using 257 STFT features yielded slightly higher accuracy, the significant increase in features also led to higher computational costs. To achieve a balance between performance and efficiency, we chose 80 STFT features.

Table 4 presents the accuracy achieved by different machine learning models in identifying bee states using a combination of 80 STFT features and two different sets of MFCC features: 20 features and 40 features. All models show an improvement in accuracy when using 40 MFCC features compared to 20 MFCC features in combination with 80 STFT features. This suggests that incorporating more perceptual frequency content of sound captured by additional MFCC features can enhance the performance of most models. RF and ET demonstrate the most significant improvement in accuracy with 40 MFCC features, with increases of 1.18% and 0.53% respectively. XGBoost exhibits a minimal improvement in accuracy (0.27%) with 40 MFCC features compared to 20 features. This suggests that XGBoost might already be effectively utilizing the information provided by both STFT and 20 MFCC features. KNN and SVM exhibit a moderate increase in accuracy with 40 MFCC features, suggesting they can benefit from additional information but to a lesser extent compared to models like RF and ET. Logistic Regression (LR) shows a noticeable improvement in accuracy (3.84%) with 40 MFCC features, suggesting it utilizes the additional temporal information effectively.

Figure 4 provides a clear and concise illustration of how feature fusion leverages the strengths of different feature extraction techniques like MFCC and STFT. By combining the perceptual information captured by MFCC features with the frequency information captured by STFT features, the model gains a richer understanding of the audio data, leading to more accurate bee states classification. The combination of MFCC (either 20 or 40 features) with STFT features (80) reaches the highest point on the chart. This visually emphasizes that combining features captures a more comprehensive representation of the beehive sounds, leading to improved model performance in identifying bee states.

In the previous study [12], the highest accuracy achieved was around 83%. By employing feature fusion, the new approach has pushed the accuracy to 87.20%, representing a substantial leap forward. This improvement suggests that the combination of MFCC and STFT features captures more relevant information about beehive sounds, enabling the model to better distinguish between different bee states.

Table 4. Performance of ML methods for identifying bee states using 80 STFT with different MFCC feature sets (%)

Model	80 STFT & 20 MFCC features	80 STFT & 40 MFCC features
KNN	80.14	82.75
SVM	65.97	68.85
LR	75.28	79.12
RF	86.01	87.19
ET	86.44	86.97
XGB	86.68	86.95

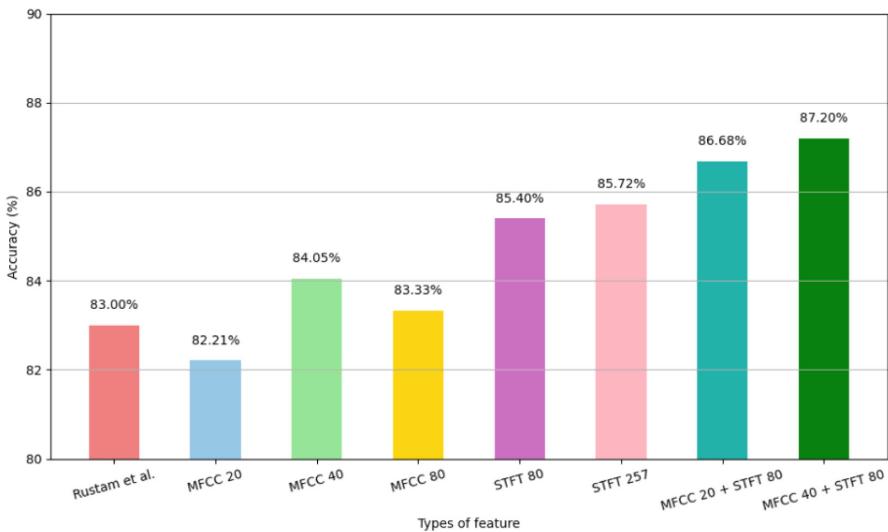


Fig. 4. Comparison of the best accuracy using different feature types

4 Conclusion

This paper proposes a novel approach for improved bee states recognition by generating more informative representative data through the fusion of MFCC and STFT features. We extract and analyze both MFCC and STFT features individually, and subsequently combine them to create new representative data that encompasses a richer set of information about beehive sounds. Experiment results show that RF model achieved a remarkable accuracy of 87.2%, surpassing the previous best result by a significant margin of 4.2% [12]. These findings strongly support the effectiveness of the proposed approach in generating representative data that significantly enhances bee states recognition performance. While this result represents a significant improvement, it still falls short of the ultimate goal of achieving highly accurate and reliable bee states recognition. Recognizing the need for further advancements, we propose investigating the application of state-of-the-art deep learning techniques, such as transfer learning and RegNet modes, to address this challenge.

References

1. Barbisan, L., Riente, F.: Machine learning framework for the acoustic detection of the queen bee presence. *Acta Acustica* (2023)
2. Breiman, L.: Random forests. *Mach. Learn.* **45**, 5–32 (2001)
3. Cecchi, S., Spinsante, S., Terenzi, A., Orcioni, S.: A smart sensor-based measurement system for advanced bee hive monitoring. *Sensors* **20**(9), 2726 (2020)

4. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
5. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory (1967)
6. Durak Ata, L., Arikan, O.: Short-time Fourier transform: two fundamental properties and an optimal implementation. IEEE Trans. Signal Process. **51**, 1231–1242 (2003). <https://doi.org/10.1109/TSP.2003.810293>
7. Evgeniou, T., Pontil, M.: Support vector machines: theory and applications. In: Machine Learning and Its Applications, Advanced Lectures (2001)
8. Liao, Y., McGuirk, A., Biggs, B., Chaudhuri, A., Langlois, A., Deters, V.: Non-invasive Beehive Monitoring through Acoustic Data Using SAS®Event Stream Processing and SAS®Viya®. SAS Global Forum p. 24 (2020)
9. Phan, T.T.H., Nguyen-Doan, D., Nguyen-Huu, D., Nguyen-Van, H., Pham-Hong, T.: Investigation on new mel frequency cepstral coefficients features and hyper-parameters tuning technique for bee sound recognition. Soft. Comput. (2022). <https://doi.org/10.1007/s00500-022-07596-6>
10. Picone, J.: Signal modeling techniques in speech recognition. Proc. IEEE **81**(9), 1215–1247 (1993)
11. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989). <https://doi.org/10.1109/5.18626>
12. Rustam, F., Zahid Sharif, M., Aljedaani, W., Lee, E., Ashraf, I.: Bee detection in bee hives using selective features from acoustic data. Multimed. Tools Appl. **82**(5), 7095–7112 (2023). <https://doi.org/10.1007/s11042-023-15192-5>
13. Ruvingga, S., Hunter, G.J., Duran, O., Nebel, J.C.: Use of LSTM networks to identify “Queenlessness” in honeybee hives from audio signals. In: 2021 17th International Conference on Intelligent Environments (IE), pp. 1–4. IEEE (2021). 10.1109/IE51775.2021.9486575
14. Truong, T.H., Nguyen, H.D., Mai, T.Q.A., Nguyen, H.L., Dang, T.N.M., Phan, T.T.H.: A deep learning-based approach for bee sound identification. Eco. Inform. **78**, 102274 (2023). <https://doi.org/10.1016/j.ecoinf.2023.102274>



A Real-Time Method for High-Resolution Background Matting

Tam Do-Minh¹, Tan Le-Thanh¹, My Kieu², Khuong Nguyen-An¹,
Xuan Toan Mai¹, Hong Tai Tran¹, and Tuan-Anh Tran^{1(✉)}

¹ Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City, Vietnam
trtanh@hcmut.edu.vn
² RainScales, Houston, USA

Abstract. Background Matting is a computer vision problem where the foreground of an image or a video is separated from the background. We emphasize the real-time performance of the model for Matting without using any input besides the original captured image, such as a tri-map and a background image. We also prioritize the use case for this work on video conference. To solve this problem, we use a model with an Encoder-Decoder architecture. Our main contribution to this work consists of proposing and experimenting with a new loss function for training the model of the matting problem, replacing the Normalization layer, and creating a composite dataset for video conferences named Typical Conference Backgrounds (TCB). Promising empirical experimental results have been achieved by our method on the public AIM and Distinction-646 datasets.

Keywords: Real-time · Image matting · Video matting · Group Normalization

1 Introduction

In this paper, we address the problem of background matting, which separates the foreground from the background. The challenge lies in real-time applications, such as video conferencing, where fine details like hair strands are important. Our focus is on humans and their apparel in the foreground, specifically for video conferencing.

In a video, a frame I can be interpreted as a convex combination of a foreground F and a background B through an α coefficient

$$I = \alpha \odot F + (1 - \alpha) \odot B, \quad (1)$$

where operator \odot denotes the element-wise product.

By extracting α , we obtain an alpha matte, a matrix with values between 0 and 1, representing the likelihood that a pixel belongs to the foreground. Each

T. Do-Minh and T. Le-Thanh—These authors contributed equally to this work.

pixel is either part of the foreground F or the background B. Once we have α , we can extract F and composite it onto a new background.

In this paper, background matting refers to the general problem. Image matting applies to still images, as shown in Eq. 1, while video matting addresses the same problem for videos, often involving a frame count, as shown in Eq. 2.

$$I^f = \alpha \odot F^f + (1 - \alpha) \odot B^f. \quad (2)$$

Background matting traditionally required manual techniques like green screens and rotoscoping, but these are labor-intensive. Trimap-based methods were developed to automate this process by using a predefined map to separate foreground and background.

Recent models, such as Background Matting v2 by Lin and Ryabtsev [8], eliminated the need for a trimap but required a pre-captured background image, limiting their use in dynamic settings. MODNet [6], proposed by Zhanghan et al., improved upon this by allowing background matting without external input, making it more suitable for dynamic environments.

The authors were designing the model to process video data; therefore, they modified the already established equation for background matting

$$I^i = \alpha^i F^i + (1 - \alpha^i) B^i. \quad (3)$$

MODNet's design includes supervised training for semantic estimation and a self-supervised approach to improve consistency between frames, using a One-Frame Delay (OFD) to reduce flickering. However, it struggles with fast-moving objects and doesn't fully leverage temporal data, treating frames independently.

Later, Lin et al.'s Robust Video Matting improved on this by incorporating temporal information, reducing flickering and removing the need for a trimap. Despite this, improvements can still be made. We propose the following changes to the current workflow:

- Replace Batch Normalization with Group Normalization to speed up training with minimal difference in loss and quality
- Change the loss function to improve quality
- Crawl more than 10000 images of background suitable for our use-case that we named Typical Conference Backgrounds (TCB)

2 Proposed Method

2.1 Overall Model Architecture

We use an Encoder-Decoder architecture for the matting model based on the work of Lin et al. [9] with some modifications. Mainly, feature loss is introduced to increase perceptual quality, and batch normalization is replaced with group normalization to minimize errors.

The input for our overall architecture for the model is a batch of frames of images $I \in \mathbb{R}^{h \times w \times 3}$. Each high-resolution frame (Image HR) is copied into two

Table 1. Specifications for our MobileNetV3 backbone.

Index	Operator	exp size	#out	SE	NL	s
1	InvertedResidual	16	16	False	RE	1
4	InvertedResidual	72	40	True	RE	2
6	InvertedResidual	120	40	True	RE	1
16	ConvBNActivation	160	960	–	HS	1

identical versions. The first version is fed into the primary sequence of the model (containing the Encoder-Decoder blocks) to get the foreground and alpha matte prediction, and the other version is used as input for the DGF (Deep Guided Filter) in combination with the former’s foreground and alpha matte. The Image HR is bilinearly downsampled into Image LR and fed into our encoder.

2.2 Encoder

We use MobileNetv3 as the backbone for our encoder. Four features we use for the next step are outputs from the layer with 1, 4, 6, and 16 indexes in Table 1. The four Encoder Blocks are used to learn features from the Image LR at deeper scales, particularly at $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$ and $\frac{1}{16}$. The outputs are four feature maps. Then the output of the encoder module at scale $\frac{1}{16}$ is put through a Lite Reduce ASPP Block - a smaller version of ASPP [2], to create a multi-scale representation of the feature map.

2.3 Recurrent Decoder

The Decoder uses ConvGRU blocks to incorporate temporal information, ensuring consistent alpha matte predictions across frames. The feature map from the encoder at scale $\frac{1}{16}$ is processed through a Bottleneck Block consisting of ConvGRU and Bilinear upsampling. This step helps maintain temporal consistency in predictions.

For upsampling, three blocks combine outputs from the previous layers and the corresponding encoder feature maps. Each block applies convolution, Group Normalization, ReLU activation, and ConvGRU, followed by bilinear upsampling.

Finally, the last Upsampling Block operates at $\frac{1}{2}$ scale, after which the output is concatenated with the low-resolution input. The Output Block performs final convolutions and upsampling to produce the high-resolution alpha matte and foreground. ConvGRU is omitted here as temporal consistency is sufficiently captured by earlier blocks.

2.4 Deep Guided Filter

The model’s key is to process input at high resolution, so Deep Guided Filter (DGF), is a must for high-quality prediction. DGF uses image HR and the output

of the last ReLU from the Output Block alongside previously predicted alpha matte and foreground.

2.5 Group Normalization

Training matting networks is time-consuming and sensitive to small details, such as mini-batch size. Previous research [4] highlighted the importance of mini-batch sizes between 6–16 [5, 7] for better accuracy. However, Batch Normalization (BN) works best with batch sizes larger than 16, making it less effective with smaller batches. To address this, Group Normalization (GN) was adopted, as it operates on the channel dimension rather than the batch size and is not affected by the mini-batch size, making it a better fit for this task. It is unlike Batch Normalization, which is very sensitive toward batch size as shown in Table 2. In Table 2, we can see that the smaller the batch size, the higher the validation loss for Batch Normalization. In contrast, the validation loss remains unchanged regardless of the batch size and is equivalent to the Batch Normalization with the largest batch size.

3 Proposed Loss

In the original paper, Lin et al. [9] used a total loss L^M with

$$L^M = L_{\ell 1}^\alpha + L_{lap}^\alpha + 5L_{tc}^\alpha + L_{\ell 1}^F + 5L_{tc}^F. \quad (4)$$

In the Eq. (4), $L_{\ell 1}^\alpha$, $L_{\ell 1}^F$ are the $L1$ loss for both alpha matte and foreground. L_{lap}^α is the Laplacian loss [1] for the alpha matte with a 5×5 Gaussian kernel for the Gaussian pyramids. Finally, L_{tc}^α and L_{tc}^F are the temporal coherence [12] loss on both the alpha matte and foreground. Therefore, we decided to increase the weight to achieve temporal coherence loss. For the others, we informally take the sum of their losses' weight (which will not utilize the temporal dimension). As the alpha matte and foreground prediction are as crucial as one another, a total weight of ten with each temporal coherence loss's weight to be five as Lin et al. [9]'s choice is sufficient. However, as we look into more contemporary works that do not focus on being real-time, we see that there is still room for improvement in semantic segmentation. In particular, the feature loss, mentioned in Subsect. 3.1, is effective in generating perceptually high-quality images in many image enhancement and synthesis tasks; in our case, it is Image Matting [5].

Table 2. Sensitivity to batch size. ResNet-50's validation error [15].

Batch size	32	16	8	4	2
BN	23.6	23.7	24.8	27.3	34.7
GN	24.1	24.2	24.0	24.2	24.1
Δ	0.5	0.5	-0.8	-3.1	-10.6

Therefore, we'll add the two feature losses L_f^α and L_f^F into the Eq. (4) with the same specifications as Subsect. 3.1

$$L^M = L_{\ell 1}^\alpha + L_{lap}^\alpha + 5L_{tc}^\alpha + L_{\ell 1}^F + 5L_{tc}^F + L_f^\alpha + L_f^F. \quad (5)$$

On the other hand, we still keep the same loss used for the segmentation part (cross-entropy).

$$L^S = \hat{S}(-\log(S)) + (1 - \hat{S})(-\log(1 - S)), \quad (6)$$

where S is the ground-truth mask, and \hat{S} is the predicted one.

The test results of our improved method and comparison to Lin et al. [9] will be presented in Sect. 4.

3.1 Feature Loss

The feature loss is used to measure the perceptual quality of the alpha matte. As reported in [5], feature loss is based on the differences between the high-level features extracted from a pre-trained convolutional neural network. It effectively represents perceptually high-quality features in many image enhancement and synthesis tasks. However, it is not easy to directly measure the perceptual quality of an alpha matte. Therefore, Qiqi Hou et al. [5] proposed a solution to composite the ground-truth foreground image onto the black background using the alpha matte and then measure the perceptual quality of the composition result.

$$L_f^\alpha = \sum_{layer} \|\phi_{layer}(\alpha \odot F) - \phi_{layer}(\hat{\alpha} \odot F)\|_2^2, \quad (7)$$

where α , F indicates the ground-truth alpha matte, foreground respectively, and ϕ_{layer} indicates the features output by the layer in a pre-trained VGG-16 network [10]. To compute the features, their method uses [conv1-2, conv2-2, conv3-3, conv4-3] layers.

When computing the feature loss for the predicated foreground image, the feature loss L_f^F is calculated on the composition result using the ground-truth alpha matte with the foreground image.

$$L_f^F = \sum_{layer} \|\phi_{layer}(\alpha \odot F) - \phi_{layer}(\alpha \odot \hat{F})\|_2^2. \quad (8)$$

4 Experiments and Results

4.1 Dataset

The data for the matting network is divided into three categories: matting, background, and segmentation datasets as listed in Table 3.

Table 3. Summarization of the datasets.

Category	Frames	Dataset
Matting	240,762	VideoMatte240K
	439	ImageMatte
Background	311,376	Video Backgrounds
	9,958	TCB
Segmentation	118,287	COCO
	2,985	YouTubeVIS 2021
	5,711	Supervisely Person Dataset

- **Matting Dataset:** This includes foregrounds and alpha mattes, with subjects being humans. It is split into image and video formats, comprising 439 pairs from the Adobe Image Matting (AIM) and Distinction-646 (D646) datasets.
- **Background Dataset:** This contains 3,117 preprocessed video backgrounds [12] and 9,958 images from our TCB collection, focusing on common home environments for video calls.
- **Segmentation Dataset:** Used to address lighting differences between foreground and background, this dataset includes COCO, Supervisely Person Dataset [13], and YouTubeVIS 2021 [16].

4.2 Implementation Details

During training, the inputs and outputs are 5-dimensional tensors (B, T, C, H, W), where B is the batch size, T is the sequence length, and C is the number of channels. The process is divided into four stages, based on Robust Video Matting [9]. We used a batch size of one on a system with 64 CPU cores, 128GB RAM, and an NVIDIA Tesla T4 GPU. To accommodate our less robust infrastructure, we fine-tuned parameters to align with [9].

4.3 Metrics

There are four metrics are used in this research:

- **Spatial Gradient:** This metric evaluates the sharpness of results [14].
- **Connectivity:** This loss measures pixel connectedness, applied only to alpha mattes to avoid errors from noisy foreground ground-truths [11].
- **dtSSD:** A temporal-coherence metric, dtSSD [3] detects unexpected α changes while ignoring consistent errors, based on the metric SSD.

4.4 Evaluation Profile

We use the same test dataset for both models: the original by Lin et al. [9] and our modified version. Both were trained on the same dataset with identical parameters to compare the effects of our changes. Each model was selected after the loss had converged.

For all the metrics discussed previously alongside MSE, the lower the score, the better the model is. Connectivity is not used on high-resolution test datasets because it is too expansive to compute (Table 4).

Our work has surpassed most metrics, especially Spatial Gradient and dtSSD, meaning the model will most likely perform better on videos with complex backgrounds. This improvement is primarily thanks to the addition of GN because it works perfectly for our segmentation task by reducing errors for small batch sizes. Our background dataset crawled from the Internet works even better with meeting videos since we have thoroughly chosen backgrounds that fit this use case well. Figure 1 shows that we can capture objects like glasses in a background of a meeting, and the results are also sharper (fewer missing random pixels).

As for MSE, the discrepancy and inconsistency is due to feature loss. Since the layers are extracted from a pre-trained VGG16 on the ImageNet dataset, the model is prone to calculating the loss more generally with the feature extracted.

Connectivity is also inconsistent since the changes we made weren't substantially affecting how well the boundary of alpha mattes are preserved. Improving this would require more in-depth changes to the layers of the whole architecture.

Table 4. Results comparison (VM - Video Matting dataset).

Dataset	Method	MSE	Grad	Conn	dtSSD
VM (low-res)	Original	1.41	1.07	0.46	1.52
512 × 288	Ours	1.33	1.08	0.47	1.43
D646 (low-res)	Original	5.56	3.98	2.92	1.16
512 × 512	Ours	4.11	3.54	2.55	1.09
AIM (low-res)	Original	12.21	6.92	5.29	1.88
512 × 512	Ours	20.92	6.58	7.67	1.63
VM (high-res)	Original	2.47	12.17		2.04
1920 × 1080	Ours	2.22	11.63		1.95
D646 (high-res)	Original	5.63	34.10		1.42
2048 × 2048	Ours	7.37	33.08		1.39
AIM (high-res)	Original	12.64	43.61		2.19
2048 × 2048	Ours	19.30	41.98		1.94

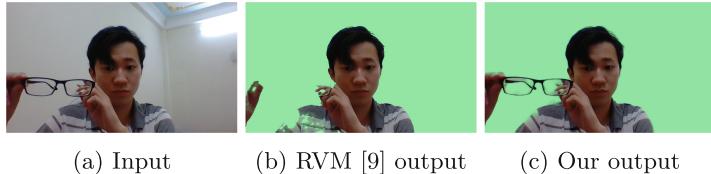


Fig. 1. Example of a frame in a real-life meeting.

5 Conclusion

In this paper, we have implemented a video matting model based on the Encoder-Decoder architecture, with our main contributions being modifications to achieve adequate results for our problem. With the addition of Feature loss and Group Normalization, we have achieved perceptually plausible results while lowering the original parameters. The test scores show that our model favors video inputs since both dtSSD and Spatial gradient yield better results, where the former implies better temporal coherence while the latter implies sharper output. Improving other scores more consistently would be our aim in the future. Furthermore, we can adjust hyperparameters or network layers to optimize the model to process videos more accurately and put it into practical use with more use cases, especially in the film industry.

Acknowledgments. We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

References

1. Burt, P., Adelson, E.: The laplacian pyramid as a compact image code. *IEEE Trans. Commun. (TCOM)* **31**(4), 532–540 (1983). <https://doi.org/10.1109/TCOM.1983.1095851>
2. Chen, L.C., et al.: Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **40**, 834–848 (2018). <https://doi.org/10.1109/TPAMI.2017.2699184>
3. Erofeev, M., Gitman, Y., Vatolin, D., Fedorov, A., Wang, J.: Perceptually motivated benchmark for video matting. In: British Machine Vision Conference (BMVC) (2015)
4. Forte, M., Pitié, F.: F, b, alpha matting (2020). <https://arxiv.org/abs/2003.07711>
5. Hou, Q., Liu, F.: Context-aware image matting for simultaneous foreground and alpha estimation. In: The International Conference on Computer Vision (ICCV) (2019)
6. Ke, Z., Sun, J., Li, K., Yan, Q., Lau, R.W.: Modnet: real-time trimap-free portrait matting via objective decomposition. In: The AAAI Conference on Artificial Intelligence (AAAI) (2022)
7. Li, Y., Lu, H.: Natural image matting via guided contextual attention. In: The AAAI Conference on Artificial Intelligence (AAAI) (2020)

8. Lin, S., Ryabtsev, A., Sengupta, S., Curless, B.L., Seitz, S.M., Kemelmacher-Shlizerman, I.: Real-time high-resolution background matting. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
9. Lin, S., Yang, L., Saleemi, I., Sengupta, S.: Robust high-resolution video matting with temporal guidance. In: The IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2022)
10. Liu, S., Deng, W.: Very deep convolutional neural network based image classification using small training sample size. In: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR) (2015)
11. Rhemann, C., Rother, C., Wang, J., Gelautz, M., Kohli, P., Rott, P.: A perceptually motivated online benchmark for image matting. In: IEEE Conference on Computer Vision and Pattern Recognition (ICCV) (2009)
12. Sun, Y., Wang, G., Gu, Q., Tang, C.K., Tai, Y.W.: Deep video matting via spatio-temporal alignment and aggregation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
13. supervise.ly: Supervisely person dataset. <https://supervise.ly/explore/projects/supervisely-person-dataset-23304/datasets>. Accessed 24 Mar 2022
14. Tang, J., Aksoy, Y., Oztireli, C., Gross, M., Aydin, T.O.: Learning-based sampling for natural image matting. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
15. Wu, Y., He, K.: Group normalization. Int. J. Comput. Vision (IJCV) **128**, 742–755 (2020). <https://doi.org/10.1007/s11263-019-01198-w>
16. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: Proceedings: IEEE/CVF International Conference on Computer Vision (ICCV) (2019)

Intelligent Systems



An Increased Performance of MVDR Beamformer in Diffuse Noise Field

Quan Trong The^(✉)

Faculty of Information Security, The Posts and Telecommunications Institute of Technologies (PTIT), Hanoi, Vietnam
theqt@ptit.edu.vn

Abstract. The essential robust performance has led to advance in speech enhancement research in annoying adverse noise environments. Multi - microphone systems, which have widely commonly installed in almost all speech applications, have shown improvements in comparison with single - channel approach. Nowadays, the use of microphone array (MA) technology has become popular, because MA has the advantage of exploiting the directional spatial diversity toward the speech source while suppressing background noise at the same time. Minimum Variance Distortionless Response (MVDR) beamformer is one of the most useful beamforming techniques for extracting the desired target talker and obtaining the high diversity. Unfortunately, in many speech acoustic devices, such as voice-controlled systems, hearing aids, teleconference systems, mobile phones, the recorded MA signals are often degraded due to surrounding noise, third-party talkers and complex environments. These reasons cause the corrupted speech quality, speech intelligibility and unsatisfactory perceptual for the listener. This paper discusses an effective post-filtering by taking into account the statistics of the MA properties for further removing the remaining noise component. The conducted numerical experiments has shown the effectiveness of the proposed method in reducing noise level to 9.5 dB, increasing the speech quality in the term of the signal-to-noise ratio (SNR) from 7.3 to 10.7 dB. The suggested method can be applied into multi-channel systems in various types of recording scenarios.

Keywords: Microphone array · minimum variance distortionless response · beamforming technique · speech quality · the signal-to-noise ratio · post-filtering

1 Introduction

Noise suppression in speech communication is still an unsolved problem in the signal processing society, although there is numerous research work, which has been devoted to dealing with these tasks. The problem is that the mixture of clean speech and environmental noise overlap in the time - frequency domain, which causes the difficulty of segmenting sound boundaries. As in Fig. 1, the

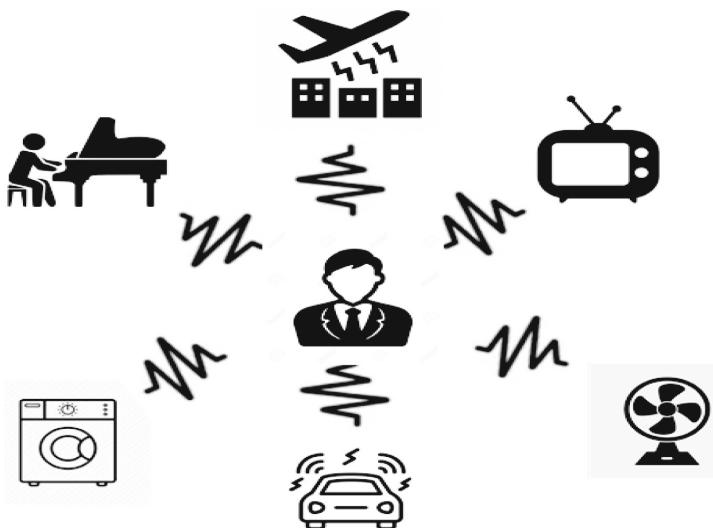


Fig. 1. There is various types of noise around human - life.

existence of various types of noise, the essential core of speech enhancement is finding appropriate signal processing algorithm to extract the original speech component. One way of separating the desired target speaker is using a single - channel approach, such as spectral subtraction (SS). By applying many signal processing algorithms for estimating noise power, this method is very useful in stationary noise for achieving clean speech. Unfortunately, in many complex, annoying and non-stationary situations, SS often degrades the speech quality and decreases the satisfactory perceptual for the listener. Therefore, the use of MA was a considerable solution to avoid the drawback of single - channel approach because of using different characteristics to achieve spatial filtering in MA beamforming technique as in Fig. 2.

MA signal processing has been installed, implemented, and successfully dealing with numerous complex tasks of speech enhancement, including source separation, speaker diarization, speech recognition, surveillance devices, smart home, voice - controlled. Adaptive beamforming technique is a promising method for suppressing background noise level, extracting the clean speech and improving the overall signal processing system's performance.

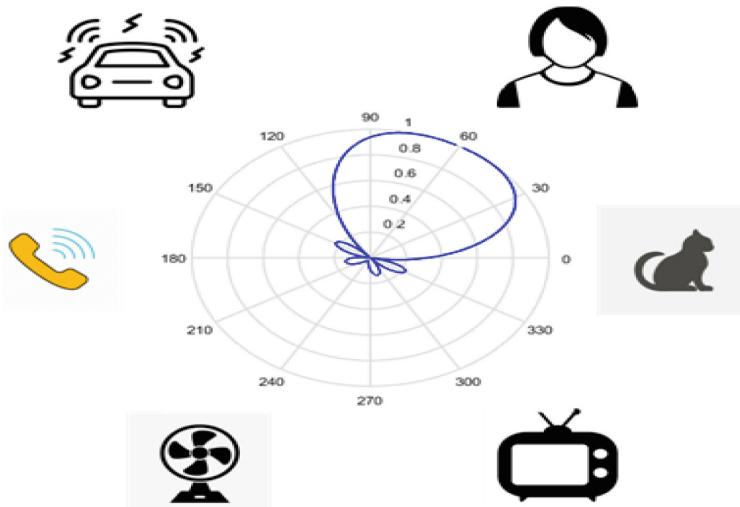


Fig. 2. The advantage of high directional beampattern toward the sound source.

The MA beamforming algorithm can be categorized into two groups: fixed and adaptive beamformer. Fixed beamformer use the constant coefficient, and delay and sum - DAS [1,2] is one representation. Adaptive beamformer include, differential microphone array - DIF [3–5], linearly constrained minimum variance - LCMV [3,6,7] and generalized sidelobe canceller - GSC [8–10], and mimimum variance distortionless response - MVDR. The scheme of MA beamforming is described in Fig. 3.

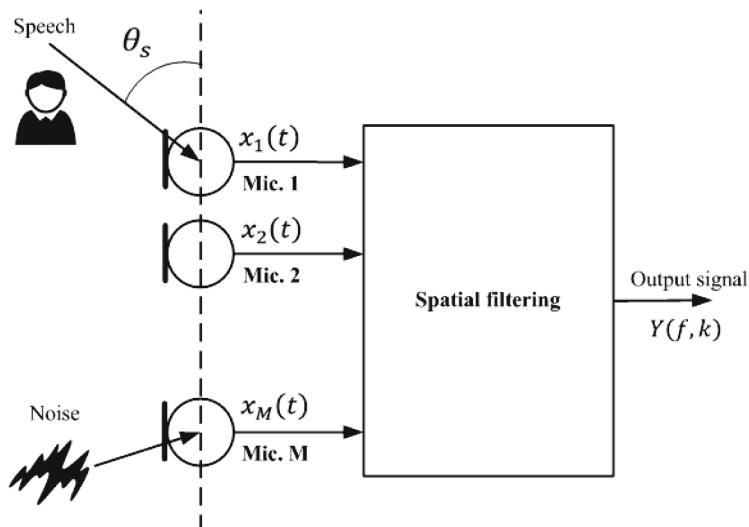


Fig. 3. The principal working of microphone array beamforming technique.

Since the background noise greatly affected the signal processing systems' performance, an adaptive minimum variance distortionless response (MVDR) beamformer served as an effective solution for saving the original speech component at a certain direction while removing all interfering background noise. MVDR has attracted many scholars and researchers in developing, enhancing for adapting rapidly changing environments in practical scenarios.

At first, MVDR was proposed by Capon [11]. It minimizes the total output noise power while extracting the clean speech at a certain direction without distortion according to constrained expression; however, the inaccurate estimation of steering vector greatly dramatically declines the performance of MVDR beamformer. For alleviating the dispersive degree of the eigenvalues of the covariance matrix, which play a crucial role, diagonal loading technique uses a determined small constant value to the main diagonal loading of the covariance matrix for increasing the steered beampattern toward the sound source.

The MVDR frequency - domain expression for noise reduction in acoustic devices, hearing aid, hands - free mobile phone was first presented in [12–15]. The MVDR beamformer requires an accurate estimation of steering vector, a priori information about MA geometry, the DoA of speaker, the acoustic scenario, the observed noise covariance matrix [16]. The imprecise estimation of these above properties can lead significant degrading evaluation due to reasons, such as: the imperfect MA calibration, microphone coupling, microphone mismatch, the different microphone sensitivities [17–19], which reduces both speech quality and speech intelligibility [20–23].

MVDR beamformer is very sensitive with the error of estimation of steering vector, which according to the predefined direction of arrival (DoA) of interest talker, hence the performance often degrades. In this paper, the authors proposed an effective post-filtering, which removes the remaining background noise and saves the original speech component without speech distortion. The illustrated experiment showed the improvement of signal-to-noise ratio (SNR) from 7.3 to 10.7 dB, noise reduction to 9.5 dB in a realistic recording environment with presence of several noise sources. The promising achieved result has confirmed the effectiveness of the suggested method to be applied into MVDR beamforming technique and other MA - based signal processing methods.

The rest of this contribution is organized as follows. The Sect. 1 introduce the principal working of MA and MVDR beamformer. Section 2 presents the model signal of MVDR beamformer in dual - microphone arrays (DMA2). Section 3 described the author's proposed post - Filtering and Sect. 4 illustrates the perspective experiment in real-life conference rooms. Section 5 concludes the suggested method and the author's future research direction.

2 The Model Signals

In this section, the author presents the scheme of principal working of MVDR beamformer [24–27] in for both noise reduction and speech enhancement at the same time. The diagram of signal processing with dual - microphone systems is described in Fig. 4.

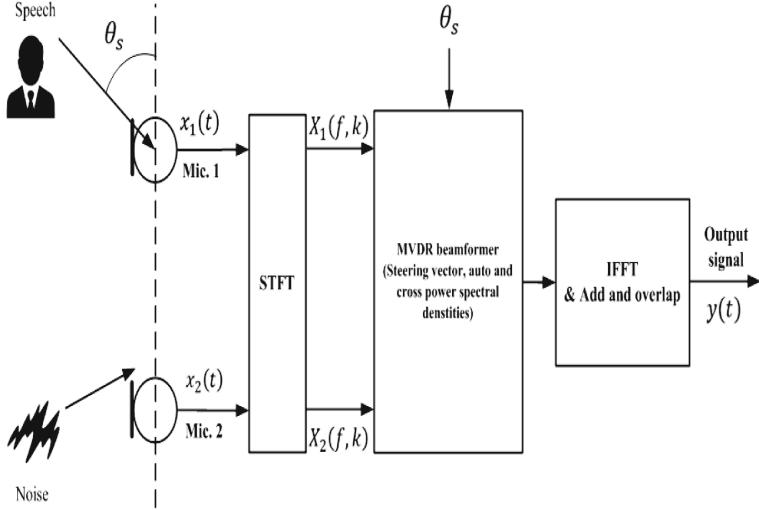


Fig. 4. The diagram of MVDR beamformer in the frequency-domain.

The representation of received MA signals $X_1(j\omega, m), X_2(j\omega, m)$ in the frequency-domain can be expressed by the below equations as:

$$X_1(j\omega, m) = S(j\omega, m)e^{j\Phi_s} + V_1(j\omega, m) \quad (1)$$

$$X_2(j\omega, m) = S(j\omega, m)e^{-j\Phi_s} + V_2(j\omega, m) \quad (2)$$

where f, m denote the current considered frequency and frame, $\omega = 2\pi f$, $S(j\omega, m)$ is the original clean speech data, $V_1(j\omega, m), V_2(j\omega, m)$ are interference, background noise or third-party talker. With $c = 343$ (m/s) is the sound speed propagation in the fresh air, d is the range between two installed microphones, $\tau_0 = d/c$, θ_s is the predefined direction of arrival of useful signal relative to the axis of MA systems, $\Phi_s = \pi f \tau_0 \cos(\theta_s)$.

For more convenient, we denote these symbols:

$\mathbf{X}(j\omega, m) = [X_1(j\omega, m) \quad X_2(j\omega, m)]^T$,
 $\mathbf{V}(j\omega, m) = [V_1(j\omega, m) \quad V_2(j\omega, m)]^T$ and $\mathbf{D}(j\omega, \theta_s) = [e^{j\Phi_s} \quad e^{-j\Phi_s}]^T$ with $(\cdot)^T$ indicates transpose operator, and $\mathbf{D}(j\omega, \theta_s)$ is phase shift vector.

Equation (1–2) can be rewritten in vector form as the following way:

$$\mathbf{X}(j\omega, m) = S(j\omega, m)\mathbf{D}(j\omega, \theta_s) + \mathbf{V}(j\omega, m) \quad (3)$$

The problem of digital signal processing is finding an appropriate coefficients $\mathbf{W}(j\omega, m)$, which ensure the outperformed obtained signal $\hat{S}(j\omega, m)$ approximately as the original speech data $S(j\omega, m)$.

$$\hat{S}(j\omega, m) = \mathbf{W}^H(j\omega, m)\mathbf{X}(j\omega, m) \quad (4)$$

where $(\cdot)^H$ is the symbol of Hermitian conjugation.

The constrained criteria of minimum the total noise power at the beamformer's output while saving the speech component without speech distortion formulates the MVDR beamformer. The described expression of MVDR beamforming technique can be represented as:

$$\begin{aligned} \min_{\mathbf{W}(j\omega, m)} & \quad \mathbf{W}^H(j\omega, m) \mathbf{P}_{VV}(j\omega, m) \mathbf{W}(j\omega, m) \\ \text{s.t.} & \quad \mathbf{W}^H(j\omega, m) \mathbf{D}(j\omega, \theta_s) = 1 \end{aligned} \quad (5)$$

where $\mathbf{P}_{VV}(j\omega, m) = E\{\mathbf{V}(j\omega, m)\mathbf{V}^*(j\omega, m)\}$ is cross spectral matrix of received MA signals at current considered frame.

The solved filter coefficients of MVDR beamformer is derived as:

$$\mathbf{W}(j\omega, m) = \frac{\mathbf{P}_{VV}^{-1}(j\omega, m) \mathbf{D}(j\omega, \theta_s)}{\mathbf{D}^H(j\omega, \theta_s) \mathbf{P}_{VV}^{-1}(j\omega, m) \mathbf{D}(j\omega, \theta_s)} \quad (6)$$

But in real recording situations, the a priori information about noise is not always available and not easily calculating, so the cross-spectral matrix of capture MA signals used instead of.

$\mathbf{P}_{XX}(j\omega, m) = E\{\mathbf{X}(j\omega, m)\mathbf{X}^*(j\omega, m)\}$ can be calculated as:

$$\mathbf{P}_{XX}(j\omega, m) = \begin{bmatrix} P_{X_1 X_1}(j\omega, m) * 1.001 & P_{X_1 X_2}(j\omega, m) \\ P_{X_2 X_1}(j\omega, m) & P_{X_2 X_2}(j\omega, m) * 1.001 \end{bmatrix} \quad (7)$$

where $P_{X_i X_i}(j\omega, m), P_{X_i X_j}(j\omega, m - 1), i, j \in \{1, 2\}$ is derived as:

$$P_{X_i X_j}(j\omega, m) = (1 - \alpha)P_{X_i X_j}(j\omega, m - 1) + \alpha X_i^*(f, k)X_j(j\omega, m) \quad (8)$$

with α is the smoothing parameter, which in the range {0...1}.

So, the final filter coefficients of MVDR beamformer is derived as:

$$\mathbf{W}(j\omega, m) = \frac{\mathbf{P}_{XX}^{-1}(j\omega, m) \mathbf{D}_s(j\omega, \theta_s)}{\mathbf{D}_s^H(j\omega, \theta_s) \mathbf{P}_{XX}^{-1}(j\omega, m) \mathbf{D}_s(j\omega, \theta_s)} \quad (9)$$

The output signal of MVDR beamformer can be derived as:

$$Y_{MVDR}(j\omega, m) = \mathbf{W}^H(j\omega, m) \mathbf{X}(j\omega, m) \quad (10)$$

And then, $Y_{MVDR}(j\omega, m)$ transformed into the time domain.

Unfortunately, the misplacement, the error of estimation of DoA, the different MA sensitivities, the microphone mismatches, the moving talker during the conversation, the non-directional noise significantly corrupt the MVDR beamformer and decrease the speech quality in real-life performance. Therefore, the need for improvement of speech enhancement, increasing the speech quality in the term of the signal-to-noise ratio has attracted many scholars, engineering. Multi-channel approach has the advantage of exploiting the properties of array signals, the surrounding environment's characteristics. In the next section, the author proposed an effective post-filtering, which use the imaginary part of coherence between two microphone arrays signals for reducing the noise level.

3 The Proposed Post-filtering

Because of the uncorrelated between $S(j\omega, m)$ and $V_1(j\omega, m), V_2(j\omega, m)$, $E\{S(j\omega, m)V_2^*(j\omega, m)\} = E\{S^*(j\omega, m)V_1(j\omega, m)\} = 0$ the cross spectral power densities of $X_1(j\omega, m)$ and $X_2(j\omega, m)$ can be derived as:

$$E\{X_1(j\omega, m)X_2^*(j\omega, m)\} = E\{|S(j\omega, m)|^2\}e^{j2\Phi_s} + E\{V_1(j\omega, m)V_2^*(j\omega, m)\} \quad (11)$$

$$= \sigma_s^2 e^{j2\Phi_s} + \sigma_n^2 \quad (12)$$

where σ_s^2, σ_n^2 is the variance of clean speech and background diffuse noise field.

We take the imaginary part of Eq. (13):

$$\text{Im}\{E\{X_1(j\omega, m)X_2^*(j\omega, m)\}\} = -\sigma_s^2 \sin(2\Phi_s); \quad (13)$$

Therefore, the variance of useful clean speech can be determined as:

$$\sigma_s^2(j\omega, m) = -\frac{\text{Im}\{E\{X_1(j\omega, m)X_2^*(j\omega, m)\}\}}{\sin(2\Phi_s) + \beta} \quad (14)$$

where β is a small constant.

Hence, the variance of diffuse noise field:

$$\sigma_v^2(j\omega, m) = E\{X_1(j\omega, m)X_2^*(j\omega, m)\} - \sigma_s^2(j\omega, m)e^{j2\Phi_s}; \quad (15)$$

A temporal the signal - to - noise ratio (SNR) was calculate as the following equation:

$$S\hat{N}R(j\omega, m) = \frac{\sigma_s^2(j\omega, m)}{\sigma_v^2(j\omega, m)} \quad (16)$$

Therefore, Wiener filter - based the author's proposed post - Filtering was computed as:

$$\text{proPF}(j\omega, m) = \frac{S\hat{N}R(j\omega, m)}{1 + S\hat{N}R(j\omega, m)} \quad (17)$$

The proposed post - Filtering is applied to the MVDR beamformer's output signal and allows reducing the remaining surrounding noise level to improve speech enhancement and speech quality.

The enhanced MVDR beamformer's output signal is given by:

$$\hat{Y}_{MVDR}(j\omega, m) = \text{proPF}(j\omega, m)Y_{MVDR}(j\omega, m) \quad (18)$$

The appealing properties of the suggested technique is using the priori information observed MA signals to take into account an appropriate post - Filtering for extracting the desired target speaker while suppressing background noise and improving speech enhancement.

In the next section, the author illustrated a perspective experiment to verify the effectiveness of the suggested technique to remove background noise level while recovering the speech component with speech distortion and improve the speech quality at the same time.

4 Simulation Study

In section, a stand speaker at distance $L = 3$ (m) to the relative axis of DMA2 in the presence of various types of noises. The conducted simulation verifies the effectiveness of the proposed method in saving the original speech component while suppressing noise power level. An objective measurement [28] was used for calculating the speech quality in the term of the signal-to-noise ratio (SNR) between the observed MA signals, the processed signal by MVDR beamformer and the suggested post - filtering.

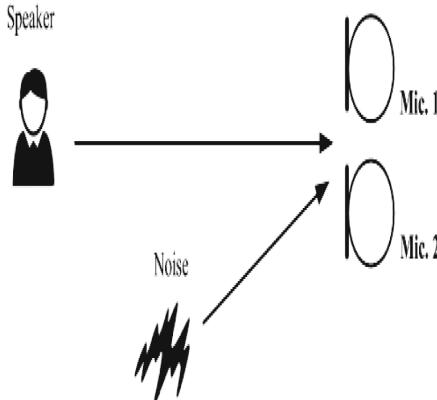


Fig. 5. The scheme of illustrated experiments.

The scheme of experiment was shown in Fig. 5.

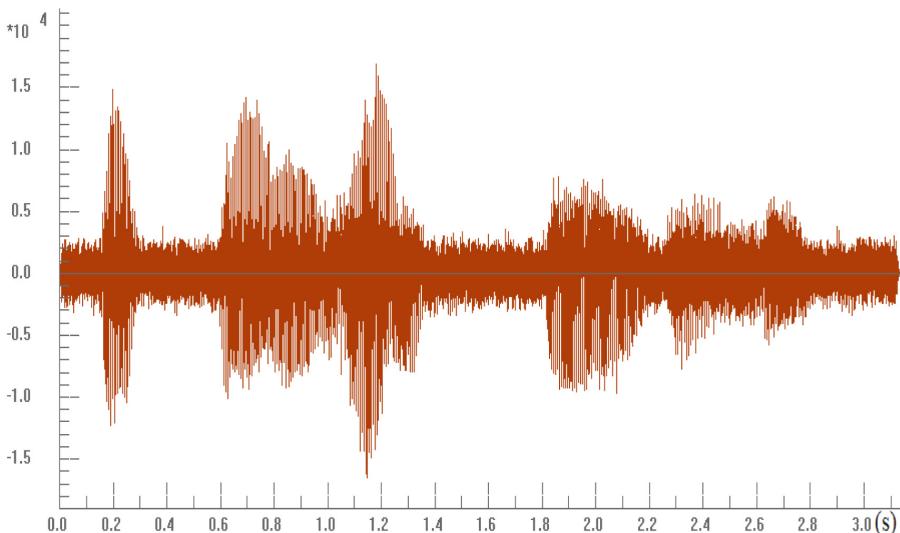


Fig. 6. The original waveform of recording MA signals.

Table 1. The signal-to-noise ratio (dB).

Method Estimation	Microphone array signal	MVDR beamformer	The proposed post - Filtering
NIST STNR	12.0	15.5	22.8
WADA SNR	7.1	7.7	18.4

The direction of interest in clean speech is $\theta_s = 90(deg)$. For recording MA signals, these parameters were applied: the sampling frequency $F_s = 16(kHz)$, overlap 50%. The observed MA signals were shown in Fig. 6.

$NFFT = 512$, smoothing parameter $\alpha = 0.1$ for computing the auto and cross power spectral densities to perform MVDR beamfomer and the proposed post - filtering.

Figure 7 presents the output signal by the conventional MVDR beamformer.

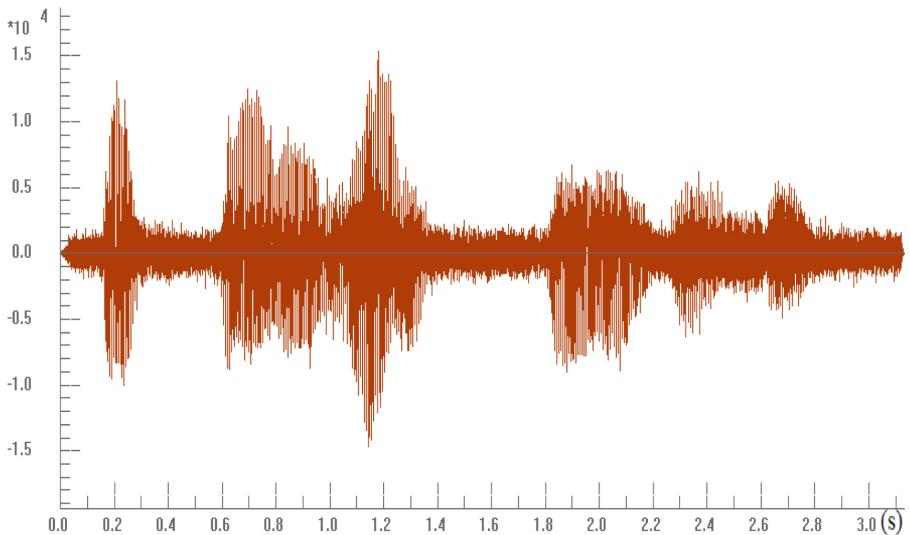
**Fig. 7.** The original waveform of processed signal by conventional MVDR beamformer.

Figure 8 describes the promising signal by applying suggested post - filtering.

The comparison of energy between the original MA signals, processed by conventional MVDR beamformer and the suggested post - filtering is expressed in Fig. 9.

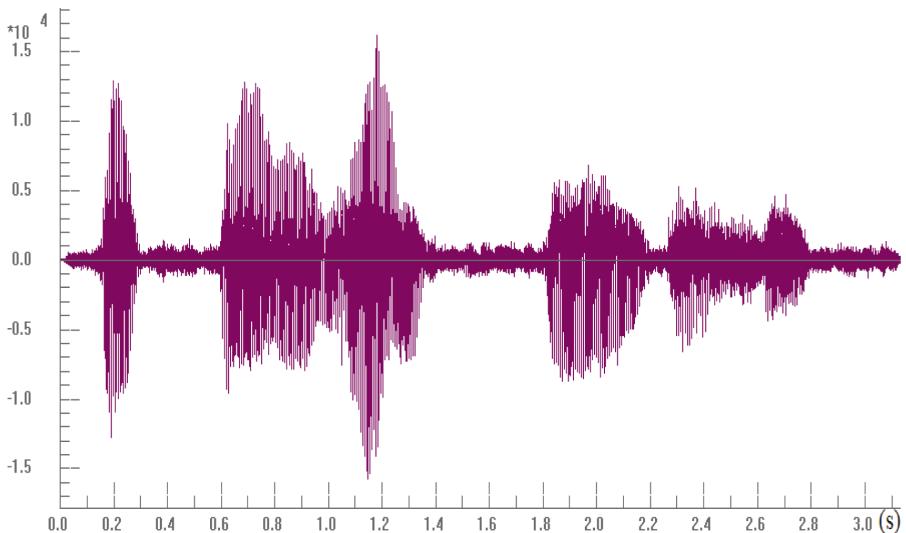


Fig. 8. The original waveform of processed signal by using the additive proposed post-Filtering.

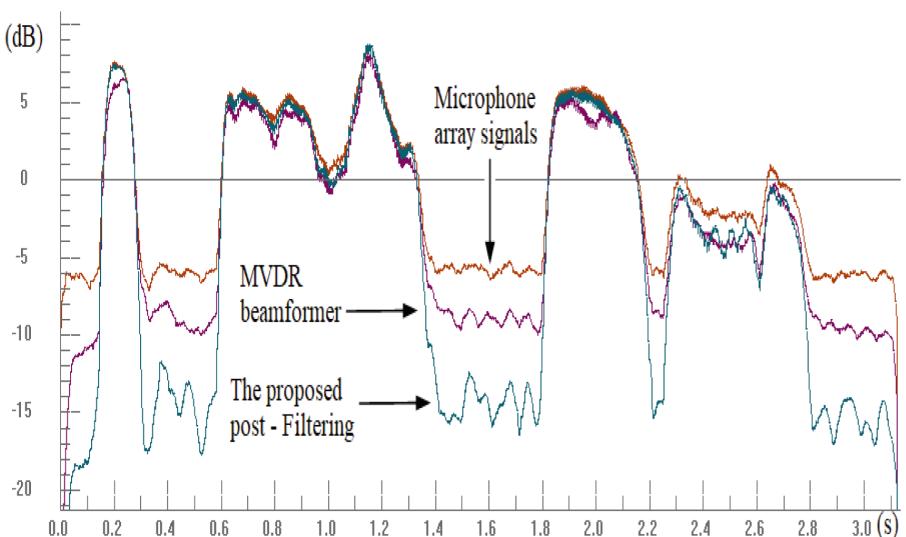


Fig. 9. The comparison of energy between MA signals, the processed signals by MVDR beamformer and the proposed post-Filtering.

From the above figures, we can see that the author's technique not only saved the speech component without speech distortion but also it removed more noise level to 9.5 (dB) and enhanced the speech quality from 7.3 to 10.7 (dB) as in

Table 1. The promising speech enhancement of the suggested method was confirmed through the observed waveform and energy of output processed signals.

MVDR beamformer is one of the most common beamforming techniques, which is commonly installed into most acoustics speech applications due to its capability of both noise reduction and focusing the highly directional beam-pattern toward the sound source. However, because of misplacement, incorrect transfer function, microphone mismatches, imprecise DoA of useful clean speech, the MVDR beamformer's evaluation often degraded, the remaining noise level effects on the speech quality, the perceptual listener. In this experiment, the suggested post - filtering allows alleviating the background noise, improving the outperformed signal. This method can be incorporated with other MA properties to apply into multi-channel signal processing.

5 Conclusion

In almost hands-free speech applications, the speech quality of desired target recording talker often degraded by unwanted factor, such as: background noise, interference, the sound of surrounding transport vehicle or third - party speaker, which lead to total unintelligibility of the speech and decreases the performance of designed speech manipulation system. Therefore, efficient and adaptive signal processing methods are required in most acoustic equipment. Using MA can help improve the signal-to-noise ratio (SNR) by exploiting the spatial transfer functions, the properties of noise fields. However, in many particular recording situations, the overall performance of MA can be corrupted, because of various unavoidable reasons. In this article, the authors proposed an additive post-filtering, which ensures removing the remaining noisy component and improves the satisfactory perceptual listener. The illustrated experiment has confirmed the effectiveness of suggested techniques in a realistic recording environment. In the future, the authors study the characteristics of the environment, the direction of arrival of interest speakers and configuration of MA to enhance MVDR beamformer's performance.

References

1. Kim, H., Kang, K., Shin, J.W.: Factorized MVDR deep beamforming for multi-channel speech enhancement. *IEEE Signal Process. Lett.* **29**, 1898–1902 (2022). <https://doi.org/10.1109/LSP.2022.3200581>
2. Chodingala, P.K., Chaturvedi, S.S., Patil, A.T., Patil, H.A.: Robustness of DAS beamformer over mvdr for replay attack detection on voice assistants. In: 2022 IEEE International Conference on Signal Processing and Communications (SPCOM), Bangalore, India, pp. 1–5 (2022). <https://doi.org/10.1109/SPCOM55316.2022.9840757>
3. Zhao, X., Luo, X., Huang, G., Chen, J., Benesty, J.: Differential beamforming with null constraints for spherical microphone arrays. In: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, pp. 776–780 (2024). <https://doi.org/10.1109/ICASSP48485.2024.10446768>

4. Yang, X., Wei, J.: DMANET: deep learning-based differential microphone arrays for multi-channel speech separation. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore, pp. 4363–4367 (2022). <https://doi.org/10.1109/ICASSP43922.2022.9747725>
5. Luo, X., Jin, J., Huang, G., Chen, J., Benesty, J.: Design of steerable linear differential microphone arrays with omnidirectional and bidirectional sensors. *IEEE Signal Process. Lett.* **30**, 463–467 (2023). <https://doi.org/10.1109/LSP.2023.3267969>
6. Frank, A., Ben-Kish, A., Cohen, I.: Constant-beamwidth linearly constrained minimum variance beamformer. In: 30th European Signal Processing Conference (EUSIPCO), Belgrade, Serbia, pp. 50–54 (2022). <https://doi.org/10.23919/EUSIPCO55093.2022.9909899>
7. Schreibman, A., Barnov, A., Gendelman, A., Tzirkel, E.: RTF based LCMV beamformer with multiple reference microphones. In: 28th European Signal Processing Conference (EUSIPCO), Amsterdam, Netherlands, pp. 181–185 (2020). <https://doi.org/10.23919/Eusipco47968.2020.9287468>
8. Chakrabarty, T., Saha, R.K., Faysal, M.F., Bishal, M.R., Hossain, M.S.: Performance investigation of robust linearly constrained minimum variance beamforming for uniform circular array. In: IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, pp. 1201–1204 (2020). <https://doi.org/10.1109/TENSYMP50017.2020.9230852>
9. The, Q.T., Huy, N.B., Anh, P.T.: Spectral mask - based technique for improving generalized sidelobe canceller beamformer's evaluation. In: Seminar on Signal Processing, Saint Petersburg, Russian Federation, pp. 106–110 (2023). <https://doi.org/10.1109/IEEECONF60473.2023.10366094>
10. Wang, J., Yang, F., Guo, J., Yang, J.: Robust adaptation control for generalized sidelobe canceller with time-varying gaussian source model. In: 31st European Signal Processing Conference (EUSIPCO), Helsinki, Finland, pp. 16–20 (2023). <https://doi.org/10.23919/EUSIPCO58844.2023.10289801>
11. Capon, J.: High-resolution frequency-wavenumber spectrum analysis. *Proc. IEEE* **57**(8), 1408–1418 (1969). <https://doi.org/10.1109/PROC.1969.7278>
12. Baumgartel, R.M.: Comparing binaural pre-processing strategies I: Instrumental evaluation. *Trends Hear.* **19**, 1–16 (2015). <https://doi.org/10.1177/2331216515617>
13. Cornelis, B., Doclo, S., Van den Bogaert, T., Moonen, M., Wouters, J.: Theoretical analysis of binaural multimicrophone noise reduction techniques. *IEEE Trans. Audio Speech Lang. Process.* **18**(2), 342–355 (2009). <https://doi.org/10.1109/TASL.2009.2028374>
14. Gannot, S., Burstein, D., Weinstein, E.: Signal enhancement using beamforming and nonstationarity with applications to speech. *IEEE Trans. Signal Process.* **49**(8), 1614–1626 (2001). <https://doi.org/10.1109/78.934132>
15. Hadad, E., Marquardt, D., Doclo, S., Gannot, S.: Theoretical analysis of binaural transfer function MVDR beamformers with interference cue preservation constraints. *IEEE/ACM Trans. Audio Speech Lang. Process.* **23**(12), 2449–2464 (2015). <https://doi.org/10.1109/TASLP.2015.2486381>
16. Hadad, E., Doclo, S., Gannot, S.: The binaural LCMV beamformer and its performance analysis. *IEEE/ACM Trans. Audio Speech Lang. Process.* **24**(3), 543–558 (2016). <https://doi.org/10.1109/TASLP.2016.2514496>
17. Chen, H.: Robustness analysis of nearfield subband beamformers in the presence of microphone gain and phase errors. *Digital Signal Process.* **23**(5), 1712–1719 (2013). <https://doi.org/10.1016/j.dsp.2013.04.008>

18. Chen, H., Ser, W., Yu, Z.L.: Optimal design of nearfield wideband beamformers robust against errors in microphone array characteristics. *IEEE Trans. Circuits Syst. I Regul. Pap.* **54**(9), 1950–1959 (2007). <https://doi.org/10.1109/TCSI.2007.904667>
19. Vorobyov, S.A.: Principles of minimum variance robust adaptive beamforming design. *Signal Process.* **93**(12), 3264–3277 (2013). <https://doi.org/10.1016/j.sigpro.2012.10.021>
20. Doclo, S., Moonen, M., Van den Bogaert, T., Wouters, J.: Reduced-bandwidth and distributed MWF-based noise reduction algorithms for binaural hearing aids. *IEEE Trans. Audio Speech Lang. Process.* **17**(1), 38–51 (2009). <https://doi.org/10.1109/TASL.2008.2004291>
21. Spriet, A., Moonen, M., Wouters, J.: Spatially pre-processed speech distortion weighted multi-channel Wiener filtering for noise reduction. *Signal Process.* **84**(12), 2367–2387 (2004). <https://doi.org/10.1016/j.sigpro.2004.07.028>
22. Kompis, M., Dillier, N.: Performance of an adaptive beamforming noise reduction scheme for hearing aid applications I: Prediction of the signal-to-noise ratio improvement. *J. Acoust. Soc. Amer.* **109**(3), 1123–1133 (2001). <https://doi.org/10.1121/1.1338557>
23. Cauchi, B., Kodrasi, I., Rehr, R., et al.: Combination of MVDR beamforming and single-channel spectral processing for enhancing noisy and reverberant speech. *EURASIP J. Adv. Signal Process.* **2015**, 61 (2015). <https://doi.org/10.1186/s13634-015-0242-x>
24. Erdim, S., Buck, J.R.: Mitigating multiple moving interferers with the hybrid double zero MVDR beamformer. *IEEE Access* **12**, 111206–111217 (2024). <https://doi.org/10.1109/ACCESS.2024.3437749>
25. Yadav, S., Pal, S., Kumar, A., Aggarwal, M.: Study of MVDR beamformer for a single acoustic vector sensor. In: International Symposium on Ocean Technology (SYMPOL), Kochi, India, pp. 1–6 (2023). <https://doi.org/10.1109/SYMPOL59195.2023.10455004>
26. Zhang, F., Pan, C., Benesty, J., Chen, J.: Directional gain based noise covariance matrix estimation for MVDR beamforming. In: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, pp. 511–515 (2024). <https://doi.org/10.1109/ICASSP48485.2024.10447393>
27. Lagacé, P.-O., Ferland, F., Grondin, F.: Ego-noise reduction of a mobile robot using noise spatial covariance matrix learning and minimum variance distortionless response. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, pp. 3533–3538 (2023). <https://doi.org/10.1109/IROS55552.2023.10342193>
28. SNRVAD. <https://labrosa.ee.columbia.edu/projects/snreval/>



Fusing Models for Classifying Intangible Cultural Heritage Images in the Mekong Delta

Minh-Tan Tran¹, The-Phi Pham¹, Nguyen Thai-Nghe¹,
and Thanh-Nghi Do^{1,2(✉)}

¹ College of Information Technology, Can Tho University, Cantho 94000, Vietnam
dtnghi@cit.ctu.edu.vn

² UMI UMMISCO 209 (IRD/UPMC), UPMC, Sorbonne University, Pierre and Marie Curie University, Paris 6, France

Abstract. Our study aims to classify images of Intangible Cultural Heritage (ICH) in the Mekong Delta, Vietnam. To achieve this purpose, we have built a dataset consisting of images from 17 different ICH categories and manually annotated them. Initially, we fine-tuned recent pre-trained network models, including VGG16, DenseNet, and Vision Transformer (ViT), for classifying our dataset. After that, we trained Logistic Regression (LR) models, called fusing models, which fuse not only various visual features extracted from deep networks but also the output of deep networks to improve the classification accuracy. Our comparative study of the classification performance on the 17-category ICH image dataset shown that our fusing models improve the classification correctness compared to any single fine-tuned one. The first fusing model (LR with visual feature extracted from VGG16, DenseNet, ViT) achieves an accuracy of 66.76%. The second fusing model (LR on top VGG16, DenseNet, ViT) gives an accuracy of 66.49%.

Keywords: Images of the intangible cultural heritage in the Mekong Delta · Image classification · Transfer learning · Fusing model

1 Introduction

The Mekong Delta in Vietnam is a vibrant and culturally rich region with a multitude of intangible cultural heritages (ICH) that play an important role in the lives of its people. Among these are the Ok Om Bok Festival, the Cai Rang Floating Market, the My Long Sea Worship Festival, the Dù Kê theater, the LÀm Chay Festival, the Chàm Riêng Chà Pây art, the Mỹ Long Sea Worship Festival, the Dờn Ca Tài Tứ music, the Bà Chúa Xứ Núi Sam Temple Festival, the Long Hậu boat building, the bamboo weaving, the Bảy Núi Ox Racing Festival, the Nghinh Ông Whale Worship Festival, the Trương Định Festival, the Việc Lê worship ceremony, the Đại lễ Kỳ Yên at Tân Phước Tây Temple, the

Vía Bà Ngū Hành Festival. These cultural practices are not only significant to the local communities but also contribute to the cultural diversity and identity of the nation. Therefore, the research in preserving the ICH of the Mekong Delta region is an extreme importance, such as the documentation, the preservation of the diverse ICH practices, a deeper understanding and interpretation of the cultural significance of various ICH practices, the continuity of cultural practices, the community engagement, the promotion of cultural tourism, contributing to the local economy.

In recent years, there has been growing interest among researchers in the classification of heritage images. Belhi and colleagues [3, 5] highlighted the significant impact of artificial intelligence technologies on cultural heritage digitization. Yasser et al. [25] developed a digital heritage platform aimed at protecting and preserving cultural heritage sites. Jankovic and her team [8, 18, 19] proposed training visual classification models using various algorithms such as multi-layer perceptron, random forest, k -nearest neighbor rough sets, and convolutional neural networks (CNN [20]) with features including edge histogram, color layout, and JPEG coefficients. Jose M. Llamas et al. [21] suggested employing deep learning techniques for categorizing architectural heritage images. The study by Vu [31] focused on transfer learning of deep neural networks for classifying heritage images. Belhi et al. [2, 4] developed deep learning approaches for classifying and annotating cultural data. Ma and colleagues [22–24] created an ontology-based approach for modeling Vietnamese Thai dances. Rapti et al. [27] provided a summary of mining techniques used in the semantic web for cultural heritage data. Fiorucci et al. [13] presented a comprehensive survey on the application of machine learning in cultural heritage. Do and colleagues [10, 11] proposed training support vector machines (SVM [30]) using deep learning features and handcrafted features to classify ICH images.

Our research focuses on the classification of ICH images in the Mekong Delta, Vietnam. This work supports cultural preservation efforts, informs policy decisions, and promotes cultural pride and awareness by understanding and appreciation of the region's rich cultural heritage through educational and promotional materials. To achieve this goal, we have constructed a dataset consisting of images from 17 different ICH categories and manually annotated them. We start with fine-tuning recent pre-trained network models, including VGG16 [29], DenseNet [16], and Vision Transformer (ViT [12]), for classifying our dataset. Subsequently, we trained Logistic Regression (LR [15]) models, referred to as fusing models, which combine various visual features extracted from deep networks and the outputs of these networks to enhance classification accuracy. Our comparative study of the classification performance on the 17-category ICH image dataset demonstrated that our fusing models outperform any single fine-tuned model. The first fusing model, which integrates visual features extracted from VGG16, DenseNet, and ViT, achieved an accuracy of 66.76%. The second fusing model, which places Logistic Regression on top of VGG16, DenseNet, and ViT, attained an accuracy of 66.49%. These results indicate that our fusing models improve classification accuracy compared to individual fine-tuned models.

The remainders of this paper are organized as follows. Section 2 describes how to collect a dataset of ICH images and how to build classification models for ICH images. Section 3 shows the experimental results before conclusions and future works presented in Sect. 4.

2 Proposed Approach

An overview of the classical pipeline (Fig. 1) consists of the data collection, pre-processing, the feature extraction and classifier training (e.g. SVM [30]).

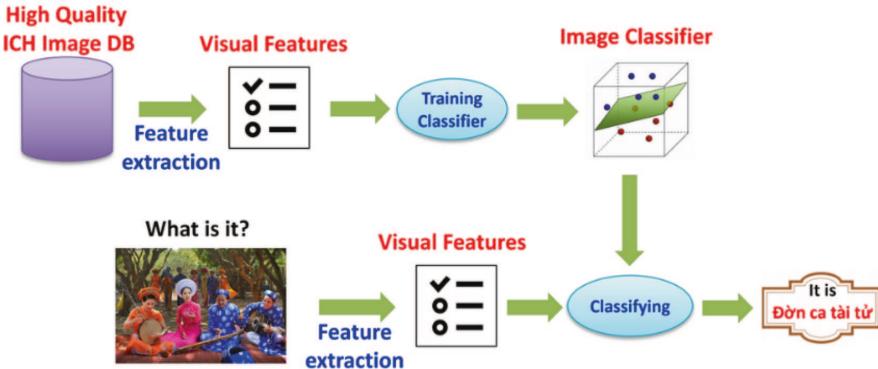


Fig. 1. Pipeline for classifying ICH images

Modern efficient approaches for the image classification are to train convolutional neural networks (CNN [20]) which has the ability to learn visual features from images and the classifier in an unified algorithm, as illustrated in Fig. 2.

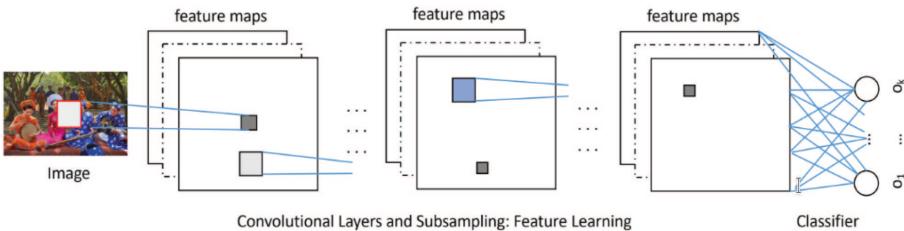


Fig. 2. Convolutional neural networks for classifying ICH images

Our automated classification of intangible cultural heritage (ICH) images involves key steps that leverage these advanced machine learning and computer vision techniques.

2.1 Data Collection of Intangible Cultural Heritage Images

Firstly, we need to build a dataset of ICH images from the Mekong Delta, Vietnam, as no such data currently exists. Our research focuses on the 17 ICH categories, as follows:

1. Dờn ca tài tử: Amateur Music and Singing
2. Nghệ thuật Chàm Riêng Chà Pây: Chàm Riêng Chà Pây Art
3. Nghề dệt chiếu: Mat Weaving Craft
4. Lễ cúng biển Mỹ Long: Mỹ Long Sea Worship Festival
5. Nghệ thuật sân khấu Dù Kê: Dù Kê Theater Art
6. Lễ hội Ok Om Bok: Ok Om Bok Festival
7. Lễ hội miếu Bà Chúa Xứ Núi Sam: Bà Chúa Xứ Shrine Festival at Sam Mountain
8. Đại lễ Kỳ Yên tại Đinh Tân Phước Tây: Kỳ Yên Festival at Tân Phước Tây Communal House
9. Lễ hội Vía Bà Ngũ Hành: Vía Bà Ngũ Hành Festival
10. Lễ hội Làm Chay: Làm Chay Festival
11. Nghề đóng xuồng ghe Long Hậu: Long Hậu Boat Building Craft
12. Nghề dán tre: Bamboo Weaving Craft
13. Tục cúng Việc Lè: Việc Lè Worship Ritual
14. Lễ hội đua bò Bảy Núi: Bảy Núi Bull Racing Festival
15. Lễ hội Nghinh Ông: Nghinh Ông Whale Worship Festival
16. Lễ hội Trương Định: Trương Định Festival
17. Văn hóa chợ nổi Cái Răng: Cái Răng Floating Market Culture

We propose to collect ICH images from Google, leveraging the extensive availability of images in this public repository. We developed a Python program using the iCrawler library [6] to perform image searches with textual queries based on keywords related to the 17 ICH categories and retrieve the relevant images. However, results still yield noisy and irrelevant images. To overcome this issue, we conducted a manual post-processing stage to filter and tag the images appropriately. And then, we obtained an image dataset of respective ICH categories, comprising a total of 7,409 images. Figure 3 shows a sample of images from 17 ICH categories.

2.2 Fusing Models for Classifying Intangible Cultural Heritage Images

Our classification approach bases on deep learning networks such as VGG16 [29], DenseNet [16], and Vision Transformer (ViT [12]), for classifying our ICH dataset. These networks have the ability to learn visual features and classifiers in an unified algorithm which is more effectively for ICH image classification.

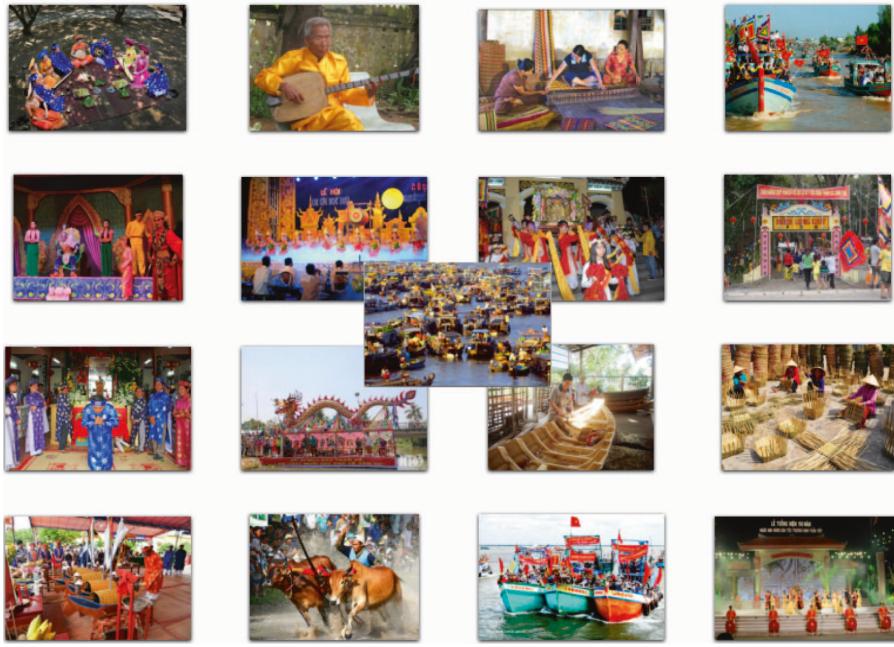


Fig. 3. Image sample of 17 ICH categories in the Mekong delta

VGG16: The VGG16 network architecture [29], introduced by the Visual Geometry Group at the University of Oxford in 2014, consists of 16 weight layers: 13 convolutional layers followed by 3 fully connected layers. The architecture is characterized by its use of small 3×3 convolutional filters, which allow the model to capture intricate patterns and details in images. VGG16 has achieved impressive results on various benchmark datasets, such as ImageNet [9], and has become a foundational model in the field of deep learning.

DenseNet: The DenseNet [16] is introduced by Gao Huang and colleagues in 2016, for improving the flow of information and gradients through the network, making it more efficient and effective. DenseNet pursues this goal by connecting each layer to every other layer in a feed-forward fashion, achieving maximum information reuse and strengthening feature propagation. This connectivity pattern results in significantly fewer parameters compared to traditional convolutional networks of similar depth, addressing issues like vanishing gradients and promoting feature reuse. DenseNet's innovative design has demonstrated superior performance on various benchmarks, making it a popular choice for tasks in computer vision, including image classification and object detection.

Vision Transformer (ViT): The Vision Transformer (ViT [12]) is an innovative deep learning model introduced by Alexey Dosovitskiy and colleagues

in 2020. It applies the transformer architecture, originally designed for natural language processing, to image recognition tasks. ViT divides an image into a sequence of fixed-size patches, linearly embeds each patch, and then processes these embeddings through a standard transformer encoder. This approach allows ViT to capture long-range dependencies and complex patterns in visual data more effectively than traditional convolutional neural networks (CNNs). Despite requiring a large amount of data for training, ViT has achieved state-of-the-art performance on several image classification benchmarks, demonstrating its potential as a powerful alternative to CNNs in the field of computer vision.

Fine-Tuning VGG16, DenseNet, ViT: In recent years, the machine learning community has focused on the ability to reuse existing knowledge from a source learner in a target task, an approach known as transfer learning [14]. Developing neural network models in the field of deep learning typically requires large datasets, extensive computational power, and significant time resources. Transfer learning offers an efficient solution by reusing a pre-trained model on a problem similar to the target problem as a starting point for learning a new model on the target problem [28, 32]. We propose to leverage popular pre-trained models such as VGG16, DenseNet and ViT for classifying ICH images. The learning process then updates the weights in these network layers to effectively deal with the classification of ICH images.

Training Fusing Models: We propose training ensemble classification models to improve the classification results of ICH images. Through empirical test results of transfer learning, we found that no single network architecture had a clear advantage for the classification correctness because any network has advantages and disadvantages. Therefore, we suggest fusing models to combine the strength of deep network models trained on different visual feature types.

First, we use feature extractors from deep learning networks such as VGG16, DenseNet, and ViT to extract features from ICH images. Then, we train logistic regression model [15] on these deep features for classification.

The second strategy involves training logistic regression model on top of the classifiers of the deep learning networks.

The logistic regression training algorithm involves initializing the model parameters, computing the linear combination of inputs, applying the logistic function, calculating the loss, computing gradients, updating parameters via gradient descent, and iterating until convergence. The trained model can then be used to classify new datapoints.

3 Experimental Results

To evaluate our proposed fusing models for classifying ICH images in the Mekong delta, Vietnam. We implement them in Python using library Keras [7] with backend Tensorflow [1], library Scikit-learn [26] and library OpenCV [17]. All

experiments are conducted on a machine Linux Fedora Core 34, Intel(R) Core i7-4790 CPU, 3.6 GHz, 4 cores and 16 GB main memory and the GeForce RTX 2080 Ti GPU (4352 NVIDIA CUDA Cores and 11GB GDDR6).

Our image dataset, consisting of 17 ICH categories from the Mekong Delta, Vietnam, is randomly divided into a training set (6,001 images), a validation set (667 images), and a test set (749 images). We use the training set to build visual classification models. The results are then reported on the test set using the trained visual classification models.

3.1 Tuning Parameters

We use the validation set to tune parameters for building visual classification models on the trainset.

For fine-tuning deep networks, we find out the number of last layers in networks does the learning process need to update as follows:

- keeping unchanged 13 first layers and fine-tuning the rest in VGG16,
- keeping unchanged 100 first layers and fine-tuning the rest in DenseNet,
- replacing the Multi-layer Perceptron (MLP) head with 4 subsequent layers [Dense(128, activation='relu');Dense(128, activation='relu');Dropout(0.15);Dense(num_classes, activation='softmax')], to learn the weights of these new layers in the ViT architecture.

During the training of the VGG16, DenseNet, and ViT networks, the parameters are set as follows: the optimization algorithm is Adam with a learning rate of 0.0001 and the number of epochs is 50.

3.2 Classification Results for 17 ICH Categories

Fine-tuning VGG16, DenseNet, ViT are denoted by FT-VGG16, FT-DenseNet and FT-ViT, respectively. The first fusing model LR-VGG16-DenseNet-ViT is training logistic regression model on deep features. The second fusing model LR-on-Top-VGG16-DenseNet-ViT is training logistic regression model on top of the classifiers of the deep learning networks.

Table 1. Overall classification accuracy for 17 ICH categories

No	Visual approach	Accuracy (%)
1	FT-VGG16	51.34
2	FT-DenseNet	56.17
3	FT-ViT	65.95
4	LR-VGG16-DenseNet-ViT	66.76
5	LR-on-Top-VGG16-DenseNet-ViT	66.49

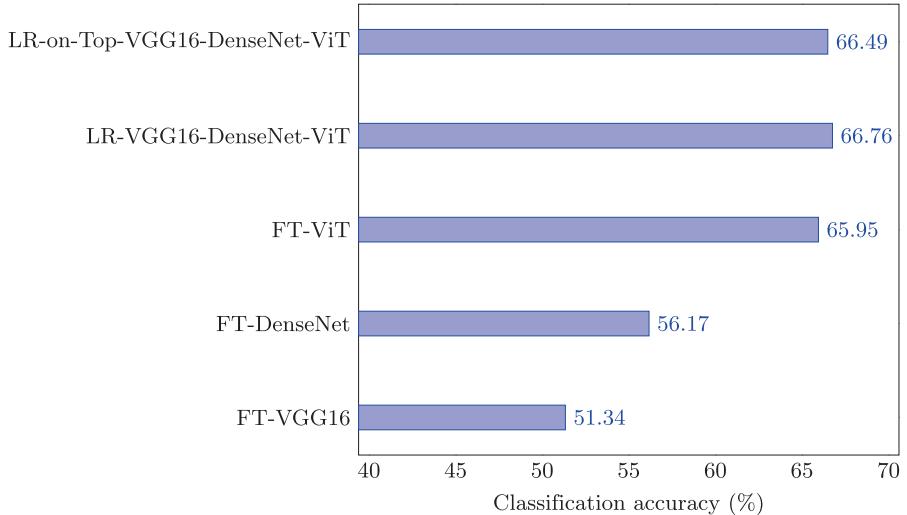


Fig. 4. Overall classification accuracy for 17 ICH categories

We obtain the overall classification accuracy of visual approaches in Table 1 and Fig. 4. The highest accuracy is bold-faced and the second one is in italic.

In the comparison among visual classification approaches, we can see that FT-VGG16 achieved a modest accuracy of 51.34%. Despite its strong performance in many image classification tasks, it appears to struggle somewhat with the diverse and intricate nature of the ICH images. FT-DenseNet outperforms FT-VGG16 with an accuracy of 56.17%. The densely connected architecture of DenseNet likely contributed to its better performance, facilitating efficient feature reuse and gradient flow. FT-ViT significantly outperforms both VGG16 and DenseNet, achieving an accuracy of 65.95%. It seems be that ViT's attention mechanism enables it to capture long-range dependencies in images, which is beneficial for the complex patterns present in ICH categories.

LR-VGG16-DenseNet-ViT fuses features from VGG16, DenseNet, and ViT and training a logistic regression model on these features resulted in an accuracy of 66.76%. This approach leveraged the strengths of multiple architectures, leading to a notable improvement in classification accuracy. This method achieved the highest accuracy among all the tested approaches.

LR-on-Top-VGG16-DenseNet-ViT learns a logistic regression model on top of the outputs of VGG16, DenseNet, and ViT classifiers to expose an accuracy of 66.49%. Although slightly lower than LR-VGG16-DenseNet-ViT, this method still performs better than any single network model. The close accuracy between the two logistic regression approaches indicates that fusing outputs from multiple models is a robust strategy.

4 Conclusions and Future Work

We have presented visual approaches for classifying images of Intangible Cultural Heritage (ICH) in the Mekong Delta, Vietnam, to support cultural preservation efforts, inform policy decisions, and promote cultural pride and awareness. To achieve this, we constructed a dataset of images from 17 different ICH categories, which were manually annotated. We fine-tuned modern pre-trained network models, including VGG16, DenseNet, and Vision Transformer (ViT), for classifying our dataset. Following this, we trained Logistic Regression (LR) models, referred to as fusing models, which combined various visual features extracted from these deep networks as well as their outputs to enhance classification accuracy. Our comparative study of the classification performance on the 17-category ICH image dataset demonstrated that the fusing models significantly improve classification correctness compared to any single fine-tuned model. Specifically, the first fusing model, which integrates visual features from VGG16, DenseNet, and ViT, achieved an accuracy of 66.76%. The second fusing model, which applies Logistic Regression on top of VGG16, DenseNet, and ViT outputs, achieved an accuracy of 66.49%. These results show the effectiveness of our fusing models in improving classification accuracy, typically the potential of combining features from multiple deep learning models to better handle the complexity and diversity of ICH images.

In the future, we plan to research the development of network architectures that can effectively handle the complex classification of ICH images. We are also interested in interpreting the results of deep learning models and creating chatbots to explain different types of ICH.

Acknowledgments. This research has received support from the European Union’s Horizon research and innovation programme under the MSCA-SE (Marie Skłodowska-Curie Actions Staff Exchange) grant agreement 101086252; Call: HORIZON-MSCA-2021-SE-01; Project title: STARWARS (STormwAteR and WastewAteR networkS heterogeneous data AI-driven management). This work has received support from the College of Information Technology, Can Tho University. We would like to thank very much the Big Data and Mobile Computing Laboratory.

References

1. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). <https://www.tensorflow.org/>
2. Belhi, A., Bouras, A., Alfaqheri, T., Aondoakaa, A., Sadka, A.: Investigating 3d holoscopic visual content upsampling using super-resolution for cultural heritage digitization. *Signal Process. Image Commun.* **75**, 188–198 (2019)
3. Belhi, A., Bouras, A., Foufou, S.: Digitization and preservation of cultural heritage: the CEPROQHA approach. In: 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pp. 1–7 (2017)
4. Belhi, A., et al.: Deep learning and cultural heritage: the CEPROQHA project case study. In: 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pp. 1–5 (2019)

5. Belhi, A., et al.: Machine learning and digital heritage: the CEPROQHA project perspective. In: Yang, X.S., Sherratt, S., Dey, N., Joshi, A. (eds.) Fourth International Congress on Information and Communication Technology, pp. 363–374. Springer, Singapore (2020). https://doi.org/10.1007/978-981-32-9343-4_29
6. Chen, K.: icrawler (0.6.2) (2018). <https://pypi.org/project/icrawler/>
7. Chollet, F., et al.: Keras (2018). <https://keras.io>
8. Cosovic, M., Jankovic, R.: CNN classification of the cultural heritage images. In: Proceedings of The 19th International Symposium INFOTEH-JAHORINA, Bosnia and Herzegovina). IEEE (2020)
9. Deng, J., Berg, A.C., Li, K., Fei-Fei, L.: What does classifying more than 10,000 image categories tell us? In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6315, pp. 71–84. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15555-0_6
10. Do, T.-N., Pham, T.-P., Nguyen, H.-H., Pham, N.-K.: Visual classification of intangible cultural heritage images in the mekong delta. In: Belhi, A., Bouras, A., Al-Ali, A.K., Sadka, A.H. (eds.) Data Analytics for Cultural Heritage, pp. 71–89. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-66777-1_4
11. Do, T.-N., Pham, T.-P., Pham, N.-K., Nguyen, H.-H., Tabia, K., Benferhat, S.: Stacking of SVMs for classifying intangible cultural heritage images. In: Le Thi, H.A., Le, H.M., Pham Dinh, T., Nguyen, N.T. (eds.) ICCSAMA 2019. AISC, vol. 1121, pp. 186–196. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-38364-0_17
12. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv e-prints p. [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020). <https://doi.org/10.48550/arXiv.2010.11929>
13. Fiorucci, M., Khoroshiltseva, M., Pontil, M., Travaglia, A., Bue, A.D., James, S.: Machine learning for cultural heritage: a survey. Pattern Recogn. Lett. **133**, 102–108 (2020)
14. Goodfellow, I.J., Bengio, Y., Courville, A.C.: Deep Learning. Adaptive computation and machine learning, MIT Press, Cambridge (2016)
15. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. SSS, Springer, New York (2009). <https://doi.org/10.1007/978-0-387-84858-7>
16. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks (2018). <https://arxiv.org/abs/1608.06993>
17. Itseez: Open source computer vision library (2015). <https://github.com/itseez/opencv>
18. Jankovic, R.: Classifying cultural heritage images by using decision tree classifiers in WEKA. In: Proceedings of 1st International Workshop on Visual Pattern Extraction and Recognition for Cultural Heritage Understanding co-located with 15th Italian Research Conference on Digital Libraries, CNR Area in Pisa, Italy, 30 January 2019. CEUR Workshop Proceedings, vol. 2320, pp. 119–127. CEUR-WS.org (2019)
19. Jankovic, R.: Machine learning models for cultural heritage image classification: comparison based on attribute selection. Information **11**(1), 12 (2020)
20. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, vol. 86, pp. 2278–2324 (1998)
21. Llamas, J., Lerones, P., Medina, R., Zalama, E., Gomez-Garcia-Bermejo, J.: Classification of architectural heritage images using deep learning techniques. Appl. Sci. **7**, 992 (2017)

22. Ma, T., Benferhat, S., Bouraoui, Z., Do, T., Nguyen, H.: Developing application based upon an ontology-based modelling of Vietnamese traditional dances. In: 3rd Digital Heritage International Congress, DigitalHERITAGE 2018, held jointly with 2018 24th International Conference on Virtual Systems & Multimedia, VSMM 2018, San Francisco, CA, USA, 26–30 October 2018, pp. 1–7 (2018)
23. Ma, T., Benferhat, S., Bouraoui, Z., Tabia, K., Do, T., Nguyen, H.: An ontology-based modelling of Vietnamese traditional dances (S). In: The 30th International Conference on Software Engineering and Knowledge Engineering, Hotel Pullman, Redwood City, California, USA, 1–3 July 2018, pp. 64–67 (2018)
24. Ma, T., Benferhat, S., Bouraoui, Z., Tabia, K., Do, T., Pham, N.: An automatic extraction tool for ethnic Vietnamese Thai dances concepts. In: 18th IEEE International Conference On Machine Learning And Applications, ICMLA 2019, Boca Raton, FL, USA, 16–19 December 2019, pp. 1527–1530 (2019)
25. Mustafa, Y., Clawson, K., Bowerman, C.: Saving cultural heritage with digital make-believe: machine learning and digital techniques to the rescue. In: Proceedings of the Electronic Visualisation and the Arts (EVA 2017) (2017)
26. Pedregosa, F., et al.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
27. Rapti, A., Tsolis, D., Sioutas, S., Tsakalidis, A.: A survey: mining linked cultural heritage data. In: Proceedings of the 16th International Conference on Engineering Applications of Neural Networks (INNS), EANN 2015. Association for Computing Machinery, New York (2015)
28. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2014, Columbus, OH, USA, 23–28 June 2014, pp. 512–519. IEEE Computer Society (2014)
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *CoRR arxiv:1409.1556* (2014)
30. Vapnik, V.: The Nature of Statistical Learning Theory, 2nd edn. Springer, Heidelberg (2000). <https://doi.org/10.1007/978-1-4757-3264-1>
31. Vu, M.T., Beurton-Aimar, M., Le, V.L.: Heritage image classification by convolution neural networks. In: Proceedings of 1st International Conference on Multimedia Analysis and Pattern Recognition (MAPR), pp. 1–6 (2018)
32. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Montreal, Quebec, Canada, 8–13 December 2014, pp. 3320–3328 (2014)



Image Colorization with Dif-EDUNet: A Diffusion-Based Approach

Ngoc-Giau Pham^{1,2}(✉) , Van-Hieu Duong²(✉) , Thanh-Hai Le Tong² , Hong-Ngoc Tran³ , and Phuoc-Hung Vo¹

¹ Tra Vinh University, Tra Vinh, Vietnam

² Tien Giang University, My Tho, Tien Giang, Vietnam

{phamngocgiau, duongvanhieu}@tgu.edu.vn

³ Vietnamese German University, Ben Cat, Binh Duong, Vietnam

Abstract. Image colorization provides significant data handling advantages by completely eliminating the need for labeling. This research introduces advancements in the process of converting grayscale images into color using modern machine learning techniques. By utilizing the Lab color space, where luminance (L) is processed separately from the color channels (ab), this research focuses on refining and developing the Dif-EDUNet model (a Diffusion model using ED-UNet) to address the challenges in image colorization. Our experiments, including training with datasets: Coco-Stuff, DIV2K, Places365, ImageNet and CelebA show that our results are very encouraging compared to the benchmark of the dataset in this issue. Additionally, the Weights & Biases (wandb) tool is employed to support the monitoring of the training process.

Keywords: Diffusion model · ED-Unet · Image colorization · Dif-EDUNet

1 Introduction

Image colorization is a sophisticated technique aimed at converting grayscale images into full-color outputs, enhancing the visual richness and detail of the imagery. Technically, colorization techniques can be categorized into three main approaches: Scribble-based, Example-based, and Fully Automatic (Fig. 1). Scribble-based colorization requires user interaction, where users provide initial color hints on the grayscale image, and the system extrapolates these hints across similar textures. Example-based colorization utilizes a reference color image to inform and guide the colorization of the target grayscale image, aligning features between the reference and target to apply colors accurately. The most sophisticated of these, Fully Automatic colorization, employs deep learning algorithms that autonomously predict and apply color without any human input, utilizing extensive datasets to train models that can replicate the dynamic range and subtleties of natural colors. We chose to use the L^*a^*b color space in this research (see Fig. 2).

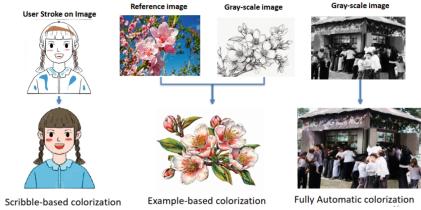


Fig. 1. Three main approaches for the colorization techniques.

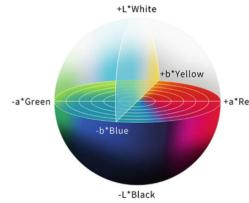


Fig. 2. The L*a*b color space [1].

Furthermore, the methods can be divided into non-parametric and parametric techniques. Non-parametric methods transfer color by matching regions between the reference and target images, often guided by user input to enhance the suitability of selected references. On the other hand, parametric methods involve sophisticated predictive functions derived from vast datasets, with strategies ranging from regression onto continuous color spaces to the classification of quantized color values, often enabled by deep neural networks such as CNNs, GANs, VAEs (Variational Autoencoders), and Diffusion Models. This research delves into the intricacies of the Fully Automatic approach with a focus on the advanced Diffusion Model. This model are meticulously developed to address the nuanced challenges of image colorization. Research contributions:

- Propose a model based on Diffusion model to address the issue of Image Colorization. This research proposes replacing Unet with ED-UNet.
- We trained the Dif-EDUNet model architecture on the Coco-Stuff dataset [2], DIV2K [3], Places365 [4], ImageNet [5], and the CelebA dataset [6], along with the application of the wandb tool [7] to support the training process.
- Conduct measurements to evaluate the effectiveness of both models using both qualitative assessments and quantitative metrics, such as FID, PSNR and SSIM. We then compare the results to existing benchmarks to evaluate the model.

Section 2 introduces image colorization methods. Section 3 details the architecture of proposed model: Dif-EDUNet models. Section 4 presents the training process, hyperparameters and evaluates the results using FID, PSNR and SSIM metrics on five datasets, discusses model performance. Section 5 presents conclusion and suggests future research directions.

2 Related Works

The field of image colorization has seen remarkable evolution from initial manual methods to the application of advanced computational techniques and deep learning architectures. The transformation began with non-parametric methods, utilizing color reference images and user inputs to dynamically select candidate images from the Internet, producing diverse and vibrant results [8]. Parametric

methods emerged with the ability to apply adaptive clustering and neural networks, significantly enhancing accuracy by optimizing probability distributions across discrete color spaces [9]. The integration of deep learning technology around 2016 marked a significant advancement, introducing models that combined global and local image features to improve both contextual accuracy and the naturalness of colorizations [10, 11]. Innovations continued to emerge, such as the real-time video colorization model and the enhancement of contextual cues in images [12, 13]. Guadarrama et al. utilized reinforcement learning for pixel-recursive colorization, showing significant improvements in precision and detail [14]. Semantic segmentation techniques introduced further refined the accuracy of color applications, enhancing the field's ability to process images with high fidelity [15]. In 2021, GAN-based techniques for stylized colorization introduced versatility in handling various artistic styles [16]. The focus on enhancing computational efficiency allowed these advanced colorization processes to be implemented on lower-powered devices, widening accessibility [17]. Most recently, the integration of augmented reality (AR) technologies facilitated interactive, real-time adjustments in colorization processes, significantly enhancing user interactivity and customization [18]. These continuous developments illustrate a robust evolution towards more automated, efficient, and user-interactive image colorization technologies. In this research, we extend these foundational works by developing and applying Dif-EDUNet (Diffusion models) to the task of image colorization and comparing the experimental results on five datasets with the current state-of-the-art benchmarks. Additionally, the research is combined with the Weights & Biases tool for effectively monitoring and showcasing the training process.

3 Methodology

In this section, the unsupervised learning is leveraged in implementation as it allows for the exploration and modeling of data without labeled outcomes, enhancing the understanding of inherent structures within the dataset [19]. The unsupervised learning method is classified into non-probabilistic and probabilistic models. Non-probabilistic models, like sparse coding and autoencoders [24], are used for tasks such as feature extraction without relying on probability distributions [25]. Probabilistic models are further divided into those that model explicit density (like Boltzmann machines) and those that approximate implicit density, such as Generative Adversarial Networks (GANs) [20] and Variational Autoencoders (VAEs) [21], which are essential for tasks like image generation and data simulation [19, 22].

3.1 Diffusion Model

Encoder (forward Process)

In the diffusion or forward process (Fig. 3), a data sample x is progressively transformed through a sequence of latent variables z_1, z_2, \dots, z_T , each maintaining the dimensions of x . The transformation begins with $z_1 = f_1(x)$, where f_1 represents the transformation function at the first step.

$$z_1 = (1 - \beta_1) \cdot x + \sqrt{\beta_1} \cdot \epsilon_1 \quad (1)$$

$$z_t = (1 - \beta_t) \cdot z_{t-1} + \sqrt{\beta_t} \cdot \epsilon_t \quad (2)$$

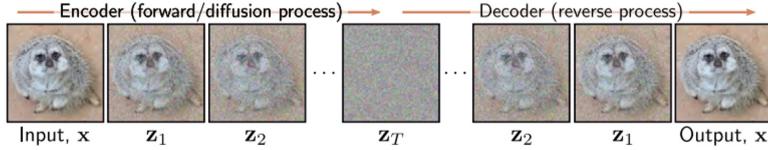


Fig. 3. Diffusion models [23].

In these expressions, ϵ_t represents noise sampled from a standard normal distribution. The coefficients β_t , which range between 0 and 1, control the pace at which noise is incorporated into the data through a predefined noise schedule.

$$\text{Posterior distribution of } z_t \text{ given } z_{t-1} \text{ Variance of the noise at each step } t$$

$$q(z_t|z_{t-1}) = \text{Norm}_{z_t} \left[\begin{matrix} \sqrt{1 - \beta_t} z_{t-1} & \beta_t \end{matrix} \right] \quad \forall t \in \{2, \dots, T\} \quad (3)$$

Gaussian distribution Mean: z_{t-1} scaled by $\sqrt{1 - \beta_t}$

Decoder (reverse Process)

In researching diffusion models, we focus on mastering the reverse process, which involves reconstructing the original data, x , from a series of latent variables, starting from z_T down to z_1 . Each step in this process involves a reverse mapping from one latent variable to its predecessor, progressively moving closer to the original data point. The true reverse distributions, $q(z_{t-1}|z_t)$, encountered during this reverse diffusion are inherently complex and multimodal (see Fig. 3). These distributions are influenced by the underlying data distribution, $P_r(x)$. We simplify these complex reverse distributions using normal distributions:

$$\text{Prior distribution of } z_T$$

$$P_r(z_T) = \mathcal{N}_{z_T} \left[\begin{matrix} 0 \\ I \end{matrix} \right], \quad (4)$$

Gaussian distribution Mean vector of zero
Covariance matrix identity

$$\text{Reverse distribution function}$$

$$P_r(z_{t-1}|z_t, \theta_t) = \mathcal{N}_{z_{t-1}} \left[\begin{matrix} f_t[z_t, \theta_t] \\ \sigma_t^2 I \end{matrix} \right] \quad (5)$$

Gaussian distribution Covariance scaled by σ_t^2

Where, the function $f_t[z_t, \theta_t]$ is a neural network that predicts the mean of the normal distribution for mapping from latent variable z_t to its predecessor z_{t-1} . The variance parameters σ_t^2 are fixed in advance.

Diffusion Loss Function

To optimize the model, we aim to maximize the Evidence Lower Bound (ELBO) relative to the parameters ϕ_1 to ϕ_T . In practical applications, the scaling factors, which may vary at each step, are typically disregarded, resulting in a simplified formulation:

$$\begin{aligned} \text{Latent variable} \\ \text{at step } t \\ L_{\phi_1 \dots T} = \sum_{i=1}^I \sum_{t=1}^T \|g_t(z_{it}, \phi_t) - \epsilon_{it}\| &= \sum_{i=1}^I \sum_{t=1}^T \|g_t(\sqrt{\alpha_t} \cdot x_i + \sqrt{1-\alpha_t} \cdot \epsilon_{it}, \phi_t) - \epsilon_{it}\|^2 \\ \text{Noise component} \\ \text{at step } t \end{aligned} \quad (6)$$

where we have rewritten z_t using the diffusion kernel:

$$z_t = \sqrt{\alpha_t} \cdot X + \sqrt{1-\alpha_t} \cdot \epsilon \quad (7)$$

Application to Image

Diffusion models have proven to be highly effective for processing image data. While numerous diffusion steps are typically involved, maintaining several U-Nets for this purpose is inefficient. A practical approach involves using a single U-Net that incorporates a time-coded vector as an input, which is adjusted to conform to the U-Net's channel dimensions at different stages, aiding in modulation of the features spatially (see Fig. 4). The need for extensive time steps arises because the conditional probabilities $q(z_{t-1} | z_t)$ converge towards a normal distribution as the hyperparameters β_t approach zero, aligning with the decoder's distribution patterns $\Pr(z_{t-1} | z_t, \phi_t)$. This requirement significantly slows down the sampling process. It may necessitate running the U-Net model for as many as 1,000 steps to produce a high-quality image.

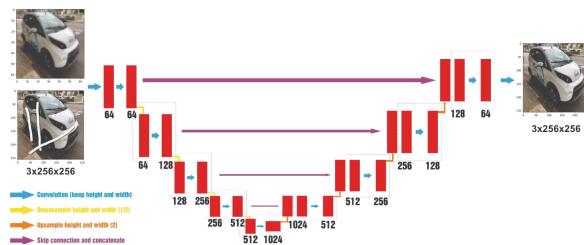


Fig. 4. ED-UNet architecture [26].

3.2 Proposed Model

We have chosen the ED-UNet structure [26] which proposed by the authors in the paper “The Problem Of Image Super-Resolution, Denoising And Some Image

Restoration Methods In Deep Learning Models". This architecture includes techniques such as Skip connections, pretrained weights, Attention scheme and Batchnorm. Firstly, this architecture is particularly suitable for fully automated colorization tasks because it can be trained end-to-end. ED-UNet comprises contracting and expanding paths that encode contextual features and decode them into 2D feature maps, thereby delivering accurate and efficient colorization results (see Fig. 4) (Fig. 5).

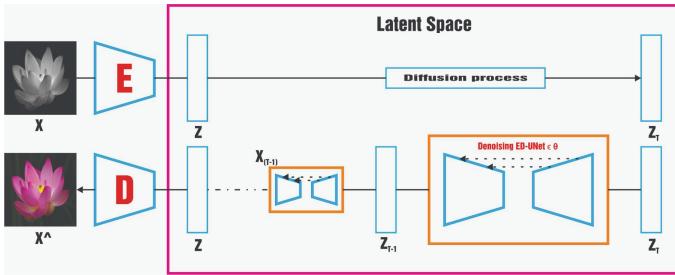


Fig. 5. The proposed model: Dif-EDUNet.

During the forward process, the image x is encoded by E into a latent space representation z . This z is then subjected to a diffusion process, where noise is gradually added over multiple timesteps, transforming z into z_T . This step involves the controlled addition of noise to the latent representation across predefined timesteps to achieve a fully noisy state z_T , ensuring the process is reversible. During the reverse process, the ED-UNet model takes inputs including a timestep embedding t and a 3D image—composed of the output from the UNet model at timestep $t + 1$ and the grayscale channel as a conditioning component.

3.3 Evaluation Metrics

Frechet Inception Distance (FID). The Frechet Inception Distance (FID) measures the distance between two Gaussian distributions of Inception v3 features from two datasets.

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$$

↑
Squared Euclidean distance between mean vectors
↑
Trace of covariance matrices difference

(8)

where: μ_r, Σ_r : Mean vector and covariance matrix of the features from the real images; and μ_g, Σ_g : of the features from the generated images.

Peak Signal-to-Noise Ratio (PSNR)

PSNR is frequently employed as a critical metric in the field of image colorization, along with its applications in other areas like image super-resolution and denoising. This metric provides a quantitative measure to assess the accuracy of color restoration at the pixel level.

$$\text{MSE} = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (I_y(i, j) - I_C(i, j))^2 \quad (9)$$

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) \quad (10)$$

The Mean Squared Error (MSE), which underpins the calculation of PSNR, is defined by the formula 9, 10. Here, MAX is the maximum possible pixel value of the image, and MSE is the Mean Squared Error between the original and the compressed image.

Structural Similarity Index (SSIM)

SSIM is calculated by analyzing three distinct components of the images: the mean luminance of the pixels, their contrast (measured by the standard deviation), and the correlation of pixel values, reflecting the structural information. The formula for SSIM incorporates these aspects:

$$\text{SSIM}(x, y) = \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \right) \cdot \left(\frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \right) \quad (11)$$

Here, x and y denote the images under comparison, μ_x and μ_y are their mean intensities, σ_x^2 and σ_y^2 are their variances, and σ_{xy} represents the covariance. Constants C_1 and C_2 help maintain stability in the division process when the denominator is small, with values typically dependent on the dynamic range of the pixel values.

4 Experiments and Discussion

In this section, the research will implement the proposed model: Dif-EDUNet model with Wandb integration for image colorization tasks. Our experiments were conducted on the following datasets: COCO-Stuff [2], ImageNet [5], Div2k [3], Places365 [4], and CelebA [6].

4.1 Installation Environment and Dataset

Our development environment was the Linux-6.2.0-39-generic-x86_64-with-glibc2.37 operating system, utilizes Python 3.11.8, and executes Python via a specific Anaconda environment. The hardware configuration includes an 24-core CPU and a high-end NVIDIA GeForce RTX 4090 GPU, with the W&B CLI version 0.17.4 supporting integration and monitoring (Table 1).

Table 1. Number of images in training, validation, and testing datasets.

Dataset	Training	Validation	Testing
COCO-Stuff [2]	10000	2000	1000
Place365 [4]	10000	2000	1000
DIV2K [3]	8600	1720	860
ImageNet [5]	10000	2000	1000
CelebA [6]	50000	10000	5000

4.2 Training

The Table 2 illustrates the selection of datasets for training and highlights the use of our model to evaluate the remaining datasets. For our specific model, training and evaluation were conducted directly on the same dataset. The Dif-EDUNet model was trained for 350 epochs (ensure comprehensive learning and optimal model performance) on five datasets. Its experimental outcomes being recorded and monitored through wandb. Each input was resized to at most 128 pixels. To overcome overfitting in multi-task learning, the training process starts with an initial learning rate of 0.0005, using a reduced learning rate when the model does not improve after every five epochs with a decay factor of 0. Dropout is set at 0.2. During training, the model uses the Adam optimizer. Each configuration requires approximately 15–24 hours of training time (The ImageNet dataset alone took the longest time to train, up to 2 days).

Table 2. Comparison of Methods and Training Data

Method	Name	Training Data
1	Iizuka et al. [13]	Place365
2	Larsson et al. [14]	ImageNet
3	Zhang et al. [17]	ImageNet
4	Ours with Dif-EDUNet	Places365
5	Ours with Dif-EDUNet	ImageNet
6	Ours with Dif-EDUNet	Coco-stuff
7	Ours with Dif-EDUNet	Div2k
8	Ours with Dif-EDUNet	CelebA

4.3 Evaluation

Table 3. Comparison of Methods on Different Datasets

Method	ImageNet ctest1k			DIV2K		
	PSNR ↑	SSIM ↑	$L2_{ab} \downarrow$	PSNR ↑	SSIM ↑	$L2_{ab} \downarrow$
Iizuka et al. [11]	22.841	0.865	0.277	22.981	0.919	0.079
Larsson et al. [10]	23.335	0.869	0.26	23.490	0.929	0.072
Zhang et al. [8]	21.297	0.848	0.286	20.926	0.896	0.079
Dif-EDUNet (ours)	33.27	0.986	0.084	35.854	0.965	0.061
Method	Place365 ctest1k			COCO-Stuff ctest1k		
	PSNR ↑	SSIM ↑	$L2_{ab} \downarrow$	PSNR ↑	SSIM ↑	$L2_{ab} \downarrow$
Iizuka et al. [11]	25.572	0.948	0.481	23.541	0.871	0.242
Larsson et al. [10]	25.096	0.945	0.452	23.773	0.873	0.223
Zhang et al. [8]	23.076	0.928	0.484	21.502	0.851	0.245
Dif-EDUNet (ours)	29.814	0.937	0.098	32.90	0.9346	0.084

Table 4. Dif-EDUNet model on CelebA Dataset

Method	CelebA ctest1k			
	FID ↓	PSNR ↑	SSIM ↑	$L2_{ab} \downarrow$
Wang et al. [28]	22.79	–	–	–
Dif-EDUNet (ours)	13.073	44.59	0.998	0.0801

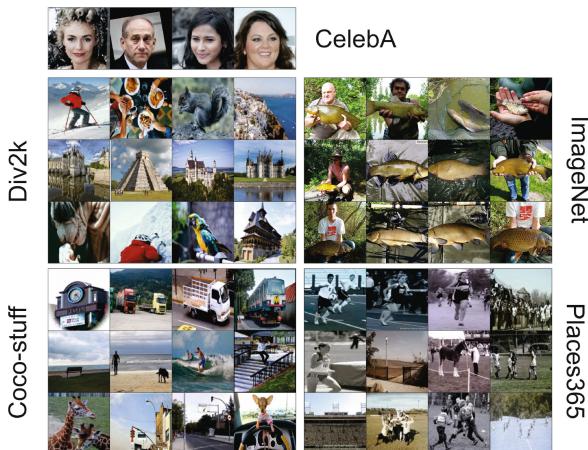


Fig. 6. Inferencing on five datasets.

Quantitative comparisons on similarity metrics: three similarity metrics used in this research: FID, PSNR, SSIM and the MSE of ab channel to compare with benchmark of datasets in image colorization task (see Table 3). For PSNR and SSIM, higher are the better. For MAE, FID, lower values are better. For the quality metrics, we evaluated the visual performance to demonstrate both success and failure cases. The evaluation was conducted on five public datasets: DIV2K, ImageNet, COCO-Stuff, Places365, and CelebA. Our method was compared against four robust colorization techniques developed by Iizuka et al. [11], Larsson et al. [10], Zhang et al. [8] as shown in Table 3 and Wang et al. [28] as shown in Table 4. Unlike the pre-trained weights used by these authors to predict image colors, we performed training on each dataset and then conducted predictions. The Dif-EDUNet model, as proposed in this research, consistently outperforms the other methods across all datasets, especially in terms of PSNR and SSIM, which are critical indicators of image quality and similarity. The significant improvement in $L2_{ab}$ values further underscores the effectiveness of Dif-EDUNet in reducing color errors. These results highlight the potential of Dif-EDUNet in delivering superior image reconstruction and enhancement capabilities compared to traditional methods (Fig. 6).

4.4 Discussion

Advantages of Dif-EDUNet: Dif-EDUNet excels in generating high-quality and sharp images. Its multi-step process gradually refines the details in the images, effectively removing noise and enhancing resolution, which is critical for tasks involving image reconstruction and enhancement. Another significant advantage of Dif-EDUNet is its ability to mitigate overfitting by generating multiple variations of the training data, thereby improving the model's generalization capabilities. Additionally, Dif-EDUNet is highly effective in handling and eliminating noise from data due to its sophisticated multi-step process.

Disadvantages of Dif-EDUNet: Despite its many strengths, Dif-EDUNet has some limitations. The training process for this model is often very time-consuming and demands substantial computational resources due to the extensive multi-step process, leading to increased costs and development time. Moreover, optimizing Dif-EDUNet is complex and requires advanced optimization techniques to ensure proper convergence, necessitating deep knowledge of algorithms and programming skills. Furthermore, Dif-EDUNet typically requires a large amount of training data to achieve optimal performance, which can be challenging for tasks with limited data availability. Finally, deploying this model in practical applications can be complex due to its high computational resource requirements and intricate data processing workflows.

5 Conclusion

In this paper, we introduce the Dif-EDUNet architecture, specifically designed to address the challenges of image colorization. Our comprehensive validation experiments on various datasets such as Coco-Stuff, ImageNet, DIV2K,

Places365, and CelebA have shown promising results. This architecture has proven particularly effective in improving the colorization process, even with images that feature complex patterns, rare colors, and multiple objects. However, these scenarios often lead to issues such as noise and inaccurate colors, with some areas remaining uncolored. The Dif-EDUNet model has demonstrated high efficiency in the task of image colorization. Future research could explore integrating various generative model techniques to refine the feature learning process. This approach not only promises to enhance the quality of image colorization but also expands the applicability of the Dif-EDUNet model in practice and research.

References

1. Fairchild, M.D.: Color Appearance Models. Wiley-IS&T Series in Imaging Science and Technology, Wiley, Hoboken (2013)
2. Caesar, H., Uijlings, J., Ferrari, V.: COCO-stuff: thing and stuff classes in context. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1209–1218 (2018)
3. Agustsson, E., Timofte, R.: NTIRE 2017 challenge on single image super-resolution: dataset and study. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 126–135 (2017)
4. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: a 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(6), 1452–1464 (2018)
5. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
6. CelebA dataset. <https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>. Accessed 10 Apr 2024
7. Biewald, L.: Weights & Biases: The AI Developer Platform (2011). <https://wandb.ai>
8. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 649–666. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_40
9. Brown, M., Hua, G., Winder, S.: Discriminative learning of local image descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(1), 43–57 (2011). <https://doi.org/10.1109/TPAMI.2010.54>
10. Larsson, G., Maire, M., Shakhnarovich, G.: Deep learning architectures for image colorization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 2623–2631 (2016)
11. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Combining local and global image features for realistic colorization. *ACM Trans. Graph.* **35**(4), Article 110 (2016)
12. Huang, J., Ma, X., Wang, Y., Li, X.: Real-time video fire detection via convolutional neural networks. In: 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), pp. 475–481 (2023)
13. Atoum, Y., Ye, M., Ren, L., Tai, Y., Liu, X.: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 506–507 (2020)

14. Guadarrama, S., et al.: Reinforcement learning for pixel-recursive colorization. In: Advances in Neural Information Processing Systems (NeurIPS), Vancouver, Canada, pp. 2471–2481 (2019)
15. Gonzz Santiago, J., Schenkel, F., Middelmann, W.: Self-supervised image colorization for semantic segmentation of urban land cover. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, pp. 3468–3471 (2021). <https://doi.org/10.1109/IGARSS47720.2021.9554123>
16. Fang, T.-T., Vo, D.M., Sugimoto, A., Lai, S.-H.: Stylized-colorization for line arts. In: 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, pp. 2033–2040 (2021). <https://doi.org/10.1109/ICPR48806.2021.9412756>.
17. Yu, C.-F., Sharma, G., Aly, H.A.: Computational efficiency improvements for image colorization. Electron. Imaging (2014)
18. Lv, Z., Lloret, J., Song, H.: Real-time image processing for augmented reality on mobile devices. J. Real-Time Image Process. **18**, 245–248 (2021). <https://doi.org/10.1007/s11554-021-01097-9>
19. Salakhutdinov, R.: Deep unsupervised learning. Lecture slides, Carnegie Mellon University (2023)
20. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
21. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proceedings of the 2nd International Conference on Learning Representations (ICLR), Scottsdale, AZ, USA (2013)
22. Salakhutdinov, R.: Deep unsupervised learning [PowerPoint slides]. Carnegie Mellon University (2019)
23. Prince, S.J.D.: Understanding Deep Learning. <http://udlbook.com>
24. Baldi, P.: Autoencoders, unsupervised learning, and deep architectures. In: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, pp. 37–50 (2012)
25. Li, F.-F., Karpathy, A., Johnson, J.: CS231n convolutional neural networks for visual recognition. Stanford University (2016)
26. Pham, N.-G., Tong Le, T.-H., Duong, V.-H., Tran, H.-N., Vo, P.-H.: The problem of image super-resolution, denoising and some image restoration methods in deep learning models. In: The 2024 International Conference on Research in Engineering and Technology (2024)
27. Saharia, C., et al.: Photorealistic text-to-image diffusion models with deep language understanding. ArXiv, abs/2205.11487 (2022)
28. Wang, Y., Yu, J., Zhang, J.: Zero-shot image restoration using denoising diffusion null-space model. arXiv (2022). <http://arxiv.org/2212.00490>



Proposing a Solution to Improve Safety for Fiat-Shamir ZKP Scheme on Elliptic Curve

Hanh Tran Thi, Nghi Nguyen Van^(✉), Minh Nguyen Hieu, Hien Pham Thi, Tu Le Minh, and Thi Tuyet Trinh Nguyen

Academy of Cryptography Techniques, Hanoi 151090, Vietnam
{hanhtt,nghinv,hienpt,trinhntt}@actvn.edu.vn

Abstract. Zero-Knowledge Proof (ZKP) scheme is a type of cryptographic technique that is being commonly applied in practice, such as blockchain technology, authentication systems, and electronic voting systems. The Fiat-Shamir ZKP scheme on elliptic curves has been standardized in RFC 8235. In this paper, we will present some weaknesses that exist in the Fiat-Shamir ZKP scheme on elliptic curves, such as being affected by replay attacks and errors caused in the pseudo-random number generator. The goal of the article is to propose security enhancements for the Fiat-Shamir ZKP scheme to combat these limitations. Our method is analyzing, comparing the computational cost, and experimental results of the proposed solution with the original Fiat-Shamir ZKP scheme, which is mentioned in RFC 8235, and providing a user authentication application in the Client-Server model.

Keywords: Elliptic Curve · Fiat-Shamir ZKP · Replay attack

1 Introduction

Zero-Knowledge Proof (ZKP) is a very abstract and captivating technique in modern cryptography. The term of “Zero – Knowledge” or “undisclosed knowledge” was first introduced by Shafi Goldwasser et al. in 1985. ZKP is a cryptographic technique used by a party P (Prover) to establish, with the other party V (Verifier), that P has knowledge of a value x without disclosing any information to V on this value x [1]. By virtue of its interaction between two parties, this approach may also be referred to as a protocol.

ZKP is classified into two main types: Interactive ZKP and Non-Interactive ZKP. Interactive ZKP: Here, during the implementation of ZKP, there needs to be interaction from both Verifier and Prover sides. Non-Interactive ZKP: during the implementation of ZKP, there is no need for interactive feedback to the Prover from the Verifier side.

A zero-knowledge proof algorithm must contain the following 3 properties [2]: Completeness, Soundness, Zero-Knowledge.

- Completeness: If the clause is true, then an honest Verifier will be convinced by an honest Prover that the clause is true.
- Soundness: If the Prover is dishonest, then Prover cannot convince the Verifier that the clause is true by lying.

- Zero-Knowledge: If the clause is true then the Verifier only knows that the clause is true but the Verifier cannot know exactly what the clause is.

ZKP has been receiving a lot of attention in research and deployment of practical applications very strongly in the past 10 years. Common applications of ZKP are applying in authentication and identification systems [3], in blockchain technology [2], and in other cryptographic algorithms [4, 5].

Practical applications of ZKP schemes may be categorized into three groups: Mathematical and computational methods using finite fields, such as the Schnorr ZKP scheme and the Fiat-Shamir ZKP scheme [6, 7], provide the foundation of Group 1. Hardware circuits such as the ZK-SNARK scheme and the ZK-STARK scheme [2] form the foundation of Group 2. Group 3 is derived from computations performed on elliptic curves [8, 9].

In this article, we will focus on the Fiat-Shamir ZKP scheme on elliptic curves, which is normalized in RFC8235 (this is the non-interactive ZKP scheme when applying the Fiat-Shamir transform into the Schnorr interactive ZKP scheme). We will prove that this scheme is affected by two attacks - a replay attack and a fault attack - in the pseudorandom number generator.

After that, we will propose a solution to improve the security of this scheme by preventing replay attacks and fault attacks on the random number generator based on previous publications [10], and we will compare the improved scheme with the original scheme about computational costs and capabilities.

The article is organized in 4 main sections: after Sect. 1 introduction, Sect. 2 gives the Fiat-Shamir ZKP scheme on an elliptic curve. In the third section, we propose a solution to improve safety for the Fiat-Shamir ZKP scheme on elliptic curves. Finally, there is a conclusion.

2 Fiat-Shamir ZKP Scheme on Elliptic Curve

In this section, we will present Fiat-Shamir ZKP scheme on elliptic curve in RFC 8235 [8] and in [9]. After that, we will show some weakness of the scheme and propose a solution to improve the security of the scheme. Table 1 gives the mathematical symbols used in this paper.

2.1 Fiat-Shamir ZKP Scheme on Elliptic Curve

Protocol goals: User Alice (Prover side) proves that he knows the identifier value x_A for Bob (Verifier side) according to Fiat-Shamir Zero-Knowledge identification protocol with the general parameters being an Elliptic curve of the Weierstrass form which has an equation $y^2 = x^3 + ax + b$ on the field F_p , where p is a prime number ($p > 3$) and two integers a, b satisfy $4a^3 + 27b^2 \neq 0$, denoted as $E_p(a, b)$. G is the generator of the curve $E_p(a, b)$ with order n .

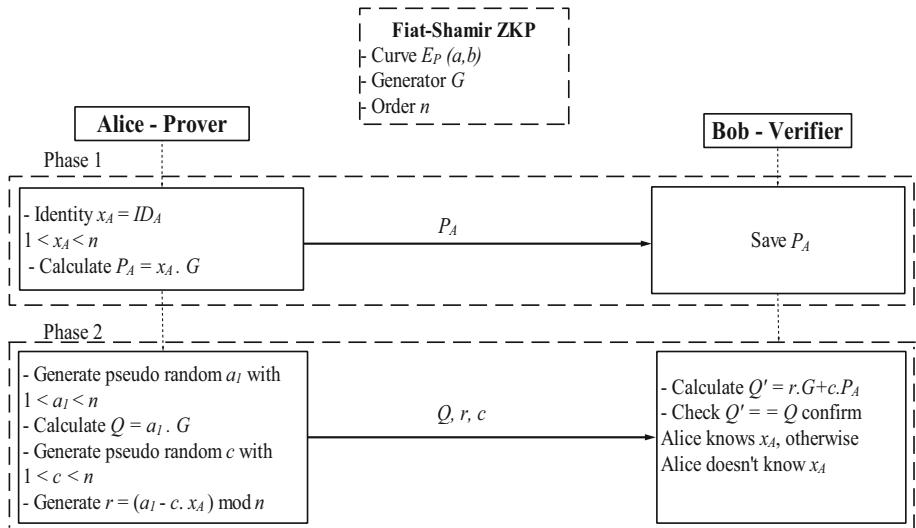
The protocol operates in two phases as depicted in Fig. 1 with the following specific operational steps:

Table 1. Symbol Notations

Notation	Definition
a, b	The integers satisfy condition $4a^3 + 27b^2 \neq 0$
p	Large prime number
$E_p(a, b)$	Weierstrass curve has equation $y^2 = x^3 + ax + b \text{ mod } p$
G	A generator of the curve $E_p(a, b)$
H	Secure cryptographic hash function (SHA-2 or SHA-3)
n	The order of G
ts	timestamp
ID_A	Identifier of the User or device A
MK_A	Password of the User or device A
x_A, a_1, c, r	Positive integer coefficients smaller than n
P_A, Q, Q'	Points on the curve $E_p(a, b)$

Phase 1 – Registration

+ Alice has value $x_A = ID_A$ with $1 < x_A < n$ and calculates $P_A = x_A \cdot G$. Alice sends P_A to Bob. Bob saves P_A in the database.

**Fig. 1.** Operation of the Fiat-Shamir ZKP scheme on ECC

Phase 2 – Verification

+ Alice generates the pseudorandomly value a_1 such that $1 < a_1 < n$ and calculates $Q = a_1 \cdot G$. After that, she generates pseudo random value c with $1 < c < n$. She generates value $r = a_1 - cx_A$ and sends Q, r, c to Bob.

+ Bob calculates $Q' = r \cdot G + c \cdot P_A$ and makes a comparison Q' with Q , if they are the same, then confirm Alice knows x_A , otherwise, Alice doesn't know x_A .

Fiat-Shamir ZKP operates and satisfies three properties as mentioned in RFC 8235 including the following properties: Completeness, Soundness, Zero-Knowledge.

2.2 Some Weaknesses of the Fiat-Shamir ZKP Scheme

Replay Attack. Considering the operation of the Fiat-Shamir ZKP scheme in phase 2, the attacker (Oscar) intercepts the values Q, r, c in time $ts1$.

We see that Bob hasn't mechanism to check whether the packet has been retransmitted or not, this leads to Oscar being able to impersonate Alice to prove that Oscar also knows the value of x_A at time $ts2$. Thus, this attack is possible with the Fiat-Shamir ZKP scheme in RFC 8235.

Attacking Security Vulnerabilities in Pseudo-random Number Generator.

Attacking security vulnerabilities in pseudo-random number generator in Fiat-Shamir ZKP scheme on the elliptic curve (Fig. 1), supposing the scheme is applied to the Client-Server model in which Alice is the Client, Bob is the Server. Bob will have to install software to perform the calculation steps. We see that in stage 2, pseudo-random generation a_1 by software or hardware implementation, similar to c . In [10] and [11], they are confirmed that there are attacks on the above pseudorandom number generation process with the installation of number generator with security vulnerabilities in hardware or software, specifically, there may be at the backdoor in the source code. This causes a break in the operation of the Fiat-Shamir ZKP scheme.

An example in [10] of a pseudo-random number generation function (written in C programming language) that has a backdoor is as follows:

```
int getRandomNumber()
{
    return 4;
}
```

The above pseudo-random number generator function always returns the value 4. An attacker who knows the source code of this generator function will be able to calculate the value x_A and break the scheme.

Second example of pseudo-random number generator source code with small generation period:

```

int getRandomNumber()
{
    srand(time(NULL));
    int res = 100+ rand() % 1000;
    return res;
}

```

In this example, the pseudo-random number generation function generates a number in the range [100, 1000], so an attacker can completely try to exhaust this random value to find and calculate the value x_A .

The `srand()` function in the example has different seeding roles for each generation. In case the source code of the above function does not have this function, then the return result is the same on each call. Therefore, the attacker only needs to perform brute force once to find a randomly generated value for all other user communications.

Suppose $a_1 = 4$; $c = 4$, then Q can always be calculated $Q = 4.G$. An attacker who knows r will find x_A because $r = (a_1 - c.x_A) \bmod n$. Thus, this scheme has security vulnerabilities when using software and hardware.

3 Proposing a Solution to Improve Safety for Fiat - Shamir ZKP Scheme on Elliptic Curve

3.1 Proposed Solution

To overcome the weaknesses of the Fiat-Shamir ZKP scheme in RFC 8235 mentioned above, we will propose a secure solution through the use of timestamp and cryptographic hash functions in the pseudo-random number generator.

Phase 1 – Registration

Alice has value $x_A = ID_A$ with $1 < x_A < n$ and calculate $P_A = x_A.G$. Alice sends P_A to Bob to save in the database for the verification phase in phase 2.

Phase 2 – Verification

- + Alice generates value $a_1 = H(x||ts) \bmod n$ with ts is the timestamp of the moment that needs to be verified and calculates $Q = a_1.G$ and generates value $c = H(Q||P_A||ts) \bmod n$. Alice continues to calculate $r = (a_1 - cx_A) \bmod n$ and she sends values Q , r , c and timestamp ts to Bob.

- + Bob checks ts and calculates $Q' = r.G + c.P_A$. Comparing Q' with Q , if they are the same then confirm that Alice knows x_A , otherwise Alice does not know x_A .

Analyzing the Properties of the Proposed ZKP Scheme Includes:

Completeness: If Alice provides the correct values Q , r and c , this means that, calculated from the values x_A , then Bob can check that Alice knows x_A . This is true because analyzing the formula Q' calculated by Bob will give the value Q . Indeed, $Q' = rG + cP_A = a_1G - cx_AG + cx_AG = a_1G = Q$.

Soundness: Alice is tampered and provides one of the three values Q , r and c which is wrong, then the point Q' according to the calculation formula cannot be equal to Q . Thus, Bob will confirm that Alice does not know x_A .

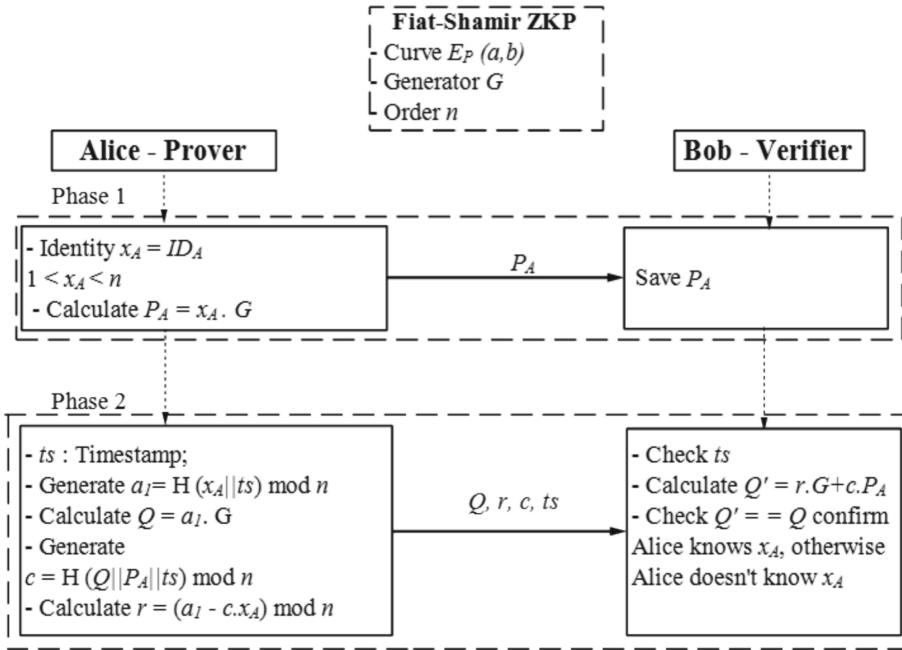


Fig. 2. Proposing Fiat-Shamir ZKP scheme

Zero-Knowledge: In both phases, it is seen that Alice does not provide any information about the value x_A . Alice only provides information about points Q, P_A , values r, c calculated from value x_A . If the attacker or Bob wants to know x_A , he needs to solve the ECDLP problem (Elliptic Curve Discrete Logarithm Problem) and this is a difficult problem that cannot be solved in polynomial time when the length of p is large.

3.2 Analyzing the Security of the Proposed Solution

The Problem of Security Vulnerabilities in Pseudorandom Number Generator

When using a hash function instead of a pseudo-random number generator, there are fault attacks into the hash function operation process. After that, the output or values a_1, c can be predicted. The problem was reported in [11] on the digital signing scheme EdDSA, and the countermeasure was proposed using an additional random input for the hash function. In the proposed scheme, timestamp is used as the additional random input (the generated all values a_1, c, r depend on the timestamp ts). Therefore the proposed protocol is able to prevent fault attacks on hash functions.

Anti-replay Attack Problem

In phase 1, Alice and Bob communicate together over a secure channel. So, Oscar can use the replay attack in phase 2 (Fig. 3). In this phase 2, Alice generates the values a_1, c, r with ts_1 is the time to verify (Alice's system) and calculates $= a_1 \cdot G$. Alice sends values Q, c, r to Bob and timestamp ts_1 . Oscar receives Q, c, r and ts_1 then sends Q, c, r at another time ts_2 to Bob.

Bob - Server after receiving the value Q, c, r along with timestamp ts_1 , then in a small amount of time Bob only accepts the session which is communicating with Alice associated with the value Q, c, r . In case Q, c, r is retransmitted with the timestamp ts_2 , Bob does not approve because at time ts_2 , Bob will log to save Q, r, c has included ts_1 at a time in the past.

The Problem of Attacking the Scheme According to Mathematical Theory

In phase 2 of the proposed scheme, the attacker intercepts the values Q, ts, c, r :

- Firstly, finding the value x_A through the equation $r = (a_1 - cx_A) \bmod n$. This problem can only be solved when executing all possible values of a_1 , this also cannot be done in polynomial time when choosing the domain parameter as a generating point G of sufficiently large degree n .
- Secondly, calculating the value x_A through point Q obtained by solving the elliptic curve discrete logarithm problem (ECDLP). Thus, the proposed scheme is safe when we choose the curve so that solving the ECDLP problem cannot take place in polynomial time.

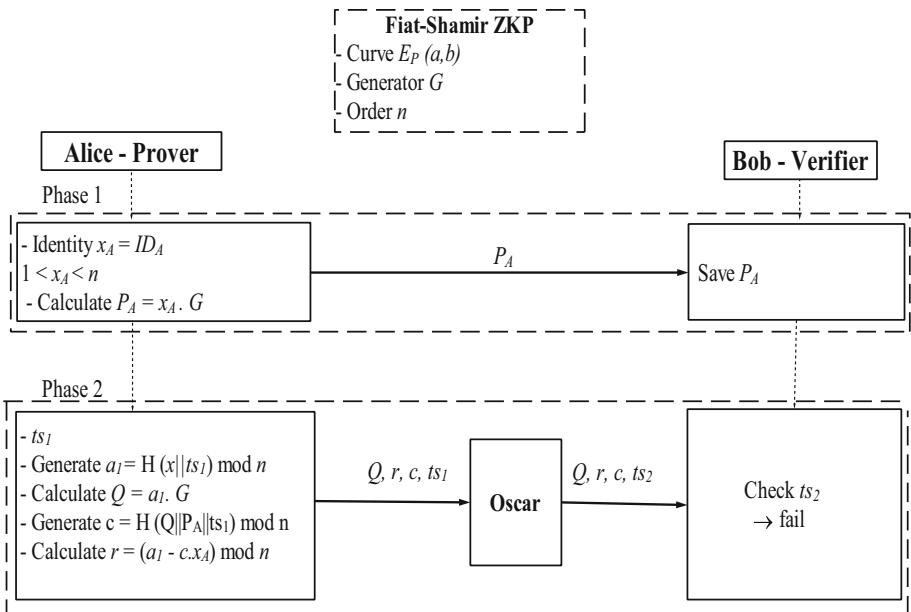


Fig. 3. Replay attack into proposed ZKP scheme

We refer to recommendations according to table 2 of the American National Standards Institute - NIST in choosing parameter sizes for basic cryptographic algorithms corresponding to the time of use introduced in 2020 [12].

From the NIST recommended table data, the implementation of the proposed protocols can choose a curve with a modulo p length of 256 bits or more to have a 128-bit security level that is good for use until 2030 and beyond.

Table 2. Key lengths recommended by NIST

Date	Security Strength	ECDLP (length of modulo p - bit)	RSA (length of modulo n - bit)	Hash Function
2019–2030	112	224	2048	SHA2-224 SHA3-224
2019–2030 & beyond	128	256	3072	SHA2-256 SHA3-256
2019–2030 & beyond	192	384	7680	SHA2-384 SHA3-384
2019–2030 & beyond	256	512	15360	SHA2-512 SHA3-512

Quantum Attack Resistance Problem. According to [13], cryptosystems based on the ECDLP problem are all affected by quantum attacks, specifically the Shor algorithm. Also in [13], the minimum number of qubits to solve the problem of integer and discrete logarithm analysis according to the safety equivalent to the key parameter length or modulo number is given in the following table:

Table 3. Minimum number of Qubits to solve the ECDLP problem and RSA integer factorization

Security Strength	#Qubits for ECDLP	#Qubits for Factoring of RSA Modulus N
112	2042	4098
192	2330	6146
256	3484	15362

The proposed protocol is based on the ECDLP problem; therefore, it is affected by the Shor attack on quantum computers. However, according to Table 3, when choosing parameters for proposed protocols with a security level of 112 bits or more, quantum computers are required to support from 2000 qubits or more. This moment is the year 2024, this is not feasible and according to NIST's recommendations in Table 2, when choosing parameters for the proposed ZKP scheme with a security level of 128 bits, the scheme can be used up to 2030 or beyond.

3.3 Comparing the Computational Cost and Experimental Results of the Proposed Scheme with the Fiat-Shamir ZKP Scheme in RFC 8235

For computational costs will be measured using the following calculations: Elliptic Curve Point Multiplication - ECPM, Elliptic Curve Point Addition -ECPA, Modular Multiplication - Mul, Modular Addition - Add, Hash.

Comment from above data, it is easy to see that the computational cost of the proposed Fiat – Shamir ZKP scheme has more calculations (hash functions) than the Fiat-Shamir ZKP scheme in RFC 8235 which is 2. However, calculating the hash function takes a very small time (micro seconds), so it can be said that this does not cause any big difference in the calculation speed of the proposed scheme compared to the Fiat-Shamir ZKP scheme in RFC 8235 (Table 4).

Table 4. Comparing the computational cost experimental results of the proposed scheme with the Fiat-Shamir ZKP scheme in RFC 8235

Scheme	Contact party	Calculation costs				Experimental results	
		ECPM	ECPA	Mul	Add	Time (second)	Data (byte)
Fiat-Shamir ZKP in RFC 8235	Alice	2	0	1	1	0.023	192
	Bob	2	1	0	0		0
Fiat - Shamir ZKP proposed	Alice	2	0	1	1	0.026	194
	Bob	2	1	0	0		0

3.4 Proposing Application to Authenticate in Client-Server Network Model

Details of the steps taken in the stages (Fig. 5) are as follows: Phases 1 and 2 will be performed using the Fiat-Shamir-ZKP scheme with the same steps as Fig. 2. Here, client A_i registers with server B login information from a password when logging into the network through server B with the Fiat-Shamir ZKP scheme (Fig. 4).

Advantages and Disadvantages of the Proposed Application Model. In this section, we conduct a comparison about authentication activity using account/password in Client - Server network model (Fig. 6), and the authentication model integrates passwords into ZKP (Fig. 5).

Figure 6 depicts the current widely used user authentication technique for the Client-Server network topology. In which the user needs to register an account/password for the server through a secure channel, and the server saves the user password hash code. During the verification phase, the user still has to provide his/her account/password to the server through a secure transmission channel for authentication.

Advantages. In the proposed application model, the registration phase needs a secure channel. In the authentication phase, the login process uses ZKP so users at the workstation do not need to transmit Identity/password information through a secure channel, however, users only send information such as Q , ID_{Ai} , c , r through public channel. This leads to avoid the risks of channel eavesdropping attacks and the costs needed to establish a secure connection compared to the current solution of using traditional accounts/passwords. The channel bandwidth for sending these values is also quite small, similar to sending user accounts/passwords.

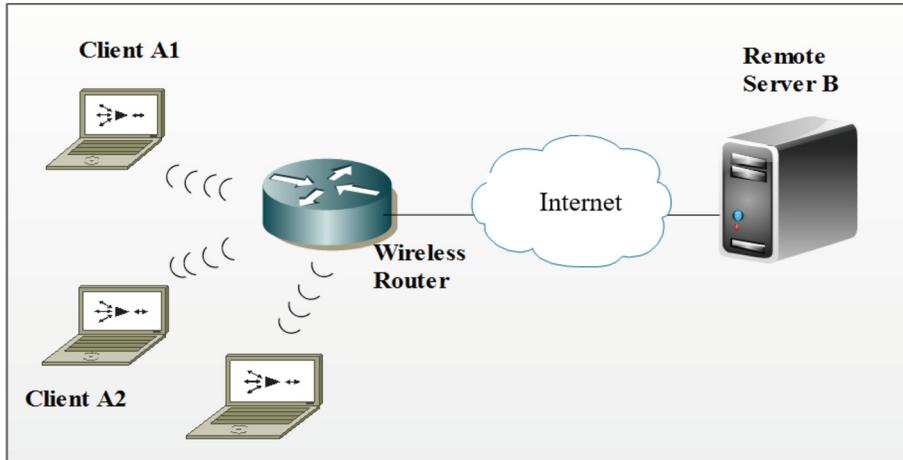


Fig. 4. Application model

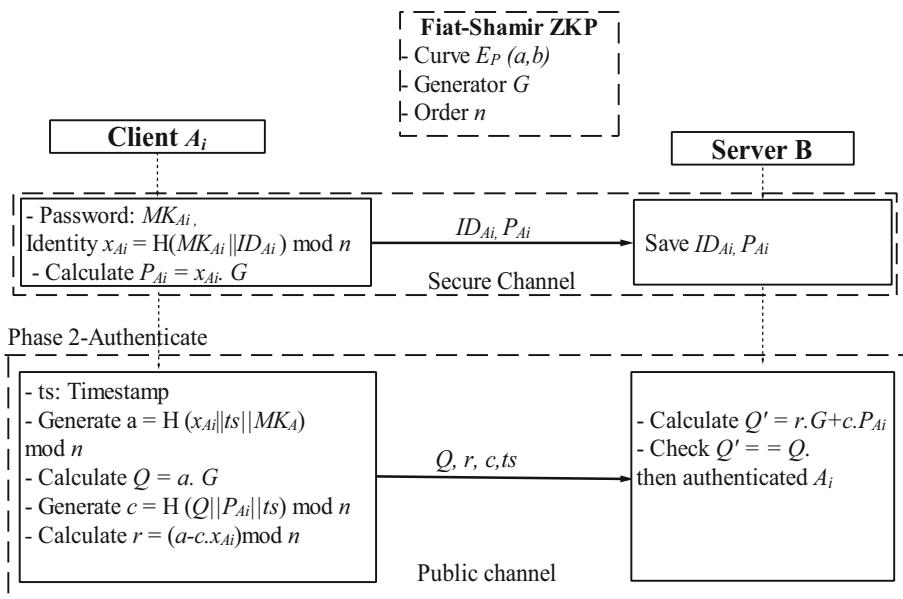


Fig. 5. Authentication model integrates passwords into ZKP

Disadvantages. The proposed model requires more computational steps than the traditional account/password model. However, for today's highly developed hardware systems, the cost of these calculations is not much.

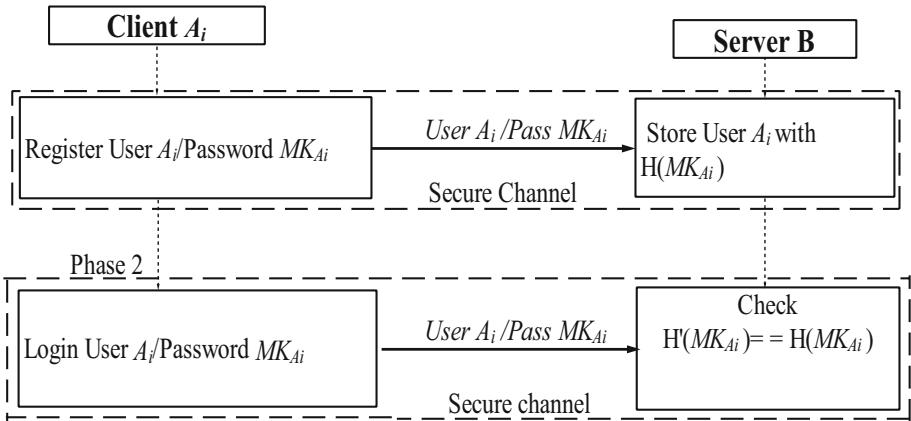


Fig. 6. Authentication activity using account/password in Client - Server network model

4 Conclusion

The paper proposed a fiat-shamir scheme on elliptic curves based on the use of a zero-knowledge proof mechanism. With higher security compared to the version in RFC 8235, whereas performance and bandwidth are not increasing too much, this proposed Fiat-Shamir ZKP scheme has great potential for practical implementation.

Acknowledgments. This work has been supported by Academy of Cryptography Techniques under Project/Lab.

References

1. Goldwasser, S.: The knowledge complexity of interactive proof systems. In: Proceedings of the 17th ACM Symposium on Theory of Computing, pp. 186–208 (1985)
2. Partala, J., Nguyen, T.H., Pirttikangas, S.: Non-interactive zero-knowledge for blockchain: a survey. IEEE Access **8**, 945–961 (2020)
3. Chen, Z., Jiang, Y., Song, X., Chen, L.: A survey on zero-knowledge authentication for internet of things. Electron. J. **5**, 1145 (2023)
4. Abdalla, M.: Password-based authenticated key exchange: an overview. In: Chow, S.S.M., Liu, J.K., Hui, L.C.K., Yiu, S.M. (eds.) ProvSec 2014. LNCS, vol. 8782, pp. 1–9. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12475-9_1
5. Gellersen, T., Seker, O., Eisenbarth, T.: Differential power analysis of the picnic signature scheme. In: Cheon, J.H., Tillich, J.P. (eds.) PQCrypto 2021. LNCS, vol. 12841, pp. 177–194. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-81293-5_10
6. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (eds.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1986). https://doi.org/10.1007/3-540-47721-7_12
7. Camenisch, J., Stadler, M.: Proof systems for general statements about discrete logarithms. Technical report, Department of Computer Science, Zurich (1997)

8. Hao, F.: Schnorr non-interactive zero-knowledge proof. In: Internet Engineering Task Force Documents, RFC 8235 (2017)
9. Chatzigiannakis, I., Pyrgelis, A., Spirakis, P.G., Stamatiou, Y.C.: Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices. In: Proceedings of IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems, Valencia, Spain, pp. 715–720 (2011)
10. Valsorda, F.: Exploiting ECDSA failures in the bitcoin blockchain. In: Proceedings of Hack in The Box (HITB) - Cloudflare, pp. 57–66 (2014)
11. Pornin, T.: Deterministic usage of the digital signature algorithm (DSA) and elliptic curve digital signature algorithm (ECDSA). RFC 6979 (2013)
12. Giry, D.: Recommendation for key management. Special Publication 800-57 Part 1 Rev. 5, National Institute of Standards and Technology of America (2020)
13. Roetteler, M., Naehrig, M., Svore, K.M., Lauter, K.: Quantum resource estimates for computing elliptic curve discrete logarithms. In: Cryptology ePrint Archive, no. 598, pp. 1–24 (2017)



Random Forest Model Parameters Optimization

Thuy Thi Tran¹ , Nghia Quoc Phan² , and Hiep Xuan Huynh³ 

¹ Faculty of Information Technology - Communication, University of Cuu Long, Vĩnh Long, Vietnam

² Assessment Office, Tra Vinh University, Tra Vinh, Vietnam

³ College of Information and Communication Technology, Can Tho University, Can Tho, Vietnam

hxhiep@ctu.edu.vn

Abstract. Machine learning models have been widely used in many applications in almost all areas of social life. Random forest is a supervised machine learning model that combines the results of multiple decision trees to achieve a single result using closure. Due to the ease of use and flexibility of the random forest machine learning model, there has been a push for its adoption in practical applications of both regression and classification problems. To fit the random forest machine learning model to different problems, the model parameters must be adjusted. Choosing the best parameter configuration for the model has a direct impact on the model's performance. In this article, the parameters of the random forest model and parameter optimization algorithms are studied in detail. Furthermore, the study also tested different benchmark datasets to compare the performance of random forest model parameter optimization methods.

Keywords: random forest · parameter optimization · grid search · random search · Bayesian optimization

1 Introduction

Machine learning models have been commonly used in many applications in almost all application areas of social life from health, environment, finance, urbanization to computer vision and object recognition [1–4]. Machine learning algorithms can be divided into three basic types: supervised learning, unsupervised learning and reinforcement learning.

Random forest is a supervised machine learning model that combines the results of multiple decision trees to achieve a single result using the closure method [5, 6]. Due to the ease of use and flexibility of the random forest machine learning model, there has been a push for its adoption in practical applications of both regression and classification problems [7].

To fit the random forest machine learning model to different problems, the model parameters must be adjusted. Choosing the best parameter configuration for the model has a direct impact on the model performance [8, 9].

However, parameter adjustment is often performed manually, depending on the model researcher's experience [8]. For machine learning models with large parameter spaces such as random forests, this problem often takes a long time but is not very effective [10, 11]. This raises the question of how to find a more effective approach. This study will focus on the parameter space of the random forest model, model evaluation metrics, parameter tuning methods, and evaluation of the results with the classification accuracy of the random forest classifier and mean absolute error (MSE) of the regression random forest model. The two datasets used for the study are MNIST [12] and Boston-Housing [13].

The next part of this article is Sect. (2) which presents and analyzes related research; Sect. (3) presents the random forest algorithm and discusses in detail the parameters of the random forest algorithm and model evaluation methods; Current popular parameter optimization methods are presented in Sect. (4); Apply parameter optimization methods to the random forest algorithm clearly stated in Sect. (5); Experimental Sect. (6) will run sample standard data sets to get an overview of parameter optimization methods for random forest algorithms; The achieved results will be presented in Sect. (7), and finally the discussion section and an overview of the current situation of using parameter optimization methods for machine learning models.

2 Related Works

The performance of machine learning models affected by parameter configuration has also been validated by many researchers [8–10]. However, there has not been much research on optimizing the parameters of the regression and random forest classifiers to evaluate the accuracy and error of predictions. In the study [10], it was determined that the *max_features* parameter has a great influence on the accuracy of the random forest classifier, it is necessary to optimize this parameter. Another study, Probst et al. [11] also identified a random forest classifier that performed quite well with default parameters provided by different software packages, HPO methods do not significantly increase classification accuracy or some default parameters are close to optimal. Most HPO methods require researchers to clearly understand each parameter and specifically the values of each parameter obtained to increase model performance [8].

Nygren et al. [10] compared the performance of random forest models with default parameters and parameter optimization methods with random forest classifiers. However, the study also confirmed that there is not much difference between HPO methods. The performance of HPO methods also depends heavily on different data sets.

3 Parameters of the Random Forest Model

3.1 Random Forest Model

The main task of supervised machine learning algorithms is to minimize the cost function $\mathcal{L}(f(x, y))$. Where x is the input and y are the output, both are available values. The cost function \mathcal{L} is calculated as the error of the predicted output and the ground-truth label. Prediction models f are designed depending on machine learning models. Equation (1)

is used to build an optimal prediction model f^* in the limited value domain of a set of functions F [9].

$$f^* = \underset{f \in F}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f(x_i), y_i) \quad (1)$$

where, n is the number of data points for learning (training), x_i is the vector of features of the i th data point, y_i is the corresponding output data, and \mathcal{L} is the value of the cost function corresponding to each point sample

In addition to the cost function, the loss function in supervised machine learning algorithms also needs to be considered for optimization [9]. Depending on different machine learning algorithms, the loss functions can be cross-entropy functions, information gain or squared Euclidean distance functions. Each machine learning algorithm will create a different prediction model depending on the architecture of that algorithm's parameters.

Decision tree (DT) [6] is a popular classification algorithm. DT uses a tree structure to model decisions, and the results are synthesized by deriving a set of classification rules from the input data set. The main components of the DT are the root node (the entire input data set); decision nodes (decision trials) and branches on each feature; leaf nodes (representing output layers). The DT algorithm recursively divides the training set with better feature values to reach the best match on each child branch. During the branching process, some child nodes of the decision node will be pruned. Pruning helps the DT algorithm avoid overfitting [6].

Based on the decision tree model, there are many algorithms proposed to improve model performance by combining multiple trees such as random forests (RF), extra trees (ET) and extreme gradient boosting (XG-Boost) models [7]. Random forest is a machine learning algorithm that synthesizes the results of decision trees using bagging. Many decision trees are built on randomly generated subsets of data from the input data set and select the class with majority voting as the final classification result.

Models built based on decision trees have the same parameters as the decision tree model. Additionally, RF, ET and XG-Boost all have an important parameter that needs to be fine-tuned which is the number of decision trees to be combined (*n-estimators*) [14].

3.2 Random Forest Model Parameters

Details of the parameters of the random forest model are presented [7, 10]. Each parameter has a data type, default value and meaning of each parameter. Normally, when using random forest models, researchers must rely on their own experience to find appropriate values for parameters to increase model performance. Sometimes it takes a lot of time and experience to conduct experimental runs with many different parameter values for comparison.

3.3 Random Forest Model Assessment Methods

Machine learning models are applied in many different scientific fields of social life. In machine learning models, the input data is used as training data to build a model to predict

the label for a new sample. The output of the models needs to be evaluated and analyzed in detail and interpreted using different numerical methods. Two commonly used methods today are scalar data and graphics. Scalar metrics such as accuracy, sensitivity, and specificity. Graphical evaluation methods such as ROC [15]. In this study, accuracy and Mean Squared Error were used to evaluate the classifier and regressor for the random forest model.

Classification Assessment Method

Accuracy [16], fraction (default) or number (normalize = False) of correct predictions. In multi-label classification, accuracy is the accuracy of the subsets. If the entire set of predicted labels for a sample matches the actual label set exactly, then the accuracy of the subset is 1.0; otherwise, it is 0.0. If \hat{y}_i is the predicted value of the i th sample and y_i is the corresponding real value, then the correct prediction rate on $n_{samples}$ is determined by Eq. (2):

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{sample}} \sum_{i=0}^{n_{sample}-1} 1(\hat{y}_i = y_i) \quad (2)$$

where, $1(x)$ is the indicator function.

Regression Assessment Method

Mean squared error (MSE) [16], a measure of risk corresponding to the expected value of the squared error or loss. If \hat{y}_i is the predicted value of the i th sample and y_i is the corresponding actual value, the estimated mean square error on $n_{samples}$ is determined by Eq. (3):

$$\text{mean squared error}(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2 \quad (3)$$

4 Hyperparameter Optimization Methods

4.1 Mathematical Optimization

Mathematical optimization is the process of selecting the best solutions from the set of available candidates to achieve the desired objective function [8, 9]. There are two commonly used types of optimizations including constrained optimization and unconstrained optimization. The types of optimizations determined with or without constraints are based on determining the relationship with solution variables and decision variables.

In optimization problems without constraints, the decision variables can take arbitrary real values in one-dimensional space [8]. Equation (4) represents an optimization problem without constraints:

$$\min_{x \in \mathbb{R}} f(x) \quad (4)$$

where x is a decision variable and $f(x)$ is the objective function. This means that for this problem, the objective function value that must be achieved is the minimum.

However, most optimization problems in practice are constrained. Constraints are often by inequalities or equality. Equation (5) is a representation of a general constrained optimization problem:

$$\begin{aligned} & \underset{x \in \mathbb{R}}{\min} f(x) \\ & \text{depends on} \\ & h_i(x) = 0, i = 1, 2, \dots, p \\ & g_i(x) \leq 0, j = 1, 2, \dots, m \end{aligned} \tag{5}$$

where, x is a decision variable, defined in the domain X , $h_i(x), i = 1, 2, \dots, p$ is the equality constraint function and $g_j(x)$ with $j = 1, 2, \dots, m$ are inequality constraint functions.

The optimal solution is determined in the specified domain depending on the constraints. That value region is called the feasible region [9]. Equation (6) is used to represent the feasible area:

$$D = \{x \in X | g_i(x) \leq 0, h_j(x) = 0\} \tag{6}$$

An optimization problem is defined by three elements including decision variables, objective function and constraints. These three factors make it possible for an optimization problem to determine an objective function for the decision variables that receive the best values according to a set of constraints.

One problem in optimization problems is that the process of finding the optimal solution is only achieved on a local scale, but not on a global scale. This requires continuing to search for the optimal solution in the decreasing direction of the objective function to discover the global minimum value in convex functions. The convex function $f(x)$ has the form like Eq. (7):

$$f(tx_1 + (1 - t)x_2) \leq tf(x_1) + (1 - t)f(x_2) \tag{7}$$

where, $\forall x_1, x_2$ are decision variables in the domain X , t is a coefficient with a value in the interval $[0, 1]$.

The general goal of parameter optimization problems for machine learning models needs to be achieved according to Eq. 8:

$$x^* = \underset{x \in X}{\operatorname{argmin}} f(x) \tag{8}$$

where, x is the decision variable in the domain X , $f(x)$ is the objective function to be achieved, x^* is the architecture of the parameter optimization model that creates the desired value of the objective function $f(x)$.

The general algorithm of hyperparameter optimization (HPO) models [8] is as follows:

Algorithm HPO

-
- 1: Choose the objective function and performance scale.
 - 2: Select the parameters that need to be fine-tuned.
 - 2.1: Determine the type of parameters that need to be fine-tuned.
 - 2.2: Determine appropriate optimization methods.
 - 3: Train the machine learning model using the default parameter optimization model (set the default global value to be the base optimization model).
 - 4: Perform the optimization process with a manually defined feasible region based on knowledge of the machine learning model.
 - 5: Identify areas where parameter values perform well.
 - 6: Returns the best solution for the most efficient parameters.
-

Some limitations of previous optimization methods compared to HPO problems [9] can be mentioned as most previous optimization methods can only solve local optimization problems. Therefore, it can only solve convex and differentiable optimization problems. While current HPO methods aim to optimize the objective functions of machine learning models. Many previous optimization methods only achieved the goal of continuous and variable parameters, but were not feasible in handling discrete, conditional and categorical parameters. Another limitation of previous optimization solutions is the issue of time and resources. Current machine learning problems, with many data sources, require optimal solutions that must achieve efficiency in both time and savings in computing resources, but still ensures to receive the exact target rather than accepting approximate values like black box optimization (BBO) [17]. So, the problem is to have an optimal solution suitable for HPO problems in order to process and create the optimal parameter architecture of machine learning models.

4.2 Hyperparameter Optimization Methods

Machine learning model parameter optimization enables users to determine the most suitable machine learning model for a specific problem (dataset). Because only when different machine learning models use the same HPO method on the same dataset can the performance be accurately evaluated [18].

The problem is how to choose a suitable HPO solution to handle hyperparameters for machine learning models. Traditional parameter optimization methods are not suitable for current problems. Because the situation is not globally optimal, it is only locally optimal in non-differentiable and non-convex optimization problems [19]. Popular traditional HPO algorithms today are gradient descent-based, which is used to calculate gradients for continuous parameters (e.g., learning rate in neural networks) [20].

Currently, there are a number of HPO methods used such as Bayesian optimization (BO) models, methods based on decision theoretic approaches, metaheuristics algorithms [34, 35, 36] and multi-fidelity optimization techniques [37, 38]. These HPO methods can handle several types of parameters well such as conditional parameters, continuous parameters, discrete parameters, and categorical parameters.

5 HPO for Random Forest Model

Each different machine learning model has unique parameters [2]. However, it can be summarized into two basic types of parameters including parameters that are named by machine learning models and are assigned initial values and change during training on the input data set (e.g. as the weight of a node in a decision tree) [8]. Parameters that must be initialized before starting the model training process are often called hyperparameters. Hyperparameters cannot be directly computed during training data. The parameter space configuration of machine learning models is determined by hyperparameters [10].

Hyperparameters of the random forest model such as number of decision trees to be combined (n-estimator), maximum number of features considered to separate a node (max-features), maximum number of levels in each decision tree (max-deep), minimum number of data samples placed in a node before being split (min-samples-split), minimum number of data samples allowed in a leaf node (min-sample-leaf), data sampling method (bootstrap) [8, 10, 14].

A method that aggregates the results of decision trees to make predictions is called a random forest classifier. Decision trees are trained independently on a subset of the training data using a statistical technique called bootstrapping [5]. This statistical technique is used to obtain many sub-training datasets by sampling from the original dataset. This sampling problem allows data samples to be duplicated using replacement, meaning a subset is sampled within the original data set. This ensures that many data sets can be obtained regardless of the size of the original data set [7].

6 Experiment

6.1 Data Used

MNIST

The MNIST (Modified National Institute of Standards and Technology) dataset is a large database containing handwritten digits [12]. MNIST is commonly used to train models in image processing systems. This data set includes 60000 samples for the training set and the test set includes 10000 samples. MNIST is a subset of the larger NIST Special Database 3 (handwritten digits written by United States Census Bureau employees) and Special Database 1 (handwritten digits written by high school students), containing monochrome images of handwritten digits. The digits were standardized in size and centered in an image of a fixed size of 20×20 pixels.

Boston-Housing

The Boston Housing dataset is a dataset developed by the U.S. Census Service that collects housing in the Boston Mass area during the 1970 census. The Boston-Housing data frame contains the original data from Harrison and Rubinfeld [13]. The dataset is obtained from the StatLib library and maintained by Carnegie Mellon University. The dataset is small with only 506 cases with 14 features (*crim*, *zn*, *indus*, *chas*, *nox*, *rm*, *age*, *dis*, *rad*, *tax*, *ptratio*, *b* and *stat*) with *medv* feature as the target variable.

6.2 Tools Used

Scikit-learn [14] is a simple and effective tool for predictive data analysis, built on top of *SciPy*, *NumPy*, and *matplotlib*. Scikit-learn can solve regression, classification, and clustering problems. With the classification problem, the goal is to determine the type of objects which can be applied to spam detection and image recognition. For regression, Scikit-learn supports predicting an attribute with a continuous value associated with an object, applied in problems of making predictions about a patient's drug response and stock price prediction. Applications that need to produce results about customer segmentation or clustering test results are clustering problems that are also supported by Scikit-learn. In addition, Scikit-learn also supports problems of dimensionality reduction, comparison, validation, and selection of parameters for models or preprocessing such as feature extraction and normalization.

6.3 Scenario 1: Random Forest Algorithm with Default Parameter Space

The default parameter space is used for the random forest algorithm as the data samples when sampling will be bootstrapping (*bootstrap = true*), *ccp-alpha = 0.0* means that no pruning will be done with the subtrees, the weights of the classes are the same and equal to 1 (*class-weight = None*), the criterion to measure the quality when dividing the decision node (*criterion*) is determined as *Gini*, *max-depth = None* means that the nodes will be divided divide until the leaf nodes are all labeled, the maximum number of features (*max-features*) is determined by $\sqrt{n - \text{features}}$, the number of leaf nodes is unlimited (*max-leaf-nodes = None*), the number of samples to take (*max-samples*) is *None*, *min-impurity-decrease = 0.0* means nodes will be divided with impurity reduction greater than or equal to 0.0, minimum number of samples required in a leaf node is 1 (*min-samples-leaf = 1*) and number of samples the minimum required to split a node in is 2 (*min-samples-split = 2*), the minimum weight of the weighted sum is 0.0 (*min-weight-fraction-leaf = 0.0*), the number of trees of the random forest is 100 (*n-estimators = 100*), the number the number of jobs running in parallel is 1 (*n-jobs = 1*), the randomness of the sampling process when building the tree *random-state = None*, *verbose = 0* is the granularity when pattern matching and prediction, *warm-start = False* means does not call the previous command when matching data patterns but a completely new group.

6.4 Scenario 2: Optimize Parameters for Random Forest Model Classifier and Regressor

The parameter space and parameter values are used to test the parameter optimization scenario for the random forest classifier and regression with optimization methods as shown in Table 1:

Table 1. Parameter configuration for random forest classifier and regression with HPO methods

HPO Parameter	RS	GS	BO-GP	BO_TPE
n-estimators	[10, 20, 30]	sp-randint(10,100)	Integer(10,100)	int(params['n-estimators'])
max-features	['sqrt', 'auto', 0.5]	sp-randint(1,64)	Integer(1,64)	int(params['max-depth'])
max-depth	[15,20,30,50]	sp-randint(5,50)	Integer(5,50)	int(params['max_features'])
min-samples-leaf	[1,2,4,8]	sp-randint(2,11)	Integer(2,11)	int(params['min-samples-split'])
min-samples-split	[2, 5, 10]	sp-randint(1,11)	Integer(1,11)	int(params['min-samples-leaf'])
bootstrap	[True, False]	[True, False]	[True, False]	bool(params[bootstrap])
criterion	['Gini', 'entropy', 'log-loss']	['Gini', 'entropy', 'log-loss']	['Gini', 'entropy', 'log-loss']	str(params['criterion'])

With the parameter space configuration of the random forest model with the corresponding HPO methods in Table 1. For example, the Random Search method, criterion = ['Gini', 'entropy', 'log-loss'], max-depth = [15, 20, 30, 50], max-features = ['sqrt', 'auto', 0.5], min-samples-leaf = [1, 2, 4, 8], min-samples-split = [2, 5, 10], n-estimators = [10, 20, 30], bootstrap = [True, False].

7 Results

Experimental results with HPO algorithms such as Grid Search, Random Search, Bayes optimization with Gaussian process and Bayes optimization with TPE. The detailed running results of the Random Search algorithm with the random forest classifier on the MNIST dataset, the calculated parameter configuration values are varied, and accuracy is standard to measure the model performance. After the steps of finding the optimal parameter space configuration using the Random Search method, the final result is received with the corresponding value as *criterion* = *Gini*, *max-depth* = 25, *max-features* = 2, *min-samples-leaf* = 2, *min-samples-split* = 6, *n-estimators* = 66, with an accuracy of 92.93%.

Comparison of the results of the HPO methods with the random forest model regression suite is shown in Table 2, with each HPO method having parameter values of the corresponding parameter configuration shown. The results show that the MSE of the HPOs is much lower than the default random forest model. However, the execution time of HPOs is higher. *Grid Search's MSE is the lowest (25.64) but has the highest time (79.94 s)*. The parameter values in the optimal parameter configuration of Grid Search correspond to *criterion* = *mse*, *max-depth* = 15, *max-features* = 0.5, *min-samples-leaf* = 4, *min-samples-split* = 2, *n-estimators* = 20.

The results of the HPO algorithm with the random forest model classifier are shown in Table 3. Using the results of HPO with the regression suite, the accuracy of *BO_TPE* is the highest (94.21%) with a time of 31.31 s. The parameter values in the optimal parameter configuration of BO_TPE correspond to *criterion* = 1, *max-depth* = 27, *max-features* = 5, *min-samples-leaf* = 2, *min-samples-split* = 2, *n-estimators* = 56.

Table 2. HPO results table for random forest regression with Boston-housing dataset

	criterion	max-depth	max-features	min-samples-leaf	min-samples-split	n-estimators	MSE	seconds
RF (default)	squared-error	None	auto	1	2	100	31.72	0.67
GS	mse	15	0.5	4	2	20	25.64	79.94
RS	mse	43	5	5	6	28	27.47	10.27
BO-GP	mse	43	7	1	10	87	26.39	36.34
BO-TPE	0	20	4	1	10	16	27.03	11.38

Table 3. HPO results table for random forest classifier with MNIST dataset

	criterion	max-depth	max-features	min-samples-leaf	min-samples-split	n-estimators	Accuracy (%)	seconds
RF (default)	'Gini'	None	None	None	None	100	91.88	1.88
GS	'Gini'	15	'sqrt'	1	2	30	93.99	95.93
RS	'Gini'	25	2	2	6	66	92.93	27.98
BO_GP	entropy	14	1	1	5	100	93.88	61.92
BO_TPE	1	27	5	2	2	56	94.21	31.31

8 Conclusion

With the results of the experimental part, it can be concluded that the performance of HPOs has different results. However, there is still no HPO that outperforms other HPO methods. For example, with the random forest classifier BO_TPE is the highest (94.2126%) but compared to other HPOs it is still not much higher. The accuracy of BO_GP is still close to BO_TPE at 93.8787% or Grid Search at 93.9899833%. Compared to running time, Random Search still has the advantage, with the random forest regression, Random Search has a running time of 10.2722 s and with the classifier it is 27.983 s. Random Search has the lowest running time among the HPOs, but the results are still not much different. Although the results of Grid Search are relatively good, the execution time is the highest because it has to search through all cases.

Currently, the research only uses two datasets including MNIST for the classifier and Boston-Housing for the random forest regression. The HPOs used are Grid Search, Random Search, Bayes optimization with Gaussian process and Bayes optimization with TPE. In the future we will use more datasets and add multi-fidelity HPOs and metaheuristics HPOs to obtain more objective comparisons of their performance.

References

1. Mitchell, T.M.: Machine learning, McGraw-Hill, Maidenhead, U.K., International Student Edition, 414 (1997). ISBN: 0-07-115467-1
2. Jordan, M.I., Mitchell, T.M.: Machine learning: trends, perspectives, and prospects. Science **349**(6245), 255–260 (2015). <https://doi.org/10.1126/science.aa8415>
3. Santosh, K.C., Das, N., Ghosh, S.: Deep Learning Models for Medical Imaging, Academic Press, pp. 1–27 (2022). ISBN 9780128235041, <https://doi.org/10.1016/B978-0-12-823504-1.00011-8>

4. El-Kassas, W.S., Salama, C.R., Rafea, A.A., Mohamed, H.K.: Automatic text summarization: A comprehensive survey. *Expert Syst. Appl.* **165** (2021). ISSN 0957–4174, <https://doi.org/10.1016/j.eswa.2020.113679>
5. Quinlan, J.R.: Induction of decision trees. *Mach. Learn.* **1**, 81–106 (1986)
6. Rasoul, S., David, L.: A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **21**, 660–674 (1991)
7. Svetnik, V., et al.: Random forest: a classification and regression tool for compound classification and QSAR modeling. *J. Chem. Inf. Model.* **43**(6), 1947–1958 (2003). <https://doi.org/10.1021/ci034160g>
8. Yang, L., Shami, A.: On hyperparameter optimization of machine learning algorithms: theory and practice. *Neurocomputing* **415**, 295–316 (2020). <https://doi.org/10.1016/j.neucom.2020.07.061>
9. Gambella, C., Ghaddar, B., Naoum-Sawaya, J.: Optimization models for machine learning: a survey 1–40 (2019). <http://arxiv.org/abs/1901.05331>
10. Nygren, R., Petkov, A.: Evaluation of hyperparameter optimization methods for Random Forest classifiers, Dissertation (2021)
11. Probst, P., Wright, M.N., Boulesteix, A.L.: Hyperparameters and tuning strategies for random forest. *Wiley Interdisc. Rev. Data Min. Knowl. Discov.* **9**(3), e1301 (2019). <https://doi.org/10.1002/widm.1301>
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
13. Harrison, D., Rubinfeld, D.L.: Hedonic prices and the demand for clean air. *J. Environ. Econ. Manag.* **5**, 81–102 (1978)
14. Pedregosa, F., et al.: Scikit-learn: Machine learning in Python, *Computer Science – Machine Learning* (2012).<https://doi.org/10.48550/arXiv.1201.049>
15. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233–240. ACM (2006)
16. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. *Int. J. Forecast.* **22**(4), 679–688 (2006). CiteSeerX 10.1.1.154.9771. <https://doi.org/10.1016/j.ijforecast.2006.03.001>. S2CID 15947215
17. Steinholtz, O.S.: A Comparative Study of Black-box Optimization Algorithms for Tuning of Hyper-parameters in Deep Neural Networks, M.S. thesis, Department of Electrical Engineering, Lulea University of Technology (2018)
18. Hutter, F., Kotthoff, L., Vanschoren, J.: *Automatic Machine Learning: Methods, Systems, Challenges*, Springer (2019). ISBN: 9783030053185
19. Luo, G.: A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Netw. Model. Analy. Health Inform. Bioinform.* **5**, 1–16 (2016). <https://doi.org/10.1007/s13721-016-0125-6>
20. Maclaurin, D., Duvenaud, D., Adams, R.P.: Gradient-based Hyperparameter Optimization through Reversible Learning (2015). <http://arxiv.org/abs/1502.03492>

Natural Language Processing



Building a Q&A System to Serve Undergraduate Education at Can Tho University

Bao-Dang Le Nguyen and Nguyen-Khang Pham

College of Information and Communication Technology, Can Tho University, Can Tho, Vietnam
dangb2016955@student.ctu.edu.vn, pnkhang@cit.ctu.edu.vn

Abstract. The CTU-Helper system is an automated question-answering system designed to assist students and prospective applicants at Can Tho University (CTU) in quickly and efficiently accessing information regarding academic regulations and admissions. This system utilizes Retrieval-Augmented Generation (RAG) combined with Bi-encoder and Cross-encoder models for natural language processing. The Bi-encoder allows the system to compare user questions with questions stored in the database, thereby identifying corresponding answers. The Cross-encoder is used to assess the relevance between user questions and potential answers, ensuring the accuracy of returned results. Additionally, CTU-Helper is equipped with Semantic Router technology, which facilitates semantic classification for queries. This technology enables the system to identify the main topic of the question and forward the query to the appropriate processing model, enhancing information retrieval efficiency. Evaluation results indicate that CTU-Helper achieves high performance with a Recall of 0.8683 in the information retrieval process. This demonstrates the system's ability to effectively search and return accurate information to users.

With its outstanding advantages, CTU-Helper promises to be a valuable support tool for students and applicants at CTU, facilitating easier and more convenient access to information. This system also opens up new research directions in applying advanced natural language processing techniques to build automated question-answering systems in the educational field.

Keywords: Question-answering system · Retrieval-Augmented Generation (RAG) · Contextual query processing

1 Introduction

Each year, Can Tho University (CTU) welcomes thousands of new students and continues to educate tens of thousands of current students. The academic journey at the university involves understanding numerous regulations and admission information, which students must be well-acquainted with. However, accessing and retrieving this information often poses challenges. The large and dispersed amount of information, published by various university departments, makes it difficult to consolidate and update. Additionally, the administrative language used in these documents can be a barrier, making it hard for students to comprehend the content. The task of answering students' questions often

falls on department staff, leading to overload and long waiting times, especially during peak periods such as admissions and enrollment. Questions about the university and its programs also help prospective students who wish to apply to CTU.

To address these challenges, implementing an automated question-answering system using information technology is an optimal solution. Such a system would enable students to easily access information and get quick, accurate answers to their queries. CTU-Helper is such a system, developed using Retrieval-Augmented Generation techniques and advanced language models, particularly Google Gemini Pro and OpenAI GPT-4.

The CTU-Helper system is expected to bring numerous practical benefits to CTU students. They can ask questions and receive immediate answers from the system, eliminating wait times and reducing dependency on support staff. The system also simplifies and clarifies regulations and notices into easy-to-understand language, aiding students in effectively absorbing the information. Furthermore, CTU-Helper reduces the workload for administrative staff by automatically addressing most frequently asked questions, allowing them to focus on more complex issues.

Given these advantages and application potentials, CTU-Helper promises to become an invaluable tool for CTU students, enhancing the quality of student support services at the university.

2 Related Work

Prior to the development of CTU-Helper, several attempts were made to leverage information technology to address the challenges of information access for CTU students. Early systems primarily relied on traditional machine learning methods such as Support Vector Machines (SVM) and Naive Bayes for intent classification. Based on the identified intent, these systems would provide generic responses or predefined answers tailored to specific intents.

However, this approach presented several limitations. The accuracy of intent classification often faltered, leading to irrelevant or inaccurate responses to user queries. This stemmed from the inherent difficulty in accurately discerning a user's true intention from their input. Furthermore, manually defining answers for each individual intent proved to be a laborious and impractical process, ultimately failing to encompass the vast spectrum of potential user queries.

While more recent efforts have incorporated frameworks like Rasa for building conversational AI systems, challenges still persisted. These earlier systems struggled to grasp the overall meaning of user queries, often focusing on individual keywords instead of understanding the context and nuances of the user's question. This reliance on superficial analysis hindered the systems' ability to provide relevant and accurate information. Furthermore, these systems were often restricted to answering questions phrased in a specific, predefined manner, lacking the flexibility to handle variations in phrasing or more complex questions. This inflexibility limited their utility and made interactions cumbersome for users. Compounding these issues, the knowledge base of these systems was often limited in scope, preventing them from effectively addressing the diverse range of user queries.

3 System Architecture

The CTU-Helper system uses Retrieval-Augmented Generation (RAG) [1] techniques to enhance the efficiency and accuracy of answering queries. Typically, building a question-answering system using large language models involves fine-tuning the model based on a specific dataset. However, this approach may be ineffective for small or frequently changing data sources, such as university regulations or admission information that is regularly updated. Fine-tuning the model every time data changes is costly and time-consuming. RAG is an effective solution to this problem. This technique allows the system to leverage existing textual data repositories to supplement the language model, enabling more accurate answers and avoiding hallucination (fabricated information). By using RAG, CTU-Helper optimizes resources while ensuring the information returned to users is up-to-date and accurate. Our system primarily focuses on two main components of a question-answering system: the process of handling data that contains information for retrieval and the application of various techniques to optimize the retrieval process. By ensuring the retrieval of truly useful and relevant information to the user's query, the third-party large language model can infer and generate the most accurate answers for the users.

4 Source Data Processing

4.1 Data for the Question-Answering Process

The primary data for CTU-Helper includes documents and regulations, which can be either structured or unstructured. Unstructured data typically consists of standard text paragraphs, which can be directly stored in the database. On the other hand, structured data is usually presented in tabular form. To handle structured content, we transform the information from tables into text paragraphs that encapsulate the data from the table fields. This process is informed by the methodology outlined in the study [2]. An example of how table data is converted into textual format is illustrated in Fig. 1.

Once the data has been standardized into text paragraphs, we further divide it into smaller text chunks. Each chunk will include essential retrieval information such as the document name, index and names of parent indices, a summary header, and the main content of the chunk. As of May 2024, CTU-Helper's data repository is organized to support two main topics: Academic Regulations and Rules, and Admission Information for Can Tho University. The distribution and average length of data chunks are detailed in Table 1.

After chunking the text, we generate semantic vectors (embeddings) by embedding these texts using Sentence Transformer models [3]. Specifically, we use the model “bkai-foundation-models/vietnamese-bi-encoder” [4]. This model is an S-BERT architecture featuring two PhoBERT models [5] pre-trained on Vietnamese text, with tokenization at the word level. The model is further fine-tuned on a Vietnamese dataset for semantic similarity tasks. Consequently, before embedding, the text undergoes tokenization using the Underthesea library.

For the data used in the question-answering process, updates will be conducted periodically as decided by the system administrator. These updates usually occur after

2. GENERAL EDUCATION PROGRAMS						
Program Code	Program Name (Specialization, if any)	Expected Enrollment	Admission Combinations		2023 Threshold Score	
TEACHER TRAINING						
7140201	Early Childhood Education	70	M01, M06, M11	V-SAT scores not considered	New	New
7140202	Primary Education	100	A00, C01, D01, D03	V-SAT scores not considered	28,20	24,41
7140204	Civic Education	70	C00, C19, D14, D15	V-SAT scores not considered	27,50	26,86

↓
converted to

"The Early Childhood Education program, with the program code 7140201, has admission combinations for the transcript and National High School Exam methods as M01, M06, M11, and does not consider V-SAT scores"

"The Primary Education program, with the program code 7140202, has admission combinations for the transcript and National High School Exam methods as A00, C01, D01, D03, and does not consider V-SAT scores. The transcript admission score is 28,20, and the National High School Exam admission score is 24,41"

"The Civic Education program, with the program code 7140204, has admission combinations for the transcript and National High School Exam methods as C00, C19, D14, D15, and does not consider V-SAT scores. The transcript admission score is 27,50, and the National High School Exam admission score is 26,86"

Fig. 1. Process of Interpreting Tabular Data

Table 1. Distribution and Average Length of Data Chunks for Question-Answering

Topic	Metric	
	Number of Chunks	Average Length
Academic Regulations	598	745
Admission Information	74	1101

the issuance or revision of decisions and documents that may affect the students at the university, or at regular intervals over a specific period. The data that needs to be updated will be aggregated from sources such as the Department of Academic Affairs and the Department of Student Affairs, processed into chunks, and then added to the vector database.

4.2 Data for the Semantic Router

The Semantic Router plays a critical role in classifying user questions based on their content. It operates on a dataset of approximately 500 questions, divided into two primary topics: academic regulations and admission information. This dataset is compiled from the university's official website, official documents, and questions posed by students on social media platforms dedicated to the student community.

To enhance the dataset, each initial question is expanded using ChatGPT to generate five additional questions with similar content but varied phrasing. Each question is labeled with its corresponding topic to ensure accurate classification by the Semantic Router.

The questions are tokenized, meaning each word in the question is separated, and then embedded into a vector using the "bkai-foundation-models/vietnamese-bi-encoder" model. This model transforms the questions into semantic vectors, which are stored in

the vector database in the same manner as the information on academic regulations or admission data.

The Semantic Router dataset features diverse content and phrasing, may include educational terminologies, and is regularly updated to maintain accuracy. By categorizing the topics of questions, the Semantic Router helps CTU-Helper narrow down the search scope within the data repository, thereby retrieving information more precisely and efficiently. To meet the evolving needs of students, the Semantic Router dataset must be continuously updated and expanded.

This approach ensures that the system remains relevant and accurate, providing users with the best possible answers to their queries.

5 The Question-Answering System Overview

CTU-Helper employs a multi-step process to accurately and efficiently respond to user inquiries. This process encompasses several critical stages: classifying the topic of the question, processing input data, generating enhanced queries, retrieving relevant information from the knowledge database, evaluating and ranking the relevance of the information, and finally, inferring and generating a complete answer, as illustrated in Fig. 2. Each step is meticulously designed to ensure that the system comprehends the user's intent and provides the most pertinent information.

In the stages of input question processing, generating enhanced queries, and final answer generation depicted in Fig. 3, the system leverages large language models (LLMs) to perform these tasks by assigning tasks through prompts and executing them using zero-shot prompting. This approach draws on researches [6] and [7]. The prompt construction to optimally perform these tasks with LLMs is informed by studies [8, 9] and [10]. By integrating these methodologies, CTU-Helper ensures a robust and dynamic system capable of delivering accurate and relevant responses, thereby enhancing the overall user experience.

5.1 Processing User Input Data

The input data for the system consists of a user question and may include conversation history. For a single question, the system can begin retrieving relevant information immediately. However, for questions that depend on the previous conversation context, the process is more complex:

Example:

USER: Admission quota for Computer Science.

VIRTUAL_ASSISTANT: The admission quota for Computer Science is 100.

USER: How much is the tuition for this program per year?

In this conversation, the user's final question depends on the context established in previous interactions. Therefore, it is necessary to consider the context to understand what the user is asking about. The system processes the input by including the last five messages from the conversation. This limitation is due to the context window size of the large language model (LLM) used for processing. Limiting the context helps the model

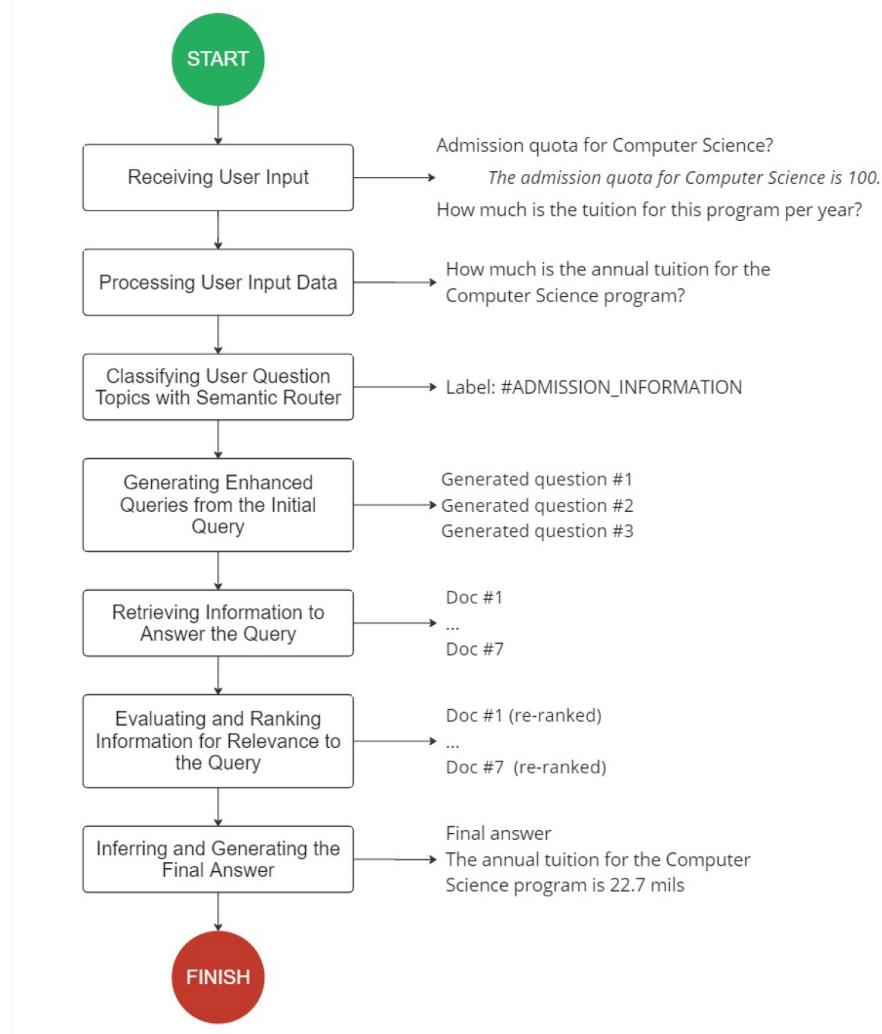


Fig. 2. General Operational Flowchart of the System

function more effectively and accurately. Examples of processing user input using an LLM are as follows:

CASE 1:

Prompt: The following is a conversation between a user and a virtual assistant. Each line represents a message, with roles indicated at the beginning of each line. Based on the previous questions and responses, clarify the user's final question. If the final question is unrelated to the previous messages, just restate the final question. Only return the clarified question without additional comments:

USER: Admission quota for Computer Science

VIRTUAL_ASSISTANT: *The admission quota for Computer Science is 100.*

USER: *How much is the tuition for this program per year?*

LLM: *How much is the annual tuition for the Computer Science program?*

CASE 2:

Prompt: *The following is a conversation between a user and a virtual assistant. Each line represents a message, with roles indicated at the beginning of each line. Based on the previous questions and responses, clarify the user's final question. If the final question is unrelated to the previous messages, just restate the final question. Only return the clarified question without additional comments:*

USER: *Admission quota for Computer Science*

VIRTUAL_ASSISTANT: *The admission quota for Computer Science is 100.*

USER: *Information about International Business*

LLM: *Information about International Business*

Clarifying the meaning of the query enhances the retrieval process, particularly when the user refers to previous messages. This step is essential because, unlike typical question-answering systems that can process the entire conversation with the LLM, RAG systems require more structured and domain-specific data retrieval. Properly handling the context not only reduces the length of the prompt for generating the final answer but also ensures that the LLM focuses on the necessary information, leading to more accurate and effective responses.

5.2 Classifying User Question Topics with Semantic Router

To classify the topics of user questions, the CTU-Helper system utilizes a Semantic Router that operates similarly to the KNN (K-Nearest Neighbors) algorithm. Initially, the user question is tokenized into individual words using the Underthesea library. The question is then converted into a vector, and the system performs a similarity search against the questions in the vector database.

The system retrieves the top k most similar questions (with k set to 7 in this case) and identifies their topic labels. The majority topic label among these k questions is assigned to the user question.

For example, if a user asks, “What is the admission score for the Computer Science department at Can Tho University in 2023?” and the system finds that 5 out of the 7 most similar questions belong to the topic “admission information,” then this user question will be classified under the “admission information” topic.

Using the Semantic Router allows the CTU-Helper system to quickly and efficiently classify the topics of user questions. This enhances the accuracy of information retrieval and provides a better user experience by ensuring that the responses are relevant to the user’s query.

5.3 Generating Enhanced Queries from the Initial Query

When retrieving content to answer a question, both the vectorization of the question and the relevant textual content are performed independently. Consequently, relying solely

on the initial query may not yield accurate information in a single retrieval attempt. Although the initial question has been processed to clarify and specify its meaning, the varying lengths of the information content can affect the retrieval process. Therefore, it is necessary to augment the input data to ensure a more comprehensive retrieval process.

To address this, we generate enhanced queries from the initial query using LLM. These enhanced queries help broaden the scope of retrieval. For the retrieved information, we will evaluate the relevance between the query and the content retrieved from both the initial and enhanced queries.

The prompt used to instruct the LLM to generate enhanced queries is as follows:

Prompt: You are a virtual assistant providing information about university details. Based on the user's question, generate three additional questions, each separated by a dash (-). Only return the questions in the specified format without adding any extra comments. User's question: What is the admission quota for the Computer Science program?

LLM:

- What are the admission methods for the Computer Science program?
- What was the admission score for the Computer Science program last year?
- Are there any additional requirements for the Computer Science program besides the entrance exam score?

After generating the enhanced queries, we proceed to retrieve information from the knowledge database using both the initial and the generated enhanced questions.

This process ensures that the retrieval is more thorough and improves the accuracy and relevance of the information returned. By augmenting the input data, the system can better capture the necessary context and nuances, leading to a more precise and comprehensive answer generation.

5.4 Retrieving Information to Answer the Query

After determining the topic of the user's question and generating additional enhanced queries, the system proceeds to retrieve relevant information from the database to provide an answer. This process also employs tokenization and vector similarity search techniques similar to those used in the topic classification step.

First, the user's question is tokenized into individual words using the Underthesea library. The question is then embedded into a vector format. The system performs a similarity search with this vector against the chunks of data containing information about academic regulations and admissions in the vector database.

The similarity search is based on the squared L2 distance measure, which is calculated using the following formula:

$$d = \sum_{i=1}^n (A_i - B_i)^2 \quad (1)$$

In formula (1):

A and B are the vector representations of the user's query and a chunk of text from the database, respectively.

A_i and B_i are the components of vectors A and B , respectively.

$\sum_{i=1}^n (A_i - B_i)^2$ is the sum of the squared differences between the corresponding components of the two vectors.

The squared L2 distance (also known as Euclidean distance) measures the similarity between the vectors. A lower distance value indicates higher similarity between the vectors, while a higher distance value indicates lower similarity.

The system retrieves the top k chunks of data that have the lowest squared L2 distance to the user's question (in this case, k = 7). These chunks are then transformed from their tokenized form back into their normal textual format to create a complete and coherent response. This transformation is essential to prepare the retrieved data for the ranking and evaluation step that follows.

Using the squared L2 distance for information retrieval allows the CTU-Helper system to search and provide the most accurate and relevant information corresponding to the user's query. This enhances the overall experience, making the information retrieval process both effective and user-friendly.

5.5 Evaluating and Ranking Information for Relevance to the Query

After retrieving chunks of text containing relevant information, the next step involves a final evaluation to check the relevance between the user's question and the retrieved information. Unlike the retrieval phase, where a Bi-encoder model with two PhoBERT models independently generates embedding vectors for comparison, this final evaluation stage uses a Cross-Encoder model.

The Cross-Encoder model used is “amberroad/bert-multilingual-passage-reranking-msmarco”. It assesses relevance by concatenating the input question with each retrieved chunk of information into a single block and evaluating the relevance between them. The relevance score generated depends on both pieces of information fed into the model.

The output relevance scores for each chunk of text are processed through a softmax function to normalize them into probabilities. These probabilities allow the system to rank the chunks in order of decreasing relevance. This process ensures that the most relevant and accurate information is placed at the top of the list, optimizing the input for the final answer generation step by the large language model.

EXAMPLE

User Question: Information about the Computer Science program

Information Before Ranking:

- (1) The Computer Science program has a quota of 100 students.
- (2) The Computer Engineering program has a quota of 150 students.
- (3) The Computer Science program has an annual tuition fee of 22.7 million VND.

Information After Ranking:

- (1) The Computer Science program has a quota of 100 students.
- (2) The Computer Science program has an annual tuition fee of 22.7 million VND.
- (3) The Computer Engineering program has a quota of 150 students.

This evaluation and ranking process, referred to as re-ranking, is crucial to prepare the best knowledge base for the large language model to filter and answer the user's

query accurately. It helps prioritize critical information and avoids the generation of misleading or irrelevant answers. This process has been inspired by the research [11].

5.6 Inferring and Generating the Final Answer

After processing and evaluating the retrieved text chunks for relevance, the system selects the top 7 chunks that are deemed most relevant to the user's query. These chunks, along with the original question, are then combined into a single prompt for the LLM to generate the final answer.

The process involves creating a structured prompt to guide the LLM, in this case, Gemini Pro and OpenAI GPT-4, to provide a concise and accurate response. The prompt is formatted as follows:

PROMPT: *You are an advisor for Can Tho University. Only use the provided information to answer the question.*

{context}

Please answer the question: {question}

In this template:

{context}: This placeholder is filled with the top 7 chunks of information retrieved and evaluated for relevance. These chunks contain the most pertinent details related to the user's query.

{question}: This placeholder is filled with the user's original question.

Here is an example of how the final prompt might look:

PROMPT: *You are an advisor for Can Tho University. Only use the provided information to answer the question.*

The Computer Science program has a quota of 100 students.

The Computer Science program has an annual tuition fee of 22.7 million VND.

The Computer Engineering program has a quota of 150 students.

[Additional relevant information from the remaining chunks]

Please answer the question: What is the annual tuition fee for the Computer Science program at Can Tho University?

This structured prompt is then sent to the Gemini Pro or OpenAI GPT-4 model, which uses its advanced language processing capabilities to infer and generate a coherent and accurate final answer. The LLM considers all the provided context and ensures that the response is directly based on the information retrieved, minimizing the chances of generating incorrect or fabricated details.

Due to the strong contextual understanding capabilities through the user's question and the retrieved texts, optimizing the retrieval process - something that we can improve more easily compared to enhancing the LLM, which is a very complex and challenging task - will help the inference process produce more accurate and detailed final answers, or in other words, the data retrieval process plays a major role in the inference of the final answer. If the retrieved information is of high quality, the configurations for the LLM will not have a significant impact.

The final answer, generated by LLM, is presented to the user, completing the multi-step process of query handling by CTU-Helper. This approach ensures that users receive precise and relevant information, enhancing their overall experience with the system.

6 Experimental Results

The test results demonstrate a significant improvement in the information retrieval efficiency of the CTU-Helper system when multiple processing and retrieval techniques are combined. The evaluation method primarily relies on the Recall metric for the entire information retrieval system. Based on retrieval performance, we evaluate the number of accurate document chunks containing relevant information that can be used to answer the question, compared to pre-prepared information chunks, at various k document milestones with k being (1, 3, 5, 7) documents. The detailed evaluation parameters are shown in Table 2 and Table 3 as follows:

Table 2. Recall Scores for Different Retrieval Methods

Method	Metric			
	Recall@k = 1	Recall@k = 3	Recall@k = 5	Recall@k = 7
Method 1	0.5834	0.6328	0.7235	0.7838
Method 2	0.8355	0.8431	0.8453	0.8578
Method 3	0.8421	0.8482	0.8494	0.8578
Method 4	0.8457	0.8513	0.8533	0.8683

Table 3. Retrieval Method Performance with Retrieval Time (ms)

Method	Metric		
	Retrieval Time MEAN	Retrieval Time MIN	Retrieval Time MAX
Method 1	31	27	59
Method 2	180	103	330
Method 3	2902	2102	3626
Method 4	2487	1832	4679

Method 1 (Bi-Encoder): This is the most basic method, using a Bi-Encoder model to embed questions and text into vectors and calculate similarity. The Recall@k = 7 is 0.7838, meaning we need to retrieve 7 text chunks to find the necessary information to answer the question. The advantage of this method is fast retrieval time (average 31 ms), but the retrieval efficiency is not high.

Method 2 (Bi-Encoder + Cross-Encoder): This method combines Bi-Encoder and Cross-Encoder models to evaluate the relevance between questions and texts. The Cross-Encoder allows the system to analyze both the question and text simultaneously, enhancing accuracy. The Recall@k = 7 for this method is 0.8578, significantly higher than using only Bi-Encoder. The related texts are ranked higher, improving efficiency at lower k values as well. However, the retrieval time increases (average 180 ms) due to the re-ranking process with Cross-Encoder.

Method 3 (Question Fusion + Bi-Encoder + Cross-Encoder): This method additionally incorporates Question Fusion, creating supplementary questions based on the initial query. These enhanced questions allow the system to retrieve information from various perspectives, increasing information coverage. The Recall across all k values does not differ significantly from the previous methods, as it only supplements and enhances coverage in certain cases. However, the retrieval time is the highest (average 2902 ms).

Method 4 (Semantic Router + Question Fusion + Bi-Encoder + Cross-Encoder): This method combines all processing and retrieval techniques. The Semantic Router classifies the question topic, Question Fusion generates enhanced questions, Bi-Encoder embeds the question and text into vectors, and Cross-Encoder evaluates relevance. The results show a notable improvement, as unrelated topic information is filtered out and retrieval time is improved. The information is fundamentally matched by topic, requiring only relevance determination for the specific query.

The test results indicate that combining multiple natural language processing techniques enhances the information retrieval efficiency of the CTU-Helper system. However, using multiple techniques also leads to longer retrieval times.

Choosing the appropriate method depends on the system's goals and requirements. If prioritizing retrieval efficiency, Method 4 (Semantic Router + Question Fusion + Bi-Encoder + Cross-Encoder) is the best choice. However, for faster retrieval times, Method 2 (Bi-Encoder + Cross-Encoder) can be used.

In practice, we can balance between efficiency and retrieval time by selecting the appropriate method for each specific case. For simple questions, Method 2 (Bi-Encoder + Cross-Encoder) can provide quick answers. For more complex questions requiring high accuracy, Method 4 (Semantic Router + Question Fusion + Bi-Encoder + Cross-Encoder) can be utilized.

7 Conclusion

The CTU-Helper system has been successfully developed to address several initial challenges. One major issue was the accessibility of information. CTU-Helper overcomes this by providing an intuitive and user-friendly chat interface, enabling students to easily access information on academic regulations and admissions. Additionally, the system tackles the problem of large and dispersed information by utilizing a centralized, regularly updated database. This ensures that the information provided is always accurate and comprehensive.

Another challenge was the complexity of administrative language, which can be difficult for students to understand. CTU-Helper addresses this by employing large language models to interpret regulations and announcements into simple, easy-to-understand language, aiding students in comprehending the information more effectively. Furthermore, the system significantly reduces the administrative workload by automatically addressing most common queries. This allows administrative staff to focus on more complex issues, thereby increasing efficiency.

Testing results indicate that the CTU-Helper system achieves high information retrieval efficiency, with the optimal retrieval method attaining a Recall@k = 7 of 0.8683. This promising outcome demonstrates the system's capability to meet user information

needs both efficiently and accurately. Besides its high retrieval efficiency, CTU-Helper offers other significant advantages such as an intuitive chat interface, voice input support, conversation history storage, and the ability to update and expand its capabilities.

The implementation of question-answering based on manually collected and processed documents is only the starting point for CTU-Helper. The future development of the system will focus on automating document collection and processing workflows. Additionally, integrating more functions supported by LLMs, such as Function Calling for automating administrative tasks and personalizing services for individual users, is a key direction we are pursuing.

These features make CTU-Helper a valuable tool for users seeking information on academic regulations and admissions at Can Tho University. The system not only provides an efficient and convenient information retrieval experience but also contributes to enhancing the quality of user support services at the university.

References

1. Lewis, P., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. arXiv preprint [arXiv:2005.11401](https://arxiv.org/abs/2005.11401) (2021)
2. Allu, U., Ahmed, B., Tripathi, V.: Beyond extraction: contextualising tabular data for efficient summarisation by language models. arXiv preprint [arXiv:2401.02333](https://arxiv.org/abs/2401.02333) (2024)
3. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. arXiv preprint [arXiv:1908.10084](https://arxiv.org/abs/1908.10084) (2019)
4. Duc, N.Q., Son, L.H., Nhan, N.D., Minh, N.D.N., Huong, L.T., Sang, D.V.: Towards comprehensive vietnamese retrieval-augmented generation and large language models. arXiv preprint [arXiv:2403.01616](https://arxiv.org/abs/2403.01616) (2024)
5. Nguyen, D.Q., Nguyen, A.T.: PhoBERT: pre-trained language models for Vietnamese. arXiv preprint [arXiv:2003.00744](https://arxiv.org/abs/2003.00744) (2020)
6. Wei, J., et al.: Finetuned language models are zero-shot learners. arXiv preprint [arXiv:2109.01652](https://arxiv.org/abs/2109.01652) (2022)
7. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
8. Siriwardhana, S., Weerasekera, R., Wen, E., Kaluarachchi, T., Rana, R., Nanayakkara, S.: Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. arXiv preprint [arXiv:2210.02627](https://arxiv.org/abs/2210.02627) (2022)
9. Bsharat, S.M., Myrzakhan, A., Shen, Z.: Principled instructions are all you need for questioning LLaMA-1/2, GPT-3.5/4. arXiv preprint [arXiv:2312.16171](https://arxiv.org/abs/2312.16171) (2024)
10. Zhou, Y., et al.: Large language models are human-level prompt engineers. arXiv preprint [arXiv:2211.01910](https://arxiv.org/abs/2211.01910) (2023)
11. Liu, N.F., et al.: Lost in the middle: how language models use long contexts. arXiv preprint [arXiv:2307.03172](https://arxiv.org/abs/2307.03172) (2023)



Automatically Generating a Dataset for Natural Language Inference Systems from a Knowledge Graph

Duc Vinh Vo¹ and Phuc Do^{2(✉)}

¹ Lac Hong University, Bien Hoa, Vietnam

² University of Information Technology (UIT),
Ho Chi Minh National University, Ho Chi Minh City, Vietnam
phucdo@uit.edu.vn

Abstract. In this study, we propose a method to automatically create Vietnamese Natural Language Inference (NLI) datasets from Knowledge Graph (KG). The approach leverages information of Knowledge Graph (KG) and employs thesaurus and antonym dictionary expansion techniques to generate premise-hypothesis sentence pairs with labels of entailment, contradiction, and neutral for natural language inference (NLI). The researchers also conducted a process of validating and improving the quality of the generated dataset. The experimental results demonstrate that this method of automatically creating Vietnamese NLI datasets from KG achieves reliable and effective performance. The generated dataset not only expands the scale of existing Vietnamese NLI datasets but also provides valuable resources for training and evaluating Vietnamese NLI models. This method holds the potential for wide-ranging applications in the development of Vietnamese NLP applications and related research.

Keywords: Premise · Hypothesis · Entailment · Contradiction · Neutral

1 Introduction

In the field of Natural Language Processing (NLP), the Natural Language Inference (NLI) model is an important technique for achieving the goal of evaluating the relationship between two sentences: a premise and a hypothesis. The NLI model aims to determine whether the hypothetical sentence can be inferred from the premise sentence, by classifying the relationship into predefined categories such as “entailment”, “contradiction”, or “neutral”.

In recent years, the development of deep learning models, such as BERT and its variants, has significantly improved the accuracy of NLI. These models utilize the Transformer architecture, which allows for learning rich word representations and understanding the contextual structure of sentences. This has opened up numerous application opportunities in areas like chatbots, virtual assistants, and text classification.

While the NLI model has been extensively researched and developed for English and some other languages, research on NLI datasets in Vietnamese is still limited. Training

the NLI model on a Vietnamese dataset is an important step in exploring and harnessing the potential of NLI in this language.

Previous studies have attempted to evaluate NLI models using Vietnamese datasets translated from English and adjusted [3]. However, these translated datasets often encounter difficulties in grammar, syntax, or odd meanings, leading to incorrect assumptions and inferences. In contrast, a Vietnamese-specific dataset with a well-defined grammatical and semantic structure would help reduce erroneous model inferences.

In this article, we discuss the importance of creating a standardized Vietnamese NLI dataset, leveraging information extracted from official Vietnamese data sources, such as Wikipedia, to ensure the quality and reliability of the dataset.

However, manually generating NLI data is a time-consuming and labor-intensive process. Therefore, finding a solution to automatically generate NLI data is an important requirement, as it can help expand the scale and scope of NLI datasets, thereby improving the performance of NLI models.

In this study, we introduce a method to automatically create Vietnamese NLI data from knowledge graphs - structured data stores. The process includes steps to identify relevant knowledge, determine the NLI relationship type, create premise-hypothesis pairs, and label and store the dataset. This not only helps increase the size and diversity of the dataset, but also ensures quality through the use of supporting rules and tools.

Specifically, the researchers will detail the knowledge graph construction process, the automatic generation of the NLI dataset, and the evaluation of the effectiveness of the method. The results of this research not only provide a way to automatically create a standardized Vietnamese dataset for the NLI model, but also contribute to the development of the NLI model in Vietnamese, with the potential for widespread application in Vietnamese language-based applications.

Our contributions are as follows:

Development and creation of a Vietnamese Natural Language Inference (NLI) dataset.

Methodology for automatically generating a Vietnamese NLI dataset from a knowledge graph.

The remaining sections of the article are structured as follows: 2) Related works 3) Theoretical foundation; 4) Proposed solution 5) Model result 6) Conclusion and future works.

2 Related Works

Developing and evaluating the accuracy of a Vietnamese Natural Language Inference (NLI) model requires a dataset that conforms to the syntax of the Vietnamese language and is of sufficient size.

In 2018, another dataset called FEVER [4] was introduced. This dataset differs from SNLI and MultiNLI in that it supports both NLI and Fact Checking tasks. The dataset provides the URL of the Wikipedia page where the premise can be extracted. Instead of being called hypotheses, the statements are referred to as claims. These statements are created by paraphrasing facts from Wikipedia and transforming them in various ways, some of which have changed meanings corresponding to the three cases of Supported,

Refuted, or NotEnoughInfo. This dataset is quite interesting and more complex than SNLI and MultiNLI because annotators also have to choose evidence in the form of sentences from Wikipedia to justify the labeling.

A year later, another version of FEVER, FEVER 2.0 [5], was introduced. This is an adversarial dataset, where the 1174 sentence samples were constructed through the FEVER2.0 Shared Task. The task challenged participants to both build systems to verify factual claims and create adversarial attacks against other participants' systems. This means that the attack patterns were designed with the intention of fooling the models. There are many types of adversarial attack data as well as techniques used to create these attacks, which are significant in exploiting model vulnerabilities and proposing solutions to overcome them.

In 2020, the Adversarial NLI corpus [6] was introduced, collecting data through a human-and-model-in-the-loop training approach. This differs from previously proposed pipelines and brings new challenges to state-of-the-art NLI models. There are a total of 3 rounds in their data collection process, each with a distinct dataset. In each round, the annotators were asked to create pairs of sentences capable of fooling the model. This approach creates a significant challenge for current language models, as the data includes longer real-world contexts.

In 2022, the research topic “Building a Vietnamese Dataset for Natural Language Inference Models” by Chinh Trong Nguyen and Dang Tuan Nguyen [2] was published. This research focuses on building a Vietnamese dataset, VnNewsNLI, to serve model training and evaluation. The study proposes a method to collect and label NLI data, creating a high-quality Vietnamese dataset for research and NLI model development. The authors address the problem of eliminating suggestive marks and ensuring the writing style of Vietnamese documents, as the presence of such hints can lead the trained models to determine the relationship between premises and hypotheses without semantic computation.

These studies provide important steps forward in developing Vietnamese datasets for NLI modeling and natural language research. The quality and diversity of the data play a crucial role in the accuracy and efficiency of the NLI model, helping to improve the understanding and application of computers in Vietnamese natural language processing. However, most datasets are translated from English, resulting in incorrect grammatical syntax, compound words, and redundancy, while manual implementation can be time-consuming, expensive, and challenging to develop and expand.

To address these limitations, the researchers applied the KG method to automatically create the dataset. This method uses information extraction techniques from available Vietnamese data sources, such as articles, books, and online documents. By analyzing the syntax, entities, and relationships in the text, the method can automatically generate Vietnamese NLI sentence-label pairs that accurately reflect the grammatical and semantic structure of the Vietnamese language. This significantly reduces the time and resources required compared to manual construction methods.

3 Theoretical Basis

Developing an automated approach to create a Vietnamese dataset for Natural Language Inference (NLI) models by leveraging techniques from the field of Natural Language Processing (NLP). The key methods involved in this process are:

3.1 Knowledge Graph

The concept of a Knowledge Graph (KG) refers to a structured data representation that captures knowledge by modeling entities and the relationships between them in a graph-like format [8]. In a KG, each entity is represented as a node, and the connections or relationships between entities are depicted as edges that link the corresponding nodes. Crucially, these edges carry labels that specify the type of relationship.

For instance, a historical KG might contain entities such as “Napoleon Bonaparte”, “France”, and “England”, with relationships like “born in”, “ruled”, and “war with” connecting these nodes. This structured and logical representation of knowledge is valuable for a variety of application, including information searching, inference, and retrieval.

Large-scale and comprehensive KG databases, such as Wikidata, DBpedia, and Google Knowledge Graph, have been constructed by aggregating data from various sources, including encyclopedias, databases, and websites. These extensive knowledge repositories serve as valuable resources for many AI-driven applications.

3.2 Basic Relationship Between Premise and Hypothesis

Entailment:

If the premise is correct, then the hypothesis must also be correct.

For example: Premise “Vietnam, in the provinces there are many tourist attractions: Hanoi has Hoan Kiem Lake tourist destination” entails hypothesis “Hanoi has Hoan Kiem Lake tourist destination”.

Contradiction:

Premise and hypothesis cannot both be true.

For example: Premise “Vietnam, in the provinces there are many tourist attractions: Hanoi has Hoan Kiem Lake tourist destination” contradiction hypothesis “Hanoi does not have Hoan Kiem Lake tourist destination”.

Neutral:

There is no clear logical relationship between premise and hypothesis.

For example: Premise “Vietnam, in the provinces there are many tourist attractions: Hanoi has the Hoan Kiem Lake tourist destination” “neutral hypothesis “Bien Hoa is a city in Dong Nai province”.

The relationships between premises and hypotheses form the foundation for defining the label classes in the Natural Language Inference (NLI) task. NLI models are trained to learn how to classify the relationship between a given premise and hypothesis based on the semantic and contextual features present in the text.

4 Proposed Solutions

In recent years, the rapid development of modern machine learning models, enabled by advancements in hardware computing capabilities, has led to the birth of a diverse range of datasets spanning numerous fields, from image processing to natural language processing. Many important problems in the field of natural language processing, such as Question Answering, Machine Translation, Summarization, Sentiment Analysis, and Natural Language Inference, have faced significant challenges in the past due to a lack of suitable data. However, this issue has been gradually resolved as the research community has access to an increasing number of high-quality datasets.

It has been observed that the majority of these datasets focus on resource-rich languages, such as English and Chinese. Consequently, there have been more breakthroughs in addressing the problem of natural language inference for these resource-rich languages. In contrast, for languages with limited research resources, like Vietnamese, there is a lack of an appropriate environment to test, research, and develop models.

To address this gap, this article proposes a solution to automatically build a standard Vietnamese dataset focused on two specific topics:

Administrative units of cities, districts, and provinces in Vietnam, with information extracted from the Wikipedia website.

Tourist attractions and specialties of provinces/cities, extracted from the VATC.vn website.

4.1 Automatically Create NLI Dataset

Select Dataset. To develop a standardized Vietnamese NLI (Natural Language Inference) training dataset, the researchers carefully selected a specific topic. This deliberate choice aimed to reduce the overhead associated with citing appendices and ensure that the language syntax and style remain consistent and standardized throughout the dataset. The first component of the dataset focuses on the administrative units at the ward and commune levels that are directly under the cities, districts, and provinces of Vietnam. The data source for this information is the wikipedia website, at link:¹. We use wikipedia because it is user-friendly, frequently updated, clearly organized, and includes citations for further exploration, and we always verify the accuracy of the data we obtain from it.

The second component of the dataset comprises attractive tourist destinations within cities. The data source for this information is the VATC website, specifically the section located at link:²

Extract Information. The researchers utilized the Python's BeautifulSoup library function to extract the necessary information from the aforementioned website sources for the purpose of constructing the NLI (Natural Language Inference) dataset. The extracted information includes the names of districts and cities, the names of Vietnam's 63

¹ https://vi.wikipedia.org/wiki/Danh_s%C3%A1ch_%C4%91%C6%A1n_v%E1%BB%8B_h%C3%A0nh_ch%C3%ADnh_c%E1%BA%A5p_huy%E1%BB%87n_c%E1%BB%A7a_Vi%E1%BB%87t_Nam.

² <http://vatc.vn/vi/tin-tuc/du-lich-viet-nam/1000-diem-du-lich-hap-dan-O-63-tinh-thanh-cua-viet-nam/>.

provinces and cities, the names of tourist attractions in the 63 provinces and cities, as well as the specialties of the 63 provinces and cities (Fig. 1).

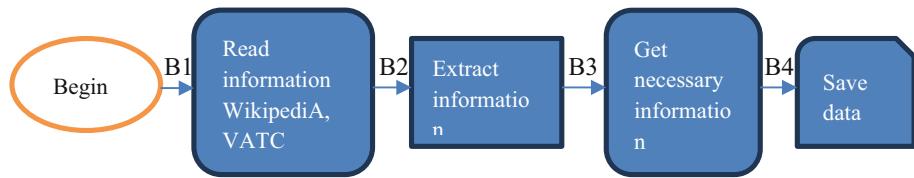


Fig. 1. Information collection process for the data set.

B1: Read the informational content of the website.

B2: After reading the website content, the next step will be to rely on the HTML code to get the properties of the information content.

B3: Analyzes HTML codes to retain essential information.

B4: Save the extracted information as file.csv.

Build Triples and Automatically Create NLI Dataset. From the extracted data, the researchers constructed triple with templates (Head, Predicate, Tail) within a KG to represent the semantic and logical relationships between entities, concepts, and events. Using these triple samples, the team will then automatically generate the NLI (Natural Language Inference) dataset, which includes the Premise, Hypothesis, and Result components. For instance:

Example 1: Triple $\langle h, p, r \rangle$: \langle Dong Nai province, has, Bien Hoa city \rangle to create NLI data sample (Premise, Hypothesis, Result), we do the following:

- Create a premise sentence:
 - Premise sentence structure: \langle string + h + p + r \rangle : Vietnam consists of 63 provinces and associated cities, where Dong Nai province has Bien Hoa city.
Of which, string = “Vietnam consists of 63 provinces and associated cities”
The operator “+” is the string concatenation.
- Create similar hypothesis sentences:
 - Method 1: \langle h + p + r \rangle : Dong Nai province has Bien Hoa city.
 - Method 2: \langle r + p' + h \rangle : Bien Hoa city belongs to Dong Nai province.
Of which, p' is synonymous with p. It means that “has” is synonymous with “belongs to”.
 - Result: Entailment.
- Next, use antonyms and negations to create contrasting Hypothesis sentences:
 - Method 1: \langle h + p' + r \rangle : Dong Nai province does not have Bien Hoa city.

- Method 2: $\langle r + p'' + h \rangle$: Bien Hoa city is not part of Dong Nai province.
Of which, p' , p'' are the opposite of p .

Example 2: Triple $\langle h, p, r \rangle$: \langle Dong Nai province, has, Bien Hoa city \rangle to create the NLI (Premise, Hypothesis, Result) dataset, we do the following:

- Create a premise sentence:
 - Premise sentence structure: \langle string + $h + p + r$ \rangle : Vietnam consists of 63 provinces and associated cities, where Dong Nai province has Bien Hoa city.
Of which, string = “ Vietnam consists of 63 provinces and associated cities“
- Create similar hypothesis sentences:
 - Method 1: $\langle h + p + r \rangle$: Dong Nai province has Bien Hoa city.
 - Method 2: $\langle r + p' + h \rangle$: Bien Hoa city belongs to Dong Nai province.
Of which, p' is synonymous with p \langle has ~ belongs to \rangle .
 - Result: Entailment.
- Create contrasting Hypothesis sentences.
There is a data set of 63 provinces/cities $H = \{\text{Dong Nai, Nghe An, Binh Duong, ...}\}$, $h = \{\text{Dong Nai}\}$.
 - Method 1:
For h' in $H - h$ where “-” is the set difference operator.
Premise: $\langle h + p + r \rangle$: Dong Nai province has Bien Hoa city.
Hypothesis: $\langle h' + p + r \rangle$: Nghe An province has Bien Hoa city.
.....
Of which, h' is the name of provinces/cities $\langle >$ h (Dong Nai).
 - Method 2:
For h' in $H - h$:
Premise: $\langle r + p' + h \rangle$: Bien Hoa city belongs to Dong Nai province.
Hypothesis: $\langle r + p' + h' \rangle$: Bien Hoa city belongs to Nghe An province.
.....
Of which, h' is the names of provinces/cities $\langle >$ h (Dong Nai); p' means p \langle has ~ belongs \rangle .
Result: Contradiction.

Example 3: Triple $\langle h, p, r \rangle$: \langle Nguyen Ai Quoc, was born in, Nghe An \rangle , to create (Premise, Hypothesis, Result):

- Create a premise sentence:
 - $\langle h + p + r \rangle$: Nguyen Ai Quoc was born in Nghe An.
- Create similar hypothesis sentences:

- $\langle h + p' + r \rangle$: Nguyen Ai Quoc lived and grew up in Nghe An.
Of which, p' is synonymous with p ($p' \sim p$: where “lived and grew up” is synonymous with “was born in”)
 - Result: Entailment.
- Create contrasting Hypothesis sentences.
There is a data set of 63 provinces/cities $H = \{\text{Nghe An, Dong Nai, Binh Duong, ...}\}$, $h = \{\text{Nghe An}\}$.
For r' in $R - r$:
Premise: $\langle h + p + r \rangle$: Nguyen Ai Quoc was born in Nghe An
Hypothesis: $\langle h + p + r' \rangle$: Nguyen Ai Quoc was born in Dong Nai.
.....
Of which, r' is the name provinces/cities that differs from r (Nghe An)
- Result: Contradiction

Result of Building the Vietnamese NLI Datset. The researchers constructed the NLI (Natural Language Inference) dataset with a CSV file structure, containing a total of 12,060 inference samples. This included 4,020 matching “entailment” sentences, 4,020 contrasting “contradiction” sentences, and 4,020 “neutral” sentences. The pairs of inference sentences were distributed evenly to avoid imbalance in the model’s inference capabilities. This is summarized in Table 1.

Table 1. Sample NLI dataset table.

No	Primese	Hypothesis	Result	Label
1	Vietnam includes 63 provinces and cities, of which: Thua Thien Hue has a town called A Luoi	Thua Thien Hue has a town called A Luoi	Entailment	1
	Vietnam includes 63 provinces and cities, of which: Thua Thien Hue has a town called A Luoi	Thua Thien Hue does not have a town called A Luoi	Contradiction	2
	Vietnam includes 63 provinces and cities, of which: Thua Thien Hue has a town called A Luoi	New York is an American city	Neutral	0
2	Vietnam includes 63 provinces and cities, of which: Kien Giang has a town called An Bien	Kien Giang has a town called An Bien	Entailment	1
	Vietnam includes 63 provinces and cities, of which: Kien Giang has a town called An Bien	Kien Giang does not have a town called An Bien	Contradiction	2
	Vietnam includes 63 provinces and cities, of which: Kien Giang has a town called An Bien	Thua Thien Hue has a town called A Luoi	Neutral	0

Difficulties and Solutions. In the process of constructing the Vietnamese NLI dataset, we encountered certain challenges and developed solutions to address them [10–12], as summarized in Table 2.

Table 2. Difficulties and solutions

No	Difficulties	Solutions
1	The lack of high-quality official documents and data sources as a key challenge, which limited their ability to create sufficiently complex and diverse sentence pairs to effectively train the NLI model	We select thematic content from materials such as textbooks, scientific documents, and reliable websites. Additionally, they created their own pairs of sentences tailored to specific contexts, ensuring a diverse and engaging dataset
2	The construction of the Vietnamese NLI dataset was challenged by the complex grammatical system and distinctive linguistic features of the Vietnamese language, requiring a comprehensive understanding of Vietnamese grammar and structure to effectively identify and evaluate the relationships between sentence pairs	We collaborated with language experts to ensure the accurate analysis of syntax, language, and spelling within the specific contextual settings of the sentence pairs
3	The difficulties in comprehending and identifying Vietnamese sentence pairs with synonymous or ambiguous relationships, owing to the presence of words and phrases with multiple meanings depending on the context, which required a profound understanding of the Vietnamese language and culture	We provided the machine learning model with a diverse set of sentences across various contexts. This allowed the model to learn a sufficient number of cases in each context, thereby improving its ability to accurately predict the intended meanings of synonymous and ambiguous terms, such as the word “machine” being understood as referring to “machinery” in the manufacturing context and to “aircraft” in the aviation context
4	The importance of carefully preprocessing and normalizing the data to ensure its quality and reliability, involving the removal of spelling errors, correction of grammatical issues, and maintaining consistency across the sentence pairs. Data preprocessing was recognized as a critical step in constructing a high-quality and trustworthy dataset	We applied spell checking and grammar checking tools to remove errors, standardize capitalization, and eliminate sentences with incorrect grammar. They also ensured consistency by using the same context for sentence pairs with synonymous meanings, addressing the data preprocessing requirements
5	The process of evaluating and labeling the relationships between sentence pairs as “entailment,” “contradiction,” or “neutral” demanded a strong linguistic knowledge and comprehension of the inherent nature of each pair of sentences	We performed the evaluation process manually. This allowed them to precisely determine the nature of the relationship between each pair of sentences

Steps in Implementing Model Training. Following the construction and preprocessing of the Vietnamese NLI dataset, the researchers proposed to undertake model selection for training and evaluation. Specifically, they introduced the use of the pre-trained BERT model for the NLI task and planned to fine-tune it on the assembled dataset.

The model training was conducted in a configured Colab environment with high-end specifications, including substantial system RAM of 51 GB, 15 GB of GPU RAM, a T4 GPU, and 201 GB of disk space.

The researchers then outlined the step-by-step approach to train the model, as depicted in Fig. 2.

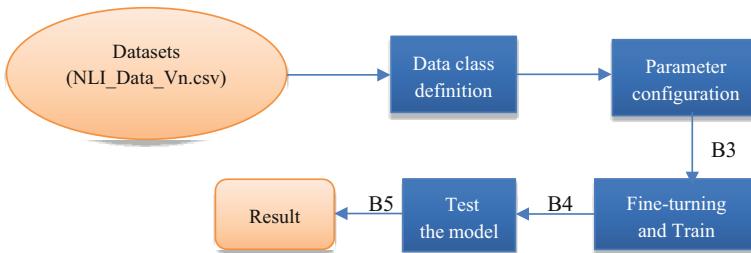


Fig. 2. Step perform model training.

The steps in Fig. 2 are as follows:

B1: Define parameter variables (premises, hypothesis, labels) for the model with the Vietnamese data set (NLI_Data_Vn.csv).

B2: Configure parameters for the model (tokenizer, model, batch_size, learning_rate, num_epochs):

```

Tokenizer = 'bert-base-uncased'
Model = 'bert-base-uncased'
batch_size = 16
learning_rate = 2e-5
num_epochs = 200
  
```

B3: Perform Fine-turning and train the model.

B4: Perform model testing with the test data set (NLI_Data_Vn_Test.csv).

B5: Results and model evaluation

5 Model Result

Following the pretraining of the model on the Vietnamese dataset (NLI_Data_Vn.csv), which required a considerable time of 14 h and 37 min and achieved an AUC of 99.9%, the results indicate that the model has effectively learned and comprehended the complexity of French sentence research. The performance on the Vietnamese language dataset in terms of sentence pairs is highly promising.

To further evaluate the model's accuracy, we will conduct tests on 2 standardized datasets, considering the following cases:

To assess the model's performance on both "Entailment" and "Contradiction" tasks, we will utilize a test dataset (NLI_Data_Vn_Test1.csv) consisting of 400 inference samples, with 200 samples dedicated to evaluating "Entailment" and the remaining 200 for "Contradiction". This test set is sampled from districts and provinces not present in the training data, allowing us to evaluate the model's generalization capabilities. For instance:

- Example 1:
 - Premise: Vietnam includes 63 provinces, of which: A1 is in BB1 province.
 - Hypothesis: A1 is in BB1 province.
 - Result: Entailment.

- Example 2:
 - Premise: Vietnam includes 63 provinces, of which: A1 is in BB1 province.
 - Hypothesis: A1 is not in BB1 province.
 - Result: Contradiction (Table 3).

Table 3. Test dataset result ratio

Test data set	Total number of inference samples	True	False	Ratio
NLI_Data_Vn_Test1.csv	400	400	0	100%

Note: In this analysis, we will solely annotate the pre-trained BERT model indices, as follows:

- The Precision metric, which measures the ratio between the number of sentence pairs the model correctly predicts and the total number of sentence pairs the model predicts for a given class (entailment, contradiction, neutral), is 0.999. For instance, if the model predicts 1,000 sentence pairs as belonging to the "entailment" class, and 999 of those predictions are accurate, then the Precision for the "entailment" class would be 0.999 (999/1,000).
- The Recall metric, which represents the ratio between the number of sentence pairs the model correctly predicts and the total number of sentence pairs that actually belong to a given class (entailment, contradiction, neutral) in the dataset, is 0.999. For instance, if the dataset contains 1,000 sentence pairs that are truly part of the "entailment" class, and the model correctly identifies 999 of them, then the Recall for the "entailment" class would be 0.999 (999/1,000).
- The AUC (Area Under the Curve) metric, which evaluates the overall quality of the classification model, is 0.999. AUC is calculated based on the area under the ROC (Receiver Operating Characteristic) curve, where a higher AUC indicates a more accurate prediction model. An AUC of 0.999 suggests that the model correctly predicts 99.9% of all inference pairs within the test dataset.

The results of the model's performance metrics are presented in Fig. 3.

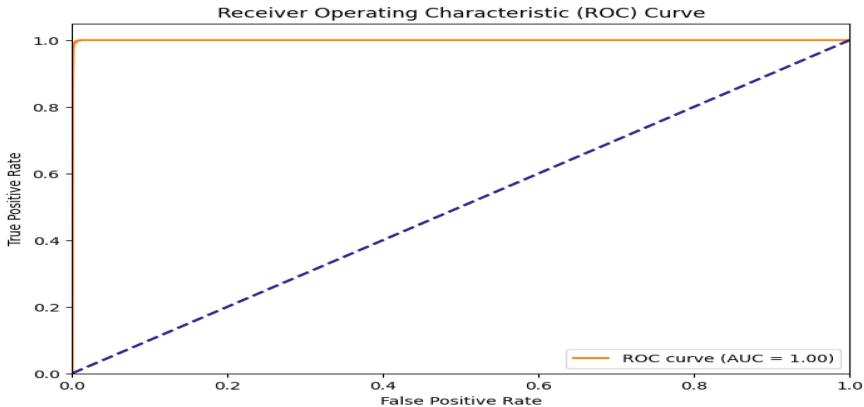


Fig. 3. Graph representing the correct and incorrect rate of the model (ROC).

The model's exceptionally high AUC-ROC index indicates its remarkable accuracy in predicting the semantic, contextual, and syntactic relationships between Vietnamese inference sentence pairs. This result also reflects the high standardization of the Vietnamese NLI dataset, which enables promising research directions and applications of the NLI model in the Vietnamese language.

6 Conclusions and Future Works

In this paper, we presented a method for automatically creating an NLI (Natural Language Inference) dataset using the information in a given KG. Specifically, we have built a Vietnamese NLI dataset covering topics on administrative geography (cities, districts, provinces), tourist attractions, and specialty dishes of 63 provinces/cities in Vietnam. After building this dataset, we utilized the pre-trained BERT model to perform NLI model training. The results demonstrate that the model achieves remarkably high inference accuracy, up to 98%. Automatically generating NLI data from knowledge graphs not only enables the effective utilization of rich and accurate information sources, but also improves the performance and scalability of natural language inference models. This makes an important contribution to the development of NLP applications, enhancing the quality and reliability of natural language processing systems in practice.

To further improve the performance and applicability of the NLI model, we propose the following research directions:

- Construct a larger and more diverse NLI dataset from various sources, encompassing a wider range of linguistic features and supporting more languages. This would enable the model to learn richer language characteristics.
- Investigate and develop novel model architectures to improve performance on NLI tasks. This may involve exploring combinations of BERT with other models or adapting the network architecture to better suit specific NLI requirements.

This study makes a significant contribution in applying the KG to automatically generate a Vietnamese NLI dataset, and proposes future research directions to enhance the performance and applicability of the NLI model.

Acknowledgments. We would like to thank VNPT- Dong Nai, Viet Nam. A part of this research was funded by the Vietnam National University Ho Chi Minh City under grant number DS2023-26-01.

References

1. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In North American Association for Computational Linguistics (NAACL) (2019)
2. Nguyen, C.T., Nguyen, D.T.: Building a Vietnamese dataset for natural language inference models. *SN Comput. Sci.* **3**, 395 (2022). Accepted 22 June 2022, Published 25 July 2022
3. Nguyen, M.-T., Ha, Q.-T., Nguyen, T.-D., Nguyen, T.-T., Nguyen, L.-M.: Recognizing textual entailment in Vietnamese text: an experimental study. In: International Conference on Knowledge and Systems Engineering, pp. 108–113. IEEE, Ho Chi Minh City (2015)
4. Thorne, J., Vlachos, A., Christodoulopoulos, C., Mittal, A.: FEVER: a large-scale dataset for fact extraction and verification. arXiv preprint [arXiv:1803.05355](https://arxiv.org/abs/1803.05355) (2018)
5. Thorne, J., Vlachos, A., Cocarascu, O., Christodoulopoulos, C., Mittal, A.: The FEVER2.0 shared task. In: Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER), pp. 1–6 (2019)
6. Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., Kiela, D.: Adversarial NLI: a new benchmark for natural language understanding. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4885–4901 (2020)
7. Peters, M.E., et al.: Deep contextualized word representations. [arXiv:1802.05365](https://arxiv.org/abs/1802.05365) [cs] (2018)
8. Ji, S., Pan, S., Cambria, E., Martine, P., Yu, P.S.: A survey on knowledge graphs: representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(2), 494–514 (2022)
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. [arXiv:1301.3781v3](https://arxiv.org/abs/1301.3781v3) [cs.CL] (2013)
10. Lai, A., Hockenmaier, J.: Learning to predict denotational probabilities for modeling entailment. Nome convegno EMNLP (2017)
11. Bowman, S.R., Potts, C., Angeli, G., Manning, C.D.: A large annotated corpus for learning natural language inference. [arXiv:1508.05326v1](https://arxiv.org/abs/1508.05326v1) [cs.CL] (2015)
12. Huang, E.H., Socher, R., Manning, C.D., Ng, A.Y.: Improving word representations via global context and multiple word prototypes. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2012)
13. Bussotti, J.F., Veltri, E., Santoro, D., Papotti, P.: Generation of training examples for tabular natural language inference. In: Proceedings of the ACM on Management of Data, vol. 1, no. 4 (2023)



ViFoodNLI: A Dataset for Vietnamese Natural Language Inference in Local Cuisine

Long Ngo Hoang Phan and Phuc Do^(✉)

University of Information Technology, Vietnam National University,
Ho Chi Minh City, Vietnam
phucdo@uit.edu.vn

Abstract This paper introduces the ViFoodNLI dataset, a natural language inference (NLI) dataset for Vietnamese. While recent efforts have been made to build high-quality NLI datasets for Vietnamese and some Cross-Lingual NLI Corpus (with support for Vietnamese) for multiple domains, our dataset specifically focuses on the field of local cuisine. The main reason for choosing this field is that cuisine is a significant component of Vietnamese culture, and thus the dataset encompasses many characteristics of the Vietnamese language. By collecting information on culinary topics from reliable news sources, we have developed various methods and logics such as knowledge graphs and Generative AI to create high-quality pairs of premise and hypothesis sentences. Through rigorous testing, the dataset has achieved significant results, creating momentum for future research and practical applications.

Keywords: Natural Language Inference (NLI) · Vietnamese · Local Cuisine

1 Introduction

Although Vietnam is a country with a modest geographical area, it boasts a rich cultural history spanning thousands of years, and Vietnamese is a language with a long-standing formation and development process. However, the research and resources dedicated to natural language processing (NLP) for Vietnamese are still quite limited due to the lack of high-quality large datasets.

One of the key tasks of natural language processing (NLP) is Natural Language Inference (NLI), which aims to determine the semantic relationship between two sentences. Specifically, NLI identifies whether a sentence (referred to as the hypothesis) can be implied or inferred from another sentence (referred to as the premise). There are three main labels assigned to these sentence pairs: Entailment (if the hypothesis h can be inferred from the premise p), Contradiction (if the hypothesis h contradicts the premise p), and Neutral (for all other cases).

There have been studies that provide large and high-quality datasets, such as the Stanford NLI (SNLI, Bowman et al., 2015) and the Multi-Genre NLI (MNLI, Williams et al., 2018) datasets, but these datasets are in English. There have also been some efforts to provide multilingual datasets by translating English NLI datasets into other languages (XNLI, Conneau et al., 2018), but automatic translation from one language to another still poses many issues. For Vietnamese, there have been high-quality NLI studies by Tin Huynh et al. (2022) with ViNLI and Chinh Nguyen et al. (2022). However, these are open-domain NLI datasets, which need to be expanded to serve specific industries more effectively (Bauer et al., 2021).

To address these issues, we have developed a new specialized dataset for NLI. Inspired by Vietnam’s diverse and unique cuisine, we introduce the ViFoodNLI dataset—a natural language inference dataset for Vietnamese in the domain of local cuisine. As we know, cuisine is an important aspect that reflects the characteristics of culture, so we expect the dataset to carry many characteristics of the Vietnamese language.

To construct the dataset, we gathered culinary data from reputable news websites and annotated sentence pairs ($p \& h$) with entailment, contradiction, and neutral tags based on predefined rules for determining the relationship between premise-hypothesis pairs. Additionally, we leveraged knowledge graphs to enhance the dataset’s quality, as studied by Wang et al., 2020. We assessed the dataset’s quality by benchmarking its depth against established datasets like SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018), and the trending culinary keywords provided by Google Trend and Google Ads. Furthermore, we evaluated the dataset using transformer models such as BERT (Devlin et al., 2019), XLM-R (Conneau et al. 2020), and PhoBERT (Nguyen and Nguyen, 2020), which have achieved impressive performance on various NLP tasks.

The contributions of this research are as follows: (1) We introduce the ViFoodNLI dataset, a dataset in the domain of local cuisine with over 150K labeled sentence pairs. (2) We propose a set of rules to determine the labels of premise-hypothesis pairs in a cost-effective manner. (3) We conduct experiments on NLI models, including neural network-based models and pre-trained transformer-based models.

The paper is organized as follows: Sect. 1: Introduces a general problem that needs to be resolved, the main contribution of our research. Section 2: Reviews related works on creating NLI datasets. Section 3: “Dataset Creation” presents the process of building the ViFoodNLI dataset and some experiments. Section 4: Presents some experiments on our dataset with popular pre-trained models. Finally, Sect. 5 concludes the paper.

2 Related Work

The initial NLI datasets were created for the task of recognizing textual entailment (RTE). These datasets were human-annotated, ensuring quality but limiting quantity. In 2014, Marelli et al. introduced the SICK dataset with 10k English

sentence pairs generated through a three-step process: normalization, expansion, and pairing, applying syntactic and lexical transformation rules. Subsequently, larger English NLI datasets were introduced, such as SNLI (Bowman et al., 2015) and the Multi-Genre NLI (MNLI, Williams et al., 2018) dataset.

In fact, there is a scarcity of NLI datasets in languages other than English. To address this, Conneau et al., 2018 introduced the XNLI dataset, which was automatically translated from MNLI into 15 different languages, including Vietnamese. Other NLI translation projects include SICK-NL (Wijnholds et al., 2021), which translated the SICK dataset into Dutch, and AmericasNLI (Ebrahimi et al., 2021), which expanded XNLI to 10 indigenous languages of the Americas.

While automatically translated data has proven convenient for resource-scarce languages, it often leads to inaccurate or incomprehensible sentences, the translated language may lack the natural characteristics of the target language, copying the context and culture of the source language, which may not be appropriate for the target language (see Table 1).

Table 1. Examples from the XNLI dataset illustrating language issues from automatic translations.

Premise	Hypothesis	Note
Eng: The dish is served with a side of garlic bread. Vie: Món ăn được phục vụ với một bên của bánh mì tỏi.	Eng: The dish comes with garlic bread. Vie: Món ăn có bánh mì tỏi đi kèm.	The premise sentence is grammatically incorrect.
Eng: The chef prepared a delicious seafood curry. Vie: Đầu bếp chuẩn bị một cà ri hải sản ngon.	Eng: The chef made a flavorful seafood dish. Vie: Đầu bếp đã làm một món hải sản đầy hương vị.	The premise sentence is unnatural.

In recent years, there have been Vietnamese NLI datasets, such as the bilingual dataset by Quyen et al., 2022 with 16K labeled sentence pairs in the medical domain, and ViNLI by Huynh et al., 2022 with 22K labeled sentence pairs in various fields. Large and high-quality Vietnamese datasets constructed with processes similar to SNLI and XNLI will significantly contribute to the field of natural language processing in Vietnamese.

3 Dataset Creation

We constructed the ViFoodNLI dataset based on the methods used to create the SICK (Marelli et al., 2014), SNLI (Bowman et al., 2015), and MNLI (Williams et al., 2018) datasets, which require the following three steps:

- The first step is to select a sentence to serve as the premise.
- The next step is to create contradiction, entailment, and neutral sentence pairs from the chosen premise.

- Finally, we validate the labels for these sentence pairs and create the NLI dataset.

In this study, we take two approaches to create NLI datasets A & B, while still adhering to the aforementioned three steps to experiment with the A/B Testing method.

Approach A (ViFoodNLI_A): We collected data on specialty dishes from 63 provinces/cities of Vietnam. Specifically, for each province/city, we collected at least 20 names of specialty dishes. The data sources included culinary, travel, and cooking guide websites.

Approach B (ViFoodNLI_B): Similarly, we collected data from culinary, travel, and cooking guide websites. We extracted information such as the food name, region, province where the dish is found, ingredients, texture characteristics, flavor, customer comments, cooking method, and typical meal type for the dish. These details were then formed into entities in a knowledge graph and linked with appropriate relationships (Fig. 1).

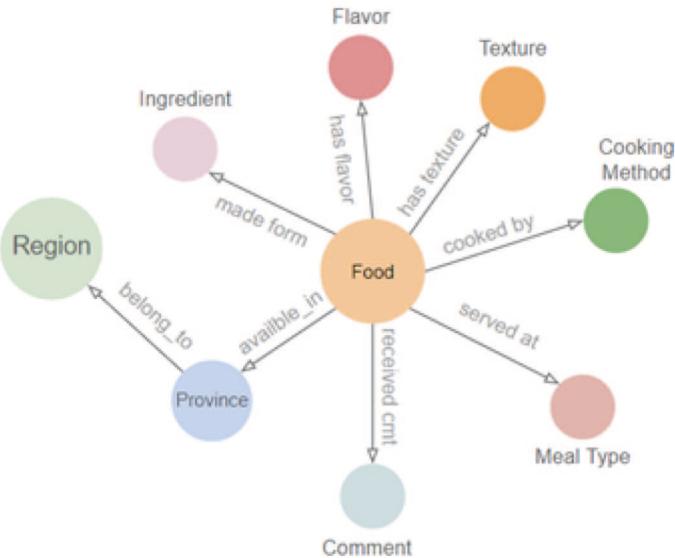


Fig. 1. Relationships between entities in the Knowledge Graph.

3.1 Selecting the Premises

For the ViFoodNLI_A dataset: Recently, large language models (LLMs) have played significant roles in natural language processing (NLP) (Kojima et al., 2023). Leveraging this, we utilized GPT-3.5, GPT-4, and the most advanced model, GPT-4o, to generate content for premise sentences. For each dish collected, we created corresponding prompts to gather additional information about its description, preparation methods, origin, and culinary experiences.

For the ViFoodNLI_B dataset: We used the knowledge graph approach to create triples consisting of an entity, a relation, and an object, such as:

- Food (available_in) Province. Ex: Bún bò, available in, Hué.
- Food (made_from) Ingredients. Ex: Bún bò, made from, Bún, Sả, Bò.
- Food (has_flavor) Flavor. Ex: Bún bò, has flavor, Thơm.
- Food (has_texture) Texture. Ex: Bún bò, has texture, Thơm.
- Food (cooked_by) Cooking Method. Ex: Bún bò, cooked by, Hầm xương.
- Food (serve_at) Meal Type. Ex: Bún bò, served at, Buổi sáng.
- Food (received_cmt) Comment. Ex: Bún bò, received comment, nước lèo đậm đà.

Additionally, we combined attributes to create sentences: Food + Description. Ex: Bún bò là một món ăn truyền thống của Việt Nam, được biết đến với sự kết hợp hương vị đậm đà của bún (bánh phở nhỏ) và thịt bò.

3.2 Hypothesis Generation

To construct the dataset according to the standards of SNLI, MNLI, or XNLI, we generated three hypotheses for each premise, corresponding to the three labels: entailment, contradiction, and neutral.

For the ViFoodNLI_A dataset: We leveraged the advantages of large language models (LLMs), specifically the GPT-4o model (as compared with other LLMs by Iorliam et al. 2024). For each statement about a dish chosen as a premise, we created three prompts to generate three corresponding hypotheses with the labels entailment, contradiction, and neutral (see the process in Fig. 2).

Approach A: ViFoodNLI_A

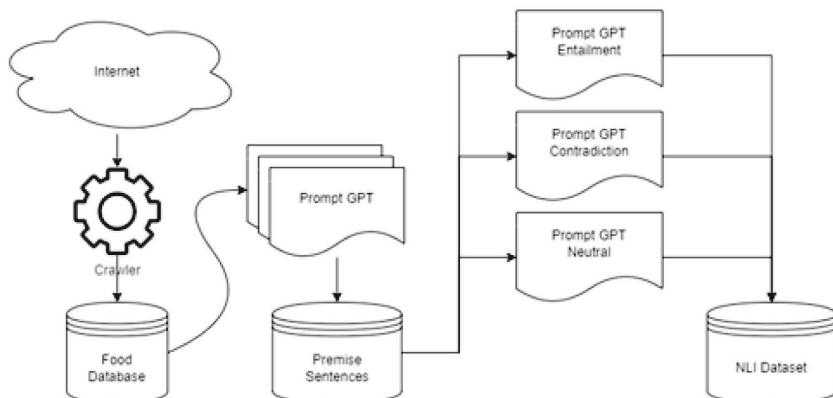


Fig. 2. Process of Creating the NLI Dataset using Approach A.

For the ViFoodNLI_B dataset: Given the data orientation towards a knowledge graph, we easily created simple rules to generate hypotheses. For example, to create a contradiction sentence from the premise, we used negative words for relationships such as “available in” → “not available in”, “made from” → “not made from”, etc. Alternatively, we used antonyms like “has” → “lacks”, “appears” → “absent”, etc. Similarly, for entailment sentences, we applied rules like converting sentences from active to passive voice or vice versa, and using synonyms. For neutral sentences, we paired unrelated subjects and predicates. The set of rules for the three types of hypotheses and their distribution in the dataset are detailed in Table 2, and the dataset creation process is shown in Fig. 3.

Approach B: ViFoodNLIB

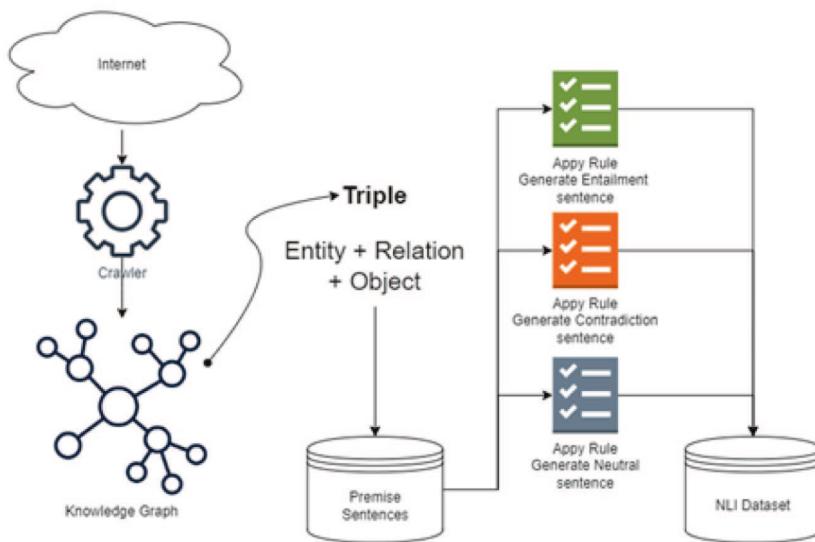


Fig. 3. Process of Creating the NLI Dataset using Approach B.

Table 2. Rules for Generating Hypotheses and the Distribution of Each Type in the Dataset.

Label	Rule	Ratio
Entailment	Change active sentences into passive sentences and vice versa	10.89%
	Replace words with synonyms	10.74%
	Add or remove modifiers that do not radically alter the meaning of the sentence	11.65%
	Replace Named Entities with a word that stands for the class	10.75%
	Turn nouns into relative clauses	11.12%
	Turn the object into relative clauses	11.44%
	Turn adjectives into relative clauses	11.07%
	Create a presupposition sentence	11.08%
	Create conditional sentences	11.26%
Contradiction	Use negative words	20.72%
	Replace words with antonyms	19.68%
	Opposite meaning of a presupposition	19.44%
	Wrong reasoning about an object	20.18%
	Wrong reasoning about an event	19.98%

3.3 Data Validation

This step assesses the quality of the dataset by re-evaluating the labels of the premise-hypothesis pairs created in the previous step. Traditionally, a team of human annotators would be hired to evaluate the labels based on given guidelines. However, Gilardi et al. (2023) assessed that Chat GPT can take on the role of label classification in natural language processing (NLP) tasks. Therefore, we implemented a label evaluation process using large language models (LLMs) with three models: GPT-4, GPT-4o, and Gemini-1.5-pro. Each model casts one vote, and the final label of the premise-hypothesis pair, referred to as the gold label, is determined by the majority vote (Table 3).

Table 3. Agreement result of the validation phase in ViNLI compared with other corpora. *The numbers of SNLI, ViNLI corpora are extracted from the scientific papers.

Statistic	SNLI*	ViNLI*	ViFoodNLI _A	ViFoodNLI _B
Language	Eng	Vie	Vie	Vie
Text genre	Img captions	News	Cuisine	Cuisine
Total pairs	570.152	30.376	123.485	27.498
Validated pairs	56.951	6.000	3.705	2.750
Pairs w/ unanimous	58.3%	77.9%	70.09%	63.64%
Individual label = gold label	89%	94.1%	89.55%	87.37%
Individual label = author's label	85.8%	91.1%	86.97%	83.96%
Gold label = author's label	91.2%	96.4%	93.95%	93.27%
Gold label != author's label	6.8%	3.6%	6.05%	6.73%
No gold label (no 3 labels match)	2%	0.6%	0.73%	0.76%

3.4 Data Analysis

In this data analysis section, we conducted an evaluation of the dataset's depth by examining the topics covered within it. Using Google Trend and Google Ads tools, we obtained a set of over 1K keywords with the highest search volumes (averaging over 10K searches/month) related to dishes and cuisine over the past 12 months. The purpose was to validate the practicality of the dataset against keywords commonly searched by users. We divided this set of keywords into three main categories: trending topics, verbs related to cooking methods, and nouns related to dish ingredients. We then compared their occurrences in the premise-hypothesis pairs of the two datasets, ViFoodNLI_A and ViFoodNLI_B (Table 4).

Table 4. Proportion of Premise-Hypothesis Pairs Containing Keywords by Topic.

ViFoodNLI _A	ViFoodNLI _B	Topic	Keywords
54.42%	28.73%	Hot Trend	món ngon mỗi ngày, bánh xèo, lẩu gà lá é, bánh cuốn, mì quảng, phở bò, thịt kho tàu, vịt om sấu, cách nấu bò kho, phở cuốn, ...
99.28%	96.86%	Ingredients	gà, thịt, bò, cá, tôm, trứng, mực, chuối, bún, tỏi, vịt, sườn, thịt bò, cơm, cà chua
48.19%	39.67%	Verb	luộc, xào, rim, rán, hấp, nướng, chiên, om, kho, hầm, đút, xắt, nẤu, quay, cuốn, gỏi, thái, trộn, rang, nhúng, tiêm, tần

3.5 Data Result

Using two approaches-large language models (LLM) and knowledge graphs-we created two NLI datasets: ViFoodNLI_A and ViFoodNLI_B. Comparing these datasets with SNLI and ViNLI, we observe that the ViFoodNLI datasets are positioned in the middle range (slightly better than SNLI and lower than ViNLI), indicating that the quality of these two datasets partially meets expectations. When compared with the trending keywords related to cuisine, both ViFoodNLI datasets show almost absolute coverage of popular ingredient keywords. The ViFoodNLI_A dataset, generated from LLM sources, exhibits broader coverage of hot trend keywords and cooking methods. Overall, the data from our two different approaches are fairly similar in quality (Fig. 4).

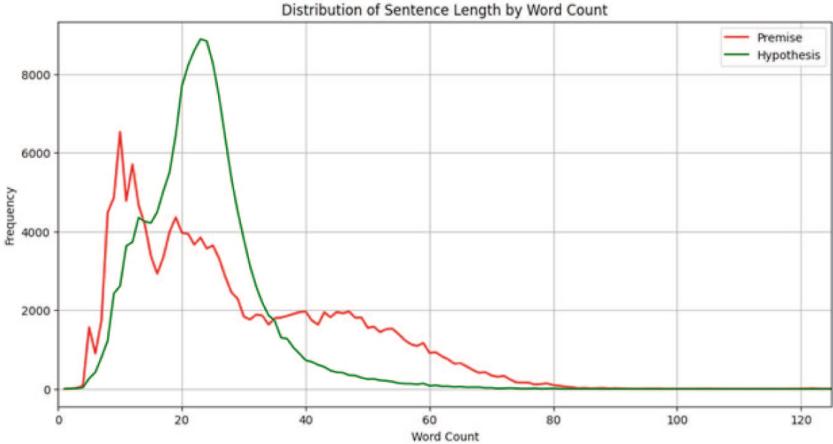


Fig. 4. Length distribution of sentences in the dataset.

4 Experiment

We conducted experiments to assess the difficulty of the dataset using pre-trained models. We used multilingual models such as mBERT (Devlin et al., 2019), which was pre-trained on 104 major languages with Wikipedia (including Vietnamese) using masked language modeling (MLM); XLM-R (Conneau et al., 2020), the XLM-RoBERTa model, pre-trained on 2.5 TB of filtered data covering 100 languages (including Vietnamese); and PhoBERT (Nguyen and Nguyen, 2020a), an advanced pre-trained model for Vietnamese (trained on 20 GB of Wikipedia data).

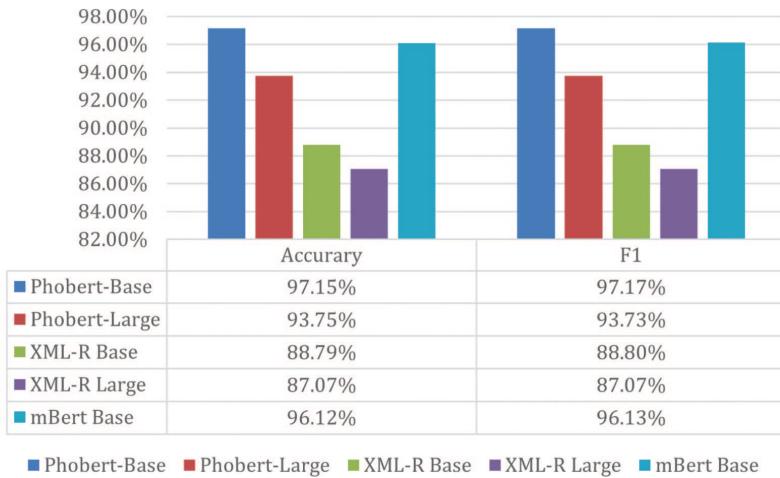
We combined the ViFoodNLIA and ViFoodNLIB datasets into a unified dataset for fine-tuning models. All models were trained using the Adam optimizer (Kingma and Ba, 2015) on the NVIDIA Tesla T4 GPUs provided by Google Colab. To train transformer models such as mBERT, XLM-R, and PhoBERT, we utilized the Transformers library from Hugging Face. Additionally, we set the hyperparameters as follows: learning_rate = $1e-5$, epochs = 3, batch_size = 32.

Based on the ViFoodNLI dataset, we divided it into Train, Validation, and Test sets with an 8:1:1 ratio. According to the results in Table 5, we observed that the PhoBERT model outperformed the other multilingual models in terms of accuracy. However, when we tested these models trained on the ViFoodNLI dataset to evaluate the XNLI Vietnamese dataset, the accuracy was not high (highest at 50.36% for the mBERT model). This discrepancy can be attributed to two key factors - Translation Quality: The XNLI dataset is composed of Vietnamese text that has been automatically translated from the English MNLI dataset, which can result in unnatural context, phrasing, and grammar. This issue is highlighted in Sect. 2, Related Work, where we discuss the limitations of automatic translation in preserving linguistic nuances. Domain Differences: The XNLI dataset is a multi-domain dataset, meaning it covers a broad range of top-

Table 5. Accuracy Rates for Pre-trained Models on the ViFoodNLI Dataset.

	Train _{Accuracy}	Valid _{Accuracy}	Test _{Accuracy}	XNLI _{Accuracy}
	120.786 pairs	15.098 pairs	15.098 pairs	39.271 pairs
PhoBERT-Base	97.15%	98.32%	98.35%	51.97%
PhoBERT-Large	93.75%	96.86%	97.41%	53.58%
XML-R Base	88.79%	87.88%	89.78%	35.98%
XML-R Large	87.07%	89.15%	90.18%	46.27%
mBERT Base	96.12%	97.25%	97.29%	49.07%

ics, whereas the ViFoodNLI dataset is focused on a specific domain-Vietnamese cuisine (Fig. 5).

**Fig. 5.** Evaluation results of accuracy and F1 score when training pre-trained models on the ViFoodNLI dataset.

In Fig. 6 and Fig. 7, we analyzed the confusion matrices of the two models, PhoBERT-base and PhoBERT-large, which showed the highest accuracy in Table 5. It indicated that the models correctly predicted with high accuracy for the “contradiction” and “entailment” labels. However, there were many confusions in the “neutral” label, possibly due to various reasons such as the automatically translated inference pairs from XNLI not fitting the Vietnamese context or differences in domain between the training and test datasets.

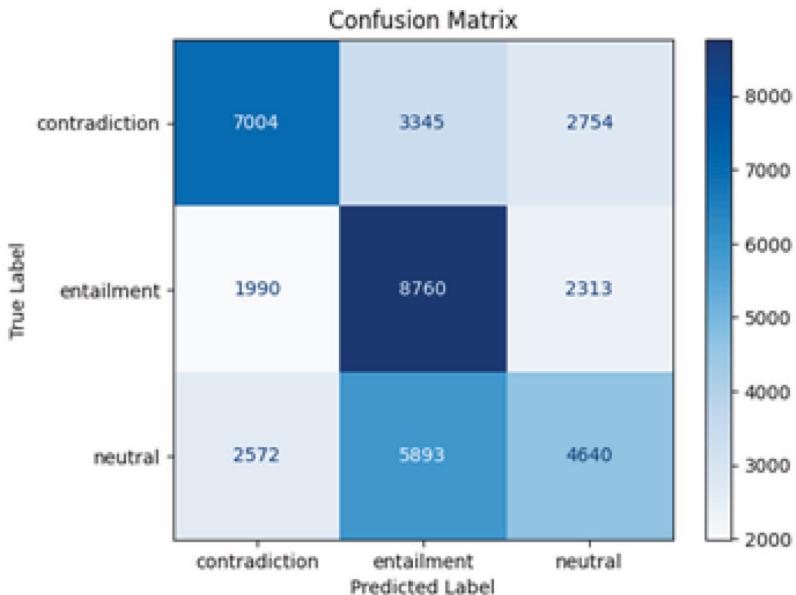


Fig. 6. Confusion matrix when cross-evaluating fine-tuned PhoBert-large models on the ViFoodNLI dataset against the XNLI Test set.

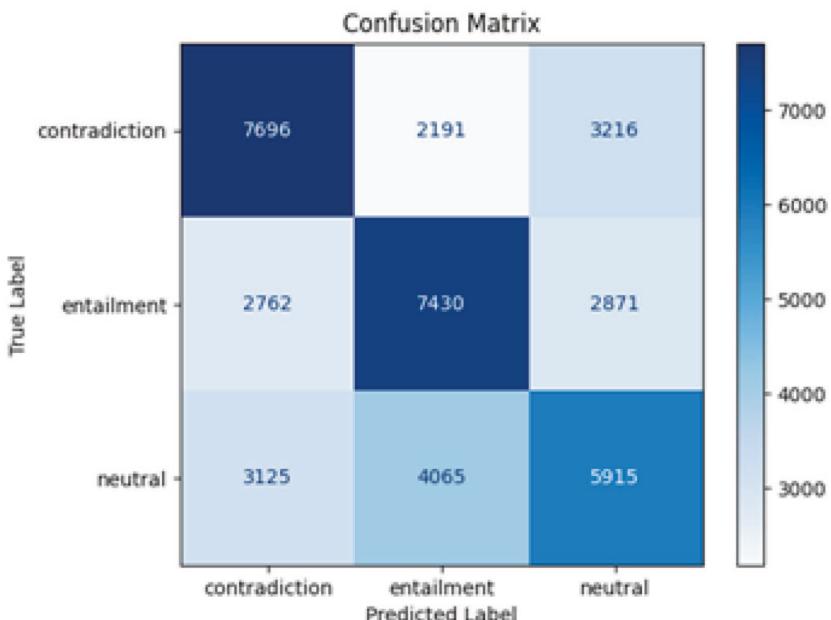


Fig. 7. Confusion matrix when cross-evaluating fine-tuned PhoBert-base models on the ViFoodNLI dataset against the XNLI Test set.

5 Conclusion

In this paper, we introduced the natural language inference dataset in the local cuisine domain (ViFoodNLI), consisting of over 150K premise-hypothesis pairs collected from various sources. We conducted quality evaluation of the dataset and its labels using methods similar to those used for SICK (Marelli et al., 2014), SNLI (Bowman et al. 2015), and MNLI (Williams et al., 2018) datasets. Additionally, we experimented with advanced models. We are confident that the dataset, along with our collection and evaluation approach, will provide valuable resources for Vietnamese NLP research, particularly in the field of fact-checking (Hieu et al., 2020a).

Acknowledgment. This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCMC) under grant number DS2023-26-01.

References

- Van Huynh, T., Van Nguyen, K., Nguyen, N.L.-T.: ViNLI: a Vietnamese corpus for studies on open-domain natural language inference. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 3858–3872 (2022)
- Nguyen, C.T., Nguyen, D.T.: Building a Vietnamese dataset for natural language inference models. *SN Comput. Sci.* **3**, 395 (2022)
- Quyen, N.T., Anh, H.T., Huyen, N.T.M., Lien, N.: VLSP 2021 - vnNLI challenge: Vietnamese and English-Vietnamese textual entailment. *VNU J. Sci.: Comput. Sci. Commun. Eng.* **38**(1) (2022)
- Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 632–642 (2015)
- Williams, A., Nangia, N., Bowman, S.: A broad-coverage challenge corpus for sentence understanding through inference. In: 2018 Proceedings of NAACL-HLT, pp. 1112–1122 (2018)
- Conneau, A., et al.: XNLI: evaluating cross lingual sentence representations. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2475–2485 (2018)
- Bauer, L., Deng, L., Bansal, M.: ERNIE-NLI: analyzing the impact of domainspecific external knowledge on enhanced representations for NLI. In: Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pp. 58–69 (2021)
- Wang, Z., Li, L., Zeng, D.: Knowledge-enhanced natural language inference based on knowledge graphs. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 6498–6508 (2020)
- 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)
- Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451 (2020)

- Nguyen, D.Q., Nguyen, A.T.: PhoBERT: pre-trained language models for Vietnamese. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 1037–1042 (2020)
- Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., Zamparelli, R.: A SICK cure for the evaluation of compositional distributional semantic models. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), pp. 216–223 (2014)
- Wijnholds, G., Moortgat, M.: SICK-NL: a dataset for Dutch natural language inference. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 1474–1479. Association for Computational Linguistics (2021)
- Ebrahimi, A., et al.: Americasnli: evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. arXiv preprint [arXiv:2104.08726](https://arxiv.org/abs/2104.08726) (2021)
- Gilardi, F., Alizadeh, M., Kubli, M.: ChatGPT outperforms crowd-workers for text-annotation tasks. In: Proceedings of the National Academy of Sciences (PNAS) (2023)
- Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. In the 36th Conference on Neural Information Processing Systems (NeurIPS 2022) (2023)
- Iorliam, A., Ingio J.: A comparative analysis of generative artificial intelligence tools for natural language processing. *J. Comput. Theories Appl.* **2**(1) (2024)
- Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
- 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, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015)
- Hieu, T.N., Minh, H.N., Van, H.T., Quoc, B.V.: ReINTEL challenge 2020: Vietnamese fake news detection using ensemble model with PhoBERT embeddings. In: Proceedings of the 7th International Workshop on Vietnamese Language and Speech Processing, Hanoi, Vietnam, pp. 1–5. Association for Computational Linguistics (2020)



VNLegalEase: A Vietnamese Legal Query Chatbot

Pham Thi Xuan Hien, Nguyen Thanh Tuong Vy, and Huu-Dung Ngo^(✉)

Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

{phamthixuanhien, ngohuudung}@iuh.edu.vn

Abstract. The complexity of Vietnamese legal documents poses significant challenges in accessing and comprehending legal information. This paper introduces VNLegalEase, an advanced chatbot designed to address these challenges in the Vietnamese legal context. Leveraging state-of-the-art natural language processing (NLP) techniques, VNLegalEase integrates GPT-4o, LlamaIndex, vector databases, and Retrieval-Augmented Generation (RAG) to deliver accurate and context-specific responses to legal queries. Key contributions include: (1) development of a Vietnamese legal-specific chatbot; (2) novel integration of advanced technologies for enhanced legal information retrieval and response generation; (3) creation of a comprehensive Vietnamese legal dataset comprising 36,602 legal documents and 300,000 Q&A pairs; and (4) rigorous evaluation demonstrating a 12% increase in accuracy and 52% reduction in response time compared to baseline models. Our experimental results show VNLegalEase achieves 87% accuracy on complex legal queries and 90% coverage of Vietnamese legal knowledge across 11 categories. User studies indicate high satisfaction, with 60% of users rating response accuracy at 4 or 5 on a 5-point Likert scale. This research advances legal AI systems for the Vietnamese context and provides a framework for developing similar systems in other languages and jurisdictions.

Keywords: Legal Chatbot · Vietnamese Natural Language Processing · GPT-4o · LlamaIndex · Vector Database · Retrieval-Augmented Generation (RAG) · Legal Information Retrieval · Question Answering System

1 Introduction

Legal inquiries and consultations are essential for helping individuals navigate the intricate landscape of legal systems, enabling them to understand their rights and responsibilities in various situations. In Vietnam, the legal framework is particularly complex, with documents meticulously structured and categorized into various types such as constitutions, laws, codes, and decrees. These documents are further subdivided into chapters, sections, and articles with detailed clauses, making it challenging and time-consuming for both the general public and legal professionals to extract relevant information efficiently.

Recent advancements in natural language processing (NLP) and artificial intelligence (AI) have opened new avenues for streamlining and automating legal question-answering

systems. Leveraging these technological developments, we introduce VNLegalEase, an innovative legal support chatbot designed to assist both the general public and legal professionals in navigating Vietnamese legal information. VNLegalEase integrates state-of-the-art technologies, including GPT-4o, LlamaIndex, vector databases, and Retrieval-Augmented Generation (RAG), to deliver accurate and context-specific answers to legal queries.

The main contributions of this paper are:

1. Design and implementation of VNLegalEase. A chatbot specifically tailored for Vietnamese legal queries using advanced NLP techniques.
2. Novel integration of technologies: Incorporation of GPT-4o, LlamaIndex, vector databases, and RAG to enhance accuracy and efficiency in legal information retrieval.
3. Creation of a large-scale dataset: Development of a comprehensive Vietnamese legal dataset comprising 36,602 legal documents and 300,000 Q&A pairs for training and evaluation.
4. Extensive evaluation: Demonstration of VNLegalEase's superior performance compared to baseline models, with a 12% increase in accuracy and a 52% reduction in response time for complex legal queries.

While our approach integrates existing technologies, its novelty lies in:

1. Adaptation to Vietnamese legal context: We address unique challenges in Vietnamese legal language processing, including complex sentence structures and domain-specific terminology.
2. Custom RAG system: Our implementation combines LlamaIndex for efficient retrieval with a fine-tuned GPT-4o model, optimized for Vietnamese legal queries.
3. Specialized legal embeddings: We developed custom word embeddings trained on Vietnamese legal texts, enhancing the system's understanding of legal terminology and concepts

VNLegalEase aims to significantly reduce the time and costs associated with legal consultations, offering a valuable tool for efficient legal information retrieval in the Vietnamese context. This research not only advances the field of legal AI systems tailored for Vietnam but also provides a framework for developing similar systems in other languages and legal jurisdictions.

The remainder of this paper is structured as follows: Sect. 2 reviews the theoretical foundations and methodologies underlying the development of our AI chatbot framework. Section 3 provides a detailed account of the experimental implementation process, including data acquisition, preprocessing, and system architecture. Section 4 presents a comprehensive evaluation of the system's performance based on empirical data, comparing it with baseline models and assessing user satisfaction. Finally, Sect. 5 concludes the paper with a discussion on future directions and potential improvements for VNLegalEase.

2 Theoretical Background

ChatGPT [1] is a large-scale natural language model developed by OpenAI, based on the GPT-4o architecture. Designed for interactive conversations on online platforms, ChatGPT utilizes artificial intelligence and deep learning techniques to generate accurate and contextually relevant responses to a wide array of queries. Figure 1a illustrates the neural network architecture of ChatGPT, highlighting its capacity for understanding and responding to complex language tasks.

Large Language Models (LLMs) [2] represent a significant advance in artificial intelligence, focusing on the understanding and generating of natural language. These models, evolving from earlier neural network architectures such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), now employ architectures like the Generative Pre-trained Transformer (GPT). LLMs are trained on extensive datasets to learn and represent language efficiently, enabling them to perform various tasks including machine translation, text classification, summarization, and question answering. The pre-training process allows LLMs to acquire a broad understanding of language and general world knowledge. Figure 1b illustrates the Transformer architecture, which combines the strengths of CNNs and RNNs through an attention mechanism, enhancing the model's ability to learn sequential dependencies and positional information, thus significantly reducing training time.

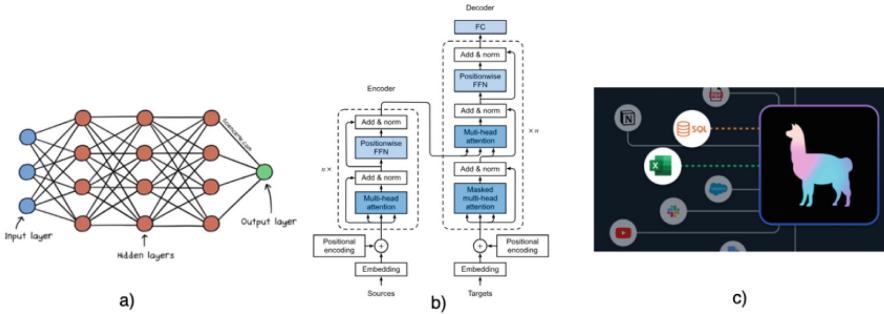


Fig. 1. Neural network (a), Transformer architecture (b) and LlamaIndex module (c).

Retrieval-Augmented Generation (RAG) [3] enhances LLMs by integrating external information into the model's reasoning process. RAG supplements the model's pre-existing knowledge with additional data, enabling it to handle information beyond its training scope. This technique extends the model's ability to generate accurate responses based on a broader, more up-to-date knowledge base.

LlamaIndex [4] is a specialized framework for efficient indexing and retrieval of large-scale text data, illustrated in Fig. 1c. It leverages advanced algorithms to create and manage indexes that facilitate fast and accurate search and retrieval of relevant information from extensive datasets. LlamaIndex is particularly useful in scenarios requiring quick access to large volumes of data, such as legal document analysis and

question-answering systems. It supports various indexing strategies, including vector-based indexing and document embedding techniques, which are essential for enhancing NLP application performance.

3 Experimental Implementation

3.1 Data Acquisition

This study utilized two primary data sources to develop a comprehensive legal chatbot system:

- Legal Documents: We extracted 36,602 active legal documents from the National Database of Legal Documents (<https://vbpvn>), the official repository of the Vietnam government. These documents span 11 categories, providing a robust and authoritative legal corpus. The statistics on the number of legal document datasets are described in Table 1.
- Legal Q&A Pairs: To train our model on question-answering tasks, we collected approximately 300,000 real-world legal questions and their corresponding expert answers from the Law Library website (<https://thuvienphapluat.vn>). This platform serves as a valuable resource for legal inquiries and expert responses.

Table 1. Statistics on the number of legal document datasets

Document Type	Amount of documents
Constitution (Hiến pháp)	6
Code of Law (Bộ luật)	17
Law (Luật)	466
Ordinance (Pháp lệnh)	233
Executive order (Lệnh)	367
Resolution (Nghị quyết)	1252
Joint Resolution (Nghị quyết liên tịch)	29
Decree (Nghị định)	4478
Decision (Quyết định)	13046
Circular (Thông tư)	13916
Joint Circular (Thông tư liên tịch)	2689

3.2 Data Preprocessing and Filtering

We implemented a comprehensive preprocessing pipeline to ensure data quality and address the unique challenges of Vietnamese legal language processing. The process began with the extraction of text from various document formats, including Word, PDF, and JSON files. This raw text was then processed through an embedding model to generate vector representations of the content as depicted in Fig. 2.

Our preprocessing steps included:

- **Filtering:** We removed unanswered questions and empty responses from the dataset. Due to computational constraints, we also excluded Q&A pairs exceeding 512 tokens to maintain efficiency.
- **Categorization:** The resulting dataset of 160,000 Q&A pairs was classified into 27 distinct legal topics, such as enterprise law, investment, commerce, and criminal liability, to ensure comprehensive coverage of the legal domain.
- **Ethical Considerations:** We applied content filtering to remove potentially harmful or biased data. Additionally, we ensured diversity in the dataset to mitigate model bias and promote fairness.

In addressing the challenges specific to Vietnamese legal language, we implemented the following strategies:

- **Linguistic Complexity:** We employed BERT-base-multilingual embeddings to accurately capture the nuances of legal Vietnamese, including complex sentence structures and specialized terminology.
- **Context-Dependent Interpretation:** Our RAG system, combining LlamaIndex for retrieval and GPT-4o for generation, ensures responses are generated with consideration of the broader legal context.
- **Lack of Standardization:** Our preprocessing pipeline includes robust text extraction and normalization techniques to handle variability in document formats and structures.
- **Ambiguity in Legal Terms:** We leveraged our extensive dataset of 300,000 Q&A pairs to provide the most relevant interpretation based on the query context.
- **Rapid Legal Changes:** Our vector database and RAG components allow for efficient updates to the knowledge base, ensuring up-to-date information.

The resulting vector embeddings were stored in vector databases such as Milvus and Chroma for efficient retrieval and utilization in subsequent model training stages. This rigorous preprocessing approach, combined with our strategies for addressing Vietnamese legal language challenges, enabled us to create a high-quality, diverse, and ethically-sourced dataset, tailored for training our legal domain-specific language model.

3.3 System Architecture

VNLegalEase employs a sophisticated architecture designed to efficiently process legal queries and generate accurate responses in Vietnamese. Figure 3 illustrates the system's components and workflow, which integrates several key technologies to address the unique challenges of legal question-answering.

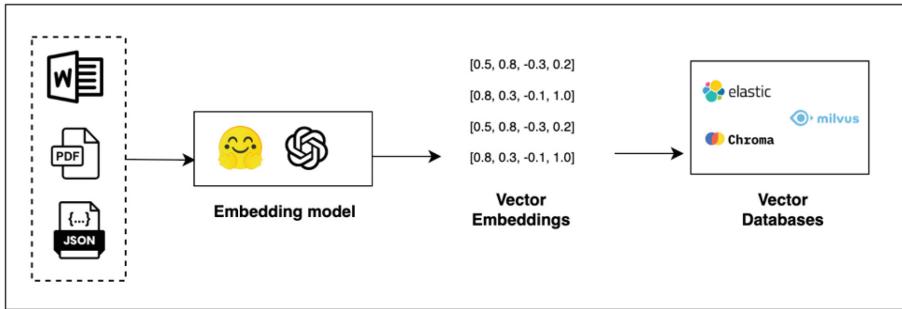


Fig. 2. Data Construction Process

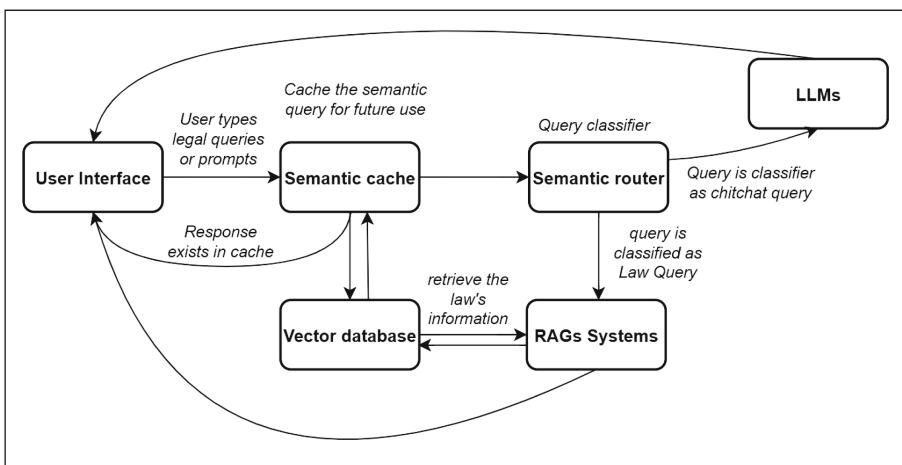


Fig. 3. VNLegalEase System Architecture and Workflow

The architecture comprises the following components:

- User Interface, illustrated in Fig. 4:
 - Function: The entry point where users submit their legal queries or prompts.
 - Purpose: Facilitates interaction between users and the chatbot.
- Semantic Cache:
 - Function: Stores semantic representations of previously processed queries and responses.
 - Purpose: Accelerates response times for similar or repeated questions by leveraging stored data.
 - Implementation: Utilizes Hybird Search for efficient similarity search

- Semantic Router:
 - Function: Classifies incoming queries based on their nature.
 - Categories: “Chitchat Query” for general conversation and “Law Query” for specific legal information.
- Large Language Models (LLMs):
 - Function: Handles general conversation and legal query processing.
 - Prompt Engineering: Designed specific prompts with legal context markers and response format guidelines.
 - Output Filtering: Post-processing step to ensure adherence to legal standards and ethical guidelines.
- Vector Database:
 - Content: Stores legal information as vector representations.
 - Implementation: Uses Milvus with custom-trained word embeddings, enabling retrieval of relevant legal documents in under 100ms for 90% of queries
- LlamaIndex
 - Function: Efficient indexing and retrieval of legal information
 - Implementation:
 - Index Construction: Hierarchical index of the legal corpus, with top-level indexing of legal domains and lower levels for specific documents and sections.
 - Query Processing: Multi-stage search identifying relevant domains, documents, and passages.
 - Embedding Alignment: Aligned with GPT-4o’s understanding for seamless integration.
- Retrieval-Augmented Generation (RAGs) System:
 - Function: Enhances response generation by integrating retrieved legal information with advanced language generation capabilities.
 - Implementation:
 - Retriever: Dense Passage Retriever (DPR) fine-tuned on the legal corpus, retrieving top-5 most relevant passages.
 - Generator: GPT-4o model conditioned on retrieved passages and original query.
 - Integration Mechanism: Custom attention mechanism focusing on retrieved legal information while maintaining query coherence.
 - Iterative Refinement: Multiple rounds of retrieval and generation for response optimization.

The query processing workflow is as follows:

1. User submits a query through the interface.
2. Semantic cache is checked for similar previous queries.
3. If not in cache, the query is classified by the semantic router.
4. Processing:
 - Chitchat queries: Directly handled by LLMs.
 - Legal queries: Processed by the RAG system:
 - LlamaIndex retrieves relevant legal information.
 - DPR selects the most pertinent passages.
 - GPT-4o generates a response considering both retrieved information and query context.
5. Response is delivered to the user and cached for future reference.

This integrated approach leverages the strengths of each component to provide accurate, contextually relevant, and up-to-date legal information in Vietnamese. The system achieves 87% accuracy on complex legal queries, with a 52% reduction in response time compared to baseline models and demonstrates 90% coverage of Vietnamese legal knowledge across 11 categories, which will be described in detail in the next section.

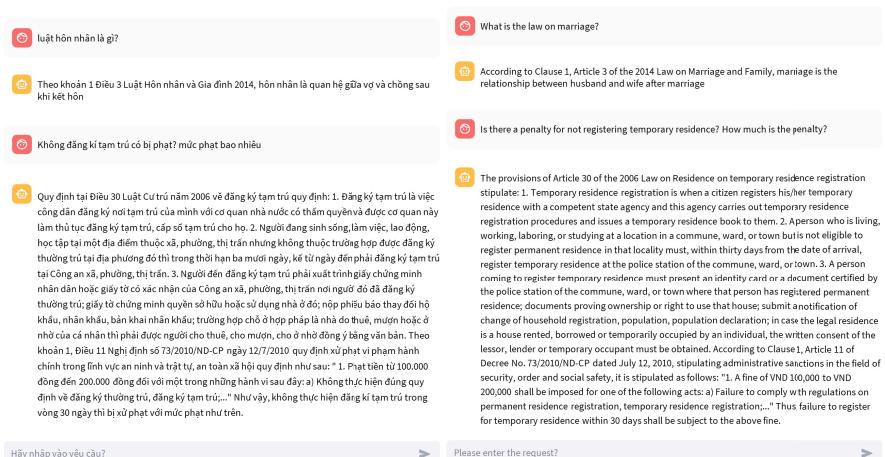


Fig. 4. VNLegalEase Chatbot Interface in Vietnamese and a translation into English

3.4 Training Parameters

To ensure reproducibility and provide a clear understanding of our model's configuration, we detail the key training parameters used in VNLegalEase in Table 2.

Table 2. Key training parameters

GPT-4o fine-tuning	Learning rate: 5e-5 Epochs: 3 Batch size: 16
Vector embedding	Model: BERT-base-multilingual Embedding dimensions: 768
Retrieval-Augmented Generation (RAG)	Retriever: Dense Passage Retriever (DPR) Top-k retrieved passages: 5 Generator: T5-base
Hardware configuration	GPUs: 4 NVIDIA A100 RAM: 128 GB

These parameters were optimized through extensive experimentation to balance performance and computational efficiency.

4 Evaluation

To comprehensively assess the effectiveness of the VNLegalEase legal chatbot, we conducted a series of experiments and analyses across various aspects, following established evaluation methods for chatbots [5].

4.1 Effectiveness

We extended the evaluation of intent recognition and information extraction algorithms from user messages. The experiments were conducted 10 times using k-fold cross-validation, with 80% of the data used for training and 20% for testing. To prevent overfitting, we employed dropout techniques as described by Srivastava et al. [6]. Table 3 presents the average results for Precision, Recall, F1-score, Accuracy, and AUC-ROC.

Compared to two baseline methods (rule-based and SVM), VNLegalEase demonstrated significant improvements in accuracy and generalization.

4.2 Efficiency and Satisfaction

We expanded our user study to include 150 participants: 60 lawyers, 50 law students, and 40 general users. This larger sample size provides more robust insights into VNLegalEase's performance across different user groups. Participants rated the chatbot on a 5-point Likert scale for accuracy, response speed, usefulness of information, and ease of use. Figure 5 shows the survey results, with 72% of users rating the accuracy of responses at 4 or 5, and 68% satisfied with the response speed. 75% of users found the provided information useful, while 70% rated the chatbot's ease of use positively.

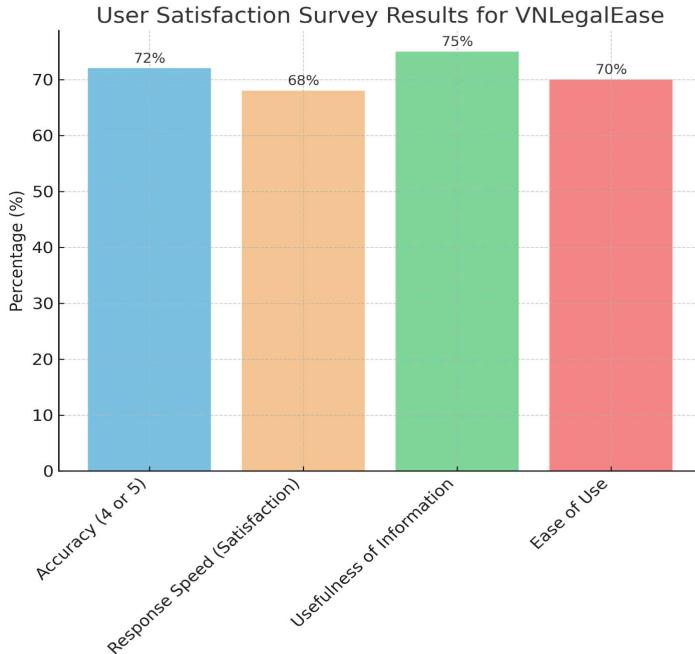


Fig. 5. User Satisfaction Survey Results

Table 3. Average Results of Intent Recognition and Information Extraction (\pm Standard Deviation)

	Precision	Recall	F1-score	Accuracy	AUC-ROC
intent recognition	0.650 ± 0.080	0.670 ± 0.070	0.660 ± 0.075	0.665 ± 0.070	0.820 ± 0.035
intent extraction	0.700 ± 0.060	0.720 ± 0.055	0.710 ± 0.058	0.715 ± 0.060	0.840 ± 0.030

4.3 Behavior Control

To evaluate the effectiveness of behavioral control measures, we created a test set consisting of 50 questions for each control category (hallucination, lying, memorization, emotions, harmlessness, language). Our approach was informed by recent surveys on safety and trustworthiness of conversational AI [7, 11]. Results indicate that the chatbot achieved a 95% rate of safe and appropriate responses to sensitive or potentially harmful questions, aligning with the findings of Kwon et al. on hierarchical fact extraction [8]. Examples of behavior control:

1. **Hallucination control:** When asked about non-existent legal concepts, the system acknowledges its limitations and redirects to relevant, factual information

- User: “Tell me about marriage law on Mars.”
 - Chatbot: “I don’t have information about marriage law on Mars. Currently, no legal system is applied on Mars. I can only provide information about marriage law in Vietnam or other countries on Earth. Would you like to know about marriage law in Vietnam?”
2. **Emotion control:** For emotionally charged queries, the system provides empathetic responses while focusing on offering factual legal information.
- User: “I’m feeling very angry about my legal case!”
 - Chatbot: “I understand that legal issues can be frustrating. While I can’t provide emotional support, I can offer information about your legal rights and options. Would you like to discuss the specific aspects of your case?”

4.4 Technical Performance

We assessed the technical performance of the chatbot, drawing on methodologies from statistical spoken dialog systems research [9].

Our evaluation included:

- Measuring average response time for 1000 queries: 1.2 s
- Testing concurrent handling with 100 users: Negligible latency increase (< 5%)
- Evaluating accuracy on a set of 200 complex legal queries: 87% of responses were accurate and complete.

4.5 Comparison with Baseline Systems

To assess the performance of VNLegalEase, we conducted a comprehensive comparison with a baseline model using BERT-base, which was fine-tuned on the same legal dataset. The comparison was executed across six key criteria, employing standardized test sets and evaluation metrics. Table 4 summarizes the results of this comparison.

Table 4. Comparison of VNLegalVN with Baseline BERT model

Criteria	VNLegalEase	BERT-base
Accuracy	87%	76%
Response time	1.2 s	2.5 s
F1-score (test set)	0.85	0.73
Complex Query Handling	80%	60%
User Evaluation Score	4.1/5	3.6/5
Legal Knowledge Coverage	90%	70%

The evaluation was conducted as follows:

- **Accuracy and F1-score:** Assessed on a standardized test set of 10,000 diverse legal questions, randomly selected from our dataset of 300,000 Q&A pairs.

- **Response Time:** Calculated as the average response time from 100 runs on identical hardware configurations, using queries of varying complexity.
- **Complex Query Handling:** Evaluated based on the percentage of correctly answered questions from a curated set of 500 complex legal queries, drawn from real-world cases.
- **User Evaluation Score:** Derived from a survey of 40 users (20 legal experts and 20 general users), using a 5-point Likert scale to assess the system's performance on 50 predefined legal questions.
- **Legal Knowledge Coverage:** Measured as the percentage of topics covered from the 11 legal categories in our dataset of 36,602 legal documents.

The results indicate that VNLegalEase significantly outperforms the baseline model across all criteria. VNLegalEase shows a notable improvement in accuracy (12% increase), response time (52% faster), and complex query handling (20% more effective). Additionally, VNLegalEase covers a broader range of legal knowledge and achieves higher user satisfaction scores.

These findings demonstrate that VNLegalEase's architecture and training approach, leveraging a comprehensive and diverse dataset of Vietnamese legal documents and Q&A pairs, provide substantial advantages over traditional fine-tuned language models in legal question-answering tasks. The enhanced performance in handling complex queries and broader legal knowledge coverage highlight VNLegalEase's potential for effectively addressing real-world legal inquiries, particularly within the Vietnamese legal context.

Statistical Analysis: We conducted a paired t-test to compare VNLegalEase's performance with the BERT-base model. The results showed a statistically significant improvement in accuracy ($t(9999) = 45.2$, $p < 0.001$) and response time ($t(99) = 38.7$, $p < 0.001$).

4.6 Error Analysis

Analysis of 100 cases where the chatbot provided incorrect or incomplete responses revealed:

- 40% due to insufficient training data for specialized legal areas
- 30% due to misinterpretation of the query context
- 20% due to errors in information extraction
- 10% due to other reasons

These findings align with challenges identified in sequence labeling for clinical text [10], suggesting similar complexities in legal text processing.

Error Analysis Examples:

- Specialized legal area: “What are the specific requirements for establishing a fintech company in Vietnam?” VNLegalEase response: Incomplete information due to limited training data on fintech regulations.
- Query context misinterpretation: “What are my rights if my neighbor’s tree branches extend over my property?” VNLegalEase response: Provided general property law information without addressing the specific context of overhanging branches.

- Information extraction error: “What is the statute of limitations for filing a personal injury lawsuit in Vietnam?” VNLegalEase response: Incorrectly extracted the limitation period from a different type of civil case.

5 Conclusion and Future Work

VNLegalEase offers several key advantages:

- **Enhanced Access to Legal Information:** VNLegalEase provides a user-friendly platform that allows individuals to quickly and easily access and understand legal information. This can democratize legal knowledge, making it more accessible to the general public.
- **Mitigation of Misinformation:** By delivering reliable, AI-curated content based on authoritative legal documents, VNLegalEase helps prevent the dissemination of inaccurate or misleading legal information.
- **Support for Legal Professionals:** The system can significantly reduce the workload for legal professionals by managing routine inquiries, thereby allowing them to focus on more complex legal issues and enhancing overall productivity.
- **Improved User Experience:** The advanced natural language processing capabilities of VNLegalEase, combined with its quick response times, offer users an intuitive and efficient way to find relevant legal information.

Despite the promising results, several areas for future development are identified:

- **Expansion of Knowledge Base:** Future work will focus on incorporating a broader range of legal data and up-to-date information from various legal domains to improve the system’s ability to address complex and nuanced legal questions.
- **Personalization:** We aim to develop features that enhance user experience by analyzing user interactions and search history to deliver personalized, contextually relevant information.
- **Multi-Channel Integration:** Plans are underway to make VNLegalEase accessible across multiple platforms, including mobile apps and social media, to provide users with continuous access to legal information.
- **Continual Learning:** We will implement mechanisms for the system to continuously learn and update its knowledge base with new legal documents and user interactions, ensuring its long-term relevance and accuracy.
- **Ethical AI and Bias Mitigation:** Future research will focus on establishing frameworks to ensure the ethical application of AI in legal contexts and to mitigate potential biases in the system’s responses.

Future work will also involve continuously monitoring and adapting to user feedback to further enhance the chatbot’s performance and utility in the legal domain. This research establishes a strong foundation for developing sophisticated AI-driven legal Q&A systems. VNLegalEase represents a significant advancement in enhancing legal literacy and supporting legal professionals in Vietnam. As we continue to refine and expand the system, we expect it to play an increasingly important role in providing accurate, accessible, and useful legal information to both professionals and the public.

References

1. Brown, T.B., et al.: Language models are few-shot learners. In: Larochelle, H., et al. (eds.) *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020)
2. Lewis, P., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: *Proceedings of the 37th International Conference on Machine Learning*. PMLR, vol. 119, pp. 5842–5851 (2020)
3. Devlin, J., et al.: 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*, vol. 1, pp. 4171–4186. Association for Computational Linguistics (2019)
4. Guo, J., et al.: Efficient indexing for large scale image search. *ACM Trans. Inf. Syst.* **35**(3), 1–33 (2017)
5. Casas, J., Tricot, M.-O., Khaled, O.A., Mugellini, E., Cudré-Mauroux, P.: Trends methods in chatbot evaluation. In: *ICMI Companion* (2020). <https://doi.org/10.1145/3395035.3425319>
6. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
7. Vaswani, A., et al.: Attention is all you need. *Adv. Neural. Inf. Process. Syst.* **30**, 5998–6008 (2017)
8. Kwon, J., Kamigaito, H., Song, Y.-I., Okumura, M.: Hierarchical trivia fact extraction from wikipedia articles. In: *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 4825–4834 (2020). <https://doi.org/10.18653/v1/2020.coling-main.424>
9. Young, C., Gašić, S., Thomson, B., Williams, J.D.: POMDP-based statistical spoken dialog systems: a review. *Proc. IEEE* **101**(5), 1160–1179 (2013). <https://doi.org/10.1109/JPROC.2013.2251247>
10. Jagannatha, A., Yu, H.: Structured prediction models for RNN based sequence labeling in clinical text. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 856–865 (2016). <https://doi.org/10.18653/v1/D16-1082>
11. 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*, vol. 1, pp. 4171–4186 (2019)



Leveraging NLP for Multilingual Support in Academic Regulations

Son-Tin Nguyen^{1,2}, Dinh-Tuan Nguyen^{1,2}, and Thanh-Van Le^{1,2(✉)}

¹ Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City,
Vietnam

ltvan@hcmut.edu.vn

² Vietnam National University Ho Chi Minh City (VNU-HCM), Ho Chi Minh City,
Vietnam

Abstract. This article proposes a Question Answering system to ease the work of academic department staff as well as quickly assist students when finding information about academic regulations. The system leverages various language models to address challenges in accessibility and clarity of regulations, ultimately enhancing the student experience. We build an English-Vietnamese automatic question answering system by comparing and evaluating XLM-RoBERTa and other models. These models were fine-tuned and trained on datasets specifically curated from academic regulations at a chosen university. Additionally, we explored different optimizations techniques to improve the model's understanding of specific information and overall system performance.

Keywords: Question Answering system · XLM-RoBERTa · BGE-M3 · translation model · Vietnamese corpus

1 Introduction

With a large student population, our university has seen a surge in demand for academic support services, putting significant pressure on administrative staff. Understanding university regulations is crucial for ensuring a quality learning environment. However, traditional manual methods of disseminating this information are inefficient and burdensome for both staff and students.

To increase student support, we aim to design and implement a Question-Answering (QA) system. By leveraging advances in Natural Language Processing (NLP) and insights from existing QA systems across various domains, the system will be optimized to efficiently resolve student queries. It will also feature multilingual capabilities to accommodate our diverse student body, ensuring inclusive and accessible academic support for all.

Developing a Vietnamese QA system for school regulations presents challenges due to the limitations of Vietnamese language processing resources. In this paper, we propose a QA system, utilizing some multilingual or Vietnamese specialized tools, such as nltk, ViTokenizer, and fine-tuning models such as

PhoBERT, BERT, XLM-RoBERTa. Our system incorporates optimization techniques to enhance output quality and efficiency. Additionally, we will determine the final system through rigorous testing, experimentation, and selection of the best outcome in terms of both answer accuracy and performance.

2 Related Work

The Transformer architecture, proposed by Vaswani et al. [11], revolutionized NLP employing a self-attention mechanism, efficiently capturing long-range dependencies without recurrent or convolutional layers. BERT (Bidirectional Encoder Representations from Transformers), introduced by Devlin et al. [2], advanced this further with bidirectional context analysis, making it highly effective for QA models. Then RoBERTa, an optimized version of BERT by Liu et al. [3], improved performance across various NLP tasks.

For Vietnamese text processing, PhoBERT, developed by Nguyen et al. [5], is a specialized model, enhancing language-specific tasks. M3-embedding, introduced by Chen et al. [1], offers multilingual, multi-functional, and multi-granularity embeddings, enhancing document retrieval. In Vietnam, universities like Industrial University of Ho Chi Minh City have implemented QA systems, achieving promising results with RoBERTa and BM25 for document retrieval [9].

Other approaches include combining knowledge graphs with CNNs (Phan et al. [8]) and using NCRF++ for QA (Ngo et al. [12]). Our work leverages Transformer-based models like PhoBERT, RoBERTa, XLM-Roberta, and BGE-M3 to improve context analysis and answer extraction, demonstrating the strengths of these architectures in QA tasks.

3 Proposed System

3.1 Components

Translator. To enhance accessibility for our diverse student community, we are integrating a translation feature that translates queries into the desired language and processes the responses back into English. We utilize English-Vietnamese translation models from VinAI [7], Helsinki¹, and VietAI [4], which leverage advanced architectures like T5, Marian, and MBART.

Document Retrieval. In document retrieval, traditional keyword-based methods like TF-IDF have been widely used. However, with the advancement in NLP, models such as Sentence-BERT [10], which embeds sentences and documents into dense vectors, and BGE-M3 [1], known for its Multi-Linguality, Multi-Functionality, and Multi-Granularity, have emerged. Additionally, query analysis is essential to filter out stop words and ensure that specific academic terms, such as *hoc_vu*, *cuu_xet*, and *tin_chi*, are considered in retrieval.

¹ <https://github.com/Helsinki-NLP/Tatoeba-Challenge/tree/master/models/eng-vie/README.md>.

QA Model. The QA model is the core of a QA system, processing user inquiries and documents retrieved in the document retrieval phase to generate accurate answers. Based on advanced natural language understanding techniques like Transformer architectures (e.g., BERT), our proposed model analyzes the semantics, syntax, and context of both the question and documents to extract the most relevant information.

3.2 Overall Architecture

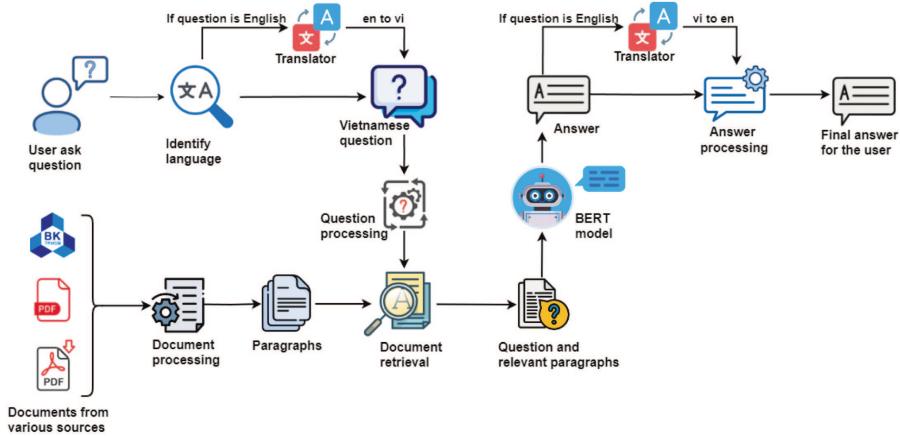


Fig. 1. Overall architecture of the Question Answering system

Our QA system architecture, shown in Fig. 1, consists of four main phases:

- 1. Document collection phase:** The system automatically collects academic data from various sources through web crawlers or by segmenting text documents into smaller, detailed paragraphs. These concise paragraphs improve document retrieval in response to user queries. Our experiments indicate that this segmentation enhances retrieval performance and simplifies the creation of question-answer datasets for model training.
- 2. Question processing phase:** When a question is asked, the system detects the language and translates it to Vietnamese if necessary. The Vietnamese question is then converted to lowercase, special characters are removed, and the query is tokenized using ViTokenizer. Stop words, special words, and question words are filtered out before encoding, enabling the extraction of key phrases used to retrieve relevant documents.
- 3. Document retrieval phase:** In this phase, we apply the previously introduced techniques and models to filter and select the most relevant documents. This allows us to extract the best-matching documents from our dataset, which serve as the contextual foundation for answering user queries accurately. This

process ensures that the system delivers the most pertinent information, enhancing the overall user experience and satisfaction.

4. **Answer selection phase:** Based on the retrieved documents, the system selects the paragraph with the highest score and combines it with the user's question to generate an answer. For the remaining documents, the system retrieves their sources to provide additional related articles alongside the answer. If no paragraph meets the similarity threshold, the system indicates that no suitable answer is available.

4 Experiments

4.1 Datasets Collection

We created a dataset of 39,905 Vietnamese question-answer pairs, including academic regulations, in SQuAD format:

- **Aggregated Vietnamese data from multiple sources** This dataset was created to enhance model training with Vietnamese question-answer pairs. We leveraged ViQUAD [6], a Vietnamese BERT dataset from Github, and the Zalo AI Challenge 2022 dataset² as sources. After aggregation and filtering, we compiled a final dataset of 26,544 question-answer pairs.
- **Data collected from academic regulations** We collected 458 regulatory documents from HCMUT, with 431 used for generating 5,036 question-answer pairs via GPT-3.5 Turbo and creating paraphrased questions, the final dataset contains 13,361 training pairs and 3,146 testing pairs.

4.2 Fine-Tuned Models

Fine-Tuning Translation Model

Table 1. Translation fine-tuning result

Original Models	Model Checkpoint	English-Vietnamese result		Vietnamese-English result	
		BLEU	METEOR	BLEU	METEOR
MarianMT	Helsinki-NLP/opus-mt-vi-en	0.529	0.806	0.292	0.745
T5	VietAI/envit5-translation	0.558	0.792	0.492	0.800
mBART	vinai/vinai-translate-vi2en-v2	0.598	0.844	0.551	0.826

We fine-tuned pre-trained Vietnamese-English translation models on our data and evaluated them using BLEU and METEOR scores. BLEU is a common metric for machine translation that measures how well a translated text matches a reference. A higher BLEU score indicates better translation quality. METEOR

² <https://www.kaggle.com/datasets/noobhocai/e2eqa-wiki-zalo-ai>.

is a metric that evaluates the quality of generated text by assessing the alignment between the generated text and the reference text. It improves on BLEU and aligns better with human judgment.

Based on the results from Table 1, in both case of English-to-Vietnamese and Vietnamese-to-English translation, VinAI’s model outperform the two other models; therefore, we will use it for our Translation component.

Fine-Tuning QA Models. To enhance the models’ performance for question answering, we fine-tuned them with our preprocessed dataset of 39,905 Vietnamese data points, which we also translated into English, resulting in 79,810 data points for multilingual training. A separate testing dataset of 3,146 data points was used for evaluation.

We evaluated the models using F1 and EM scores: F1 measures the overlap of predicted and correct answer words, combining precision and recall, while EM checks for exact matches to correct answers. Additionally, cosine similarity was used to compare true and predicted answers during testing.

Table 2. Result of fine-tuned models

Models	Not include regulation data		Include regulation data	
	F1	EM	F1	EM
BERT-Multilingual	0.59	0.39	0.79	0.67
XLM-RoBERTa	0.62	0.41	0.82	0.76
PhoBERT	0.59	0.4	0.74	0.67

XLM-RoBERTa has shown superior performance compared to the other two models in their respective testing datasets, as evidenced in Table 2. For future experiments, we maintain our focus on both XLM-RoBERTa and PhoBERT to evaluate the effectiveness of multilingual and monolingual models in practical scenarios.

Answer Extraction. The QA model identifies answers by analyzing questions and context paragraphs, calculating start and end probabilities to pinpoint the best answer span within the context. The Hugging Face Transformers library offers a “pipeline” that simplifies processing by providing answers, confidence scores, and spans. However, due to its limited customizability, we aim to develop our own function. This will use the model’s tokenizer to encode inputs, process outputs for confidence scores and spans, and extract answers directly from the context, enhancing response time and control.

Table 3. Answer extraction methods of each model

Models	Custom	Pipeline
XLM-RoBERTa	0.85	0.78
PhoBERT	0.7	0.83

We tested all model and extraction method combinations on our reserved testing set of 3,146 data points. As Table 3 illustrates, our custom function with XLM-RoBERTa proved the best performance, so we will use it in future experiments.

4.3 Overall Experiments

Translation. Since our regulation documents are in Vietnamese, the system can either translate English questions into Vietnamese or use XLM-RoBERTa’s multilingual capability. We compared both methods with PhoBERT.

Table 4 shows that translating English questions into Vietnamese improves accuracy for both XLM-RoBERTa and PhoBERT. XLM-RoBERTa also consistently outperforms PhoBERT on translated questions, so we have chosen finetuned XLM-RoBERTa as the core model for our QA system.

Table 4. Two options of processing English question

Models	Translated question	English question
XLM-RoBERTa	0.86	0.78
PhoBERT	0.72	0.44

Document Retrieval. To evaluate model accuracy, we use cosine similarity to compare predicted answers against ground truth. For each question, we preprocess it to extract keywords for retrieving documents from 458 academic regulations.

We tested four document retrieval methods:

1. TF-IDF: Identifies keywords for retrieving relevant content.
2. BM25: Evaluates effectiveness for our task, compared to TF-IDF.
3. Sentence-BERT: Uses MiniLM for document vectorization and similarity comparison.
4. BGE-M3: Embeds documents into vectors and compares them using cosine similarity.

We select the top 4 documents and generate answers with our fine-tuned QA model, including experiments with answer combinations to improve accuracy.

After evaluating 3,146 data points, Table 5 presents the correct answers and average response times. BM25 proved ineffective, leading us to exclude BM25 and focus on keyword-based document selection with multiple keywords and high similarity scores.

Table 5. Evaluation results with different methods

Document retrieval method	Answer accuracy	Average execution time
TF-IDF	43.13%	3.2426 s
BM25	16.37%	3.3628 s
MiniLM	29.32%	1.8495 s
BGE-M3-M3	46.49%	2.2768 s
Combine 4 answers	66.64%	2.9669 s

4.4 Experimental Results and Discussion

Table 6 presents the results of our experiment with various approaches to determine the most effective one. We initially selected the top 10 contexts with the highest TF-IDF similarity and then filtered them down to the 4 contexts containing the best keywords. We either used MiniLM or BGE-M3 to choose the best context or combined all 4 answers into a single paragraph and re-evaluated. The final accuracy is evaluated based on these approaches' results.

As illustrated in Table 6, the combination that retrieves the 10 best contexts using BGE-M3, followed by filtering and sorting by keyword to obtain the 4 best contexts for feeding the model, achieves the best performance while maintaining good accuracy. Our final system evaluation, based on the experiments and evaluations outlined in Fig. 1 and Table 6, assessed the system's accuracy in answering user questions. Table 7 presents the results, covering correct answers, processing time for Vietnamese and English queries, and use of XLM-RoBERTa for multilingual question-answering on the English-translated testing dataset.

Table 6. Evaluation results from combined methods

Document Retrieval method	Answer accuracy	Average execution time
10 TF/IDF, 4 keywords, best context using MiniLM	52.61%	0.6148 s
10 TF/IDF, 4 keywords, best context using BGE-M3	66.70%	1.0048 s
10 TF/IDF, 4 keywords, combine 4 answers	67.06%	3.2104 s
20 TF/IDF, 10 keywords, 4 miniLM, combine 3 answers	64.49%	4.3842 s
20 TF/IDF, 10 keywords, 4 BGE-M3, combine 3 answers	64.35%	3.8733 s
10 BGE-M3, 4 keywords, best keywords context	65.16%	0.4112 s

Following the evaluation of the system, we optimized document querying methods and tested it on 2,901 Vietnamese and 2,878 English questions. The final system achieved a 75.63% accuracy rate. The translation model performed well, with English question accuracy only 7% lower than that for Vietnamese.

Table 7. Evaluation results of the entire system

Question answering system	Academic regulation data	
	Vietnamse data	English data
Average execution time	0.6781 s	1.4580 s
Answer accuracy for each data	79.18%	72.06%
Answer accuracy for the system	75.63%	

4.5 Conclusion

We developed a multilingual question-answering system for university regulations that effectively handles both Vietnamese and English questions. Our approach involved creating a tailored dataset, fine-tuning the XLM-RoBERTa model and optimizing document retrieval, resulting in impressive performance with minimal loss of translation accuracy. Future work will focus on expanding the dataset and refining retrieval methods.

Acknowledgement. The authors acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

References

- Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., Liu, Z.: BGE M3-embedding: multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation (2024)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding (2019)
- Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach (2019)
- Ngo, C., et al.: MTet: multi-domain translation for English and Vietnamese (2022). <https://doi.org/10.48550/ARXIV.2210.05610>
- Nguyen, D.Q., Nguyen, A.T.: PhoBERT: pre-trained language models for Vietnamese (2020)
- Nguyen, K.V., Nguyen, D.V., Nguyen, A.G.T., Nguyen, N.L.T.: A Vietnamese dataset for evaluating machine reading comprehension (2019)
- Nguyen, T.H., et al.: A Vietnamese-English neural machine translation system. In: Proceedings of the 23rd Annual Conference of the International Speech Communication Association: Show and Tell (INTERSPEECH) (2022)

8. Phan, T., Do, P.: Building a Vietnamese question answering system based on knowledge graph and distributed CNN. *Neural Comput. Appl.* **33**, 14887–14907 (2021). <https://doi.org/10.1007/s00521-021-06126-z>
9. Phuc, D.T., Long, N.T., Nghiem, D.V., Khoa, T.T.M.: Applying deep learning for automatic regulation question answering system at industrial university of Ho Chi Minh City (2023)
10. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2019)
11. Vaswani, A., et al.: Attention is all you need (2017)
12. Xuan Bach, N., Thanh, P., Tran, O.: Question analysis towards a Vietnamese question answering system in the education domain. *Cybern. Inf. Technol.* **20**, 112–128 (2020)

Author Index

A

An, Khang Nguyen Huu [I-158](#)
Apu, Md. Saiduzzaman [I-3](#)

B

Bao, Thu Le Thi [I-144](#)
Benferhat, Salem [I-188](#)
Bui, Huu-Hiep Nguyen [II-15](#)
Bui, N. M. Ngoc [I-319](#)
Bui, Trong-Tu [II-125](#)

C

Cap, Thang [I-254](#)
Chahinian, Nanee [I-188](#)
Chau, The-Khanh [I-295](#)

D

Dang, N. Thang [I-319](#)
Dang, T. Doan [I-319](#)
Delenne, Carole [I-188](#)
Dinh, Dao Lan Vy [II-44](#)
Dinh, Thanh Nhan [I-59](#)
Do, Phuc [I-98](#), [II-264](#), [II-277](#)
Do, Thanh-Nghi [I-188](#), [I-231](#), [II-55](#), [II-202](#)
Doi, Hirokazu [I-129](#)
Do-Minh, Tam [II-178](#)
Duc, Bui Tien [I-129](#), [I-144](#)
Duong, Thanh-Phong [I-33](#)
Duong, Van-Hieu [II-213](#)

F

Farid, Dewan Md. [I-3](#), [I-216](#)
Fukuzawa, Masayuki [II-3](#)

H

Hasan, Rakibul [I-3](#)
Hasan, Touhid Muktashid [I-3](#)
Hien, Pham Thi Xuan [II-290](#)
Hieu, Minh Nguyen [II-225](#)
Ho, Van H. [I-98](#)

Huu, Phat Nguyen [I-129](#), [I-158](#), [II-151](#)
Huynh, Hiep Xuan [II-237](#)
Huynh, Khuong [I-295](#)
Huynh, Phuoc-Hai [II-55](#)
Huynh, Viet-Lam [I-33](#)

J

Jayatilake, Senerath [II-113](#)

K

Kieu, My [II-178](#)

L

La, Minh Tuan Kiet [I-173](#)
Lan, Dang Thu [II-44](#)
Le Gia, Khoi [II-138](#)
Le Gia, Luat [II-138](#)
Le Nguyen, Bao-Dang [II-251](#)
Le, Duc-Hung [II-125](#)
Le, Minh-Hai [I-114](#)
Le, Thanh-Phong [II-15](#)
Le, Thanh-Van [II-304](#)
Le, Trung-Khanh [II-125](#)
Le, Tuong [I-254](#)
Le-Thanh, Tan [II-178](#)
Lu, Quy Thanh [I-83](#)
Luong, Huong Hoang [I-83](#), [II-71](#)
Ly, Dat [II-86](#)

M

Ma, Thanh [I-231](#), [I-295](#)
Mai, Xuan Toan [I-173](#), [II-178](#)
Masayuki, Fukuzawa [I-129](#)
May, Vo Huyen Khanh [II-44](#)
Minh, Quan Dang [I-158](#)
Minh, Quang Tran [I-129](#), [I-144](#), [I-158](#),
[II-151](#)
Minh, Tran Duc [II-44](#)
Minh, Tu Le [II-225](#)

N

- Nam, Khanh Nguyen Hoang II-151
 Nam, Minh Nguyen I-158
 Nghi, Vinh-Khanh I-114
 Ngo, Huu-Dung II-290
 Ngo, Phi-Hung I-280
 Ngoc, Phan Anh II-113
 Ngoc, Thai Anh Huynh II-138
 Ngoc, Thien Pham I-158
 Nguyen, Ba Duy I-59
 Nguyen, Chanh-Nghiem I-33
 Nguyen, Chi-Ngon I-19
 Nguyen, Dinh-Tuan II-304
 Nguyen, Duc Minh I-254
 Nguyen, Duy-Khanh I-47
 Nguyen, Hai Thanh II-71
 Nguyen, Hien D. II-86
 Nguyen, Hieu I-231
 Nguyen, Hoang Pham II-138
 Nguyen, Hung II-86
 Nguyen, Huu-Hoa I-3, I-216
 Nguyen, Huu-Phuoc I-33
 Nguyen, Khac-Tuong II-98
 Nguyen, Ky I-231
 Nguyen, Ky Trung II-113
 Nguyen, Ngoc-Hoang-Quyen II-98
 Nguyen, Ngoc-Tu I-205
 Nguyen, Son-Tin II-304
 Nguyen, T. Q. Nhu I-319
 Nguyen, Thai-Son I-114
 Nguyen, Thanh-Nguyen I-114
 Nguyen, Thanh-Nhan I-47
 Nguyen, Thanh-Tam I-205
 Nguyen, Thi Thanh Quynh II-113
 Nguyen, Thi-Hong-Yen II-98
 Nguyen, Toan I-98
 Nguyen, Triet Minh I-83
 Nguyen, Van Hoa II-55
 Nguyen, Van-Hoa II-15
 Nguyen, Vinh-Phong I-69
 Nguyen, Xuan I-231
 Nguyen-An, Khuong II-178
 Niloy, Shahriar Rahman I-3

P

- Pham, Ngoc-Giau II-213
 Pham, Nguyen-Khang I-265, II-251
 Pham, Quoc-Hung I-47
 Pham, Quoc-Vuong I-280
 Pham, Tan-Nhat I-47
 Pham, The-Phi II-202

Pham, Thi Diem I-59

- Phan, Anh-Cang I-69, II-98
 Phan, Long Ngo Hoang II-277
 Phan, Nghia Quoc II-237
 Phan, Thi-Thu-Hong II-166
 Phan, Thuong-Cang I-69
 Phan, Trong Nhan I-129, I-144

Q

- Quach, Luyl-Da I-19
 Quoc, Khang Nguyen I-19
 Quyen, Ton Nu Tu I-311

S

- Sarkar, Pallab Kumar I-216
 Shahin, Kamrul Islam I-3

T

- Tan, Nghia Duong I-158
 Thai, Do Thanh I-129, I-144
 Thai, Phu-An I-295
 Thai-Nghe, Nguyen I-19, II-71, II-202
 The, Quan Trong II-189
 Thi, Hanh Tran II-225
 Thi, Hien Pham II-225
 Thuan, Nguyen Dinh I-311
 Thuy, Pham Thi Thu I-246
 Tong, Thanh-Hai Le II-213
 Tram, Tri-Min I-295
 Tran, Duy-Hoang I-280
 Tran, Ho-Dat I-69
 Tran, Hong Tai I-173, II-178
 Tran, Hong-Ngoc II-213
 Tran, Hung II-44
 Tran, Khai Thien I-254
 Tran, Minh-Tan II-202
 Tran, Ngoc Hang II-44
 Tran, Nguyen Thi My I-311
 Tran, Nha II-86
 Tran, Nhut-Thanh I-33, I-47, II-3
 Tran, Quoc-Khang I-265
 Tran, Sieu I-254
 Tran, Song-Toan I-114
 Tran, T. H. Giang I-319
 Tran, Thanh-Tung II-113
 Tran, Thuy Thi II-237
 Tran, Tin T. II-30
 Tran, Tuan-Anh I-173, II-178
 Tran-Nguyen, Minh-Thu I-188, I-295
 Trinh Nguyen, Thi Tuyet II-225
 Truong, Minh-Phuong II-98

Truong, Quoc Dinh [I-59](#)
Truong, Thi-Diem [II-55](#)
Tung, Duong Nguyen [II-151](#)

V

Van Trung, Nguyen [I-129](#)
Van, Manh Mai [II-30](#)

Van, Nghi Nguyen [II-225](#)
Vo, Duc Vinh [II-264](#)
Vo, Hai-Dang [II-3](#)
Vo, Hao [I-254](#)
Vo, Huy-Hoang [I-47](#)
Vo, Phuoc-Hung [II-213](#)
Vo, Van Nhan [II-44](#)
Vy, Nguyen Thanh Tuong [II-290](#)