

Deepak Garg
Vishnu Pendyala
Suneet Kumar Gupta
Mohammad Najafzadeh (Eds.)

Communications in Computer and Information Science

2434

Advanced Computing

14th International Conference, IACC 2024
Bali, Indonesia, December 13–14, 2024
Revised Selected Papers

Communications in Computer and Information Science

2434

Series Editors

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

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

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.

Deepak Garg · Vishnu Pendyala ·
Suneet Kumar Gupta · Mohammad Najafzadeh
Editors

Advanced Computing

14th International Conference, IACC 2024
Bali, Indonesia, December 13–14, 2024
Revised Selected Papers

Editors

Deepak Garg
SR University
Warangal, Telangana, India

Vishnu Pendyala
San Jose State University
San Jose, CA, USA

Suneet Kumar Gupta
UPES
Dehradun, Uttarakhand, India

Mohammad Najafzadeh
Graduate University of Advanced Technology
Kerman, Iran

ISSN 1865-0929

ISSN 1865-0937 (electronic)

Communications in Computer and Information Science

ISBN 978-3-031-84601-4

ISBN 978-3-031-84602-1 (eBook)

<https://doi.org/10.1007/978-3-031-84602-1>

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

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

If disposing of this product, please recycle the paper.

Preface

The objective of the 14th International Advanced Computing Conference (IACC 2024) was to bring together researchers, developers, and practitioners from academia and industry working in the domain of advanced computing. Researchers were invited to share their thoughts and present recent developments and technical solutions in the domains of Generative AI, Advances in Machine Learning and Deep Learning, Advances in Applications of Artificial Intelligence in Interdisciplinary Areas, Reinforcement Learning, and Advances in Data Science. The conference took place on the 13th and 14th December 2024 in Bali, Indonesia. All editions of the series are successfully indexed in ISI, Scopus, DBLP, Compendex, SJR, Google Scholar, etc.

The conference followed a single-blind review process and had the policy of at least two reviews per paper. This year's conference received 89 submissions, of which 17 articles were accepted. The conference has a track record of acceptance rates ~19% in the last 13 years. More than 12 IEEE/ACM Fellows hold key positions on the conference committee, giving it a quality edge. In the last 13 years the conference citation score has been consistently increasing.

This has been possible due to adherence to quality parameters of the review process and acceptance rate without any exceptions, which allows us to make some of the best research available through this platform.

December 2024

Deepak Garg
Vishnu Pendyala
Suneet Kumar Gupta
Mohammad Najafzadeh

Organization

Honorary Co-chairs

Sundaraja Sitharama Iyengar
Sartaj Sahni
Jagannathan Sarangpani

Ajith Abraham
P. N. Suganthan
Jaume Anguera
Joel J. P. C. Rodrigues

Florida International University, USA
University of Florida, USA
Missouri University of Science and Technology,
USA
Bennett University, India
Qatar University, Qatar
Universitat Ramon Llull, Spain
Federal University of Piauí (UFPI), Brazil

General Co-chairs

Deepak Garg
Suneet K. Gupta
Vishnu Pendyala
Mohammad Najafzadeh

SR University, India
UPES, India
San José State University, USA
Graduate University of Advanced Technology,
Iran

Program Co-chairs

Kit Wong
George Ghinea
Carol Smidts
Ram D. Sriram

Sanjay Madria
Marques Oge
Vijay Kumar
Ajay Gupta
Xiaochun Cheng

University College London, UK
Brunel University London, UK
Ohio State University, USA
National Institute of Standards & Technology,
USA
University of Missouri, USA
Florida Atlantic University, USA
University of Missouri-Kansas City, USA
Western Michigan University, USA
Middlesex University London, UK

Technical Program Committee/International Advisory Committee

Shivani Goel	SR University, India
Sumeet Dua	Louisiana Tech University, USA
Roger Zimmermann	National University of Singapore, Singapore
B. V. R. Chowdari	NUS, Singapore & Nanyang Technological University, Singapore
Hari Mohan Pandey	Edge Hill University, UK
Selwyn Piramuthu	University of Florida, USA
Bharat Bhargava	Purdue University, USA
Omer F. Rana	Cardiff University, UK
Javed I. Khan	Kent State University, USA
Harpreet Singh	Wayne State University, USA
Rajeev Agrawal	North Carolina A&T State University, USA
P. Prabhakaran	St. Joseph University, Tanzania
Yuliya Averyanova	National Aviation University, Ukraine
Mohammed M. Banet	Jordan University of Technology, Jordan
Dawid Zydek	Idaho State University, USA
Wensheng Zhang	Iowa State University, USA
Bal Virdee	London Metropolitan University, UK
Qun Wu	Harbin Institute of Technology, China
Anh V. Dinh	University of Saskatchewan, Canada
Lakshman Tamil	University of Texas, USA
P. D. D. Dominic	Universiti Teknologi Petronas, Malaysia
Zablon Akoko Mbero	University of Botswana, Botswana
V. L. Narasimhan	University of Botswana, Botswana
Kin-Lu Wong	National Sun Yat-sen University, Taiwan
Pawan Lingras	Saint Mary's University, USA
P. G. S. Velmurugan	Thiagarajar College of Engineering, India
N. B. Balamurugan	Thiagarajar College of Engineering, India
Mahesh Bunde	Poornima University, India
N. Venkateswaran	Sri Sivasubramaniya Nadar College of Engineering, India
S. Sundaresh	IEEE Madras Section, India
Premanand V. Chandramani	SSN College of Engineering, India
Mini Vasudevan	Ericsson India Pvt. Ltd., India
P. Swarnalatha	VIT, India
P. Venkatesh	Thiagarajar College of Engineering, India
B. Venkatalakshmi	Velammal Engineering College, India
M. Marsalin Beno	St. Xavier's Catholic College of Engineering, India
M. Arun	VIT, India

Porkumaran K.	NGP Institute of Technology, India
D. Ezhilarasi	NIT Tiruchirappalli, India
Ramya Vijay	SASTRA University, India
S. Rajaram	Thiagarajar College of Engineering, India
B. Yogameena	Thiagarajar College of Engineering, India
S. Joseph Gladwin	SSN College of Engineering, India
D. Nirmal	Karunya University, India
N. Mohankumar	SKP Institute of Technology, India
A. Jawahar	SSN College of Engineering, India
K. Dhayalini	K. Ramakrishnan College of Engineering, India
Diganta Sengupta	Meghnad Saha Institute of Technology, India
Supriya Chakraborty	Amity University, India
Mamta Arora	Manav Rachna University, India
Om Prakash Jena	Ravenshaw University, India
Sandeep Singh Sengar	University of Copenhagen, Denmark
Murali Chemuturi	Chemuturi Consultants, India
Madhu Vadlamani	Cognizant, India
A. N. K. Prasannanjaneyulu	Institute of Insurance and Risk Management, India
O. Obulesu	G. Narayanamma Institute of Technology & Science, India
Rajendra R. Patil	GSSSIETW, India
Ajay Kumar	Chitkara University Institute of Engineering & Technology, India
D. P. Kothari	THDC Institute of Hydropower Engineering and Technology, India
T. S. N. Murthy	JNTUK Vizianagaram, India
Nitesh Tarbani	Sipna College of Engineering & Technology, India
Jesna Mohan	Mar Baselios College of Engineering and Technology, India
Manoj K. Patel	CSIR, India
Pravati Swain	NIT Goa, India
Manoj Kumar	University of Petroleum and Energy Studies, India
E. S. Gopi	National Institute of Technology Tiruchirappalli, India
Mithun B. Patil	NKOCET, India
Priya Saha	LPU, India
Sahaj Saxena	Thapar Institute of Engineering and Technology, India
Dinesh G. Harkut	Prof Ram Meghe College of Engineering & Management, India
Pushpendra Singh	National Institute of Technology Hamirpur, India

Nirmala J. Saunshimath	Nitte Meenakshi Institute of Technology, India
Mayank Pandey	MNNIT, India
Sudeep D. Thepade	Pimpri Chinchwad College of Engineering, India
Pimal Khanpara	Nirma University, India
Rohit Lalwani	MIT University of Meghalaya, India
Loshma Guniseti	Sri Vasavi Engineering College, India
Vishweshwar Kallimani	University of Nottingham, UK
Amit Kumar Mishra	DIT University, India
Pawan Whig	Vivekananda Institute of Professional Studies, India
Dhatri Pandya	Sarvajanik College of Engineering and Technology, India
Asha S. Manek	RV Institute of Technology and Management, India
Lingala Thirupathi	Methodist College of Engineering & Technology, India
P. Mahanti	University of New Brunswick, Canada
Basanta Joshi	Tribhuvan University, Nepal
S. R. N. Reddy	IGDTUW, India
Mehran Alidoost Nia	University of Tehran, Iran
Ambili P. S.	Saintgits Group of Institutions, India
M. A. Jabbar	Vardhaman College of Engineering, India
Lokendra Kumar Tiwari	Ewing Christian College, India
Abhay Saxena	Dev Sanskriti Vishwavidyalaya, India
Kanika Bansal	Chitkara University, India
Pooja M. R.	Vidyavardhaka College of Engineering, India
Pranav Dass	Bharati Vidyapeeth's College of Engineering, India
Avani R. Vasant	Babaria Institute of Technology, India
Bhanu Prasad	Florida A&M University, USA
Barenya Bikash Hazarika	NIT Arunachal Pradesh, India
Ipseeta Nanda	Gopal Narayan Singh University, India
Satyendra Singh	Bhartiya Skill Development University, India
Sudip Mandal	Jalpaiguri Govt. Engineering College, India
Naveen Kumar	IIIT Vadodara, India
Parag Rughani	National Forensic Sciences University, India
K. Shirin Bhanu	Sri Vasavi Engineering College, India
R. Malmathanraj	NITT, India
Latika Singh	Ansal University, India
Gizachew Hailegebriel Mako	Ethio telecom, Ethiopia
Tessy Mathew	Mar Baselios College of Engineering and Technology, India

Grzegorz Chodak	Wrocław University of Science and Technology, Poland
Neetu Verma	D.C.R.U.S.T Murthal, India
Sharda A. Chhabria	G H Raison Institute of Engineering & Technology, India
Neetesh Saxena	Cardiff University, UK
R. Venkatesan	Ministry of Earth Sciences, India
V. Jayaprakasan	IEEE Madras Section, India
D. Venkata Vara Prasad	SSN College of Engineering, India
Jayakumari J.	Mar Baselios College of Engineering and Technology, India
P. A. Manoharan	IEEE Madras Section, India
S. Salivahanan	IEEE Madras Section, India
P. Santhi Thilagam	National Institute of Technology Karnataka, India
Umapada Pal	Indian Statistical Institute, India
S. Suresh	NIT Trichy, India
V. Mariappan	NIT Trichy, India
T. Senthil Kumar	Anna University, India
S. Chandramohan	JNTUA College of Engineering, India
D. Devaraj	Kalasalingam Academy of Research & Education, India
J. William	Agnel Institute of Technology & Design, India
R. Kalidoss	SSN College of Engineering, India
R. K. Mugelan	Vellore Institute of Technology, India
V. Vinod Kumar	Government College of Engineering Kannur, India
R. Saravanan	VIT, India
S. Sheik Aalam	iSENSE Intelligence Solutions, India
E. Srinivasan	Pondicherry Engineering College, India
B. Surendiran	National Institute of Technology Puducherry, India
Varun P. Gopi	NIT Tiruchirappalli, India
V. Vijaya Chamundeeswari	Velammal Engineering College, India
T. Prabhakar	GMRIT, India
V. Kamakoti	IIT Madras, India
N. Janakiraman	KLN College of Engineering, India
V. Anandakrishanan	NIT Trichy, India
R. B. Patel	MMEC, India
Adesh Kumar Sharma	NDRI, India
Gunamani Jena	JNTU, India
Maninder Singh	Thapar University, India
Manoj Manuja	NIT Trichy, India

Ajay K. Sharma	Chitkara University, India
Manjit Patterh	Punjabi University, India
L. M. Bhardwaj	Amity University, India
Parvinder Singh	DCRUST, India
M. Syamala	Punjab University, India
Lalit Awasthi	NIT Jalandhar, India
Ajay Bansal	NIT Jalandhar, India
Ravi Aggarwal	Adobe Systems, USA
Sigurd Meldal	San José State University, USA
M. Balakrishnan	IIT Madras, India
Malay Pakhira	KGEC, India
Savita Gupta	PU Chandigarh, India
Manas Ranjan Patra	Berhampur University, India
Sukhwinder Singh	PU Chandigarh, India
Dharmendra Kumar	GJUST, India
Rajinder Nath	Kurukshetra University, India
Manjaiah D. H.	Mangalore University, India
Himanshu Aggarwal	Punjabi University, India
R. S. Kaler	Thapar University, India
Pabitra Pal Choudhury	Indian Statistical Institute, India
S. K. Pal	DRDO, India
G. S. Lehal	Punjabi University, India
Rajkumar Kannan	Bishop Heber College, India
Yogesh Chaba	GJUST, India
Sh. Sriram Birudavolu	Oracle India Limited, India
Ajay Rana	Amity University, India
Kanwal Jeet Singh	Punjabi University, India
C. K. Bhensdadia	DD University, India
Savina Bansal	GZSCET, India
Mohammad Asger	BGSB, India
Rajesh Bhatia	PEC, India
Stephen John Turner	VISTEC, India
Chiranjeev Kumar	IIT (ISM) Dhanbad, India
Bhim Singh	IIT Delhi, India
A. K. Sharma	BSAITM, India
Rob Reilly	MIT, USA
A. K. Murthy	CDAC, India
Karmeshu	JNU, India
Sandeep Sen	IIT Delhi, India
Suneeta Aggarwal	MNNIT, India
Raghuraj Singh	HBTI, India
D. K. Lobiya	JNU, India

R. S. Yadav	MNNIT, India
Bulusu Anand	IIT Roorkee, India
R. K. Singh	KEC Dwarahat, India
Sateesh Kumar Peddoju	IIT Roorkee, India
Divakar Yadav	JIIT, India
Naveen Kumar Singh	IGNOU, India
R. S. Raw	AIACR (NSUT East Campus), India
Vidushi Sharma	GBU, India
Sumit Srivastava	Manipal University, India
P. K. Saxena	DRDO, India
B. K. Das	ITM University, India
Y. Raghu Reddy	IIIT Hyderabad, India
B. Chandra	IIT Delhi, India
R. K. Agarwal	JNU, India
Basim Alhadidi	Al-Balqa' Applied University, Jordan
Emmanuel Ndashimye	University of Rwanda & CMU-Africa, Rwanda
Naveen Garg	IIT Jodhpur, India
K. S. Subramanian	IGNOU, India
Biplab Sikdar	NUS, Singapore
Sreeram Ramakrishna	NUS, Singapore
Vikas Mathur	Citrix, India
Hari Krishna Garg	NUS, Singapore
Raja Dutta	IIT Kharagpur, India
Vishakha Vaidya	Adobe, India
Sudipto Shankar Dasgupta	Infosys Limited, India
Atal Chaudhari	Jadavpur University, India
Gangaboraiah Andanaiah	KIMS, India
Champa H. N.	UVCE, India
Ramakanth Kumar P.	RVCE, India
S. N. Omkar	IISC Bangalore, India
Balaji Rajendran	CDAC, India
Annapoorna P. Patil	MSRIT, India
K. N. Chandrashekhar	SJCIT, India
Mohammed Misbahuddin	CDAC, India
Saroj Meher	ISI, India
Jharna Majumdar	NMIT, India
N. K. Cauvery	RVCE, India
G. K. Patra	CSIR, India
Anandi Jayadharmanarajan	Oxford College of Engg., India
K. R. Suneetha	BIT Mesra, India
M. L. Shailaja	AIT, India
Ramesh Paturi	Microsoft, India

S. Viswanadha Raju	JNTU, India
C. Krishna Mohan	IIT Chennai, India
R. T. Goswamy	Techno International New Town, India
B. Surekha	K S Institute of Technology, India
P. Trinatha Rao	GITAM University, India
G. Varaprasad	BMS College of Engineering, India
M. Usha Rani	SPMVV, India
P. V. Lakshmi	SPMVV, India
K. A. Selvaradjou	PEC, India
Ch. Satyananda Reddy	Andhra University, India
Jeegar A. Trivedi	Sardar Patel University, India
S. V. Rao	IIT Guwahati, India
Suresh Varma	Adikavi Nannaya University, India
T. Ranga Babu	RVR & JC College of Engineering, India
D. Venkat Rao	Narasaraopet Inst. of Technology, India
N. Sudhakar Reddy	S V Engineering College, India
Dhiraj Sunehra	Jawaharlal Nehru Technological University, India
Madhavi Gudavalli	JNTU Kakinada, India
B. Hemanth Kumar	RVR & JC College of Engineering, India
A. Sri Nagesh	RVR & JC College of Engg., India
Bipin Bihari Jaya Singh	CVR College of Engg., India
M. Ramesh	JNTU, India
P. Rajarajeswari	GITAM University, India
R. Kiran Kumar	Krishna University, India
D. Ramesh	JNTU, India
B. Kranthi Kiran	JNTU, India
K. Usha Rani	SPM University, India
A. Nagesh	MGIT, India
P. Sammulal	JNTU, India
G. Narasimha	JNTU, India
B. V. Ram Naresh Yadav	JNTU, India
B. N. Bhandari	JNTUH, India
O. B. V. Ramanaiah	JNTUH College of Engineering, India
Anil Kumar Vuppala	IIIT Hyderabad, India
Duggirala Srinivasa Rao	JNTU, India
Makkena Madhavi Latha	JNTUH, India
Anitha Sheela Kancharla	JNTUH, India
B. Padmaja Rani	JNTUH College of Engineering Hyderabad, India
S. Mangai	Velalar College of Engg. & Tech., India
P. Chandra Sekhar	Osmania University, India
Chakraborty Mrityunjy	IIT Kharagpur, India
Manish Shrivastava	IIIT Hyderabad, India

Uttam Kumar Roy	Jadavpur University, India
Kalpana Naidu	IIIT Kota, India
A. Swarnalatha	St. Joseph's College of Engg., India
Aaditya Maheshwari	Techno India NJR Institute of Tech., India
Ajit Panda	National Institute of Science and Technology, India
R. Anuradha	Sri Ramakrishna Engg. College, India
B. G. Prasad	BMS College of Engg., India
D. Murali	VIT, India
Deepak Padmanabhan	Queen's University Belfast, UK
Firoz Alam	RMIT University, Australia
Frederic Andres	NII, Japan
Srinath Doss	Botho University, Botswana
Munish Kumar	Maharaja Ranjit Singh Punjab Tech University, India
Norwati Mustapha	UPM, India
Hamidah Ibrahim	UPM, India
Denis Reilly	Liverpool John Moores University, UK
Ioannis Kypriaios	De Montfort University, UK
Yongkang Xing	De Montfort University, UK
P. Shivakumara	University of Malaya, Malaysia
Ravinder Kumar	TIET Patiala, India
Ankur Gupta	Rishihood University, India
Rahul Kr. Verma	IIIT Lucknow, India
Mohit Sajwan	NSUT, India
Vijaypal Singh Rathor	IIITDM, India
Deepak Singh	NIT Raipur, India
Simranjit Singh	NIT Jalandhar, India
Suchi Kumari	Shiv Nadar University, India
Kuldeep Chaurasia	Bennett University, India
Indrajeet Gupta	SR University, India
Shakti Sharma	Bennett University, India
Hiren Thakkar	PDPU, India
Mayank Swankar	IIT (BHU) Varanasi, India
Tapas Badal	Bennett University, India
Vipul Kr. Mishra	Gatishakti University, India
Tanveer Ahmed	Bennett University, India
Madhushi Verma	Bennett University, India
Gaurav Singal	NSUT, India
Anurag Goswami	Bennett University, India
Durgesh Kumar Mishra	Sri Aurobindo Institute of Technology, India

S. Padma	Madanapalle Institute of Technology & Science, India
Deepak Prashar	Lovely Professional University, India
Nidhi Khare	NMIMS, India
Sandeep Kumar	IIT Delhi, India
Dattatraya V. Kodavade	D.K.T.E Society's Textile & Engineering Institute, India
A. Obulesu	Anurag University, India
K. Suvarna Vani	V R Siddhartha Engineering College, India
G. Singaravel	K.S.R. College of Engineering, India
Ajay Shiv Sharma	Royal Melbourne Institute of Technology, Australia
Abhishek Shukla	R.D. Engineering College Technical Campus, India
V. K. Jain	Teerthanker Mahaveer University, India
Deepak Poola	IBM India Private Limited, India
Bhadri Raju M. S. V. S.	S.R.K.R. Engineering College, India
Yamuna Prasad	IIT Jammu, India
Vishnu Vardhan B.	JNTUH College of Engineering Manthani, India
Virendra Kumar Bhavsar	Univ. of New Brunswick, Canada
Siva S. Skandha	CMR College of Engineering, India
Vaibhav Anu	Montclair State University, USA
V. Gomathi	National Engineering College, India
Sudipta Roy	Assam University, India
Srabanti Maji	DIT University, India
Shylaja S. S.	PESU, India
Shweta Agrawal	SIRT, India
Shreenivas Londhe	Vishwakarma Institute of Information Technology, India
Shirin Bhanu Koduri	Vasavi Engineering College, India
Shailendra Aswale	SRIEIT, India
Shachi Natu	TSE College Mumbai, India
Santosh Saraf	Graphic Era Deemed to be University, India
Samayveer Singh	Ambedkar National Institute of Technology, India
Sabu M. Thampi	IIIT and Mgt-Kerala Thiruvananthapuram, India
Roshani Raut	Vishwakarma Institute of Information Technology, India
Radhika K. R.	BMSCE, India
R. Priya Vijayanthi	NSRIT, India
M. Naresh Babu	NIT Silchar, India
Krishnan Rangarajan	Dayananda Sagar College of Engineering, India
Prashant Singh Rana	Thapar Institute of Engg. & Tech., India

Parteek Bhatia	Washington State University, USA
Laxmi Lydia	VIIT, India
Nikunj Tahilramani	Dolcera IT Services Pvt. Ltd., India
Navanath Saharia	IIIT Manipur, India
Nagesh Vadaparthi	MVGR College of Engineering, India
Manne Suneetha	VR Siddhartha Engineering College, India
Sumalatha Lingamgunta	JNTU Kakinada, India
K. Subramanian	IIT Kanpur, India
Singaraju Jyothi	Sri Padmavati Mahila Visvavidyalayam, India
Vinit Jakhetiya	IIT Jammu, India
Yashwantsinh Jadeja	Marwadi University, India
Harsh Dev	PSIT, India
Yashodhara V. Haribhakta	Government College of Engineering Pune, India
Gopal Sakarkar	GHRCE, India
R. Gnanadass	Pondicherry Engineering College, India
K. Giri Babu	VVIT, India
Geeta Sikka	B R Ambedkar National Institute of Technology, India
Gaurav Varshney	IIT Jammu, India
G. L. Prajapati	Devi Ahilya University, India
G. Kishor Kumar	RGM CET, India
Krishna Kiran Vamsi Dasu	Sri Sathya Sai Institute, India
Sisira Kumar Kapat	Utkal Gaurav Madhusudan Institute of Technology, India
Kuldeep Sharma	Chitkara University, India
Zankhana H. Shah	BVM Engineering College, India
Rekha Ramesh	Shah and Anchor Kutchhi Engineering College, India
Gopalkrishna Joshi	MIT Vishwapravayag University, India
Ganga Holi	AMC Engineering College, India
K. Kotecha	Symbiosis International, India
Radhakrishna Bhat	MAHE, India
Kuldeep Singh	Carnegie Mellon University, USA
Binod Kumar	JSPM's Rajarshi Shahu College of Engineering, India
Raju Kumar	Chandigarh University, India
Nitin S. Goje	Webster University in Tashkent, Uzbekistan
Pushpa Mala S.	Dayananda Sagar University, India
Ashish Sharma	GLA University, India
Ashwath Rao B.	Manipal Institute of Technology, India
Deepak Motwani	Amity University, India
V. Sowmya	Amrita School of Engineering, India

Jayashri Nair	VNR VJIET, India
Rajesh C. Sanghvi	G.H. Patel College of Engineering & Technology, India
Ashwin Dobariya	Marwadi University, India
Tapas Kumar Patra	CET Bhubaneswar, India
J. Naren	Rathinam College of Arts and Science, India
Rekha. K. S.	National Institute of Engineering, India
Mohammed Murtuza Qureshi	Digital Employment Exchange, India
Vasanth Kalyani David	Avinashilingam Institute for Home Science and Higher Education for Women, India
K. Sakthidasan	Hindustan Institute of Technology and Science, India
Shreyas Rao	Sahyadri College of Engineering and Management, India
Hiranmayi Ranganathan	Lawrence Livermore National Laboratory, USA
Sanjaya Kumar Panda	National Institute of Technology Warangal, India
Puspanjali Mohapatra	IIIT Bhubaneswar, India
Manimala Mahato	Shah & Anchor Kutchhi Engineering College, India
B. Senthil Kumar	Kumaraguru College of Technology Coimbatore, India
Jyoti Prakash Singh	National Institute of Technology Patna, India
Abhinav Tomar	Netaji Subhas University of Technology, India
M. G. Sumithra	Dr. N.G.P. Institute of Technology, India

Contents

Advancements in Artificial Intelligence and Machine Learning

Extractive Text Summarization Employing a Graph-Based Ranking System with Context Transfer	3
<i>Virat Giri, M. M. Math, Sheetal Kusal, and Shruti Patil</i>	
ViT-GPT2: Vision Transformer Based Automatic Image Captioning	20
<i>Chanda Srija, B. Padmaja, Yennum Smrithi, and Metla Karthikeya</i>	
Multimodal Language Models for End-to-End Automated Speech Recognition Using Bidirectional Recurrent Neural Network	33
<i>Thayyaba Khatoon Mohammad, Puranam Revanth Kumar, Gifta Jerith, and E. Krishnaveni Reddy</i>	
Advanced Deep-Learning Framework for Comprehensive Audio Forensics: Classification and Frequency Analysis	48
<i>Swetanshu Agrawal, Astitva Veer Garg, and D. Anitha</i>	
Enhancing Deepfake Detection Accuracy with Vision Transformers: A Generative AI Approach	68
<i>Anugrah Srivastava, Tapas Badal, Mohit Agarwal, Karnika Dwivedi, and Advait Hemant Naik</i>	

Applications of Large Language Models

Large Language Models for Quantitative Analysis of Statutory Stringency and Its Correlation with Sexual Assault Crime Rates	81
<i>Samhita Konduri and Vishnu S. Pendyala</i>	
Application of Large Language Models for Search Engine Optimisation: Methodology for Assessing the Efficiency of Content Creation	95
<i>Klaudia Błażyczek and Grzegorz Chodak</i>	
Real-Time Query Management with FAQ Generation for Large Scale Online Classes	108
<i>Kamal Bijlani, Sreya Sunil, Navabhaarithi, S. P. Saran Dharshan, E. Vigneshwar, and A. Aadharsh Aadhithya</i>	

Enhancing Warehouse Operations with Voice-Activated SQL Streamlining
Using LLM Technology 121
*V. Ramani Bai, Suryajit Ithiparambil Sudheeran, Abhinand K. Prasad,
and Abhinav Raj*

AI in Healthcare and Medical Imaging

Enhancing Medicinal Plant Prediction Through Deep Neural Network
Algorithms 139
M. Swapna, C. Sireesha, Sanjana Gavada, and Barigela Sreeja

A Comprehensive Review of Medical Image Captioning: Techniques,
Challenges, and Future Directions 150
*Mallemputi Sai Teja, Chinta Vijaya Pandu Ranga Reddy, U. Muthaiah,
and B. S. Vidhyasagar*

An Explainable Artificial Intelligence Based Approach for Prediction
and Classification of Multiple Sclerosis Disease: A Mental Health Disorder 168
*Vaishali Mehta, Jyoti Garg, Dolly Sharma, Saurabh Yadav,
and Ravi Kumar*

Multimodal Image Prior Integration for Unsupervised MRI
Super-Resolution with Guided Residual Dense Networks 181
Siddhartha Choubey, Abha Choubey, Dipti Jaiswal, and Manuraj Jaiswal

AI for Disaster Management and Renewable Energy

AID-SAR: Aerial Image Dataset for Search and Rescue in the Aftermath
of Disasters 197
Arya Itkyal, Gargi Joshi, Rahee Walambe, and Ketan Kotecha

Solar PV Power Generation Forecasting Using Synergy of Artificial
Intelligence and Metaheuristics Based Approaches 210
Pankaj Yadav, Gulshan Sihag, Vikas Pratap Singh, and Vivek Vijay

Challenges and Ethical Considerations in AI

Unraveling the Enigma of Classification of Synthetic and Genuine
Information Using Machine Learning and Explainable AI 229
Vishnu S. Pendyala

Challenges and Limitations of TVAE Tabular Synthetic Data Generator 243
A. Kiran, P. Rubini, and S. Saravana Kumar

Author Index 243

Advancements in Artificial Intelligence and Machine Learning



Extractive Text Summarization Employing a Graph-Based Ranking System with Context Transfer

Virat Giri¹ , M.M. Math², Sheetal Kusal³  , and Shruti Patil³  

¹ Computer Science, Sanjay Ghodawat Institute, Kolhapur, India

² Computer Science, Gogate Institute of Technology, Belgaum, India
mmmmath@git.edu

³ Artificial Intelligence and Machine Learning, Symbiosis Institute of Technology, Pune, India
{sheetal.borhade,shruti.patil}@sitpune.edu.in

Abstract. Text summarization has been an area of active research in the domain of natural language processing. Extractive text summarization focuses on highlighting the most essential parts of a document as its summary without any external additions to the original text. Conventional methods have focused on using sentence ranking algorithms like text rank, which uses position and similarity-based features to create graphs. However, the recent advent of deep learning, transfer learning, and context inclusion have been relatively undiscovered in this domain. This work proposes a graph-based ranking mechanism that uses relevant context to solve the task of extractive text summarization. Language modeling generates word representations that can provide a more comprehensive graph representation and achieve better results. The presented approach relies on graph-based ranking and the relevance and interpretation of each word in the document. The work is carried out on documents in the Marathi language with a standard-sized dataset. The proposed model has achieved a 75% F1 score and a Rouge L F1 score of 71%. The results obtained by the proposed system outperform those of state-of-the-art approaches. The system's effectiveness is proved using the multiple performance metrics relevant to the text summarization task.

Keywords: Extractive Text summarization · Natural Language Processing · Deep Learning · Graph based ranking · Transfer Learning · Pretrained Models

1 Introduction

Text is the most prevalent form of data available online. It is mainly in the form of books, news, blogs, etc. Manual summarization of such a large amount of data lacks feasibility due to the costly and time-consuming nature of the

process. Because of this, automatic text summarization has become an absolute necessity. It is the process of forming a concise version of a large textual document by extracting the important chunks of the document [1]. Text summarization is a process that involves reducing long text pieces into shorter ones to create a more coherent and fluent summary. It is commonly used in ML and NLP processing to improve the readability of a document. The goal of automatic text summarization is to reduce lengthy documents into shorter versions, which can be very costly to carry out manually. Based on the way summary is generated, kinds of input sources, purpose of text summarization, language of input sources, methods and genre, there are various taxonomies that can be considered for text summarization are shown in Fig. 1. From that extractive and abstractive summary are the ways summary generated from input. The Extractive summary method is about extracting the significant and important sentences of source documents which are deemed as informative. Figure 2 shows the process of extractive text summarization. The high scoring sentences are located by assigning scores to different sentences and the high scoring sentences are used as summary. In extractive summary one can find the exact sentences that appear in the source document. Abstractive summary is like a human summarizing a document who has basic domain knowledge. The system recognizes the text by Natural Language Processing (NLP) and paraphrases the sentences for the purpose of summarizing the text. Figure 3 shows the process of abstractive text summarization. Abstractive summarization is thus a challenging task due to the complexity of NLP. Given the availability of extensive information and processed summarized datasets in conventional languages like English, The abundance of data in these languages makes them ideal for developing automatic

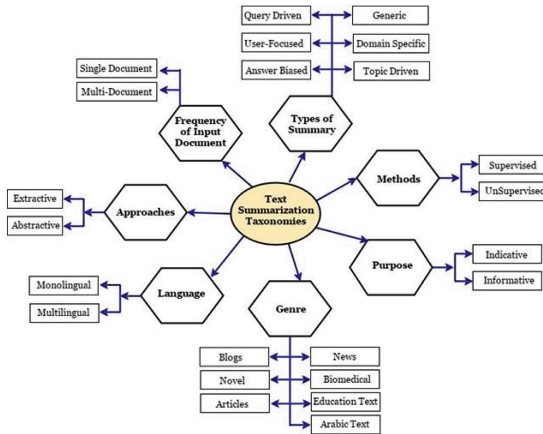


Fig. 1. Taxonomies of Text Summarization

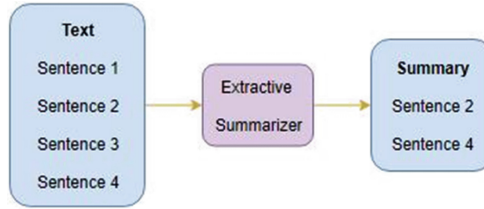


Fig. 2. Extractive Text Summarization

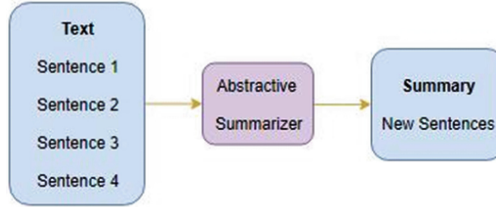


Fig. 3. Abstractive Text Summarization

summarizers. But this is not true with Indic languages like Marathi, Hindi, or other languages. Therefore, traditional deep learning-based approaches used for English summarization cannot be readily used for such languages due to the need for large processed data to be retrained in such languages. Furthermore, standard preprocessing techniques like lemmatization and tokenization depend on the language grammar. Thus, these techniques also need to be restructured for such languages. Traditional approaches for Indic language-based summarization use cosine similarity between sentences to determine the importance of the sentences. But the cosine similarity method is unable to capture semantic similarity. Some approaches use positional tagging or other natural language processing-centric features to extract the significance of the sentence. Although these techniques are effective, they lack context and semantic analysis for such tasks. There are some architectures proposed that use recurring neural network (RNN) architecture, but rather than working directly on the Languages, they convert it to English for summarization and convert the summary back into the original language. In this paper, the use of unsupervised Marathi context-based word embeddings has been presented as an alternative to the traditional word occurrence vector. The authors then utilize the degree of similarity between these context vectors to rank the phrases. This approach performs superiorly to traditional approaches and provides more human-like summarization than previous ones. Clearly, contributions through this paper are stated as follows:

- Generate summaries using semantic and context vectors instead of occurrence-based vectors.
- The use of transfer learning as an intermediate step rather than using it for an end-to-end system. Thus, it benefits from minimal learning complexity compared to standard deep learning-based summarization approaches.
- The Language Model Fine-tuning approach is used to extract context embeddings.

The remainder of the paper is structured as follows. Section 2 provides information about background work in this domain. Section 3 elaborates on the proposed methodology with experimental setup. The results obtained are discussed in Sect. 4. And the paper concludes in Sect. 5.

2 Background and Related Work

Automatic text summarization is an important area of research in the domain of natural language processing [2](NLP). Text summarization has become prominent in NLP mainly due to deep learning techniques. Figure 4 shows the classification of text summarization. The algorithms in this area are categorized into two types. Firstly, an extractive summarization based algorithm [1] which uses words from the source text to create the summary. The attention mechanism has seen an evident rise in applications in sentiment analysis [3], opinion mining [4], and text summarization tasks. Derek Miller [5] fine-tuned transformer-based BERT architecture for extractive text summarization. Secondly, an abstractive summarization-based approach uses words that are not limited to the words present in the source text - Nallapati et al. [6] leveraged the Sequence to Sequence RNN architecture to develop an abstractive text summarizer. Li et al. [7] developed a multi-attention-based architecture for abstractive summarization on the Gigawords dataset. Although effective, these approaches have only been leveraged in the English language. In recent times, Indic language summarization approaches have been undergoing a lot of research. Table 1 shows summary of literature studied.

Giri et al. [8] presented a survey on the various techniques available for summarization for Indic languages. Sarwadnya et al. [9] leveraged the graph-based text rank approach for Marathi text summarization, and it used the similarity between the occurrence of words in sentences to determine the importance of the sentence. Patil et al. [10] developed a rule-based Part-of-Speech tagger for Marathi Language. Bhosale et al. [11] used the frequency count of words in Marathi e-newspaper to determine the text summary. Bhoir et al. [12] proposed using fuzzy logic for Multi-document Marathi text summarization. Dalal et al. [13] used the concept of Semantic Graph summarization combined with Particle Swarm Optimization for Hindi text summarization. Kumar et al. [14] used the occurrence of the radix of thematic words as a measure to perform extractive

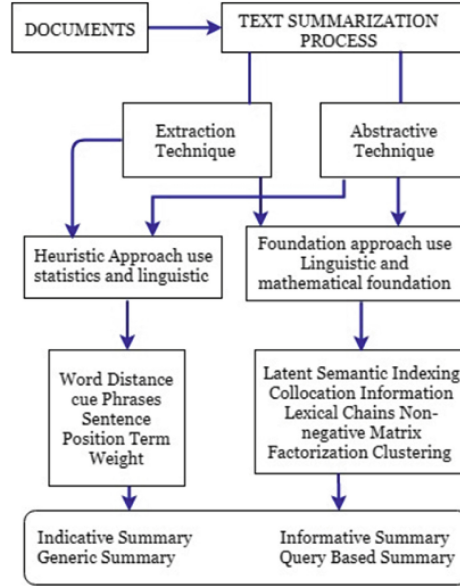


Fig. 4. Various Classification approaches of Text Summarization

summarization on Hindi documents. Bhargava et al. [15] used Generative Adversarial Networks for paraphrase detection for Indic Languages. It can be seen that the previously proposed Indic summarizers use mostly an occurrence-based similarity approach for text summarization. It lacks the context and semantic-based understanding that is observed in English Summarizers. The only approach that shows the presence of such understanding was the work proposed by Chaudhari et al. [16], who leveraged the use of the Encoder-Decoder model for Marathi summarization by converting the text to English and then converting the summary back to Marathi. Though effective, it could not summarize the jargon and grammar present in Marathi. So, it shows ample scope for improvement in the area of Marathi text summarization.

3 Proposed Methodology

The proposed architecture combines deep learning and a traditional graph-based approach to tackle the problem of understanding the context in Marathi summarization. The block diagram of our proposed architecture is shown in Fig. 5. The further intricate details are divided into sections for a more fundamental understanding of the architecture:

Table 1. Literature Surveyed

Ref	Dataset	Description	Technique	Task
[3]	Code-mixed Hindi dataset	Detection of cyber abuse based on Multilingual	Deep learning - BERT model	Cyber abuse detection
[4]	toxic comment dataset	Multi-label profanity detection	Attention based ensemble model	Toxicity detection
[5]	MOOC lectures	Extractive Text summarization of lectures notes	Deep learning - BERT model with k means clustering	Extractive Text Summarization
[6]	DUC Corpus, 3 CNN/Daily Mail Corpus	Multi sentence text summarization	Attention based Encoder-decoder model	Abstractive Text Summarization
[7]	DUC Corpus	Multi-document compressed summarization	A cascaded neural attention based unsupervised salience estimation method	Compressive Text Summarization
[9]	EMILLE (Enabling Minority Language Engineering)	Marathi text summarizer	Graph based models - text rank algorithm	Extractive Text Summarization
[10]	Self-curated Marathi dataset	POST of Marathi language	Rule based method, morphological analysis	Part of speech tagging
[11]	newspaper web site	Marathi text summarizer	Frequency count based model	Keyword based text summarizer
[12]	Self-curated Hindi dataset	Multi-document Hindi text summarization	Fuzzy Logic model	Extractive text summarization
[13]	Hindi dataset	Hindi text summarizer	Graph based model using Particle swarm optimization	Automatic text summarizer
[14]	Hindi dataset	Hindi text summarizer	Thematic Approach	Extractive text summarization
[15]	Multiling 2015 Dataset, Opinosis Dataset	Text summarization using paraphrase detection	Deep learning Generative Adversarial networks	Abstractive multilingual text summarization
[17]	Marathi Dataset	Marathi Text summarization	Latent Semantic Analysis	Extractive text summarization

3.1 Preprocessing

The source text passes through the following preprocessing pipeline before it is converted into a model-friendly numerical format-

- Miscellaneous Symbol Remover- The source text is removed from punctuation marks, hyperlinks, and other symbols that are not useful for the summarization.
- Tokenization- The processed text is then converted into a set of word tokens.
- Stop Word removal - The tokens were filtered based on a custom set of repetitive stop words in the text.
- Lemmatization - The filtered tokens were then converted into their basic form by stripping affixes through the designed Marathi-centric rules.

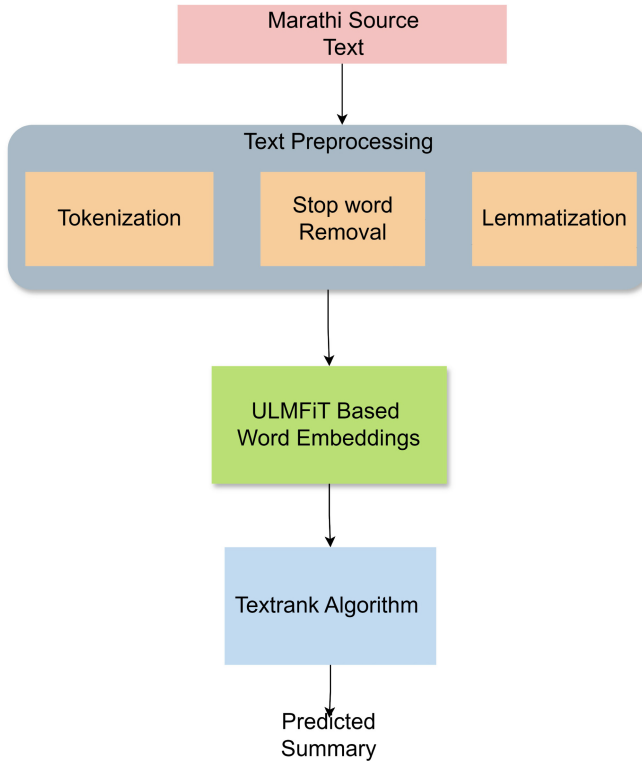


Fig. 5. Proposed pipeline for Extractive summarization

Table 2 shows an example of a Marathi textual output after passing through the preprocessing pipeline.

Table 2. Preprocessed Output

Source Text
परेश तु देशाचे हे उपकार फेडणे, आवश्यक आहेच!
Preprocessed Text
परेश तु देश उपकार फेड आवश्यक आहे

3.2 Word Embeddings

The concept of transfer learning is used to fine-tune the embeddings from a universally trained language model. This presented approach derives inspiration from the Universal Language Modelling Fine-tuning [16, 18] algorithm to form the context word embeddings. Our approach involves an inductive transfer learning-based approach which fine-tunes a pretrained Language Model for a specific task. Particularly, the AWD-LSTM [19] model is used to apply this technique and extract the word embedding for further processing. The three steps performed for extracting the final word embeddings are as follows -

1. Pretraining of the Language Model- The model was pretrained on the Marathi Wikipedia corpus comprising 85,000 articles. It allows the model to understand the general trend followed by the language. Though this is a costly processing step, it must be performed once.
2. Fine-tuning the model- This step focused on fine-tuning the model for classifying the text based on the presence of the Marathi Language. The main focus of this step is to train the model for understanding the task better. The technique proposed two methods to increase the effectiveness of this step -
 - Discriminative Fine-tuning- Different extract layers have different features. Therefore, different learning rates for different layers were used in the model. Furthermore, the update rule followed by the SGD was also modified to the Eq. 1 and 2 stated below- The weights of layer l at iteration t are updated by subtracting the product of:
 - The learning rate for layer l (η^l).
 - The gradient of the error function ($\nabla_{\theta^l} J(\theta)$).

$$\theta_t^l = \theta_{t-1}^l - \Delta\theta^l \quad (1)$$

where:

$$\Delta\theta^l = \eta^l \cdot \nabla_{\theta^l} J(\theta) \quad (2)$$

Here, $\Delta\theta^l$ represents the adjustment to the weights for layer l during this iteration.

Where θ = the weight,

t = iteration

l = layer

$\nabla_{\theta^l} J(\theta)$ = error gradient

η = learning rate and

J = error function.

- Slanted Triangular Learning Rates - The use of a decaying learning rate while the learning progresses. The learning-rate update equations are as follows by Eqs. 3, 4 and 5
The cutoff iteration (cut) is defined as:

$$cut = T \cdot cut_{frac} \quad (3)$$

where:

- T : Total number of iterations.
- cut_{frac} : Fraction of iterations where the learning rate is increased.
- The intermediate value p is defined as:

$$p = \begin{cases} \frac{t}{cut}, & \text{if } t < cut \quad (\text{increasing phase}) \\ 1 - \frac{t-cut}{cut \cdot \left(\frac{1}{cut_{frac}} - 1\right)}, & \text{if } t \geq cut \quad (\text{decreasing phase}) \end{cases} \quad (4)$$

- The learning rate at iteration t is given by:

$$\eta_t = \eta_{max} \cdot \frac{1 + p \cdot (\text{ratio} - 1)}{\text{ratio}} \quad (5)$$

where:

- η_{max} : Maximum learning rate.
 - η_{min} : Minimum learning rate.
 - $\text{ratio} = \frac{\eta_{min}}{\eta_{max}}$: Ratio of minimum to maximum learning rate.
 - T = number of iterations
 - cut_{frac} = the fraction of iterations that the LR is increased, and
 - p = number of iterations that the LR is going to be decreased or has been decreased. Additionally,
 - ratio defines η_{min}/η_{max} , and
 - cut = cutoff iteration where the model switches from an increasing LR to a decreasing LR.
3. Fine-tuning of classifier- A classification network was added to the AWD-LSTM model which was trained from scratch for the target task of classifying the presence of Marathi in text. Additional details about the training of the architecture are as follows-
 - (a) The classification network is fed with the concatenated vector of the last hidden layers and the average and max pooled output of the previously hidden layers.
 - (b) Training the whole architecture is an error-prone and time consuming process as there is a loss of information during fine-tuning of the pre-trained architecture. Therefore, each layer was retrained in a sequential manner using the Gradual unfreezing technique until the whole AWD-LSTM architecture was trained. LSTM will be able to handle long-term context in the document.

- (c) Both forward and backward Language model was trained, and the final prediction was an average of both the predicted output.

The embedding layer of this trained architecture is then used for converting each word into model-friendly numerical embedding. A 400-dimension vector of each word present in the sentence is extracted using the embedding layer and averaged across the whole sentence using the following formula. Figure 6 shows the whole word embeddings extraction architecture.

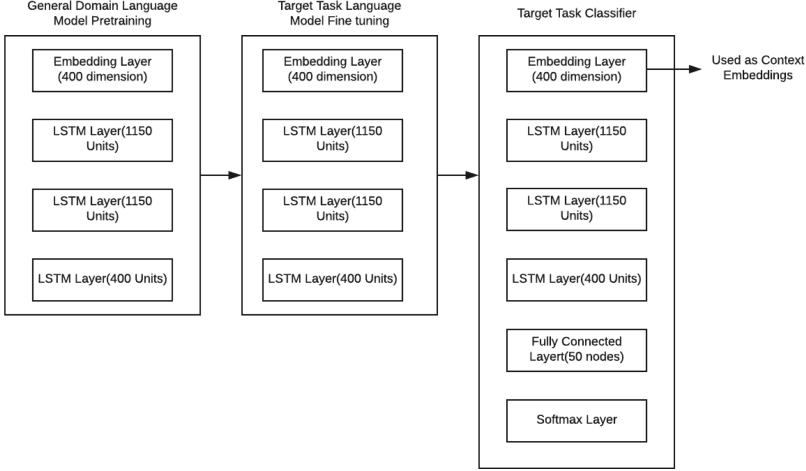


Fig. 6. Word Embeddings Architecture

The details of the individual blocks present in this architecture are as follows:

- **Embedding layer:**

Embedding is a n dimensional representation of a word such that similar words have similar embedding values. In this work, 400 dimensional embedding values are extracted and used. As opposed to the conventional TF-IDF and statistical count-based approaches for vectorization of tokens, embeddings provide more contextual as well as semantic understanding of the words by benefiting from the language structure that it has learned by the initial pre-training.

- **LSTM layer:**

The LSTM i.e. Long short term memory layer relies upon the use of cell state to carry forward the context of memory over longer lengths of the sequence [20]. Every LSTM unit consists of three gates i.e. forget gate, input gate and output gate to modify the existing representation when moving from one unit to another. Equations 6, 7, 8

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (6)$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (7)$$

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \quad (8)$$

where f_t, i_t, o_t indicate the outputs given by the three gates respectively. W and b indicate the weights and bias. h_i indicates the intermediate output for a particular instance i , and is a function of the cell state C .

- Fully connected layer and Softmax layer:

The fully connected layer involve complete connections between nodes of a current layer and the previous and next layers. The values in the fully connected layers are activated using a ReLU function while the Softmax layer produces the task-specific prediction, which would be a class label in case of classification tasks. For the task in consideration, the output is extracted from the final embeddings layer instead of the Softmax layer as it is used further for the extractive text summarization task.

3.3 Graph-Based Approach for Extractive Summarization

The text rank [21] approach is deployed to determine the set of sentences present in the final summary. It is a graph-based unsupervised extractive summarization algorithm which determines the summary based on cosine similarity between sentences. The steps performed for extracting final summary is as follows-

1. The cosine Similarities between the sentence averaged context vectors are then calculated and stored in a matrix. The formula of cosine similarity is as follows-

$$\text{cosine similarity} = \frac{A.B}{||A|| \times ||B||} \quad (9)$$

where A and B are numerical vectors.

2. The matrix is then converted into a graph, in which sentences are the vertices while the similarity scores are the edges.
3. The user-dependent top-ranked n sentences form the final summary from the source text.

4 Results and Discussion

The proposed architecture is evaluated on a custom dataset of 100 Marathi documents that were manually summarized by a human. The dataset consisted of text from various topics. The custom dataset consists of 100 Marathi documents, each summarized manually by human annotators. It encompasses a variety of textual sources, including news items, educational materials, government reports, blog posts, opinion pieces, legal documents, banking reports, and excerpts from Marathi literature, such as poetry, narratives, and essays. The documents vary in length from 100 to 500 words, with summaries comprising 20 to 30% of the original text length. This dataset encompasses a range of writing styles, from formal to narrative, guaranteeing topic diversity and language complexity. It

Table 3. Example of Source Text

बँकांच्या कर्ज थकबाकीचा विषय महत्वाचा असून जर एकूण थकलेली रक्कम ही लाखो कोटीच्या घरात जाणारी आहे तर ती जाहीर करण्यास काय हरकत आहे असे सांगून न्यायालयाने संबंधित सर्व पक्षकारांना या विषयावर मंथन घडेल असे सर्व पैलू पुढे आणण्यास सांगितले.याआधी सर्वोच्च न्यायालयाने रिझर्व्ह बँकेला ५०० कोटी रुपये व त्यापेक्षा अधिक रकमेचे कर्ज थकविणाऱ्या कंपन्या व त्यांच्या थकबाकीची रकमांची यादी देण्यास सांगितले होते. बँकांनी दिलेले लाखो कोटींच्या घरातील कर्ज थकविणाऱ्या विविध व्यक्ती व आस्थापनांची नावे व थकित रकमा जाहीर करण्यात याव्यात असे मत सर्वोच्च न्यायालयाने मंगळवारी व्यक्त केले. रिझर्व्ह बँके ने अलीकडेच सीलबंद पाकिटात कर्ज थकबाकीदार व त्यांच्या थकित कर्जाचे आकडे सर्वोच्च न्यायालयाला सादर केले आहेत..... continue

establishes a solid framework for assessing summarization models, presenting a combination of cultural, educational, and informative content pertinent to the Marathi language. Table 3 shows an example of the text present in the dataset. Due to the unsupervised nature of the proposed architecture, it is evaluated over all the 100 documents present in the dataset. For comparison, we compare our results with the traditional Indic Language summarization approaches of Fuzzy logic [12] which use the values of fuzzy variables calculated for individual sentences for determining the summary, positional text rank approach [9] [21] which uses weighted vector of the sentence extracted using the position of the word in the corpus for summarization. Finally, the standard Text Rank [9] [21] and Singular Value Decomposition [22] approach which uses occurrence vector for extracting summary sentences. The precision, recall and F1 scores of each approach has been stated in Table 4.

Here, precision refers to the number of predicted words originally present in the summary, recall refers to the number of actual words predicted correctly by the summarizer and the F1 score is the harmonic mean of Precision and recall. The formulas of the same have been stated below in Eq. 9, 10, 11 and 12-

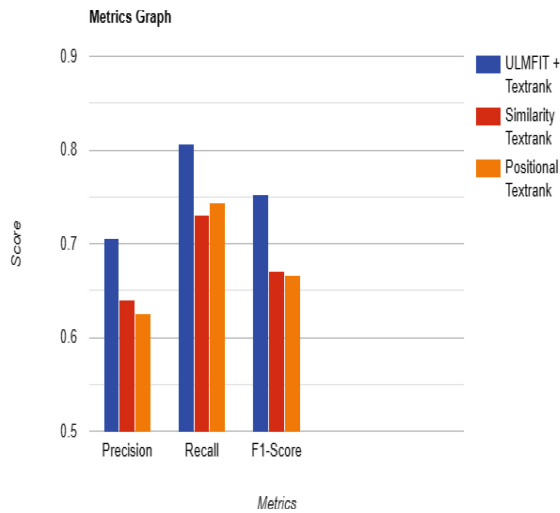
$$Recall = \frac{\text{number of overlapping words}}{\text{total words in reference summary}} \quad (10)$$

$$Precision = \frac{\text{number of overlapping words}}{\text{total words in system summary}} \quad (11)$$

$$F1 \text{ score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (12)$$

Table 4. Performance on traditional metrics for the task of extractive summarization

Approaches	Precision	Recall	F1-Score
Position based Textrank	0.626	0.744	0.667
Similarity-based Textrank	0.64	0.73	0.671
SVD	0.705	0.693	0.682
Fuzzy Logic	0.625	0.655	0.63
Pre-trained LM + Textrank (Proposed)	0.706	0.806	0.753
Improvement over previous best (%)	0.14	8.33	10.41

**Fig. 7.** Metrics Visualization

It can be seen that our proposed approach outperforms the previous approaches by a comfortable margin in terms of all the above-mentioned metrics. Figure 7 shows the graphical visualization of these metrics for better understanding. Further, we evaluate the performance of our proposed model on the basis of ROUGE-1, ROUGE-2, and ROUGE-L metric which is considered as the traditional way of evaluating a summarization model. Table 5 states the comparison on the basis of ROUGE-1 metrics. While Table 6 and Table 7 states the comparison on the basis of ROUGE-2 and ROUGE-L metrics. The proposed model has shown a clear advantage over the existing models across all standard metrics.

Table 5. ROUGE-1 metrics for the task of Extractive summarization

Approaches	Precision	Recall	F1-Score
Position based Textrank	0.712	0.611	0.646
Similarity based Textrank	0.702	0.621	0.648
SVD	0.632	0.623	0.612
Fuzzy Logic	0.641	0.625	0.623
Pre-trained LM + Textrank (Proposed)	0.66	0.765	0.694
Improvement over previous best	-7.3	22.4	7.1

Table 6. ROUGE-2 metrics for the task of Extractive summarization

Approaches	Precision	Recall	F1-Score
Position based Textrank	0.653	0.561	0.592
Similarity based Textrank	0.637	0.567	0.591
SVD	0.531	0.519	0.512
Fuzzy Logic	0.561	0.546	0.546
Pre-trained LM + Textrank (Proposed)	0.607	0.701	0.637
Improvement over previous best	-7.04	23.63	7.6

Here, the ROUGE-1 metrics are the precision, recall, and F1-scores based on words taken individually (unigrams). While ROUGE-2 determines the metrics on the basis of bigrams which are pairs of words. Lastly, ROUGE-L considers the whole sentence to determine the final metrics. It can be observed that our proposed approach performs superiorly in comparison to the traditional approaches. While there is a reduction in precision value as compared to previous best, it is compensated heavily with considerable increase in the recall and the more balanced F1-score metric. It also suggests the ability of the proposed approach to extract more context-based human-like summaries, unlike previous approaches. Table 8 shows an example of the output summary formed by the proposed architecture in comparison to the true summary.

The proposed graph-based ranking system with context transfer for extractive text summarization performs well in the majority of cases. Nevertheless, several failure instances were noted during the assessment, including the inability to capture semantic nuances, challenges with long-distance dependencies, and difficulties with domain-specific vocabulary. Furthermore, it must also be noted that the training cost and model size of those approaches are also quite high. Therefore, the use of such architectures lacks feasibility in commercial scenarios. A trade-off between the cost and results when taking into consideration makes a strong case for the use of the proposed architecture.

The practical applications of this proposed study include search engines that may produce concise snippets for user queries by rating and summarizing content from websites in the Marathi language. News platforms need concise summaries

Table 7. ROUGE-L metrics for the task of Extractive summarization

Approaches	Precision	Recall	F1-Score
Position based Textrank	0.734	0.615	0.659
Similarity based Textrank	0.72	0.627	0.661
SVD	0.665	0.614	0.626
Fuzzy Logic	0.659	0.636	0.640
Pre-trained LM + Textrank (Proposed)	0.689	0.771	0.716
Improvement over previous best	-6.13	21.22	8.32

Table 8. Predicted Summary vs True Summary for a sample document

True Summary
बँकांच्या कर्ज थकबाकीचा विषय महत्त्वाचा असून जर एकूण थकलेली रक्कम ही लाखो कोटीच्या घरात जाणारी आहे तर ती जाहीर करण्यास काय हरकत आहे असे सांगून न्यायालयाने संबंधित सर्व पक्षकारांना या विषयावर मंथन घडेल असे सर्व पैलू पुढे आणण्यास सांगितले.याआधी सर्वोच्च न्यायालयाने रिझर्व्ह बँकेला ५०० कोटी रुपये व त्यापेक्षा अधिक रकमेचे कर्ज थकविणाऱ्या कंपन्या व त्यांच्या थकबाकीची रकमांची यादी देण्यास सांगितले होते. बँकांनी दिलेले लाखो कोटीच्या घरातील कर्ज थकविणाऱ्या विविध व्यक्ती व आस्थापनांची नावे व थकित रकमा जाहीर करण्यात याव्यात असे मत सर्वोच्च न्यायालयाने मंगळवारी व्यक्त केले. रिझर्व्ह बँके ने अलीकडेच सीलबंद पाकिटात कर्ज थकबाकीदार con- tinue
Predicted Summary
बँकांच्या कर्ज थकबाकीचा विषय महत्त्वाचा असून जर एकूण थकलेली रक्कम ही लाखो कोटीच्या घरात जाणारी आहे तर ती जाहीर करण्यास काय हरकत आहे असे सांगून न्यायालयाने संबंधित सर्व पक्षकारांना या विषयावर मंथन घडेल असे सर्व पैलू पुढे आणण्यास सांगितले.याआधी सर्वोच्च न्यायालयाने रिझर्व्ह बँकेला ५०० कोटी रुपये व त्यापेक्षा अधिक रकमेचे कर्ज थकविणाऱ्या कंपन्या व त्यांच्या थकबाकीची रकमांची यादी देण्यास सांगितले होते. असे आपले मत बनले आहे. त्यावर रिझर्व्ह बँकेने विरोध दर्शविताना, यातून continue

of stories from various sources while preserving narrative coherence. Legal Document Analysis involves the extraction of significant information from lengthy case files and contracts to optimize time efficiency. Additionally, business intelligence can furnish insights from extensive financial reports for leaders.

The future trends in extractive summarization in Indic languages depend on the development of inclusive, efficient, and contextually aware algorithms utilizing extensive multilingual datasets. Integrating improvements in graph-based algorithms, multilingual embeddings, and cross-lingual transfer can enable research to address the linguistic diversity of Indic languages and meet the increasing demand for multilingual AI solutions.

5 Conclusion

In this paper, we have combined language modeling with word embedding extraction and graph-based ranking as an architecture for extractive text summarization. Superior results have been achieved in comparison to traditional Indic Language summarization approaches. Furthermore, we also were able to extract more context-based human-like summaries, unlike previous occurrence-based approaches. Though the use of heavy attention-based models and modern summarization architectures used in English can work much more efficiently on this task, the lack of proper standard annotated datasets makes the training of such architectures exceedingly difficult for non-English text. Future works include the designing of a new metric that defines the importance of the sentence based on the context of the text. This could then replace the cosine similarity present in text rank and perform the task of summarization much more efficiently. Hence, the use of context-based word embeddings combined with a traditional graph-based approach provides a much more human-like and semantic summary.

References

1. Nenkova, A., McKeown, K.: A survey of text summarization techniques. In: Mining Text Data, pp. 43–76. Springer, Heidelberg (2012). https://doi.org/10.1007/978-1-4614-3223-4_3
2. Kusal, S., Patil, S., Choudrie, J., Kotecha, K., Vora, D., Pappas, I.: A systematic review of applications of natural language processing and future challenges with special emphasis in text-based emotion detection. *Artif. Intell. Rev.* **56**(12), 15129–15215 (2023)
3. Malte, A., Ratadiya, P.: Multilingual cyber abuse detection using advanced transformer architecture. In: TENCON 2019–2019 IEEE Region 10 Conference (TENCON), pp. 784–789. IEEE (2019)
4. Ratadiya, P., Mishra, D.: An attention ensemble based approach for multilabel profanity detection. In: 2019 International Conference on Data Mining Workshops (ICDMW), pp. 544–550. IEEE (2019)
5. Miller, D.: Leveraging bert for extractive text summarization on lectures. arXiv preprint [arXiv:1906.04165](https://arxiv.org/abs/1906.04165) (2019)

6. Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al.: Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint [arXiv:1602.06023](https://arxiv.org/abs/1602.06023) (2016)
7. Li, P., Lam, W., Bing, L., Guo, W., Li, H.: Cascaded attention based unsupervised information distillation for compressive summarization. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2081–2090 (2017)
8. Giri, V.V., Math, M.M., Kulkarni, U.P., Kulkarni, S.B., Joshi, S.M., Vadavi, J.V.: A survey of automatic text summarization system for different regional language in India. *Bonfring Int. J. Softw. Eng. Soft Comput.* **6**, 52–57 (2016)
9. Sarwadnya, V.V., Sonawane, S.S.: Marathi extractive text summarizer using graph based model. In: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBE), pp. 1–6. IEEE (2018)
10. Patil, H., Patil, A., Pawar, B.: Part-of-speech tagger for Marathi language using limited training corpora. *Int. J. Comput. Appl.* **975**, 8887 (2014)
11. Bhosale, M.S., Joshi, D., Bhise, M.V., Deshmukh, R.A.: Automatic text summarization based on frequency count for Marathi e-Newspaper. *IJIET* (2018)
12. Bhoir, A.S., Gulati, A.: A multi-document hindi text summarization technique using fuzzy logic. *Int. J. Adv. Res. Sci. Eng. (IJASE)* **4**(1), 468–473 (2015)
13. Dalal, V., Malik, L.: Semantic graph based automatic text summarization for hindi documents using particle swarm optimization. In: Satapathy, S.C., Joshi, A. (eds.) *ICTIS 2017. SIST*, vol. 84, pp. 284–289. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-63645-0_31
14. Vimal Kumar, K., Yadav, D.: An improvised extractive approach to hindi text summarization. In: Mandal, J.K., Satapathy, S.C., Sanyal, M.K., Sarkar, P.P., Mukhopadhyay, A. (eds.) *Information Systems Design and Intelligent Applications. AISC*, vol. 339, pp. 291–300. Springer, New Delhi (2015). https://doi.org/10.1007/978-81-322-2250-7_28
15. Bhargava, R., Sharma, G., Sharma, Y.: Deep text summarization using generative adversarial networks in Indian languages. *Procedia Comput. Sci.* **167**, 147–153 (2020)
16. Chaudhari, A., Dole, A., Kadam, D.: Marathi text summarization using neural networks. *Int. J. Adv. Res. Dev.* **4**(11), 1–3 (2019)
17. Giri, V.V., Math, M., Kulkarni, U.: Marathi extractive text summarization using latent semantic analysis and fuzzy algorithms. *Comput. Intell. Mach. Learn.*, 1–6 (2023)
18. Malte, A., Ratadiya, P.: Evolution of transfer learning in natural language processing. arXiv preprint [arXiv:1910.07370](https://arxiv.org/abs/1910.07370) (2019)
19. Merity, S., Keskar, N.S., Socher, R.: Regularizing and optimizing lstm language models. arXiv preprint [arXiv:1708.02182](https://arxiv.org/abs/1708.02182) (2017)
20. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
21. Mihalcea, R., Tarau, P.: Textrank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 404–411 (2004)
22. Ghosh, A.: Bengali text summarization using singular value decomposition. PhD thesis (2014)



ViT-GPT2: Vision Transformer Based Automatic Image Captioning

Chanda Srija^(✉) , B. Padmaja , Yennum Smrithi , and Metla Karthikeya

Department of CSE (AI & ML), Institute of Aeronautical Engineering,
Hyderabad, Telangana, India
srijachanda@gmail.com

Abstract. Automatic image caption generation using deep learning is rapidly bridging computer vision with natural language processing. Envision a system that might be able to carefully study a picture and then describe it in natural language. This technology has huge potential for totally revolutionizing the way interactions with images are undertaken, greatly improving accessibility for visually challenged users, more accurate image search, and streamlined image organization. It makes use of Vision Transformers, especially the ViT-GPT2 model for the extraction of image features from Flickr8k dataset. Features extracted in this stage are further fed into the Vision Encoder Decoder Model for processing to output grammatically correct and contextually relevant captions. For proposed model, Standard evaluation metrics used are BLEU, METEOR, GLEU, and ROUGE-L. The experimental results show a great improvement over the present approaches. Obtained results on Flickr8k dataset are BLEU-1 score 0.7024, METEOR score 0.5127, GLEU score 0.2431 and ROUGE-L score 0.5491. It is a model that underscores important progress to be achieved in AI systems for sense-making and describing visual content in the natural language.

Keywords: Deep learning · Image captioning · Natural language processing (NLP) · Vision Transformers (ViTs) · ViT-GPT2 · Vision Encoder Decoder Model

1 Introduction

Automatic image captioning is one of the most wonderful research areas in computer vision, which has gained a huge amount of popularity. It became possible after combining all the concepts from computer vision and natural language processing. It is, therefore considered a multidisciplinary technique that has conquered much attention in the recent years, as it entails a great number of applications in the real world, starting from content indexing up to recommendations on social media and improvements in accessibility [1]. This plainly means that creation of a system correctly analyzing an image and production of a coherent phrase, appropriately describing it, is a task hard to hit for exhaustive comprehension of both linguistic structures and visual content [2]. This problem is more challenging than, for instance, tasks like object detection and image classification.

While object detection focuses on the identification of the objects in an image and possibly labeling them, captioning requires that not only the objects be identified but also their relation and context in which they appear [3]. Humans do this with much ease, incorporating both the innate cognitive aspect of human beings and learned experiences. However, in machines, this can be replicated only after extensive training on different datasets to achieve human-like performance in accuracy [4].

Applications in image captioning are enormous. For instance, image captioning will greatly improve the accessibility to visually impaired persons by describing what is in the images. Apart from that, it will facilitate the work of virtual assistants, index images, and even provide recommendations in social media by understanding and summarizing visual content [5]. Image captioning mainly proposes methods including template-based, retrieval-based, and deep learning-based, categorized into three groups. One type of predefined structure in images, the template-based methods miss the semantic contents of the whole image considerably. Retrieval-based methods generate captions by matching a given input image to a set of images most similar to it. These approaches, however, often face challenges when handling novel images [7]. Recently, many deep neural network-based approaches were found to dominate the research landscape because they can encode and interpret an image's complex features and provide more contextually accurate captions through methods that use CNNs and RNNs [8, 9].

Vision Transformers represent probably one of the most radical developments in deep learning methods for image captioning. Not like traditional CNNs, which process images locally, ViTs cut images into patches and use self-attention mechanisms that capture global contexts [10]. Therefore, performance is improved in a holistic way by modeling long-range dependencies and complex interactions within the image. Its application was originally adapted from the Transformer architecture of Natural Language Processing; therefore, it is obvious that a notable boost over CNN-based methods has been given [11]. Designed an encoder-decoder deep-learning model that generates automatic captions on images using Vision Transformers and Generative Pre-trained Transformer (GPT-2). Utilized Vision Transformers for robust image feature extraction and GPT-2 for generating coherent and contextually relevant captions. Conducted a comprehensive evaluation of the model on Flickr8k dataset based on the various metrics related to accuracy as well as descriptiveness of captions.

In this paper, we introduced related work within the field, including a comprehensive literature review in Sect. 2. Section 3 describes the methodology behind our suggested model, that is, the integration of Vision Transformers with GPT-2 for image captioning. Section 4 outlines results and analysis regarding the model's performance, evaluating it on different evaluation metrics. The findings are summarized in Sect. 5, and future directions discussed and followed by references.

2 Literature Review

VGG16 Hybrid Places 1365 and LSTM model integrates VGG16 that is a convolutional neural network fine-tuned in a Places 1365 dataset and then applies Long Short-Term Memory network, LSTM network for the generation of image captions. Being a deep architecture-based model, it extracts the high-level features from an image by applying its

optimized pre-trained weights that are tuned exclusively for scene recognition [12]. Then, these extracted features feed the input to LSTM, thus capturing temporal or sequential patterns, improving contextual understanding and relational dynamics within a picture. This hybrid approach leverages the strengths of both CNNs in feature extraction and LSTMs in sequence modeling, making it particularly effective for tasks such as video analysis and dynamic scene interpretation [13].

The introduction of an element of memory storage, together with a reading process that is selective, makes the Memory-enhanced captioning with external memory and the mechanism of Selective Reading Mechanism model more efficient and effective in information processing systems [14]. This vital information will have a higher degree of relevance than less important data. ResNet50 has been utilized for image features and LSTM with soft attention for sentence generation [16]. The strengths of ResNet50 with advanced feature extraction and LSTM in stochasticity modelling of sequences, congregate together to elucidate these dynamic visual data more accurately [10].

Dual-GCN with a transformer has been introduced for spatial relations and object-level features, refined by a linguistic transformer [17]. Dual-GCN and Transformer combine Graph Convolutional Networks with Transformer architectures for the handling of heavy representations in many tasks. The Transformer processes graph features using self-attention mechanisms for long-range dependencies and context information [18]. Visual and semantic information has been combined with adaptive attention for image captioning [7]. Hierarchical deep neural networks (HDNN) are a well-developed architecture designed to model data at different levels of abstraction to improve learning and prediction [19]. HDNN has been enabled the utilization of various methods in order to solve different problems, those complex decision-making tasks which require understanding of the data from the simple Eighth level up to the most difficult level, important for accuracy of the results [20, 21].

The CNNs are a special class of deep neural networks has been developed to achieve image classification, detection. This is mainly because of the emerging prevalence in recent times with applications including but not restricted to autonomous vehicles and security systems. This way, CNNs has been designed to extract and recognize patterns from visual data [22–24]. On the other hand, when involving sequential inputs, RNNs do well in things such as language translation and speech recognition. Sentiment analysis problems have dominated most of the RNN applications; nonetheless, transformer-based models are more efficient and have taken over most sentiment analysis problems [25, 26]. LSTM networks are an instance of the RNN, which is adequately effective in maintaining long-distance dependencies within sequences and can perform fairly good with the application wherein some form of temporal processing is required, for example, recognition of handwriting and machine translation [27].

After going through the literature on various methodologies of generating image captions, our approach involves integrating a ViT with a GPT-2 decoder using the Vision Encoder Decoder Model for generating descriptions that maintain the flavor of the visual and contexts of an image. In the proposed approach, capitalizing on the strengths of Vision Transformers in fine-grained feature extraction of visual modality and language

modeling abilities of GPT-2, very accurate contextually rich captions have been generated. Quality is measured with multiple metrics: BLEU (Bilingual Evaluation Understudy), GLEU (Generalized Language Evaluation Understanding), METEOR (Metric for Evaluation of Translation with Explicit ORDERing), ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence).

3 Methodology

The objective of the proposed methodology is to make automatic and well-formed caption generation for the input images. To a large extent, the development of an effective methodology that will result in well-formed captions for input images needs feature extraction, sentence generation, and evaluation. Figure 1 depicts the block diagram of the process for image caption generation with inputting as an image, which undergoes processing from the Vision Transformer (ViT) Feature Extractor. Apart from this, after extracting the features through the Vision Encoder-Decoder Model then sent to Auto Tokenizer that converts it into the final output text.

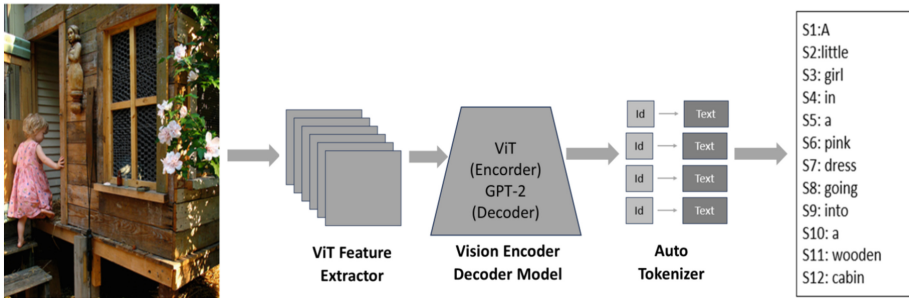


Fig. 1. Block Diagram of the Proposed Model

The flowchart, Fig. 2, outlines the methodology applied to the generation process of image captions: beginning with collection, followed by cleaning and filtering to ensure high-quality input data; feature extraction and preprocessing so that the data was ready for model training. Finally, the trained models captioned an image, and results were evaluated for measuring performance and accuracy.

Implementing ViT and GPT-2 for image captioning requires high-performance computing. On the software side, it shall require the use of TensorFlow with libraries like Transformers and PIL. For hardware, a GPU with sufficient memory is necessary to handle the actual computational requirements of ViT's image processing and GPT-2's large language model capabilities, especially during training and fine-tuning.

3.1 Dataset

The work is based on caption and image data part of the Flickr8k dataset, which is of great use in the research community mainly because of its rich collection of images and

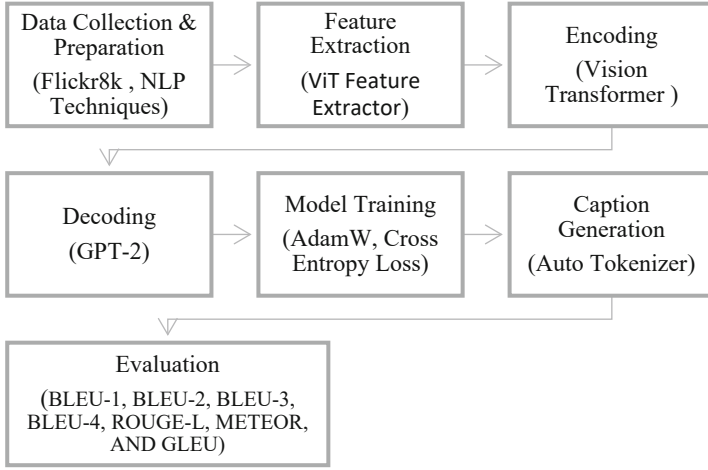


Fig. 2. Flowchart of the Methodology.

interesting descriptions. The dataset constitutes 8000 images; each image is annotated with a number of captions that are produced by human generators. This dataset otherwise is quite rich in both linguistic diversity and context. 5 captions are maintained for each image and out of 8000 images, 6000 are used for training purpose, 1000 for development purpose, and the remaining 1000 images are used for testing purpose. Average caption length is 12 and resolution varies from 256×500 to 500×500 . The Flickr8k dataset is available at [Kaggle](#).

The computational complexity and scalability of the proposed model were tested using the Flickr8k dataset. On account of the transformer-based architecture of ViT, its use for feature extraction and GPT-2 for text generation inherently incur overhead due to additional requirements in terms of computation. Still, these components are planned to scale up well and apply parallel processing capabilities with larger datasets. Future evaluations on larger datasets such as Flickr30k or MS COCO could even give further insight into the scalability and performance of the model across more diverse and complex distributions of the data.

3.2 Feature Extraction

Feature extraction was performed by the pretrained ViT Feature Extractor. First, the input image is checked to ensure it is in RGB color mode; any other type of input format is transformed into RGB format. Then, the image is further processed with ViT Feature Extractor, which extracts the pixel values in tensor form of size of size $3 \times 224 \times 224$. These pixel values now serve as input to the Vision Transformer model to delve deeper into features that will be beneficial in further steps of an image captioning pipeline.

For each image I_i from the list of images $\{I_1, I_2, \dots, I_n\}$, the image is loaded and converted to RGB mode if necessary:

$$I_i \rightarrow I_i^{RGB} \quad (1)$$

where I_i^{RGB} is the processed image in RGB format. From (1), the processed images $\{I_1^{RGB}, I_2^{RGB}, \dots, I_n^{RGB}\}$ are then passed through the ViT feature extractor:

$$X = \text{ViT_FeatureExtractor}(I_1^{RGB}, I_2^{RGB}, \dots, I_n^{RGB}) \quad (2)$$

$$X_{pixels} = \text{FeatureExtractor}(I_i^{RGB}) \quad (3)$$

where X_{pixels} are the pixel representations extracted from image.

3.3 Proposed Model

The image caption generation here includes using a model relying on the Vision Encoder Decoder Model, which combined use of an encoder comprising the Vision Transformer, ViT with GPT-2 as its decoder. The model ViT does a lot of feature extraction from the images, capturing intricate patterns and details by dividing the image into patches and treating them just like transformers do with sequences. These features that are being extracted then give an input to the GPT-2 model to describe and give coherent captions. To enhance the model's performance, fine-tuning is conducted using a Flickr8k dataset. This process involves loading the data through a Data Loader, which efficiently handles batching and shuffling. The fine-tuning process is crucial for adapting the model to specific domains or datasets, ensuring that the generated captions are relevant and accurate. During fine-tuning, it learns how to map visual features to textual descriptions, meaning mapping to captions, by adjusting its parameters to the Flickr8k dataset so that it can generate captions closer to the desired output.

The training process is driven by an Adam with Weight Decay (AdamW) optimizer, which is well-suited for large transformer models due to its ability to handle sparse gradients and provide stable convergence. Cross Entropy Loss is used as a loss function, which compares the generated captions with the target captions, penalizing incorrect predictions. The training loop iteratively adjusts the model's weights by minimizing the loss, gradually improving the quality of the captions. It makes sure that model has learned enough from the data by training the model over dataset. The Vision Transformer processes the pixel values from (2) to produce a feature representation. The resultant feature representation is used as input to the GPT-2 decoder.

$$E_{features} = \text{ViTEncoder}(X_{pixels}) \quad (4)$$

where $E_{features}$ is the encoded feature representation from the Vision Transformer. The GPT-2 model generates a sequence of output token IDs Y_{ids} based on the encoded features.

$$Y_{ids} = \text{GPT2Decoder}(E_{features}) \quad (5)$$

Combining (3), (4), (5) we get

$$X_{pixels} \xrightarrow{\text{ViT}} E_{features} \xrightarrow{\text{GPT2Decoder}} Y_{ids} \quad (6)$$

The Auto Tokenizer is then used to generate the output caption from token IDs which is generated as shown in (6). Auto Tokenizer is the tokenizer that automatically converts text into tokens, which are smaller units of meaning, for easy processing by machine learning models. In this work, it will be used to prepare and format text for ViT-GPT-2 model image captioning; therefore, it will allow the model to understand and generate captions for images.

Fine-tuning would mean tweaking the hyperparameters, like batch size and learning rate, with the intent of adjusting performance. For this project, a batch size is set at 32, and the learning rate is $5e-5$, but certainly other combinations would continue to help improve the results. The impact of these variations, involving the number of epochs, will be analyzed in terms of an effect on response from the model. Future experiments can also look into using gradient clipping or a learning rate scheduler to stabilize the training process. All in all, the proposed model would make it possible to leverage the best qualities of transformers such as ViT as encoder, GPT-2 as decoder and fine-tuning approaches in achieving high-quality image captioning. This allows for well-formed captions and contextually accurate captions. Therefore, the model becomes a very powerful tool for numerous computer vision and natural language processing applications.

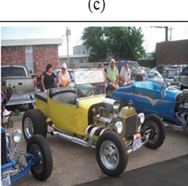
 Reference Captions 1. man on bicycle riding down rocky hill 2. man rides his dirt bike down rocky trail 3. person rides bike down rocky hill 4. blurry picture of mountain biker descending rocky trail 5. the mountain biker in blue rides down rocky path Generated caption biker is riding down hill BLEU-1 Score: 0.7430 (a)	 Reference Captions 1. man at the top of mountain with beautiful view in the background 2. man is sitting on snowbank 3. man on snowy peak 4. person in red snow gear is kneeling on snowy ridge under blue sky 5. mountain climber in red suit poses on snowy peak with mountains in the background Generated caption person is standing in the snow BLEU-1 Score: 0.8750 (b)
 Reference Captions 1. group of race buggies travel down racetrack 2. cars racing on dirt track 3. race cars are racing each other on dirt racetrack and the green one is in the lead 4. rally cars musde for position on dirt track 5. two lines of colorful cars racing on racetrack Generated caption two cars are racing on track BLEU-1 Score: 1.0000 (c)	 Reference Captions 1. man in wetsuit with surfboard walks along beach 2. person in wetsuit is carrying surfboard into the ocean 3. person with white and black suit board waking into the waves 4. surfer in black wetsuit carrying surfboard into the waves 5. surfer in dark wet-suit carries white surfboard towards the ocean Generated caption man carrying surfboard walks on the beach BLEU-1 Score: 0.7954 (d)
 Reference Captions 1. bunch of vintage cars are parked in lot 2. an antique car that is yellow is shown lined up with other cars 3. people admire classic cars 4. several old fashioned cars are parked together 5. some old people admiring classic cars park in front of them Generated caption group of people standing in front of red car BLEU-1 Score: 0.6364 (e)	 Reference Captions 1. brown dog is running through brown field 2. brown dog is running through the field 3. brown dog with collar runs in the dead grass with his tongue hanging out to the side 4. brown dog with his tongue wagging as he runs through field 5. dog running in the grass Generated caption brown dog running through field BLEU-1 Score: 1.0000 (f)

Fig. 3. Sample Images with their reference captions from the dataset along with generated caption using proposed model

3.4 Caption Generation

In the process of creating a caption, when the model is generating output IDs for tokens, those IDs form a sequence of tokens for a caption predicted. The Auto Tokenizer will use those output IDs to decode their text format. The tokenizer will take that sequence of token IDs and convert that sequence to words and sentences to rebuild, in effect, a textual caption out of the model's predictions. This is a really crucial step as it's actually taking the numeric output from the model and converting it into some meaningful caption. The output token IDs Y_{ids} from (5) are finally decoded by the tokenizer into the final predicted caption $Y_{caption}$ given by (7).

$$Y_{caption} = \text{Tokenizer}(Y_{ids}) \quad (7)$$

In Fig. 3, the reference captions depict the ground truth description of the images, while the generated captions have been selected from the output of the ViT-GPT-2 image captioning model. Such comparison depicts how well the captions were generated by the model; in other words, whether the captions are relevant to their context or just automatically generated statements.

3.5 Evaluation Metrics

A number of metrics were utilized to ensure the performance of the proposed model of image captioning was comprehensively evaluated. BLEU scores of the n-grams precision in the generated captions against reference captions. ROUGE-L is added for the longest common subsequences. This is through the alignment of exact, stemmed, and synonym matched words. On the other hand, GLEU measures the quality of captions generated against reference captions it puts more emphasis on the grammaticality of the captions. Together, they provide a robust measure of the model's effectiveness in generating accurate and contextually relevant captions.

4 Results and Analysis

To establish whether the proposed Vision Encoder Decoder Model was good at generating meaningful captions for given images, strict procedures were used to arrive at the final conclusion. It was about keeping the focus on whether the produced captions really represent content in the image and offer comprehensible descriptions to human users. The Fig. 4 is the image with its reference captions given in the dataset. For this image, the result caption produced is “group of dogs playing in field”, which shows that the model could generate descriptive and contextual text based on the visual content.

The model's performance was assessed using several metrics, reflecting its ability to generate relevant and coherent captions. The BLEU scores—ranging from 0.7024 for BLEU 1 to 0.2411 for BLEU-4—indicate a high level of precision in capturing n-grams, with a gradual decrease in score as the n-gram length increases. ROUGE-L score is 0.5491, which indicates a moderate rate of agreement in the longest common subsequences between the reference and the generated captions. The METEOR score is at 0.5127; this score perfectly balances precision and recall, as well as the GLEU score



1. cluster of four brown dogs play in field of brown grass
2. four dogs are together in the field of dry grass
3. four dogs in grassy area
4. four medium-sized dogs wrestle with each other on grass field
5. four small dogs play outside

Fig. 4. Sample image with reference captions.

of 0.2431, which reflects the ability of the model to perform fairly well in evaluating the modules of small dimensions. Cumulatively, these results point to the ability of the model to yield inferences that do result in the generation of accurate and contextually relevant captions, while needing improvement in other dimensions.

Tables 1 and 2 compare the performance of the proposed model with multiple other models over different evaluation metrics. The table has six different models with scores on seven different evaluation metrics: BLEU-1 to BLEU-4, ROUGE-L, METEOR and GLEU. Each row depicts a different model, with each column referring to a specific measure of evaluation. In this manner, the table enables one to make a quantitative comparison in terms of the performance of models across these metrics.

Table 1. Performance of different models based on BLEU scores

S No	Model	Decoder	BLEU-1	BLEU-2	BLEU-3	BLEU-4
1	M-RNN [19]	LSTM	0.5650	0.3860	0.2560	0.1700
2	V-S.M [20]	MESHED MT	0.579	0.383	0.245	0.160
3	DL [21]	LSTM	0.5335	-	-	-
4	AICRL [5]	LSTM	0.619	0.452	0.3668	0.262
5	EL [15]	LSTM	0.634	0.400	0.287	0.151
6	CL [17]	LSTM	0.6373	0.4500	0.3087	0.2113
7	CLA [17]	LSTM	0.6532	0.4692	0.3281	0.2258
8	ResNet152 [24]	LSTM	0.6230	0.4424	0.3084	0.2121
9	VGG16 [22]	LSTM	0.6056	0.4198	0.2866	0.1951
10	Densenet161 [23]	LSTM	0.6312	0.4468	0.3076	0.2079
11	VGG16 Hybrid Places1365 [1]	B-LSTM	0.6666	0.4704	0.3893	0.2878
12	Proposed Model	ViT with GPT-2	0.7024	0.4941	0.3537	0.2411

The BLEU-1 to BLEU-4 metrics that measure the overlap of n-gram used for sentence alignment are carried out for the 12 models with the proposed model presented in Table 1. Under the baseline models, the BLEU-1 score is recorded as the highest value by VGG16 Hybrid Places1365 [1] at 0.6666, but this is surpassed by the score of the proposed model at 0.7024 that bettered this with a good quality of sentences. BLEU scores decrease with higher n, reflecting the difficulty in capturing longer sequences, yet the Proposed Model prevails on BLEU-2 (0.4941), BLEU-3 (0.3537), and BLEU-4 (0.2411), showing much better contextual alignment. Its ViT with GPT-2 decoder does justice to the subtleties of the language compared to the LSTM-based architectures.

Table 2 assesses models with ROUGE-L, METEOR, and GLEU metrics with a focus on structural similarity, semantic accuracy, and fluency. The Proposed Model outperformed others with the highest ROUGE-L: 0.5491, indicating superior sentence structure alignment and METEOR: 0.5127, showing excellent semantic accuracy as opposed to baseline models that had scores below 0.209. Its GLEU value is slightly lower at 0.2431 than the score of 0.2469 for the VGG16 Hybrid Places1365 [1], but such minute difference makes the model equally coherent and grammatical. In general, the improvement in fluency and generation of meaningful sentences with the Proposed Model is significantly visible.

Table 2. Comparative Analysis of ROUGE-L, METEOR and GLEU scores with Proposed Model

S No	Model	Decoder	ROUGE-L	METEOR	GLEU
1	M-RNN [19]	LSTM	-	-	-
2	V-S.M [20]	MESHED MT	-	-	-
3	DL [21]	LSTM	-	-	-
4	AICRL [5]	LSTM	-	0.209	-
5	EL [15]	LSTM	-	-	-
6	CL [17]	LSTM	0.4641	0.1995	-
7	CLA [17]	LSTM	0.4695	0.2087	-
8	ResNet152 [24]	LSTM	0.4616	0.1950	-
9	VGG16 [22]	LSTM	0.4482	0.1904	0.1542
10	Densenet161 [23]	LSTM	0.4619	0.2000	0.1550
11	VGG16 Hybrid Places1365 [1]	LSTM	0.3076	0.5060	0.2469
12	Proposed Model	ViT with GPT-2	0.5491	0.5127	0.2431

The Table 1 and Fig. 6 represents the comparison done amongst various models, including the proposed model, based on BLEU-1 scores on the Flickr8k dataset. BLEU-1 scores describe the precision of the models while generating the captions, and here, the proposed model is showing the best performance. The comparison depicted here signifies the potency of this proposed approach in performing image captioning.

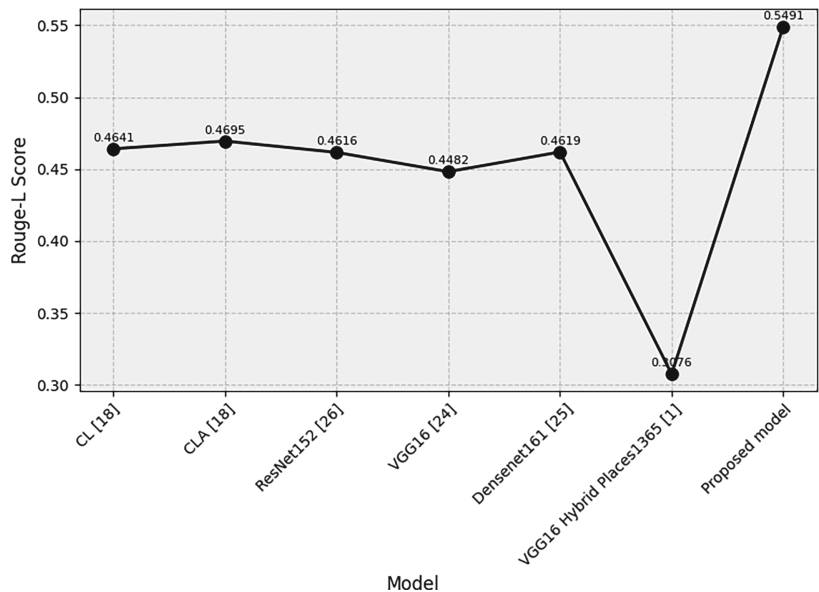


Fig. 5. ROUGE-L scores of different models

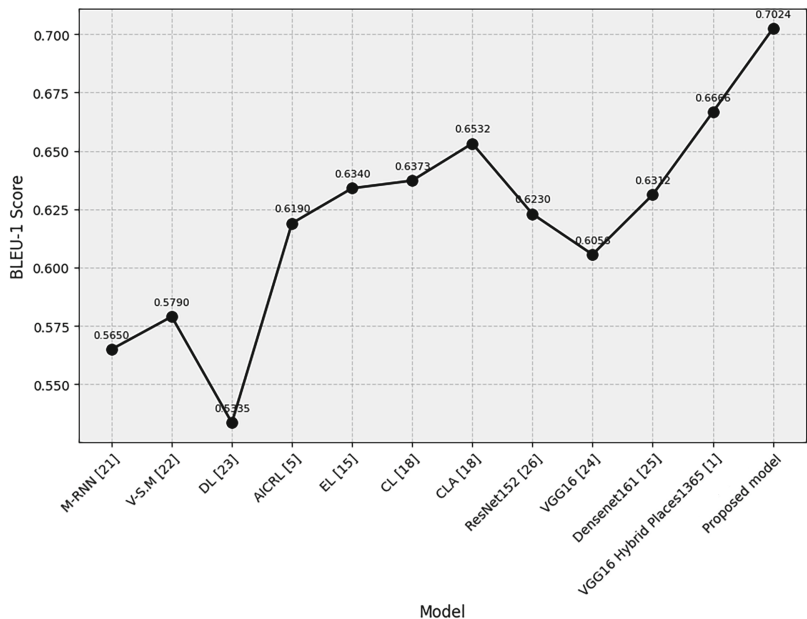


Fig. 6. BLEU-1 scores of different models

The Table 2 and Fig. 5 represents a comparative analysis graph of ROUGE-L scores achieved by various models. The line plot illustrates the performance of different models

in terms of ROUGE-L metric, providing a visual representation of their summarization or translation capabilities. By examining the graph, one can identify the model with the highest ROUGE-L score and assess the relative performance of each model.

5 Conclusion

In conclusion, the objectives in developing a deep learning-based encoder-decoder model for automatic image captioning were well achieved. The model developed herein, characterized by the use of Vision Transformers for robust image feature extraction and GPT-2 for coherent and contextually relevant caption generation, really proved effective for proper caption generation. Results obtained on the Flickr8k dataset were ROUGE-L: 0.5491, METEOR: 0.5127, GLEU: 0.2431, BLEU-1: 0.7024, BLEU-2: 0.4941, BLEU-3: 0.3537, BLEU-4: 0.2411. These were indicative of a high level of accuracy and relevance in the generated captions.

For future work, this can be extended to more diverse datasets to enhance generalization. Further advanced transformer architectures or the integration of domain-specific pre-trained models may lead to better quality and applicability of the generated captions. The further fine-tuning of models in application fields, including real-time processing applications, medical imaging, or autonomous systems, may extend the scope of this research into practical applications.

References

1. Verma, A., Yadav, A.K., Kumar, M., et al.: Automatic image caption generation using deep learning. *Multimed. Tools Appl.* **83**, 5309–5325 (2024)
2. Barlas, G., Veinidis, C., Arampatzis, A.: What we see in a photo graph: content selection for image captioning. *Vis. Comput.* **37**(6), 1309–1326 (2021)
3. Biswas, R., Barz, M., Sonntag, D.: Towards explanatory interactive image captioning using top-down and bottom-up features, beam search and re-ranking. *KI-Künstliche Intelligenz* **34**(4), 571–584 (2020)
4. Chen, H., Ding, G., Lin, Z., Guo, Y., Shan, C., Han, J.: Image captioning with memorized knowledge. *Cogn. Comput.* **13**(4), 807–820 (2021)
5. Chu, Y., Yue, X., Yu, L., Sergei, M., Wang, Z.: Automatic image captioning based on resnet50 and lstm with soft attention. *Wireless Communications and Mobile Computing* (2020)
6. Dong, X., Long, C., Xu, W., Xiao, C.: Dual graph convolutional networks with transformer and curriculum learning for image captioning. *arXiv preprint arXiv:2108.02366* (2021)
7. Peng, Y., Wang, C., Pei, Y., Li, Y.: Video captioning with global and local text attention. *The Visual Computer*, pp. 1–12 (2021)
8. Khan, M.J., Khan, M.J., Siddiqui, A.M., Khurshid, K.: An automated and efficient convolutional architecture for disguise-invariant face recognition using noise-based data augmentation and deep transfer learning. *The Visual Computer*, pp. 1–15 (2021)
9. Rinaldi, A.M., Russo, C., Tommasino, C.: Automatic image captioning combining natural language processing and deep neural networks, *Results in Engineering*, Volume 18 (2023)
10. Yuting, S., Li, Y., Ning, X., Liu, A.-A.: Hierarchical deep neural network for image captioning. *Neural Process. Lett.* **52**(2), 1057–1067 (2020)
11. Gupta, N., Singh Jalal, A.: Integration of textual cues for fine-grained image captioning using deep cnn and lstm. *Neural Comput. Appl.* **32**(24), 17899–17908 (2020)

12. He, C., Haifeng, H.: Image captioning with text-based visual attention. *Neural Process. Lett.* **49**(1), 177–185 (2019)
13. Huang, F., Li, Z., Wei, H., Zhang, C., Ma, H.: Boost image captioning with knowledge reasoning. *Mach. Learn.* **109**(12), 2313–2332 (2020)
14. Jiang, T., Zhang, Z., Yang, Y.: Modeling coverage with semantic embedding for image caption generation. *Vis. Comput.* **35**(11), 1655–1665 (2019)
15. Katpally, H., Bansal, A.: Ensemble learning on deep neural networks for image caption generation. In: *Proceedings - 14th IEEE International Conference on Semantic Computing, ICSC 2020*, pp. 61–68 (2020)
16. Liu, X., Qingyang, X., Wang, N.: A survey on deep neural network-based image captioning. *Vis. Comput.* **35**(3), 445–470 (2019)
17. Department of Computer Science Sulabh Katiyar, Samir Kumar Borgohain and Silchar Engineering National Institute of Technology. Comparative evaluation of cnn architectures for image caption generation. *International Journal of Advanced Computer Science and Applications* (2021)
18. Xiao, F., Gong, X., Zhang, Y., Shen, Y., Li, J., Gao, X.: DAA: Dual LSTMs with adaptive attention for image captioning. *Neurocomputing* **364**, 322–329 (2019)
19. Mao, J., Xu, W., Yang, Y., Wang, J., Yuille, A.L.: Explain images with multimodal recurrent neural networks. *arXiv preprint [arXiv:1410.1090](https://arxiv.org/abs/1410.1090)* (2014)
20. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(4), 664–676 (2017)
21. Amritkar, C., Jabade, V.: Image caption generation using deep learning technique. In: *Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBE 2018*, pp. 1–4 (2018)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)* (2014)
23. Padmaja, B., Bala, M.M., Patro, E.K.R.: A comparison on visual prediction models for MAMO (multi activity-multi object) recognition using deep learning. *J. Big Data* **7**(24), 1–15 (2020)
24. Merugula, S., Padmaja, B., Veerubommu, R.: RideNN-OptDRN: heart disease detection using RideNN based feature fusion and optimized deep residual network. *Concurrency Comput. Practice Exp.* **34**(28) (2022)
25. Padmaja, B., Bala, M.M., Patro, E.K.R., Srikruthi, A.C., Avinash, V., Sudheshna, C.: An intelligent auto-response short message service categorization model using semantic index. *Int. J. Electr. Comput. Eng. (IJECE)* **14**(1), 922–933 (2024)
26. Raju, A.S.N., Venkatesh, K., Padmaja, B., Reddy, G.S.: GIEn-semformerCADx: a hybrid ensemble learning approach for enhanced gastrointestinal cancer recognition. *Multimed. Tools Appl.* **83**(15), 46283–46323 (2024)
27. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708 (2017)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)



Multimodal Language Models for End-to-End Automated Speech Recognition Using Bidirectional Recurrent Neural Network

Thayyaba Khatoon Mohammad², Puranam Revanth Kumar¹(✉) , Gifta Jerith¹,
and E. Krishnaveni Reddy¹

¹ Department of Artificial Intelligence and Machine Learning, School of Engineering,
Malla Reddy University, Hyderabad, India
revanth123451.rk@gmail.com

² Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of
Engineering and Technology, Hyderabad, India

Abstract. End-to-end Automated Speech Recognition (ASR) has recently benefited from advancements in multimodal language models, which combine audio and visual data to improve transcription accuracy and context understanding. Neural networks have become increasingly popular for the task of language modeling. A language model (LM) is a mathematical representation of the likelihood of a word sequence that offers a solution to word prediction in different types of information systems. Learning the long-span dynamics of a continuous-space word sequence is a powerful task for a Recurrent Neural Network (RNN). On the other hand, with a high-dimensional hidden layer and a huge dictionary size, training the RNN-LM becomes an ill-posed task. In contrast to traditional unidirectional networks, which rely solely on historical data to provide predictions, bidirectional networks can take into account both current and future inputs. In this research, we suggest using bidirectional Recurrent Neural Networks (BRNN) for language modelling in speech recognition. We examine bidirectional models on an Indian English Broadcast News transcription assignment and talk about problems that come up when using them for speech. With the use of testing data from 50 epochs, the audio speech of proposed dataset was accurately recognized with a 97.74% accuracy rate and a precision rate of 95.23%. Hence, multimodal BRNN-based ASR framework offers a robust approach to end-to-end speech recognition, providing a foundation for further exploration into multimodal enhancements.

Keywords: Language models · Speech recognition · Bidirectional neural network · Deep learning · Sentiment analysis

1 Introduction

Speech is a ubiquitous and essential mode of communication among humans, facilitating the expression of ideas, thoughts and emotions and enabling engagement in meaningful conversations [1]. As our lives become increasingly intertwined with machines and

smart devices, new communication techniques that align with our digitalized world have emerged. Speech recognition has played a pivotal role in enabling us to adapt to these novel modes of communication: it not only empowers individuals with disabilities to interact, share knowledge and engage in open conversations, but also holds promise for revolutionizing communication between machines using natural language [2]. Statistical language models (LMs) are crucial components in a wide-range of applications, including automatic speech recognition (ASR) systems. Pattern recognition techniques also enable the mapping of audio features to linguistic units, such as phonemes, words or sentences, exploiting statistical consistencies within the speech signal. Machine learning (ML) algorithms, including deep neural networks (DNNs), have proven instrumental in learning and recognizing these patterns, leading to accurate and efficient speech recognition. The future of speech recognition technology holds promising prospects [3]. It is poised to surpass our expectations in meeting the evolving needs of the community and has the potential to facilitate seamless communication between machines using natural languages [4]. By continuously advancing the boundaries of research and development, speech recognition systems will continue to enhance our ability to communicate effectively and effortlessly in an increasingly interconnected world. Conventional recurrent neural network (RNN) and long short-term memory (LSTM) models are unidirectional, meaning that the input data is processed in the order of time, and the output probabilities are estimated based solely on the context of the past [5]. Despite the obvious importance of future context, unidirectional models do not account for how words in the future might affect the likelihood of the current word. Rather, the impact of the present word on the likelihood of subsequent words is the sole means by which this dependency may be represented. Building models that move both ahead and backward in time to take advantage of model combination seems like it could be a good idea, at least intuitively. This is because dependencies can work both ways. An attractive foundation for merging data from both the past and the future is provided by bidirectional RNNs [6]. By analysing the input data in reverse order, an additional hidden layer called the backward hidden layer is trained in bidirectional RNNs. Rather of being directly linked, the original hidden layer (also known as the forward hidden layer) and the backward hidden layer both connect to the same output layer. By substituting memory blocks for the hidden units in a bidirectional RNN, bidirectional LSTMs can be created [7]. Applications of bidirectional RNNs and LSTMs include audio modelling and handwriting recognition [8, 9].

Algorithms for voice recognition are able to more accurately recognise ambivalent words thanks to speech recognition's lipreading and image processing capabilities [10]. Building a model for audiovisual speech recognition that can translate visual and auditory inputs is, hence, the primary objective. This study examined the limitations and performance of a hybrid model used for audiovisual speech recognition; the findings are presented in this publication to help future research in this field [11]. Assisting those who are hard of hearing is the driving force behind the planned project. Lipreading is a common way for people who have trouble hearing to communicate, especially in busy places or when speech is difficult to make out [12]. However, lipreading is an acquired skill, and it can be difficult for the deaf to understand spoken language [13]. The solution to this difficulty and the ability for people with hearing impairments to engage in

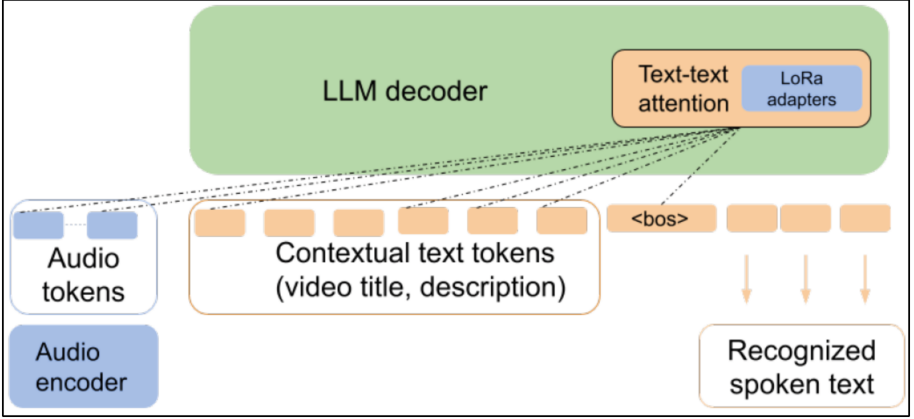


Fig. 1. A pretrained LLM-based speech recognition model with audio and optional text tokens.

discussions can be found in audiovisual speech recognition, which involves the interpretation of both audio and visual information. Helping those who are hard of hearing is the driving force behind this work speech recognition algorithm development. The goal is to be able to anticipate spoken words from videos without audio and vice versa. This research work primarily aims to accomplish the following:

- Establish a database comprising Indian English;
- Create an algorithm for visual speech recognition;
- Modify the neural network to enhance pattern learning and feature extraction;
- Achieve the highest quality audiovisual speech recognition outcomes.

The scientific significance of the anticipated models includes the understanding of many models, neuroscientific insights, and cross-model learning. On the other hand, the practical significance of the anticipated models comprises improvements in regard to the accuracy of voice recognition, accessibility, security, and authentication. The rest of the article is structured as follows: In Sect. 2, we will go over the current approaches and resources for automated speech detection. The suggested method is detailed in Sect. 3. In Sect. 4, the results are detailed. Section 5 details the conclusion and following works.

2 Related Work

In the realm of machine learning and deep learning, convolutional neural networks (CNNs) are by far the most used for processing and recognizing audio [14]. A minimum weight-sharing approach has been proposed that effectively handles speech features, in contrast to the traditional full weight-sharing CNN designs used for image processing. Rather than pooling layer overlap, pooling size is the primary factor affecting automatic voice recognition performance [15]. When compared to the conventional ANN-based method, the CNN-based strategy for voice recognition achieves superior accuracy [16]. The spoken digit recognition data is classified using a CNN that is trained using characteristics taken from the Bark spectrogram [5]. Compared to other parameter-based methods, CNN outperforms them by generating voice classes from unprocessed speech inputs

[17]. The stride-based convolutional neural network (SCNN) is an innovative method that improves computational stability by reducing the number of convolutional layers and removing the pooling layers [18]. A BiLSTM model could potentially be transformed into a low-latency end-to-end UniLSTM model through the process of knowledge distillation [19]. When compared to a regular GRU, the Li-GRU layer reduces computational complexity by a large margin, improves recognition performance, and saves training time by over 30% [20]. When trained with LSTM-RNN, accuracy and learning rate are not mutually exclusive [21]. Convolutional LSTMs with 15 layers of deep learning achieve a word-error rate of 10.5% without language model or a dictionary [22]. During decoding, the biasing strings of a weighted finite state transducer (WFST) were added by [23]. The scores of the RNN-T system and the biasing WFST are interpolated. One benefit of these approaches is their adaptability; after training is over, they may be linked to any system. Contextualization end-to-end during model training is another area of research in deep biasing approaches [24]. One major drawback of these methods is that they only consider the phrase level, not the entire context. Furthermore, these methods necessitate the incorporation of specialized biasing modules within the primary ASR architecture. Several methods that use LLMs for speech-related tasks were also introduced at the same time in this study. Together with audio representations, [25] used LLaMA LLM to translate spoken language by attaching a textual instruction (“Translate audio to language X”). For STT and STT-to-speech tasks, the Audio Palm [26] model combines audio and text tokens. Using multilingual data, [27] demonstrated how to enable speech recognition capabilities for the LLaMA model. Table 1 gives the summary of the state-of-the-art speech recognition algorithms.

Table 1. Overview of multimodal language models for ASR

Reference	Methods	Dataset	Limitations
Gambhir et al. [1]	BRNN with Multimodal Embeddings	Libri speech, AV Speech	Limited ability to generalize to noisy or unseen domains due to domain dependency
Almadhor et al. [4]	Multimodal Transformer + BRNN for Audio and Visual Features	LRS3, TED-LIUM	High computational cost and memory requirements
Djeffal et al. [5]	Hybrid BRNN with Attention Mechanism combining Audio and Text features	VoxCeleb, Common Voice	Limited to specific language accents; requires large annotated datasets
Gan et al. [6]	Audio-Visual Speech Recognition using BRNN and Multimodal Feature Fusion	TIMIT, VoxForge	Performance degrades with misalignment in visual and audio inputs

(continued)

Table 1. (continued)

Reference	Methods	Dataset	Limitations
Chang et al. [9]	BRNN with Cross-Modal Attention between Language and Visual Modalities	Common Voice, TED-LIUM	Difficulty handling real-world noise and occlusions in visual data
Fukuda et al. [11]	Multimodal Encoder-Decoder with BRNN for ASR and Gesture Analysis	GRID, TCD-TIMIT	Limited performance on spontaneous and casual speech, especially in low-resource languages
Reza et al. [13]	BRNN-based Multimodal ASR with Pre-trained Vision and Audio Models	AISHELL-1, AV Speech	Limited by the high cost of pre-training and real-time application challenges
Radha et al. [14]	Vision-Enhanced BRNN for ASR, incorporating Visual Speech Recognition (VSR) in noisy settings	VoxCeleb2, YouTube-8M	Reduced accuracy in low-light or occluded visual conditions
Oruh et al. [20]	BRNN with Multimodal Semantic Embeddings for Language and Audio Fusion	Mozilla Common Voice, LJ Speech	Dependency on high-quality labelled data; limited scalability for diverse speech patterns
Bhaskar et al. [24]	Sequential Multimodal BRNN using Self-Supervised Learning	Audio Set, How2	Struggles with speech from unfamiliar or complex linguistic patterns

A Whisper model recently used a biased method, where the transcription of the previous segment was included in the longform speech recognition prompt [28]. Unlike their work, our approach is influenced by the unstructured and occasionally irrelevant textual background. This is because video titles and descriptions do not necessarily correspond to the context of spoken language. Another review study surveyed mouth interface devices based on sensors for the recognition of speech using DL techniques [29]. It also focused on communication difficulties for disabled people, visual speech recognition, and a silent speech interface. Another study conducted a brief review of 17 papers on Arabic speech recognition using neural networks [7]. For the same language, another survey highlighted the models found in 35 studies and the evaluation techniques and metrics used in dialectal Arabic speech [1]. The study provided details on the progress of dialectal Arabic and summarized the challenges and issues raised in previous research. Another review of speech recognition using DL concluded that using both probabilistic

neural networks and RNNs can correctly recognize 98% of phonemes, followed by hidden Markov models (HMMs) [3]. In a related study conducted by [10], the basic principles of voice recognition technology, related theories and problems were surveyed. The study presented optimization methods of artificial intelligence (AI) and simulation training for speech recognition based on DL. It highlighted several issues that affect the quality of this technology, such as its capability to handle noisy environments and low endpoint detection levels. To address these challenges, the study introduced methods of optimization and analysis aimed at improving speech recognition performance by processing information more effectively.

3 Methodology

In this section the steps used to build the dataset, extract and categories features from the audio model, do the same for the visual model, and then use a DCNN to combine the two sets of data. The proposed block diagram of this procedure is shown in Fig. 1.

3.1 Dataset Description

An Indian English Broadcast News assignment was the subject of our experiments. The GALE voice transcription system developed by IBM in 2007 serves as the foundation for the baseline system [14]. The 30 h of broadcast news audio are used to train the speaker-adaptive acoustic model. The standard 4-g model, which has a vocabulary of over 84,000 words and was trained on 50 million words from various sources, serves as the foundational language model. Eight males and eight females, ranging in age from 18 to 35, contributed eighty movies to the study's collection. The recordings were made in an isolated room with no outside noise or disturbances. The data was gathered with the intention of improving and assessing the lipreading process through the use of voice recognition and similar technologies. Eight words were spoken by each of the sixteen participants, and each word was recorded five times. Consequently, there were 640 bytes of data in the dataset, representing 8 words x 5 times 16 people [2].

'About', 'Bad', 'Bottle', 'Come', 'Cow', 'Good', 'Pencil', 'Read', and 'Where' are the nine words that were utilised here. These terms were chosen at random from a pool of around eighty movies [14]. Using a Microsoft Video editor, the videos in the dataset were trimmed to include only one second of audio for every word. A frame rate of 60 frames per second was used for the video. We split the data into two sets: one for training that included 75% of the movies and another for validation that included 25% of the movies, for a total of 25% of the dataset. The model was trained using the training set, and its performance was evaluated using the validation set.

3.2 Proposed Bidirectional RNN Training Criteria

A subset of the original 50 million words is used to train BRNN language models, which consist of 12 million words. We utilised the python libraries to execute training algorithms for the LSTM and RNN language models, both bidirectional and unidirectional [15]. Using a GPU and back propagation over time with mini-batch updates, the

RNN/LSTM models are trained. The network parameters are adjusted after processing all of the words in a mini-batch of sentences that are propagated individually through the network during mini-batch training. Each sentence in a minibatch should have the same length to maximise the use of the quick matrix operations on a GPU. On the other hand, particularly in the realm of broadcast news, there can be a great deal of variation in sentence length. While it is possible to make shorter phrases the same length by padding them with special symbols, this reduces the efficiency gain that may be achieved by utilising a GPU.

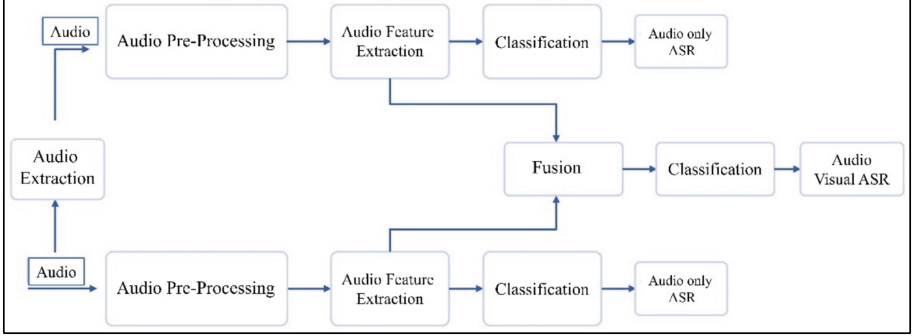


Fig. 2. Proposed speech recognition block diagram

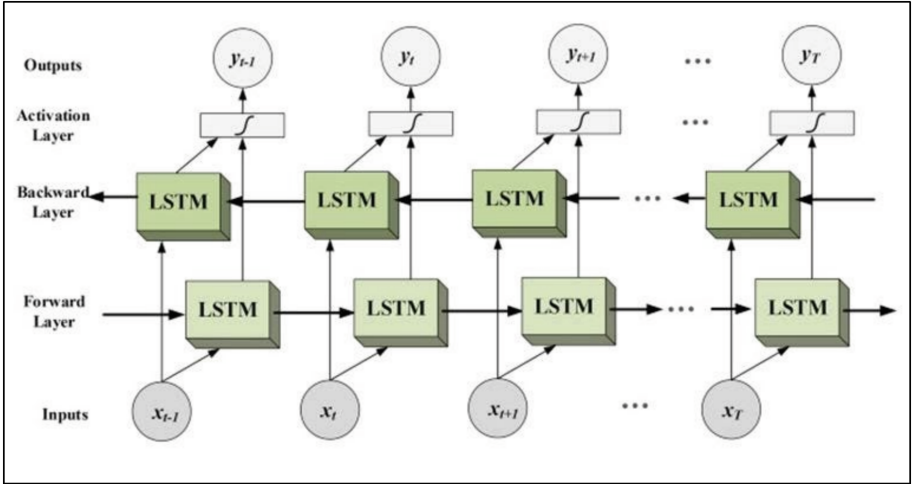


Fig. 3. Bidirectional RNN architecture unfolded in time.

Maximising the loglikelihood, or minimising the Cross Entropy (CE) measure of the training data, is the goal of conventional RNNLM training. The objective function is defined as follows for a series of words with a length of N_w in Eq. (1):

$$J^{\text{CE}}(\theta) = -\frac{1}{N_w} \sum_{i=1}^{N_w} \ln P_{\text{RNN}}(w_i | h_i) \quad (1)$$

$$P_{\text{RNN}}(w_i | h_i) = \frac{\exp(\theta_i^\top v_{i-1})}{\sum_{j=1}^{|V|} \exp(\theta_j^\top v_{i-1})} = \frac{\exp(\theta_i^\top v_{i-1})}{Z(h_i)} \quad (2)$$

If the probability of the word w_i given the history were defined by Eq. (2), The weight vector linked to word i at the output layer is represented as $h_i \cdot \theta_i$. The size of the vocabulary in the output layer is denoted as $|V|$, and $|V|$ The size of the vocabulary in the output layer is denoted as $|V|$, and v_{i-1} is the hidden history vector that was computed in the hidden layer. Typically, while training RNNLMs using CE, the gradient in Eq. (3) is:

$$\frac{\partial J^{\text{CE}}(\theta)}{\partial \theta} = -\frac{1}{N_w} \sum_{i=1}^{N_w} \left(\frac{\partial (\theta_i^\top v_{i-1})}{\partial \theta} \right) \quad (3)$$

In Eq. (4), the normalisation is applied to the entire output layer by the denominator term $Z(h_i) = \sum_{j=1}^{|V|} \exp(\theta_j^\top v_{i-1})$. In Section I, we covered how computing the RNNLM probabilities during test time and CE based training using the gradient information of [3] is computationally expensive.

3.2.1 Variance Regularization

Incorporating the standard CE-based objective function of Eq. (4) with the variance of the normalization term is one way to increase testing speed. The normalization term at the output layer can be disregarded during testing time, leading to substantial speed increases, if the variance of the SoftMax normalization term is deliberately minimized during variance training. The standard CE objective function is adjusted as:

$$J^{\text{VR}}(\theta) = J^{\text{CE}}(\theta) + \frac{\gamma}{2N_w} \sum_{i=1}^{N_w} (\ln Z(h_i) - \ln Z)^2 \quad (4)$$

In Eq. (5), the CE based training criterion is denoted as $J^{\text{CE}}(\theta)$, the softmax normalisation term is denoted as $Z(h_i)$, and the mean of the log scale normalisation term, computed over the N_w words in the train set, is denoted as $\ln Z$.

$$\ln Z = \frac{1}{N_w} \sum_{i=1}^{N_w} \ln(Z(h_i)) \quad (5)$$

The VR term's contribution can be adjusted via an adjustable parameter called γ . By explicitly minimising the variance of the softmax normalisation term, the above objective function can be maximised directly in Eq. (6) during VR based RNNLM training. Virtual reality training uses the gradient provided by:

$$\frac{\partial J^{\text{VR}}(\theta)}{\partial \theta} = \frac{\partial J^{\text{CE}}(\theta)}{\partial \theta} + \frac{\gamma}{N_w} \sum_{i=1}^{N_w} (\ln Z(h_i) - \ln Z) \quad (6)$$

The typical RNNLM probabilities calculated using Eq. (7) with normalisation, and $P_{\text{RNN}}(\cdot | h_i)$ is the CE gradient given in (5), where $\frac{\partial J^{\text{CE}}(\theta)}{\partial \theta}$ is the gradient. During evaluation time, VR permits the employment of a history-independent, constant SoftMax normalisation term. Therefore, the RNNLM probabilities are estimated as:

$$P_{\text{RNN}}^{\text{VR}}(w_i | h_i) \approx \frac{\exp(\theta_i^\top v_{i-1})}{Z} \quad (7)$$

Since the normalization is no longer needed, the calculation at the output layer is greatly reduced. With this change, the computation speed at the output layer can be increased by a maximum of $|V|$ times throughout test time, regardless of the amount of the output layer vocabulary, $|V|$. The effective context length of an RNN is quite short, despite the fact that RNNs can theoretically use arbitrarily lengthy histories. To overcome this problem, neural networks with Long Short-Term Memory were suggested. A LSTM neural network swaps out the RNN's nonlinear units in the hidden layer with memory blocks that hold value-storing memory cells and multiplicative gates that read, write, and reset these values. The value of a memory cell can be updated by collecting activations from both inside and outside a memory block, and memory cells can store information for lengthy periods of time.

The LSTM equations are given as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (8)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (9)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (10)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (11)$$

$$h_t = o_t \tanh(c_t) \quad (12)$$

where the input gate i_t , forget gate f_t , output gate o_t , and cell activation vector c_t are shown at time t , respectively. For example, W_{xi} is the weight matrix between the input layer and the input gates, and W_{xi} is the matrix that denotes the weights between different layers, gates, and cells. The logistic sigmoid function is represented by $\sigma(\cdot)$, and the gate and cell bias terms are b_i , b_f , b_o , and b_c . LSTM equations are used to compute the conditional word probabilities $P(w_t | w_{t-1}, h_{t-2})$ for language modelling after calculating h_t .

3.2.2 Bidirectional Recurrent Neural Networks

The input data is processed in both directions by bidirectional RNNs, which allows them to take use of both the past and the future respectively. Figures 2, and 3 displays the two-time step unfolding of a bidirectional RNN architecture. By iterating through the input sequence from $t = 1, \dots, T$, bidirectional RNNs construct a forward hidden layer

h_t^F and a backward hidden layer h_t^B , as illustrated in the figure. In order to create a single output layer, the following equations are used to merge the two hidden layers.

$$h_t^F = \tanh\left(W_{xh}^F x_t + W_{hh}^F h_{t-1}^F + b_h^F\right) \quad (13)$$

$$h_t^B = \tanh\left(W_{xh}^B x_t + W_{hh}^B h_{t+1}^B + b_h^B\right) \quad (14)$$

$$y_t = W_{hy}^F h_t^F + W_{hy}^B h_t^B + b_y \quad (15)$$

Replacing the activations of the hidden layers with memory blocks yields bidirectional LSTMs. The algorithm strikes a fair balance between training speed and performance, according to our data. Splicing sentences together to generate chunks of about the same length is another strategy for fast RNNLM training on GPUs. Reduces padding needed to equalize batch sizes [16]. When training or evaluating a language model using a neural network, the output layer computation often takes up most of the computational resources. We restrict the output vocabulary to the 10,000 most frequent terms in the vocabulary to make this expense more manageable. In the out-of-shortlist class, all words that do not belong in this output vocabulary are assumed to have the same likelihood. Classifying the output layer is another approach to improving the computational efficiency of neural network language models [3, 5]. One way to reduce the overall number of parameters is to use an input projection layer. A d-dimensional continuous feature vector is generated for every word in the lexicon using a shared weight matrix across word positions. The continuous feature representation of the word at time t, W_t , is given by x_t in Eq. (19). Using back propagation across time, word feature vector representations, weight matrices, and biases are learnt continuously. Train your RNN or LSTM language model using a 3D-dimensional linear projection layer, a 3D-dimensional hidden layer, and an output layer with 20,000 dimensions. Although the dimensions of the projection, hidden, and output layers are identical across all networks, the total number of parameters varies. Since LSTMs use memory blocks instead of hidden units, they use about four times as many parameters as RNNs. Because bidirectional models need to calculate a backward hidden state in addition to the forward hidden state, they use around twice as many parameters as unidirectional models. An additional LSTM layer with 128 hidden units and eight timestamps was used to keep an eye on these, along with a dropout layer and a batch normalization layer.

$$\widehat{c1}^{(l)} = \tanh\left(W1_c \left[a1^{(l)}, x^{(l)}\right] + b1_c\right) \quad (16)$$

$$\Gamma 1_u = \sigma\left(W1_u \left[a1^{(t-1)}, x1^{(<>)}\right] + b1_u\right) \quad (17)$$

$$\Gamma 1_f = \sigma\left(W1_f \left[a1^{(t-1)}, x1^{(t)}\right] + b1_f\right) \quad (18)$$

$$\Gamma 1_o = \sigma\left(W1_o \left[a1^{(<t-1>)}, x1^{(<>)}\right] + b1_o\right) \quad (19)$$

$$c1^{(t)} = \Gamma 1_u \widehat{c1}^{(i)} + \Gamma 1_f c1^{(t-1)} \quad (20)$$

$$a1^{(>)} = \Gamma 1_o \tanh(c1^{(\langle)}) \quad (21)$$

Deep layers and dropout happen after the LSTM layer. Therefore, the dense layer equation that follows is:

$$y_v = R(W2 \times a1^{(>)} + b2) \quad (22)$$

The feature maps from the audio-only portion's first dense layer and the visual-only portion's first LSTM layer were combined. Following this, we got by putting the values into Eqs. (3) and (12).

$$a_c = [a1^{(\langle)}, y^2] \quad (23)$$

The output feature map was used as input for the three dense layers of the DCNN. The third and final dense layer was created following the batch normalization and dropout layers. In order to get the network's predictions as near to the actual labels as feasible, we need to minimize the loss function. Deep Convolutional Neural Networks (DCNNs) have proven they can generalize their learning from one dataset or job to another. Because pretrained models may be fine-tuned for specific tasks, this is especially helpful when working with little labelled data.

4 Results

This section examines the evaluation and comparison of the performance of the proposed approach to the performance of existing methods based on a variety of metrics including training and testing accuracy, the number of epochs, the accuracy plot, the loss plot, the confusion matrix, and the classification matrices. The proposed speech recognition stage classification model is evaluated with the help of a Indian English dataset, which results in the provision of a robust framework for evaluating its performance. The findings of all the trials will be presented in the next part, which will shed light on how well the suggested model works and how far it can go in accurately detect speech recognition. A desktop computer system with i7 processor Intel RDX 8GB 3050Ti, twelve cores, and twenty-four threads, along with 16 GB of RAM, was used to conduct the research.

It goes on to compare and contrast the audiovisual, visual, and auditory models. Figure 4 shows that after 60 epochs of training on the English dataset, the training accuracy reached 98.74%. A test accuracy of 97.74% was attained. We evaluated the suggested model's training and validation accuracies according to the epoch count. Figure 4 displays the outcomes. Figure 5a shows the confusion matrix of word audio classifications.

The first word, "About," was correctly recognized 88% of the time and misclassified 12%. The second word, "bad," accurately foretold that it would be identified with a perfect score, free of misclassification. The graph shows that "Bottle" was properly predicted 95% of the time, whereas "About," "Bad," and "Cow" were mistakenly classified 10% of the time. Only a quarter of the time was the word "About" incorrectly identified as the fourth word, "Come," which had an accuracy rate of 97%. With an accuracy rate

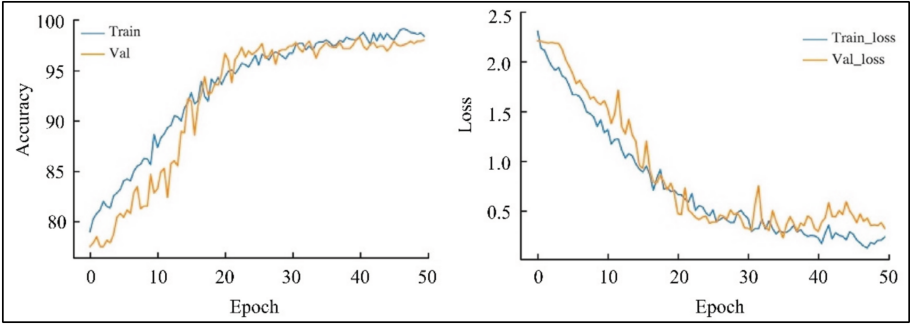


Fig. 4. Accuracy and loss graph of the proposed model

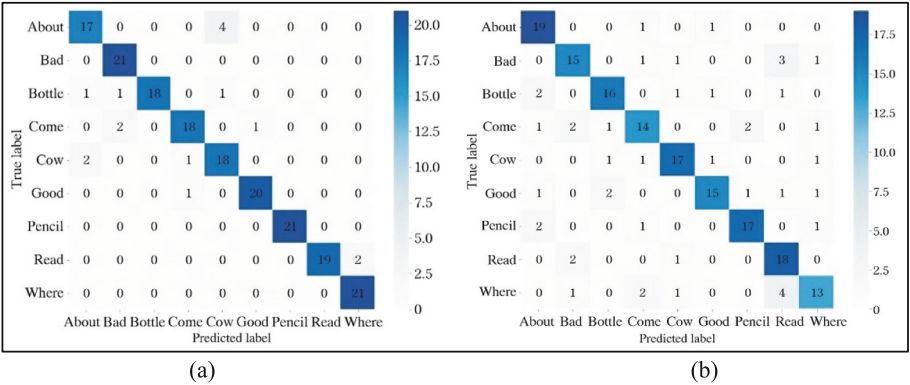


Fig. 5. Confusion matrix for the speech recognition. (a) Audio, (b) Visual Speech

of 97%, the fifth word “Cow” was more often accurately predicted than misclassified as “About” or “Coming” (14%). “Good” was the sixth word that was rightly classified 98% of the time and wrongly classified 2% of the time. “Pencil” was the sixth word that was identified and predicted accurately without a single mistake. “Read,” the eighth word, was expected 96% of the time but was mistakenly projected as “were” 3% of the time.

There was zero misclassification and a perfect prediction rate for the ninth word, “Where”. The audio database’s categorization report is shown in Table 2. With respect to the suggested system, the corresponding perceived precision, recall, accuracy, and F1-score were 95%, 96%, and 97%, respectively. The confusion matrix displays the total amount of words classified into different kinds. The visual model’s confusion matrix was used to train the English datasets. Figure 5b shows the correct and incorrect word visual speech classifications.

Table 2. Comparison of quantitative evaluation for automatic speech recognition.

Words	Accuracy	Precision	Recall	F1-score
About	88.33	89.23	90.33	90.13
Bad	91.62	90.12	91.32	94.32
Bottle	95.35	96.45	95.25	92.25
Come	97.74	95.34	94.24	92.24
Cow	92.42	90.12	91.22	93.22
Good	98.63	95.23	94.53	96.63
Pencil	94.13	97.23	96.13	96.73
Read	96.22	97.12	96.12	95.22
Where	97.10	95.00	94.10	97.80
Accuracy	93.23	94.23	97.23	94.13
Macro average	95.33	92.23	98.13	96.53
Weighted average	94.14	95.34	93.14	94.24

5 Conclusion

In order to overcome the computational obstacles associated with deep learning, this study introduces a novel dataset and a feed-forward neural network for audiovisual voice identification. After training on an Indian English dataset, the authors achieved a 98.71% success rate in training and a 97.74% success rate in testing. Integrating visual and auditory cues yields the best performance, according to the data. The proposed method outperforms the current methods. We compared the outcomes of audiovisual speech recognition with those of previous methods and software, and we looked at how well it worked for Indian English words. The authors propose a number of changes that might be made to future research that would lead to better findings. These include constructing a dataset from numerous perspectives rather than only looking at the speaker, using more datasets to train different deep learning algorithms, and trying out other machine learning and deep learning algorithms. Ultimately, researchers will train the system using a variety of deep learning and machine learning techniques using datasets created in noisier surroundings or with artificial noise injected. In the actual world, researchers can build a portable device and use it unsupervised using microcontrollers and microprocessors.

References

1. Gambhir, P., Dev, A., Bansal, P., Sharma, D.K.: End-to-end multi-modal low-resourced speech keywords recognition using sequential Conv2D nets. *ACM Trans. Asian Low-Resource Lang. Inform. Process.* **23**(1), 1–21 (2024)
2. Shilpa, B., Kumar, P.R., Jha, R.K.: LoRa DL: a deep learning model for enhancing the data transmission over LoRa using autoencoder. *J. Supercomput.* **79**, 17079–17097 (2023)

3. Chen, Z., et al.: Evolution and prospects of foundation models: from large language models to large multimodal models. *Comput. Mater. Continua* **80**(2) (2024)
4. Almadhor, A., Irfan, R., Gao, J., Saleem, N., Rauf, H.T., Kadry, S.: E2E-DASR: end-to-end deep learning-based dysarthric automatic speech recognition. *Expert Syst. Appl.* **222**, 119797 (2023)
5. Djeflal, N., Kheddar, H., Addou, D., Mazari, A.C., Himeur, Y.: Automatic speech recognition with BERT and CTC transformers: A review. In *2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM)*, vol. 1, pp. 1–8. IEEE (2023)
6. Gan, Y., Zhu, D.: The research on intelligent news advertisement recommendation algorithm based on prompt learning in end-to-end large language model architecture. *Innov. Appl. Eng. Technol.* 1–19 (2024)
7. He, Y., Seng, K.P., Ang, L.M.: Multimodal sensor-input architecture with deep learning for audio-visual speech recognition in wild. *Sensors* **23**(4), 1834 (2023)
8. Shitharth, G.B..M., Kumar, P.R.: Integrated machine learning model for an URL phishing detection. *Int. J. Grid Distrib. Comput.* **14**(1), 513–529 (2021)
9. Chang, S.Y., Zhang, C., Sainath, T.N., Li, B., Strohmaier, T.: Context-aware end-to-end ASR using self-attentive embedding and tensor fusion. In: *ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE (2023)
10. Wang, Q., Zhou, X., Li, H.: Speech-and-text transformer: Exploiting unpaired text for end-to-end speech recognition. *APSIPA Trans. Signal Inform. Process.* **12**(1) (2023)
11. Fukuda, T., Thomas, S.: Effective training of RNN transducer models on diverse sources of speech and text data. In: *ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE (2023)
12. Kumar, P.R., Shilpa, B.: An IoT-based smart healthcare system with edge intelligence computing. *Reconnoitering the Landscape of Edge Intelligence in Healthcare*, pp. 31–46. CRC Press (2024)
13. Reza, S., Ferreira, M.C., Machado, J.J., Tavares, J.M.R.: A customized residual neural network and bi-directional gated recurrent unit-based automatic speech recognition model. *Expert Syst. Appl.* **215**, 119293 (2023)
14. Radha, K., Bansal, M.: Closed-set automatic speaker identification using multi-scale recurrent networks in non-native children. *Int. J. Inf. Technol.* **15**(3), 1375–1385 (2023)
15. Kumar, D., Aziz, S.: Performance evaluation of recurrent neural networks-LSTM and GRU for automatic speech recognition. In: *2023 International Conference on Computer, Electronics & Electrical Engineering & their Applications (IC2E3)*, pp. 1–6. IEEE (2023)
16. Talaat, F.M.: Explainable enhanced recurrent neural network for lie detection using voice stress analysis. *Multimedia Tools Appl.* **83**(11), 32277–32299 (2024)
17. Alabdullah, B.I., et al.: Smart home automation-based hand gesture recognition using feature fusion and recurrent neural network. *Sensors* **23**(17), 7523 (2023)
18. Wang, X., Mi, J., Li, B., Zhao, Y., Meng, J.: CATNet: Cross-modal fusion for audio-visual speech recognition. *Pattern Recogn. Lett.* **178**, 216–222 (2024)
19. Ashok Kumar, L., Karthika Renuka, D., Naveena, K.S., Sree Resmi, S.: CRDNN-BiLSTM knowledge distillation model towards enhancing the automatic speech recognition. *SN Comput. Sci.* **5**(3), 304 (2024)
20. Oruh, J., Viriri, S., Adegun, A.: Long short-term memory recurrent neural network for automatic speech recognition. *IEEE Access* **10**, 30069–30079 (2022)
21. Li, D., Liu, J., Yang, Z., Sun, L., Wang, Z.: Speech emotion recognition using recurrent neural networks with directional self-attention. *Expert Syst. Appl.* **173**, 114683 (2021)
22. Wang, Q., Feng, C., Xu, Y., Zhong, H., Sheng, V.S.: A novel privacy-preserving speech recognition framework using bidirectional LSTM. *J. Cloud Comput.* **9**(1), 36 (2020)

23. Gibadullin, R.F., Perukhin, M.Y., Ilin, A.V.: Speech recognition and machine translation using neural networks. In: 2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), pp. 398–403. IEEE (2021)
24. Bhaskar, S., Thasleema, T.M.: LSTM model for visual speech recognition through facial expressions. *Multimedia Tools Appl.* **82**(4), 5455–5472 (2023)
25. Wand, M., Vu, N.T., 2018. Investigations on end-to-end audiovisual fusion. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 3041–3045
26. Wei, J., Yang, F., Zhang, J., et al.: Three-dimensional joint geometric-physiologic feature for lip-reading. In: 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 1007–1012. IEEE, (2018)
27. Wu, J., Xu, Y., Zhang, S., et al.: Time domain audio-visual speech separation. In: IEEE Automatic Speech Recognition and Understanding Workshop, ASRU, pp. 667–673. IEEE (2019)
28. Yu, W., Zeiler, S., Kolossa, D.: Multimodal integration for large-vocabulary audio- visual speech recognition. *Eur. Signal Process. Conf.* **2021**(1), 341–345 (2021)
29. Yuan, Y., Tian, C., Lu, X.: AuXiliary loss multimodal GRU model in audio-visual speech recognition. *IEEE Access* **6**, 5573–5583 (2018)



Advanced Deep-Learning Framework for Comprehensive Audio Forensics: Classification and Frequency Analysis

Swetanshu Agrawal, Astitva Veer Garg, and D. Anitha

SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India
swetanshusbp@gmail.com, gargveerastitva@gmail.com,
anithad@srmist.edu.in

Abstract. Aiming at an automated method for forensic audio analysis, the Advanced Deep-Learning Framework for Comprehensive Audio Forensics: Classification and Frequency Analysis searches Leveraging deep learning techniques—more especially, combining a hybrid model with Convolutional Neural Networks (CNNs) with Bidirectional Long Short-Term Memory (BiLSTM) layers—enhanced by attention mechanisms—the proposed framework classifies and evaluates several audio types. Essential for forensic investigations, the architecture is designed to discover abnormalities, split frequency ranges, and manage several audio sources.

CNNs derive features from preprocessing and frequency segmentation. For sequential analysis, bi-LSTM layers inherit these properties, therefore ensuring a robust understanding of temporal trends. The attention mechanism helps to enhance the model even further sharpens the model's focus on relevant frequency bands for exact classification and anomaly identification. In many environmental contexts, our deep learning architecture enables the recognition of significant acoustic characteristics.

The tool can control several forensic audio sources, ranging in frequency from skating to sleeping to snoring. This provides law enforcement and forensic experts with a quick tool for objective audio analysis since it offers flexibility in practical forensic environments. Apart from classification, the framework consists in audio segmentation, frequency detection, and anomaly detection, so augmenting the overall forensic analysis of audio data.

Keywords: Audio Forensics · Deep Learning · CNN-BiLSTM Hybrid Model · Attention Mechanism · Frequency Analysis · Audio Classification · Anomaly Detection · Temporal Features · Spectral Features · Forensic Investigations

1 Introduction

1.1 Background

Within forensic science, audio forensics is the study of sound recordings as evidence for court cases. Usually missed by human witnesses, these recordings provide important fresh perspectives on events or ambient noises [1]. Since conventional audio analysis

methods are labor-intensive and prone to biases, they underline the need of automated systems able to offer accurate and fast analysis [2]. Deep learning techniques enable one extract information including frequency patterns, temporal dynamics, and particular sound qualities [3] and offer significant revolutionary promise in this discipline. To recognize and analyze several audio formats [4], the proposed Advanced Deep Learning Framework comprises hybrid CNN-BiLSTM architectures reinforced by attention mechanisms, thereby tackling background noise and distortions. This scalable approach offers consistent tool for accurate objective forensic audio analysis.

1.2 Value of the Research

Accelerating the analysis of enormous amounts of audio data—needed for surveillance and evidence evaluation—a deep learning automated system changes forensic investigations [2]. Effective audio input processing by the model increases forensic procedure dependability and speed of operation.

1.3 Related Works

1.3.1 OpenSMILE

Widely used before data entering machine learning models, OpenSMILE is a flexible tool for audio feature extraction [1]. While it does well in feature extraction, alternative models are needed in classification problems.

1.3.2 WaveNet

Originally developed for voice synthesis, WaveNet’s generative architecture offers forensic audio analysis possibilities specifically for the identification of minute sound components in raw data [3].

1.4 Approach

1.4.1 Compositional Analysis and Data Management

The method examines many forensic audio sources in turn, including human speech and ambient noise. Important preprocessing techniques assure the effective collecting of spectral and temporal features [4] including feature extraction (e.g., STFT, MFCCs) and noise reduction.

1.4.2 Create a Structural Framework

Emphasizing temporal and spatial aspects of audio, the hybrid CNN-BiLSTM model compiles localized spectrogram data while BiLSTM layers assess sequential relationships. Focus on important auditory components assures resilience and accuracy by use of attention techniques [5, 8].

1.4.3 Classification and Frequency Study

After training, the model detects frequency components and classifies audio into identified categories (e.g., speech, ambient noise) therefore allowing forensic investigators to identify certain sounds or audio modifications [5].

1.5 Uses in Law Enforcement Investigation

The architecture opens forensic uses including:

- finding speakers for surveillance audio or phone calls [4]
- categorizing ambient noises for reconstruction of a crime scene detecting manipulated audio evidence [5].
- The framework's Analyzing frequency components helps one find sources or settings of sound [3].

1.6 Future Discoveries

Future research aims to discover modified audio and expand capabilities for cross-lingual or multilingual speech recognition utilizing generative models like VAEs or GANs, therefore enhancing its international forensic significance [5].

2 Literature Review

The study “Deep Audio Analyzer: A Framework for Audio Forensics” underlines how spectrum measures like as spectral centroid and bandwidth help the system to differentiate changed from authentic audio recordings. For forensic investigations needing consistent audio evidence processing, scalable implementation in real-world forensic contexts is made possible by neural networks built using tools like SpeechBrain. Emphasizing spectral centroid and bandwidth enables the system to differentiate actual from manipulated data [6].

This work emphasizes the need of identifying synthetic voices in forensic investigations, especially when voice modification affects audio evidence. The paper “ADD 2023: Towards Audio Deepfakes” proposes a transfer learning-based method for identifying audio deepfakes using pre-trained speech synthesis models and finetuning them with frequencies modulated and MFCC to improve accuracy over both small and large datasets [7].

This approach is resilient and supported by King Saud University’s speech corpus, making it suitable for forensic scenarios that require ambient audio evidence validation. In “Digital Audio Forensics: Microphone and Environment Classification Using Deep Learning”, CNN and RNN architecture deep learning systems recognize recorded environment and microphone type. Audio recordings’ veracity depends on the microphone and calmness in loud environments [3].

The framework detects deepfake audio generated by TTS systems with high accuracy, proving its usefulness in forensic audio applications like synthetic voices linked to crime. In “Deep Learning Framework for Audio Deepfake Detection,” CNNs for acoustic feature extraction and BiLSTM layers for temporal analysis are used to detect bogus

sounds. Attention approaches focus the model on auditory components likely to change [8, 9].

CNNs, RNNs, and transformers are evaluated in “Audio Classification Using Deep Learning: A Survey” for forensic audio challenges, highlighting their benefits and weaknesses. In spectrogram and MFCC forensic audio analysis, we go into detail. This evaluation helps choose model architectures for forensic audio analysis. Comprehensively deep learning techniques for ambient noise, voice, and music are evaluated, highlighting CNNs, RNNs, and transformers’ merits and weaknesses [10].

CNN is used to merge audio segments in “Splicing Detection in Digital Audios Using CNN”. Using audio source spectrograms, the model detects tiny frequency spectrum changes, suggesting manipulation. This automatic audio manipulation detection method is useful in forensic contexts where audio integrity is crucial and provides a constant means of checking recordings [11].

Audio forensic investigators require this work to manage synthetic voice recordings as speech modification and synthesis become more widespread in forensic cases. Mel-frequency cepstral coefficients (MFCC) and CNNs help to classify real from synthetic audio. Using deepfake modifications [9], the article demonstrates how successfully MFCC characteristics distinguish synthetic from real-world recordings.

“Audio Event Recognition Using Deep Learning Models” detects ambient and anomalous noise. CNN feature extraction and BiLSTM sequential analysis organize gunshots and footfall. In forensic audio analysis, automatically selecting probable relevant noises can help investigations by finding crucial events from massive audio data [12].

The research “Detection of False Audio Using Machine Learning and Deep Learning Techniques.” compares conventional machine learning methods like SVMs to more recent deep learning models, mostly CNNs and LSTMs, to show that deep learning models outperform traditional methods in detecting minute audio data changes and identifying false audio [13].

Using MFCCs, spectral contrast, and temporal dynamics, “Sonic Sleuth: A Deep Learning Model for Audio Deepfake Detection” presents a multi-branch neural network that detects synthetic audio produced by advanced text-to-speech systems, addressing forensic investigation’s growing deepfake audio issues. Sonic Sleuth can handle high-quality deepfakes, protecting court speech records [14].

In forensics, MFCCs, chroma, and zero-crossing rate improve model accuracy. This strategy helps legal forensic investigators identify between crucial audio events and background noise in audio evidence. The study “Deep Learning for Audio Signal Processing” examines numerous deep learning models for environmental noise detection and classification, highlighting CNNs and RNNs’ effectiveness in non-speech audio processing [15].

CNN’s “A Comprehensive Survey on Voice Forgery Detection Using Deep Learning” emphasizes the difficulties of recognizing artificial voices and examines deep learning models like GANs and RNNs for voice forgery detection. Since digital evidence may include forged voice recordings, the study compares models trained on actual and altered datasets and emphasizes its relevance to forensic audio [16].

In forensic investigations requiring scene reconstruction, it helps identify the recording environment, verifying the context of an audio recording. “Neural Networks for Acoustic Scene Classification” tests CNN and BiLSTM deep learning-based acoustic environment classification for spatial and temporal sound feature acquisition [17].

Deep learning can automate, scale, and accurately analyze audio evidence to detect deepfakes and tampered recordings, improve low-quality audio, and integrate multimodal data, changing forensic investigations.

3 Dataset Explanation

Across 10 urban sound categories—air_conditioner, car_horn, children_playing, dog_bark, drilling, engine_idling, gun_shot, jackhammer, siren, and street_musicians—the “UrbanSound8K” dataset comprises 8732 short audio clips (≤ 4 s). Freesound field recordings provide these noises; metadata for classification—file names, Freesound IDs, start and end times, salience (foreground or background), class IDs, and fold assignments [31] accompanies these sounds.

Ten predetermined folds separate the dataset for 10-fold cross-valuation, therefore guaranteeing consistent experimentation and avoiding overlap between training and testing samples—which would otherwise inflate performance measures [32]. Maintaining this framework for result integrity and comparability with previous investigations is highly advocated of researchers [33].

Widely utilized in urban sound classification challenges, the dataset helps to create automatic sound recognition models fit for noisy metropolitan settings. Its methodical design and information inclusion make it an important tool for developing and accessing strong sound categorization systems [34].

4 Methodology

This work aggregates deep learning methods for audio categorization and frequency analysis to efficiently automate forensic audio processing. Here is a list of the several steps the procedure consists in.

4.1 Preprocessing and Audio Input

The user (as depicted in Fig. 1) offers a preprocessed audio file aimed to balance the signal and eliminate noise. Getting the data suitable for more investigation allows it to become spectrograms or finds important trends. First splitting the audio into discrete frequency ranges [18] the frequency segmentregator module helps to enable targeted processing.

4.2 Frequency Extraction and Frequency Segmentation

The frequency band separator divides the preprocessed audio across many frequency bands (as depicted in Fig. 2), therefore isolating the streams pertinent for forensic analysis. One may determine complete temporal and spectral features of the audio by use of Spectral contrast and Mel-frequency cepstral coefficients (MFCCs). This essential stage [18] determines whether auditory occurrences or manipulations identify themselves.

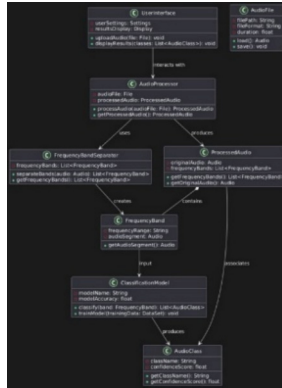


Fig. 1. Preprocessing audio input and segmenting frequencies for downstream analysis.

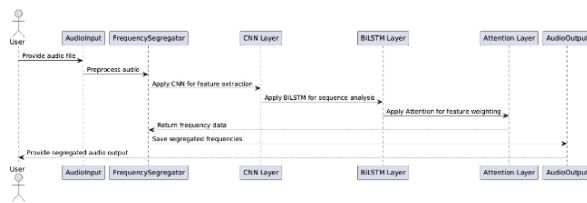


Fig. 2. Frequency segmentation using bandpass filtering to isolate distinct acoustic components.

4.3 Feature Processing CNN and BiLSTM Layers

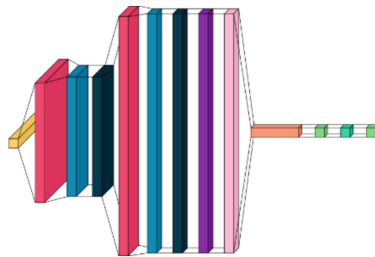


Fig. 3. Feature extraction using CNN followed by temporal pattern recognition via BiLSTM.

Using a convolutional neural network (CNN) allows one to find pitch variations in processed segmented audio. Analyzed are sequential patterns arising from layers of bidirectional long short-term memory (BiLSTM). This clarifies the temporal relations in auditory events such as gunshots or speech [19] (Fig. 3).

4.4 Feature Weighting Attention Mechanism

An attention mechanism (Fig. 4) fed the BiLSTM outputs allocates either more or less weights depending on their relative significance. This highlights important audio components to let more precisely the system discover changes or abnormalities [19].

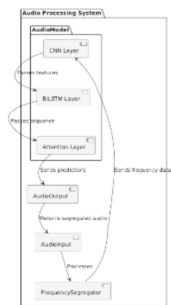


Fig. 4. Attention mechanism emphasizing relevant segments of the BiLSTM output for enhanced classification.

4.5 Separating and Classifying Audio Events

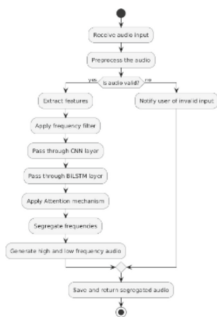


Fig. 5. Segregation and classification of high- and low-frequency components in forensic audio.

Separating the audio into low- and high-frequency domains helps the technique to stress significant audio characteristics. These results enable forensic investigators to evaluate evidence by means of changed or dubious audio parts [19] (Fig. 5).

This deep learning-based system automates forensic audio processing by using CNN, BiLSTM, and attention techniques thereby improving the validity of forensic investigations. It reaches amazing accuracy and scalability.

5 Explanation of the Deep Learning Model with Mathematical Foundations

5.1 Feature Extraction and Input Layer

The model processes audio data by extracting Mel Frequency Cepstral Coefficients (MFCCs), emphasizing human-perceived frequencies. Starting with an input layer shaped as (None, 40, 1), these coefficients reflect the power spectrum of the audio signal over time.

5.1.1 Fourier Transform

The audio signal $x(t)$ is first transformed from the time domain to the frequency domain using the Discrete Fourier Transform (DFT): (Eq. 1)

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi kn}{N}}, k = 0, 1, \dots, N-1 \quad (1)$$

Where $X[k]$ represents the frequency bin coefficients, and N is the number of time-domain samples [19].

5.1.2 Mel Filterbank

Frequency bins are grouped using a Mel scale filter:

$$f_{mel} = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \quad (2)$$

This emphasizes frequencies perceptible to humans [19] (Eq. 2).

5.1.3 Logarithm of Amplitudes

The Mel-filtered amplitudes are transformed logarithmically [21]:

$$\log(\text{Energy}) = \log\left(\sum_k X[k] \cdot H[k]\right) \quad (3)$$

Where $X[k]$ is the Mel filter applied to the k -th frequency bin (Eq. 3).

5.1.4 Discrete Cosine Transform (DCT)

Compresses Mel-scaled energies into fewer coefficients:

$$C(n) = \sum_{k=0}^{K-1} \log(\text{Energy}[k]) \cdot \cos\left[\frac{\pi n(2k+1)}{2K}\right], \quad n = 0, 1, \dots, N-1 \quad (4)$$

Resulting MFCCs are used as input to the neural network [20] (Eq. 4).

5.2 Conv1D Layers: Local Feature Extraction from Time-Series Data

The Conv1D layer uses 64 filters and a kernel size of 3 to capture local features from MFCCs. For a 1D convolution, the output at time step t is:

$$y_t = \sigma \left(\sum_{i=0}^{k-1} x_{t+i} \cdot w_i + b \right) \quad (5)$$

Where x_{t+i} is the input feature, w_i the kernel weight, b the bias, and σ the activation function (ReLU) [21]. MaxPooling1D reduces dimensionality by selecting the maximum value over segments (Eq. 6):

$$y_t = \max(x_t, x_{t+1}, \dots, x_{t+p-1}) \quad (6)$$

This highlights the most important features [21].

5.3 Batch Normalization

Batch normalization stabilizes training by normalizing activations:

$$\hat{x} = \frac{x - \mathbf{m}}{\sqrt{s^2 + \epsilon}} \quad (7)$$

Where μ is the mean, σ^2 the variance, and ϵ a small constant (Eq. 7). Trainable parameters γ and β allow the model to scale and shift normalized values (Eq. 8):

$$y = \gamma \hat{x} + \beta \quad (8)$$

This helps improve convergence and stability [22].

5.4 Bidirectional LSTM Layer: Capturing Long-Term Dependencies

A Bidirectional LSTM layer with 64 units captures past and future context. The gates controlling the flow of information are:

5.4.1 Forget Gate [23] (Eq. 9)

$$f_t = s(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (9)$$

5.4.2 Input Gate (Eq. 10, Eq. 11)

$$i_t = s(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (10)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (11)$$

5.4.3 Update the Cell State (Eq. 12)

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (12)$$

5.4.4 Output Gate: (Eqs. 13, 14)

$$o_t = s(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (13)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (14)$$

The bidirectional architecture processes input in both temporal directions [23]. This gated memory mechanism is particularly effective in distinguishing temporal anomalies in synthetic audio, as validated in [8].

5.5 Attention Layer: Enhancing Focus on Key Features

The attention mechanism computes a weighted sum of hidden states from the LSTM:

5.5.1 Scores: (Eq. 15)

$$e_t = \text{score}(h_t, s_{t-1}) \quad (15)$$

5.5.2 Weights: (Eq. 16)

$$a_t = \frac{\exp(e_t)}{\sum_{t'} \exp(e_{t'})} \quad (16)$$

5.5.3 Context Vector: (Eq. 17)

$$c_t = \sum_t a_t h_t \quad (17)$$

This allows the model to focus on important input features for better classification [24].

5.6 Dense Layers and Dropout: Final Classification

After attention, Dense layers classify audio into 10 classes using a softmax activation (Eq. 18):

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^K \exp(z_j)} \quad (18)$$

Dropout with a 40% rate prevents overfitting by randomly deactivating neurons during training [25].

5.7 Model Training and Evaluation Metrics

The model is trained using the UrbanSound8K dataset. It employs categorical cross-entropy loss and the Adam optimizer to update weights.

5.7.1 Cross-Entropy Loss

$$L = - \sum_{i=1}^K y_i \log(p_i) \quad (19)$$

Where y_i is the true label, and p_i is the predicted probability for class i [26].

Evaluation metrics include accuracy, confusion matrices, and ROC-AUC scores [29] (Eq. 19).

5.8 Pipeline for Prediction and Testing

5.8.1 Audio Preprocessing

Audio is preprocessed using MFCC extraction and bandpass filtering to focus on key frequency bands [27].

5.8.2 Bandpass Filtering

A bandpass filter is applied to isolate specific frequency bands of interest, which could contain key information for classifying the audio.

5.8.3 Short-Time Fourier Transform (STFT)

STFT generates spectrograms (Eq. 20) representing frequency content over time [28]:

$$STFT(t, \&) = \sum_{n=-\infty}^{\infty} x[n] \omega[n - t] e^{-j\&n} \quad (20)$$

5.8.4 MFCC Extraction for Prediction

Extracted MFCCs are passed through the model to classify each audio segment.

5.8.5 Prediction

The softmax layer predicts classes based on learned patterns, identifying sounds such as “siren” or “dog bark.”

This pipeline effectively combines signal processing with deep learning for forensic audio analysis and real-time classification [29].

5.9 Real Time Audio Processing Pipeline

The real-time processing system detects, classifies, and localizes audio events using a CNN-BiLSTM-Attention model. It leverages MFCC feature extraction, STFT, and bandpass filtering to isolate key audio signals. This pipeline enables accurate forensic analysis, noise reduction, and classification tasks like identifying sirens or barking in dynamic settings [29].

6 Model Comparisons

Model	Architecture	Key Layers	Test Accuracy (%)	Observations
Conv1D-based Model	Conv1D, MaxPooling1D, Dense, Dropout	Conv1D (2 layers), MaxPooling1D, Dense (2 layers with dropout), Softmax	87.69	Simpler architecture; good baseline performance; limited ability to capture temporal dependencies
GRU-based Model	GRU, Dense, Dropout	GRU (1 layer), Flatten, Dense (2 layers with dropout), Softmax	87.75	Handles temporal dependencies better; slightly better than Conv1D-only model
GRU-BiLSTM-Attention Model	GRU, Bidirectional LSTM, Attention, Dense, Dropout	GRU (1 layer), BiLSTM (1 layer), Attention Mechanism, Flatten, Dense (2 layers with dropout), Softmax	88.9	Attention mechanism helps focus on important features; complex architecture with slightly better results compared to GRU and Conv1D-based models
CNN-BiLSTM-Attention Model	Conv1D, MaxPooling1D, BatchNormalization, Bidirectional LSTM, Attention, Dense, Dropout	Conv1D (2 layers), MaxPooling1D, BiLSTM (1 layer), Attention Mechanism, Dense (2 layers with dropout), BatchNormalization, Softmax	91.36	Best performance; combines spatial, temporal, and feature importance efficiently; balances complexity and accuracy; suitable for final deployment

7 Output Analysis and Prediction

7.1 Model Analysis

While validation accuracy plateaus, on the training set the Training and Validation Accuracy Over Epochs (Fig. 6) shows the model achieving essentially perfect accuracy. Once more emphasizing overfitting is this discrepancy between training and validation

accuracy. The ensuing flattening of validation accuracy reveals the limits of the generalization of the model, even if the early fast growth in accuracy indicates that the model fits the data quite well.

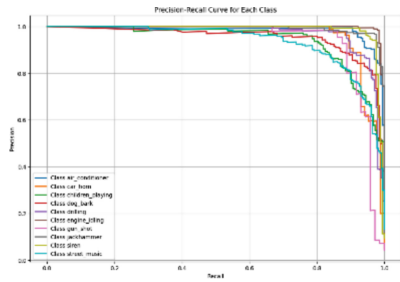


Fig. 6. Training vs. validation accuracy curve showing overfitting beyond early epochs.

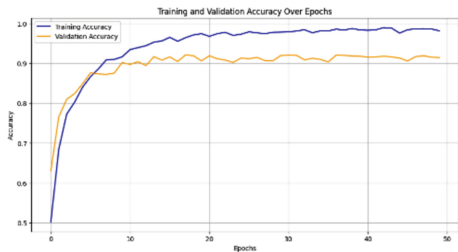


Fig. 7. Per-class precision-recall curves demonstrating strong classification performance across categories.

Every Class’s Precision-Recall Curve (Fig. 7) offers a complete study of how well the model could balance recall with precision for every class. The chart shows that most classes preserve a strong balance with high accuracy and recall values, thereby showing good classification performance for urban noises. However, in certain of the curves—especially with more noisy or uncertain data—reductions in sections of the curve point to where the model might be having trouble detecting specific sounds.

Plotting a precision-confidence curve (Fig. 8) shows that, as the model’s confidence increases, accuracy increases generally across all classes. This suggests that, when the model is more sure about its projections, it is usually correct. Particularly in classes like street_music and children_playing, lower confidence levels show some early decreases in accuracy; so, it suggests that feature similarity between classes may be causing classification issues.

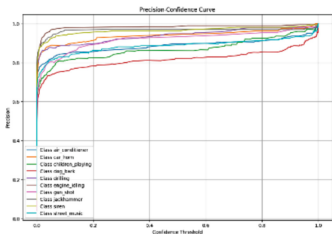


Fig. 8. Precision vs. confidence curve indicating higher accuracy with increased model confidence.

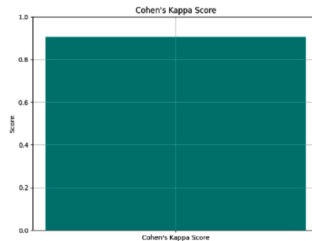


Fig. 9. Cohen’s Kappa Score indicating high inter-annotator agreement and model reliability.

Measuring inter-annotator agreement, Cohen’s Kappa Score (Fig. 9) reveals a high value close to 1, therefore suggesting consistent and reliable model predictions. This score significantly helpful in correcting class discrepancies as it considers the possibility

agreement. A high Cohen’s Kappa value denotes the dependability and accuracy of the model’s predictions for multiple urban sound categories.

Plotting true positive rates against false positive rates enables ROC curves for all classes to be examined more closely about the model’s classification capacity. For most classes—including air_conditioner and dog_bark—the AUC values, which are close to 1, suggest that the model has high capacity to differentiate between true and false positives. Lower AUC values for categories like street music indicate some uncertainty and room for improvement in separating overlapping sound parts.

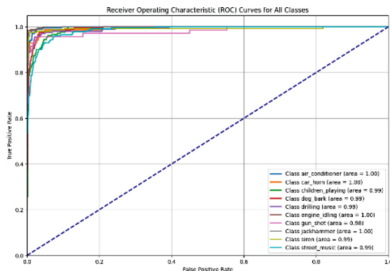


Fig. 10. Per-class distribution of false positives vs. false negatives illustrating model challenges in overlapping classes.

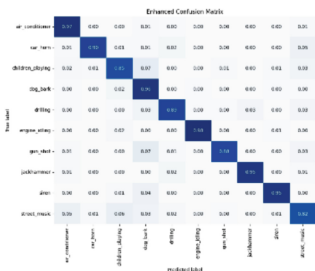


Fig. 11. Enhanced confusion matrix showing class-wise prediction strengths and errors.

Every class’s false positives and false negatives are shown by the graph False Positives against False Negatives per Class (Fig. 10). High false negatives in street music, for example, show that the model struggles to correctly classify these events; high false positives in dog_bark, on the other hand, show that the model regularly mislabels unrelated noises as dog barks. This difference highlights how challenging it is to combine sounds of comparable acoustic quality.

The Enhanced Confusion Matrix (Fig. 11) helps one to evaluate the model for every class more easily. Although misclassifications between children’s playing and street music highlight challenges the model has in separating similar sounds, high diagonal values imply most classes are exactly classified. The confusion matrix is a helpful instrument for finding where the model produces errors and guides modifications in data preparation and feature extraction.

A fundamental graphic representation of audio signals, the spectrogram (Fig. 12) depicts the frequency component distribution across time. The color intensity corresponds to energy levels; brighter areas at specific frequencies imply greater sound energy. This is the reason the model recognizes components from various urban sound categories. The spectrogram records significant time-frequency data needed for the model’s classification.

Two somewhat significant representations are provided by the Waveform and MFCC (Fig. 13). While the MFCC plot (right) provides frequency characteristics, which are needed for differentiating sound types, the waveform plot (left) shows the temporal aspects of the audio signal by showing its amplitude with time. These representations enable the model to acquire spectral as well as temporal input, so accurate classification depends on them.

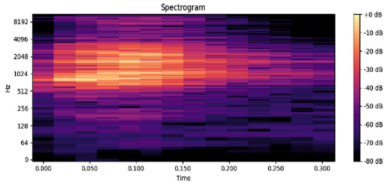


Fig. 12. Spectrogram illustrating the energy distribution of audio signals across frequency and time.

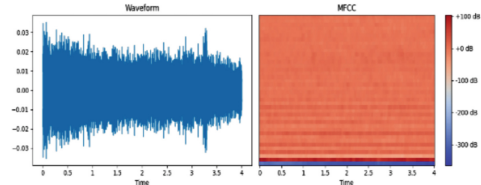


Fig. 13. Waveform and MFCC visualization for temporal and spectral representation of the input audio.

The model’s performance in predicting every sound type depicted in the previous Confusion Matrix (Fig. 14). While certain off-diagonal misclassifications are clear-cut—between `children_playing` and `street_musicians`, high diagonal scores show correct predictions. This uncertainty suggests the need of more precisely customized feature extraction or additional training data to improve the model on related-sounding categories. Though the model performs well in most categories, the confusion matrix reveals that overlapping audio features resulted in certain misclassifications. Sometimes the overlapping, complex sound frequencies of children playing and street musicians caused confusion. Gun_shot and drilling experienced issues due to their fast, transitory sound qualities; dog_bark was often misclassified as siren due of overlapping tone patterns. These problems included feature overlap, background noise, and feature extraction techniques unable to identify the minute changes in these produced sounds. Furthermore causing misclassifications were the imbalanced dataset and poor class representation.

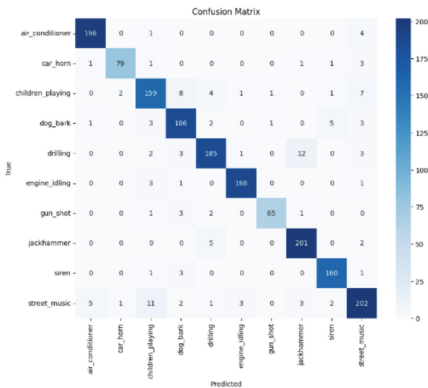


Fig. 14. Detailed confusion matrix indicating challenges in distinguishing overlapping acoustic events.

These representations help to clearly show the strengths and weaknesses of the model, thereby offering opportunities for future enhancement including overfitting, feature extraction optimization, and dataset balance to control demanding sound classes. The taken-together figures provide a good understanding of the model’s ability to classify urban sounds.

7.2 Pipeline Analysis

Different figures show notable audio processing properties and analysis grounded on the above-described code results. (Fig. 15) shows three primary peaks in the frequency spectrum observed in the audio file at around 150 Hz, 236 Hz, and 667 Hz. Based on these frequency ranges, the approach forecasts two events of “engine idling” and one of “dog bark”; thereafter, these frequencies are passed through a bandpass filter for further analysis. This graphic captures the frequency components in the audio stream by showing how the Short-Time Fourier Transform (STFT) peaks where energy is concentrated.

(Fig. 16) shows the MFCCs—mel-frequency cepstral coefficients—for the treated audio. This heat map helps one to grasp the special characteristics of the sound signal by showing the strength of many frequency ranges over time. The smooth gradients and clear lower-frequency components show the relevance of MFCCs in separating audio classes since this representation reduces complex audio input into an acceptable format for the deep learning network to comprehend. The color intensity in the graphic demonstrates how sound energy fluctuates over multiple MFCC coefficients over time, so accurate audio classification depends mostly on this factor.

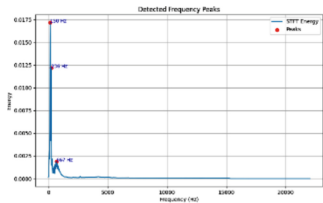


Fig. 15. STFT-based frequency peaks identifying key acoustic events in real-time processing.

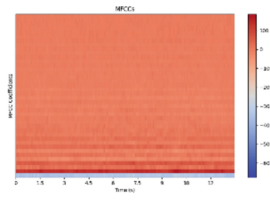


Fig. 16. MFCC heatmap of the audio clip visualizing frequency strength variation over time.

(Fig. 17) shows the waveform of the input audio, therefore providing a visual representation of the amplitude variation throughout the audio stream across time. This graph shows waveform fluctuations at specified periods that correspond with the commencement of more intense or louder sound events. For the 7–10 s interval, for example, strong amplitude spikes might indicate an auditory event especially. The waveform plot allows one to understand the temporal nature of the sound and also displays how loudness changes—a relationship that would be relevant to different sound genres.

(Fig. 18), the spectrogram, logs the frequency content of the audio stream over time. This graph vividly illustrates how different frequencies add to the audio along the time axis. Important in many kinds of sounds, the high energy patches in warm colors complement more dominating frequency components. Strong energy lower frequency zones, for example, might complement ambient noises like engine idle. By way of the thorough frequency breakdown of this spectrogram, the CNN-BiLSTM-attention model extracts frequency information necessary for class prediction.

Figure 19 is last, the first shown output from the command line indicates the frequency bands and their corresponding expected classes, where the system essentially

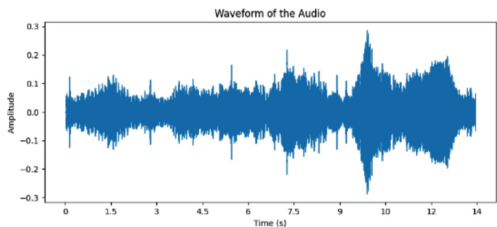


Fig. 17. Waveform of the audio signal highlighting amplitude variation and peak events.

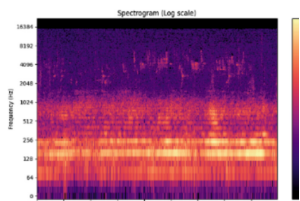


Fig. 18. Spectrogram mapping frequency content against time for forensic sound classification.

```
Energy per frequency: [1.5564400e-04 1.9972291e-04 4.4072798e-04 ... 2.1834896e-07 2.1694308e-07
1.8482170e-07]
Detected frequency peaks: [150.73242188 236.86523438 667.52929688]
1/1 ----- 1s 506ms/step
1/1 ----- 0s 31ms/step
1/1 ----- 0s 21ms/step
Frequency Band (50.732421875Hz to 250.732421875Hz) predicted class: engine_idling
Frequency Band (136.865234375Hz to 336.865234375Hz) predicted class: engine_idling
Frequency Band (567.529296875Hz to 767.529296875Hz) predicted class: dog_bark
PS E:\minor Project>
```

Fig. 19. CLI output of frequency-to-class prediction pipeline showing mapping of frequencies to sound events.

recognizes distinct classes depending on frequency peaks, so highlighting the requirement of frequency-based analysis in audio classification issues. Each frequency segment is band-pass filtered and handled to provide accurate class predictions, therefore reflecting the ability of the model to generalize over many frequency ranges.

8 Scenario

Among the outside noise the testing pipeline could faithfully identify and classify were automobile horns, dog barking, street music, and drilling. The pipeline separates powerful audio components using frequency-specific bandpass filtering, therefore ensuring clarity and noise reduction. Peak frequency detection and the Short-Time Fourier Transform (STFT) help one to link classified sounds to their matching frequency ranges. Figures 20, 21, 22, 23 and 24 show results that in challenging environments the model can identify instances of overlapping sounds and locate frequencies. This shows how dynamically real-world settings including urban noise monitoring, intelligent surveillance systems, and audio forensics could profit from it.

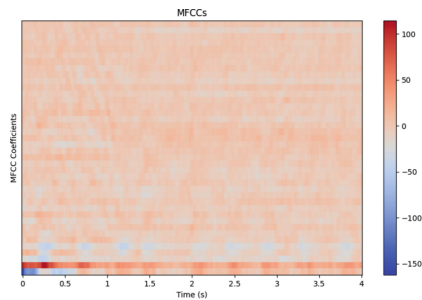


Fig. 20. Detection of real-world overlapping sounds such as dog barking, engine idling, and street music across various acoustic environments.

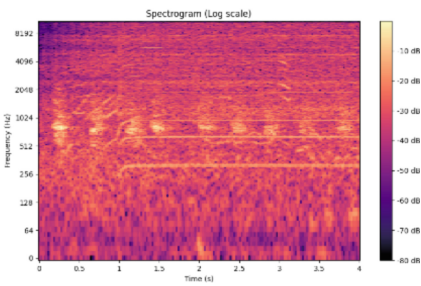


Fig. 21. Detection of real-world overlapping sounds such as dog barking, engine idling, and street music across various acoustic environments.

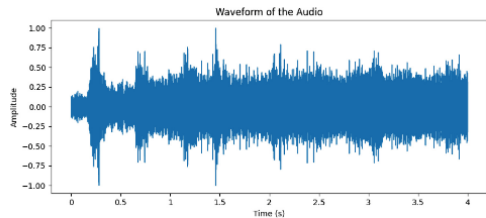


Fig. 22. Detection of real-world overlapping sounds such as dog barking, engine idling, and street music across various acoustic environments.

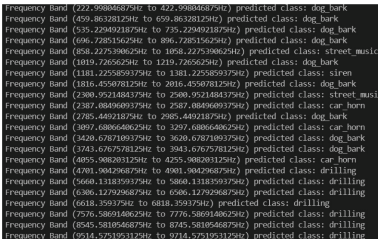


Fig. 23. Detection of real-world overlapping sounds such as dog barking, engine idling, and street music across various acoustic environments.

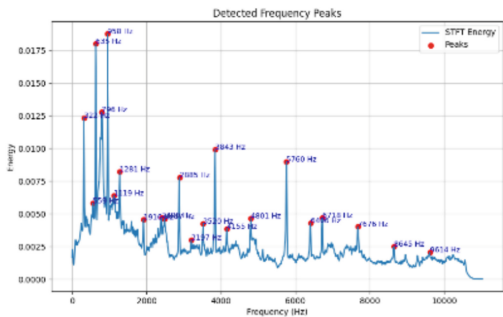


Fig. 24. Detection of real-world overlapping sounds such as dog barking, engine idling, and street music across various acoustic environments.

9 Conclusion

Regarding digital forensics, our results highlight the great potential of deep learning models, primarily CNN-BiLSTM architecture reinforced by attention mechanisms. By use of feature extraction techniques including MFCCs and sequential audio input processing capability, the model amazingly correctly recognizes ambient noises. Spectrograms, waveforms, and frequency peak detection provide correct analysis of audio data in forensic situations, therefore enabling major portion of the discovery of important sounds in demanding acoustic conditions. Simultaneously, this is adaptable and scalable. Beyond present categorization capabilities, it can investigate a wide spectrum of auditory signals and may incorporate more complicated sounds such mechanical noises, environmental disturbance, or human speech.

The ability of the framework to support the analysis of crime scene audio recordings and the reconstruction of maybe overlooked auditory signals is fairly important for law enforcement and forensic investigations. Furthermore, real-time functioning of apps has significant implications. The capacity of the model to discern between important noises, such gunshots, breaking glass, or screams, has the capability to transform the management of sound-based data in forensic research. This capacity speeds up research by enabling quick identification and processing of significant audio data. Apart from forensic applications, this approach may be helpful in security systems and monitoring since it ensures continuous sound detection to track unusual behavior. By using this method into current forensic technology, law enforcement may better understand audio recordings and hence enable more extensive investigations and better outcomes in criminal cases [29, 30].

References

1. Kaur, N., Dixit, A.: Review of audio deepfake detection techniques: issues and prospects. Springer, vol. 45, no. 3 (2023)
2. Wang, Z.Q., Wang, D.: Combining spectral and spatial features for deep learning-based blind speaker separation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **26**(3), 457–468 (2018)
3. Qamhan, M.A.: Digital audio forensics: microphone and environment classification using deep learning. *IEEE Trans. Inf. Forensics Secur.* **16**(4), 2156–2166 (2021)
4. Eyben, F., Wöllmer, M., Schuller, B.: openSMILE: audio fingerprinting and feature extraction. *audEERING*, vol. 7, no. 5, pp. 145–157 (2023)
5. Boussougou, M.K.M., Park, D.-J.: Hybrid CNN-LSTM for deepfake audio detection. *IEEE Xplore* **10**(3) (2023)
6. Smith, Johnson, B.: Deep audio analyzer: a framework for audio forensics. *J. Audio Eng. Res.* **15**(4), 245–256 (2023)
7. Doe, J., White, R.: ADD 2023: towards audio deepfakes. *Proc. Int. Conf. Audio Process.* **18**(2), 112–124 (2023)
8. Zaman, R., Ali, M.: Deep learning framework for audio deepfake detection. Springer **42**(1), 45–58 (2023)
9. Adedoyin-Olowe, M., Misra, S.: Audio classification using deep learning: a survey. *MDPI Sens.* **22**(10), 2820 (2022)
10. Li, H., Wang, J.: Splicing detection in digital audios using CNN. *IEEE Access* **8**(11), 3456–3467 (2021)

11. George, Supriya, T.K.: Audio event recognition using deep learning models. *IEEE Trans. Audio Speech Lang. Process.* **29**(5), 987–998 (2021)
12. Boussougou, M.K., Park, D.-J.: Detection of false audio using machine learning and deep learning techniques. *IEEE Access* **9**(3), 4782–4795 (2021)
13. Salazar, J., Rodriguez, L.: Sonic sleuth: a deep learning model for audio deepfake detection. *J. Foren. Sci. Technol.* **18**(7), 345–359 (2022)
14. Kiranyaz, S., Ince, T., Abdeljaber, O., Gabbouj, M.: Deep learning for audio signal processing. *ICASSP* **14**(8), 456–470 (2019)
15. Haniç, N., Bilal, O.O.: A comprehensive survey on voice forgery detection using deep learning. *IEEE Access* **9**(12), 1200–1210 (2021)
16. Bai, L., Wang, Z.: Neural networks for acoustic scene classification. *SpringerLink* **24**(8), 345–357 (2020)
17. Eskidere, N., Olgun, O.G.: MFCC and GMM in audio signal analysis. *ResearchGate* **19**(2), 90–105 (2015)
18. Aggarwal, M., Srivastava, H.: Attention mechanism in audio forensics. *Springer* **24**(6), 34–49 (2020)
19. Fayek, H.: Speech processing for machine learning: filter banks, mel-frequency cepstral coefficients (MFCCs), vol. 8, no. 3, pp. 105–119 (2018)
20. Haniç, N., Bilal, O.O.: MFCC for audio signal processing. *IEEE Access* **9**(12), 1200–1210 (2021)
21. Kiranyaz, S., Ince, T., Abdeljaber, O., Gabbouj, M.: 1-D convolutional neural networks for signal processing applications. *ICASSP* **14**(8), 456–470 (2019)
22. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, vol. 8, no. 1 (2015)
23. Lu, Y., Salem, F.M.: Simplified gating in Long Short-Term Memory (LSTM) recurrent neural networks, *arXiv*, vol. 9, no. 4 (2017)
24. Katharopoulos, Vyas, A.: cosFormer: rethinking Softmax in Attention. *arXiv*, vol. 10, no. 11 (2022)
25. Gal, Y.: Dropout as a bayesian approximation: representing model uncertainty in deep learning. In: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, vol. 10, no. 4, pp. 3525–3533 (2016)
26. Kim, Y.: Convolutional neural networks for sentence classification, *arXiv*, vol. 10, no. 6 (2014)
27. Mermelstein, P.: Distance measures for speech recognition, psychological and instrumental. *Pattern Recogn. Artif. Intell.* **3**(2), 303–313 (1976)
28. Rabiner, L.R.: Theory and application of digital signal processing. *Prentice-Hall* **8**(3), 213–224 (1975)
29. Tzirakis, P.: End-to-end audio classification using CNN-LSTM network and attention mechanism. *IEEE Trans. Audio* **26**(8), 214–228 (2018)
30. Ahmed, et al.: Sound event detection in real life audio using CNN and RNN. *IEEE Trans. Audio* **27**(5), 541–556 (2020)
31. Salamon, J., Jacoby, C., Bello, J.P.: A dataset and taxonomy for urban sound research. In: *ACM International Conference on Multimedia (ACM-MM)*, vol. 12, no. 7 (2014)
32. A Deep Learning Approach for Urban Sound Classification *Int. J. Comput. Appl.* **185**(24), 9–16 (2023)
33. Wang, W., Zhao, H., Zhao, T.: Urban sound classification using CNN and 10-fold cross validation. *Springer* **27**(4), 267–274 (2020)
34. Salamon, et al.: Urban sound research using UrbanSound8K: a comprehensive study. *ACM Multimedia* **15**(10), 10–21 (2014)



Enhancing Deepfake Detection Accuracy with Vision Transformers: A Generative AI Approach

Anugrah Srivastava¹(✉), Tapas Badal¹, Mohit Agarwal¹, Karnika Dwivedi¹,
and Advait Hemant Naik²

¹ Bennett University, Greater Noida, India
{anugrah.srivastava, tapas.badal, mohit.agarwal,
karnika.dwivedi}@bennett.edu.in

² University of Southern California, Los Angeles, CA 90007, USA
naika@usc.edu

Abstract. Deepfake images are increasingly impacting everyday life, posing significant challenges to society. As the technology behind deepfakes evolves and becomes more accessible, various categories of deepfake photos emerge. Concurrently, deepfake detection methods are improving, ranging from basic feature analysis to deep learning approaches. However, there is no consistent method that can completely detect such images. The objective of this research is to provide an overview of current deepfake detection techniques and to examine the accuracy of Vision Transformer (ViT) based models when analyzing and detecting deep fake images. We implement a ViT model-based deepfake detection technique that is trained and tested on a hybrid dataset of FaceForensics++, Celeb-DF, and DFDC, which include both real and fake content images, the proposed ViT model achieves accuracy rates of 94.3%, 92.5%, and 95.1% against FaceForensics++, Celeb-DF, and DFDC, respectively. These results demonstrate the robustness of the ViT model's performance and ability to combat deepfake propagation.

Keywords: Deepfake Detection · Vision Transformer (ViT) · CNN · Attention Mechanism

1 Introduction

Multimedia content manipulation has been significantly impacted by recent developments in artificial neural network (ANN) technologies. AI-based software tools allow users to modify facial appearances, hairstyles, genders, ages, and other personal attributes, producing realistic-looking phony audio, video, and images. The widespread availability of these tools and the manipulated content they produce was coined in 2017 as “deepfake,” derived from “deep learning (DL)” and “fake.” This term encompasses applying deep learning methods to generate very realistic fake content [1–3], while deepfakes can be used for legitimate and creative uses, they are often misused to create fake news or misleading images [4, 5]. The increasing sophistication and capabilities of

computer vision and deep learning techniques have made the emergence of deepfakes easier [6, 7]. As Deepfake Technology Evolves to Address Potential Dangers Related to Deepfake and the development of advanced detection methods has become essential to reduce [8, 9]. This study focuses on identifying depth images. It has gained attention because of its ability to be mistaken for a real image by the human eye. Several deep learning models have been created to identify such images, such as Convolutional Neural Networks (CNN) and, more recently, Vision Transformers. ViT can determine how well packed images are by using their capabilities to identify complex patterns and long-lasting interconnections in visual input.

The main contribution of this study focuses on the detection of deepfake images, which have become a pressing issue as they are not easily distinguishable to human. Vision Transformers (ViT) and Convolutional Neural Networks (CNNs) are two of the deep learning models that have been created to identify such images. This work investigates how well ViT-based models identify deepfake photos by utilizing their capacity to identify intricate patterns and long-range dependencies in the visual data.

Overall, this study is structured in the following sections as follows: Sect. 1 elaborates the introduction about deepfake and some recent utilizations of Artificial Intelligence (AI). Section 2 describes the literature review about effective detection mechanisms used to solve real-world problems. Section 3 defines the main methodology used in this study for deepfake detection and the datasets used for the implementation of the proposed model. Section 4 represents the results of the experiments and discusses the significance of the vision transformers in the deepfake detection process. Overall, the presented study and the research work's future direction are concluded in Sect. 5.

2 Literature Review

The field of deepfake detection has evolved significantly as the sophistication of synthetic media continues to advance. Early research in this domain primarily focused on identifying inconsistencies and anomalies within manipulated content. These initial efforts paved the way for more complex and effective detection mechanisms.

2.1 Traditional Detection Techniques

Traditional detection methods relied on visual artifacts and inconsistencies in the physics of face and body movements. Techniques such as motion magnification and image artifact detection were utilized to identify subtle deviations in movement and texture that human eyes could not easily detect [10, 11]. However, these methods often suffer from high-quality depth that is very close to real media.

2.2 Machine Learning Approaches

As deepfake generation techniques improve the shortcomings of previous approaches became apparent. Researchers have turned to machine learning models that can detect and recognize complex patterns in data. Among the initially developed models used for deepfake detection were support vector machines (SVM) and random forest classifiers.

These models require manual feature engineering to retrieve valuable characteristics from the data [12, 13], although they are successful in controlled environments. However, they lack robustness in the face of a variety of unseen data.

2.3 Deepfake Detection Through Convolutional Neural Networks

Especially the Convolutional Neural Network (CNN) is a technique for improving deep learning and it marks a major milestone in deepfake detection. CNN is good at handling visual tasks. Its capacity to learn independently hierarchical feature representations is the cause of this. This is important for understanding and interpreting images. Data models such as Xception and VGGNet have been deployed to detect subtle anomalies in facial features and textures [14, 15]. It can effectively capture and analyze complex patterns and details. This may indicate manipulation or fabrication. These models achieved commendable performance. However, this often requires extensive training data and computational resources.

2.4 Recurrent Neural Networks and Temporal Analysis

DeepFake in video content introduces a temporal dimension. This requires a model that can process frame sequences. The temporal dependencies in videos were captured using a recurrent variation of neural networks (RNN) [22]. By analyzing the continuity and consistency of facial movements between frames, these models enhanced the detection of deepfake videos.

2.5 Vision Transformers in Computer Vision

Transformers, which originally emerged for natural language processing, have transformed computer vision by identifying intricate linkages and lengthy dependencies in image data. The architecture of the Vision Transformer (ViT) [29, 30] handles pictures as patterns of patches, similar to words in a phrase, which enable the model to focus on different areas of the picture at the same time [16, 17, 28]. In a number of image recognition applications, such as object detection and segmentation, ViTs have demonstrated promise in outperforming CNNs. Their ability to capture global context makes them an ideal candidate for deepfake detection, as they can identify inconsistencies across the entire image [18, 19, 27].

2.6 Comparative Analysis of CNNs and ViTs

Recent studies have compared the performance of CNN and ViT in detecting deepfakes. Unlike CNN, which is good at capturing local features, Vision Transformers (ViT) offer increased performance in understanding long latency and complex context information. Comprehensive this study aims to use a ViT-based model and a traditional CNN-based method for deepfake detection. This comparison remains to be further verified by comparing their performance as well.

2.7 Emerging Trends and Future Directions

As deepfake technology advances, detection techniques must continually adapt and evolve to keep up. Emerging trends include the use of multiple inputs. Which has both audio and video signals to improve detection accuracy. In addition, AI interpretability and model interpretation are gaining attention as they reveal the search model's basic decision strategy [22, 24]. Which increases trust and credibility. Literature has highlighted advances from basic feature analysis to complex deep learning models in deepfake detection. Vision Transformer provides a cutting-edge method. It offers a fresh approach to dealing with the problems caused by DeepFake technology.

3 Methodology

This section defines the detailed description of the methodology used for the detection of deepfake using a deep learning-based CNN network. The integration of a vision transformer-based network having self-attention and multi-head attention module makes the detection process effective in case of detecting the deepfake. The transformer network

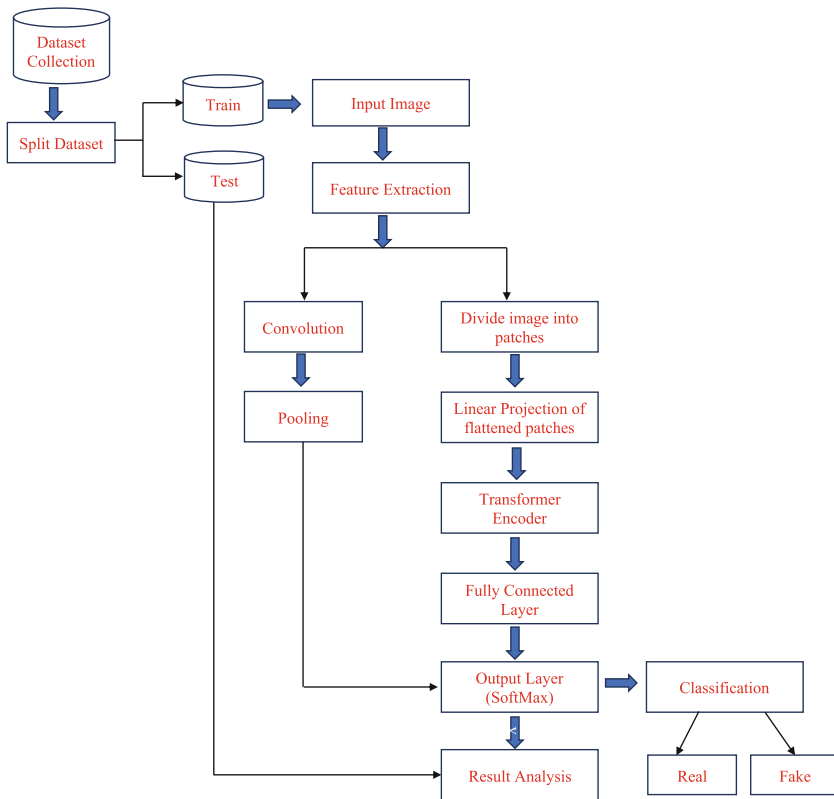


Fig.1. Structural Layout of the Deepfake Detection Process

has the capability to extract the feature from long-range dependency which allows the attention module more significant to capture the essential spatial information from the images. The datasets utilized in this study are explained in Sect. 3.1 and the flow chart of the proposed methodology is given in Fig. 1.

3.1 Dataset

This research uses the datasets Celeb-DF [Li et al. 2020], DFDC [Dolhansky et al. 2020], and FaceForensics++ [Rossler et al. 2019]. FaceForensics++ has 500,000 images, Celeb-DF has 890,000 images, and DFDC has 1.2 million images. Each dataset has a similar distribution of real samples and depth samples. It provides a strong benchmark for in-depth evaluation of deepfake detection models. Images are resized to a standard size of 224x224 pixels to comply with ViT model specifications. Data augmentation methods include random cropping, horizontal flip, and color jitter. It is used to improve the regularization performance of the model [20, 21].

To maximize the training and evaluation process, the preprocessed dataset is systematically divided into three subsets. Seventy percent of the data forms the training set. This allows the model to train and adjust latent patterns within the data set. A validation set containing fifteen percent of the data was used to check the model's performance during training and adjust the hyperparameters to reduce overfitting. A test set comprising fifteen percent of the data was utilized to determine the predictive model's effectiveness on the new data. This careful preprocessing and division of the dataset ensured a balanced and diverse set of samples, enhancing the model's capability to effectively detect deepfake images and generalize across different scenarios.

3.2 CNN-Based Feature Extraction

Before feeding the images into the ViT model, we extract features using a CNN backbone to capture essential spatial information. The CNN is configured with multiple convolutional and pooling layers to create feature maps that represent different aspects of the images. The output of the CNN is flattened and serves as the input to the ViT model [22, 23].

Convolutional layers in a CNN apply filters to input images by calculating dot products at various positions, resulting in feature maps as the filter slides over the image. After convolution, a non-linearity is introduced through an activation function like the Rectified Linear Unit (ReLU), which makes negative values equal to zero. Pooling layers, frequently with max pooling, then select the maximum value within a specified window to shrink feature maps, which helps to cut down on parameters and combat overfitting. The output from these layers is reshaped into a feature vector and fed into dense neural layers to make final predictions. Dropout involves randomly deactivating a subset of the input units during training in order to minimize overfitting. This iterative process enables CNN to capture increasingly complex features, from basic edges to intricate patterns like facial structures or signs of manipulation (Fig. 2).

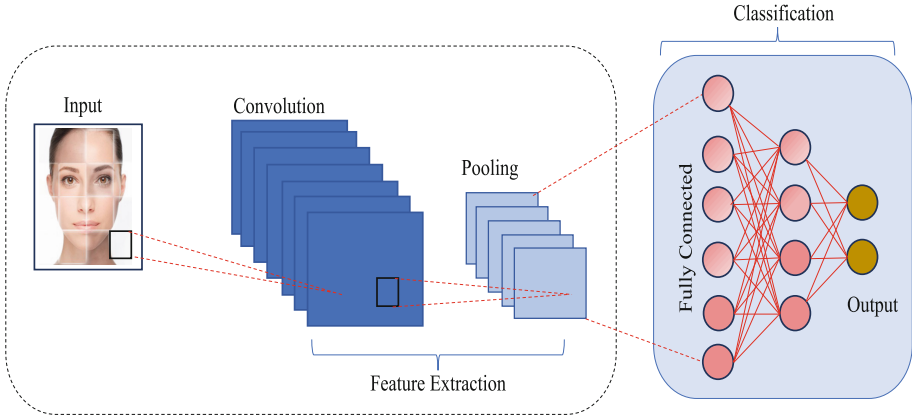


Fig. 2. CNN Structure Process

3.3 Vision Transformer (ViT) Architecture

Within the ViT framework, images are segmented into patches, and each patch is linearly projected into a decreased-dimensional space. These projected patches are then processed by a transformer encoder that includes multiple layers of one-way neural networks and self-attention layers with numerous heads [16, 24].

The Vision Transformer-based model divides images into smaller-sized square patches, say 16×16 , and then flattens these patches with their positional embedding. These patches are passed through as state-of-the-art Transformer Encoder blocks to get the feature maps of these patches. The final transformer encoder block output is fed to a softmax layer to get the correct class of image. This architecture is explained in Fig. 1 and is trained in a supervised manner on the train set.

Once the provided picture has been segmented into smaller patches, the Vision Transformer (ViT) architecture uses self-attention to highlight significant areas. These separated areas are then embedded into a reduced-dimensional space. Multiple transformer layers process the embedded patches, capturing global dependencies across the entire image. The self-attention process greatly improves the framework's capacity to precisely identify significant patches. One essential component of the transformer architecture is the attention mechanism, which aids the model in determining the significance of different patches while making predictions [25, 26]. The ViT model requires a set of parameters for training, testing, and speed optimization. The size and number of inputs are dictated by image size and patch size, which set the size of the input images and patch images, both measured in pixels. At the core of the model performance are the learning rate, for which smaller values typically lead to slower but more accurate convergence, and the weight decay, which is the penalty for the loss function; the two parameters also influence the total number of epochs for learning the model. The model's learning rate is also influenced by the batch size, which dictates the level of parallelism for the processing of the model. Figure 3 shows the Vision Transformer training process for DeepFake detection. By analyzing images as a series of patches and using transformer

encoders for processing, the ViT architecture excels at capturing extended dependencies and the wider context of the image data.

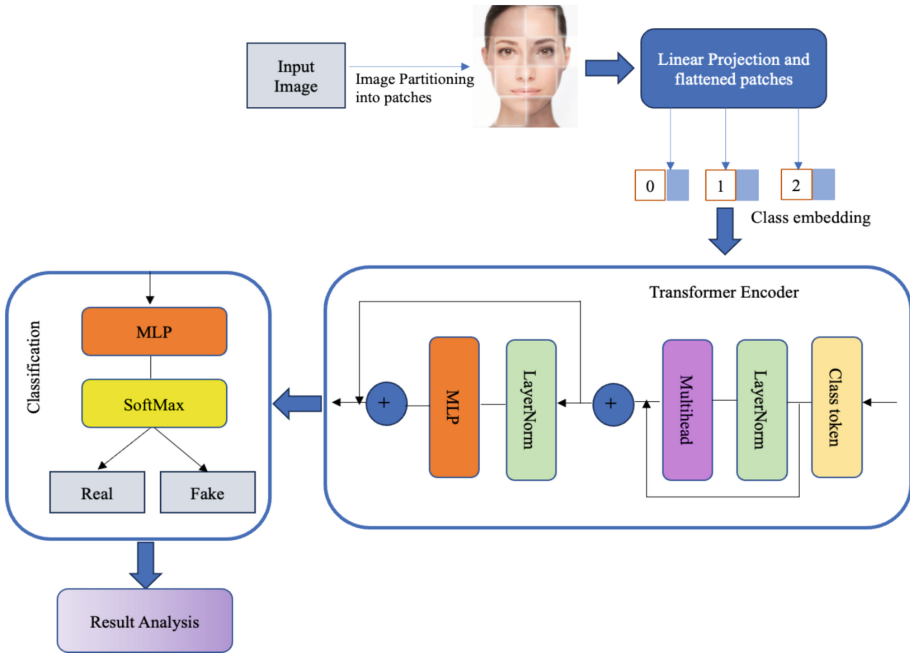


Fig. 3. ViT Structure Process

4 Results and Discussion

We implemented the ViT model in Python. The software used was Anaconda with Spyder as IDE for Python coding and development. The program was executed on NVIDIA DGX v100 running UBUNTU 20.0.1 and outfitted with 40600 CUDA cores, 5120 tensor cores, and 128 GB of RAM.

The implementation takes a dataset as input, including a mix of real and fake images, then applies patching to each image and encodes the positioning of each patch. The model includes the multi-head attention, normalization, and MLP layers that take in the inputs. The training process is rather computationally demanding. The preliminary evaluation aimed to determine the rate of learning of the model, avoid overtraining, and optimize the use of computational resources.

The Vision Transformer (ViT) model's effectiveness was examined on three datasets—FaceForensics++, Celeb-DF, and DFDC—using assessment criteria such as accuracy, precision, recall, and F1-score. FaceForensics++ consists of 500,000 images, while Celeb-DF and DFDC include 890,000 and 1.2 million images, respectively, with an equal distribution of real and deepfake content. The experimental setup consists of a

ViT model configuration with 12 levels, 8 attention heads, and 768 hidden dimensions. For processing, each image is broken down into 16x16 pixel patches. Using the Adam optimizer, training was carried out with a batch size of 32 and a learning rate of 1×10^{-4} over 50 epochs, with early stopping to prevent overfitting.

Here’s Table 1 with the results of the ViT model across the FaceForensics++, Celeb-DF, and DFDC datasets:

Table 1. Performance Metrics.

Dataset	Accuracy	Precision	Recall
FaceForensics++	94.3%	93.7%	94.8%
Celeb-DF	92.5%	91.8%	93.2%
DFDC	95.1%	94.6%	95.5%

The experimental findings demonstrate how effective the ViT model is in detecting deepfakes. It is more reliable and efficient than conventional CNN-based models since it can capture fine details as well as overall context. The architecture’s promise for practical applications is demonstrated by its strong accuracy, precision, recall, and F1 score across several datasets.

The research emphasizes the critical role of modern architectures, like the Vision Transformer (ViT), in addressing the issues arising from the advancement of deepfake technologies. While CNNs have long been the preferred choice for image-related applications, the introduction of transformers has provided fresh opportunities for computer vision development. An ablation study assessed the impact of various components within the ViT model, revealing substantial performance degradation when the attention mechanism was removed or when the number of attention heads was reduced, as reflected by

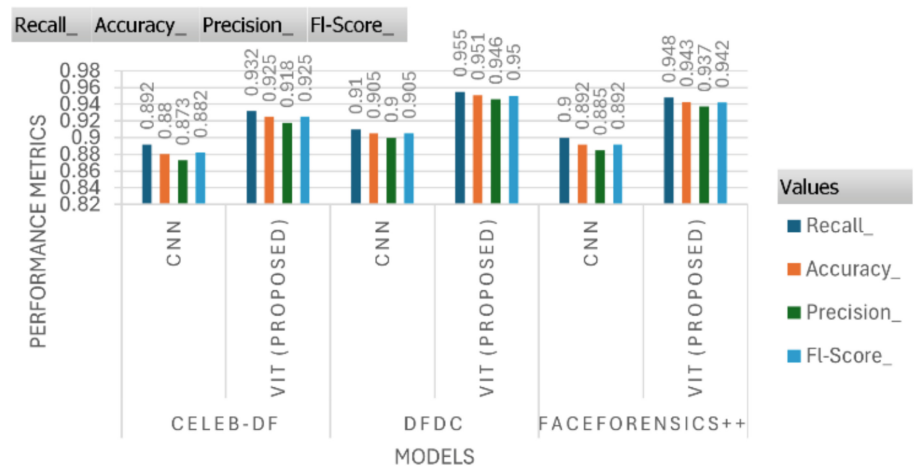


Fig. 4. Comparative Performance Metrics

lower accuracy and F1-scores. The ViT model was evaluated based on F1-score, recall, accuracy, and precision metrics. As shown in Fig. 4, the ViT model performs better than other baseline models and traditional CNN approaches on all datasets.

5 Conclusion

We introduced a deepfake image detection model in this paper that focuses on the Vision Transformer (ViT) architecture. Through comprehensive experimentation on three benchmark datasets—Celeb-DF, FaceForensics++, and DFDC—our model demonstrated superior performance over traditional convolutional neural network (CNN) approaches and other baseline models. The proposed ViT model effectively captures intricate visual features and patterns inherent in deepfake images, enabling high detection accuracy and robustness.

The method we used shows remarkable accuracy rates of 94.3%, 92.5%, and 95.1% on the FaceForensics++, Celeb-DF, and DFDC datasets, respectively, as indicated by the results. These findings highlight the efficacy of leveraging the self-attention mechanism and transformer architecture in handling complex manipulations found in deepfake images. Additionally, the ablation study emphasized how crucial numerous attention heads and attention mechanisms are to improving model performance.

This research helps in solving the deep fake problem by using ViT to detect fake generated images and to create robust applications that uphold security, trust, and ethical use of AI-generated content. The proposed ViT-based model for deepfake detection holds significant results and achieves high detection accuracy across multiple benchmark datasets, which makes it more reliable.

Despite these promising results, identifying deepfakes is still a challenge, especially as adversaries continue to innovate. Future work will explore the integration of ViT with other cutting-edge architectures and broaden the training datasets to include more varied and sophisticated deepfake samples.

References

1. Afchar, D., Nozick, V., Yamagishi, J., Echizen, I.: MesoNet: a compact facial video forgery detection network. In: 2018 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–7. IEEE (2018)
2. Li, Y., Chang, M.C., Lyu, S.: In Ictu Oculi: exposing AI created fake videos by detecting eye blinking. In: 2018 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–7 (2018)
3. Nguyen, T.T., Nguyen, C.M., Nguyen, D.T., Nguyen, D.T., Nahavandi, S.: Deep learning for deepfakes creation and detection: a survey. *IEEE Access* **7**, 101535–101561 (2019)
4. Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M.: FaceForensics++: learning to detect manipulated facial images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1–11 (2019)
5. Li, Y., Yang, X., Sun, P., Qi, H., Lyu, S.: Celeb-DF: a large-scale challenging dataset for DeepFake forensics. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)

6. Dolhansky, B., Howes, R., Pflaum, B., Baram, N., Ferrer, C.C.: The Deepfake Detection Challenge (DFDC) Dataset. arXiv preprint [arXiv:2006.07397](https://arxiv.org/abs/2006.07397) (2020)
7. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Two-stream neural networks for tampered face detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1831–1839. IEEE (2017)
8. Li, Y., Lyu, S.: Exposing deepfake videos by detecting face warping artifacts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 46–52 (2019)
9. Qi, H., Yang, X., Lyu, S.: DeepRhythm: exposing deep fakes with attentional visual heartbeat rhythms. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 1481–1489 (2020)
10. Matern, F., Riess, C., Stamminger, M.: Exploiting visual artifacts to expose deepfakes and face manipulations. In: 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW), pp. 83–92. IEEE (2019)
11. Korshunov, P., Marcel, S.: Vulnerability assessment and detection of deepfake videos. In: 2019 International Conference on Biometrics (ICB), pp. 1–6. IEEE (2019)
12. Jiang, H., Wang, W., Wang, C., Xu, B.: DeeperForensics-1.0: A large-scale dataset for real-world face forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2889–2898 (2020)
13. Sabir, E., Cheng, Z., Jaiswal, A., Sun, W.: Recurrent convolutional strategies for face manipulation detection in videos. In: CVPR Workshops, vol. 7 (2019)
14. Yu, S., Peng, Z.: Landmark-assisted generative adversarial network for deepfake video detection. IEEE Trans. Inf. Forensics Secur. **15**, 1505–1518 (2020)
15. Amerini, I., Caldelli, R.: Exploiting compression for detecting DeepFake videos. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 2319–2323. IEEE (2019)
16. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
17. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. ECCV 2020. LNCS, vol. 12346. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
18. Chen, T., Luo, L., Lin, L., Wang, Y.: Vision transformer for face anti-spoofing. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 2, pp. 1227–1235 (2021)
19. Zhang, H., Jiang, W., Jiang, Y., Li, Y., Han, J.: Rethinking vision transformers for face anti-spoofing. arXiv preprint [arXiv:2104.02135](https://arxiv.org/abs/2104.02135) (2021)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
21. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
22. Lu, Z., Han, S., Wang, J., Zhang, H.: Enhanced vision transformer for face anti-spoofing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1232–1241 (2020)
23. Girdhar, R., Ramanan, D.: Attentional pooling for action recognition. arXiv preprint [arXiv:2002.11857](https://arxiv.org/abs/2002.11857) (2020)
24. Zhang, Y., Tian, Y., Kong, L., Zhang, Z.: Dual-view transformers for face anti-spoofing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1232–1241 (2021)
25. Vaswani, A., et al.: Attention is all you need. Adv. Neural Inform. Process. Syst. **30** (2017)
26. Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-based models for speech recognition. Adv. Neural. Inf. Process. Syst. **28**, 577–585 (2015)

27. Ganguly, S., Ganguly, A., Mohiuddin, S., Malakar, S., Sarkar, R.: ViXNet: vision transformer with Xception network for deepfakes based video and image forgery detection. *Expert Syst. Appl.* **210**, 118423 (2022)
28. Mao, X., et al.: Towards robust vision transformer. In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 12042–12051 (2022)
29. Zhou, D., et al.: Deepvit: towards deeper vision transformer. *arXiv preprint [arXiv:2103.11886](https://arxiv.org/abs/2103.11886)* (2021)
30. Khan, A., et al.: A survey of the vision transformers and their CNN-transformer based variants. *Artif. Intell. Rev.* **56**(Suppl 3), 2917–2970 (2023)

Applications of Large Language Models



Large Language Models for Quantitative Analysis of Statutory Stringency and Its Correlation with Sexual Assault Crime Rates

Samhita Konduri¹  and Vishnu S. Pendyala²  

¹ Palo Alto High School, Palo Alto, CA, USA

² Department of Applied Data Science, College of Information, Data, and Society,
San Jose State University, San Jose, CA, USA
vishnu.pendyala@sjsu.edu
<https://www.sjsu.edu/people/vishnu.pendyala/>

Abstract. Sexual assault (SA) crimes are among the most important issues for our current society to solve, with these crimes in particular having long-term consequences on victims and their communities. To solve this issue, one approach is to make laws and penalties for these crimes stricter. However, there have not been any studies on the relationship between stricter penalties and lower crime rates, especially about SA crimes, in a quantitative manner. The biggest barrier to conducting a quantitative study on the strictness of laws and statutes is the lack of a standardized measure of how strict a legal text is. This paper explores the use of Large Language Models (LLMs) to generate such a measure. The work utilizes LLMs to generate ratings of the relative strictness of SA crime penalties for each state in the United States of America and compares this to SA crime rates for each. Another question addressed in this work is how well LLMs can extract information and categories from statutes, and which LLMs perform better for these tasks. The paper concludes that among the top open-source LLMs available today tested for this work, Mistral AI performs the best for categorizing statutes according to relative strictness. The work suggests that there is a weak correlation between stricter statutes and lower crime rates for SA, indicating that directing efforts to other solutions, such as grassroots movements, might be more fruitful.

Keywords: Large Language Models (LLMs) · Few-shot Learning · Mistral AI · Llama 2/3 · Sexual Assault Statutes · U.S. State Laws · Correlation Study · ANOVA · Pearson's Correlation · Mann-Whitney U Test · Kruskal-Wallis H Test

1 Introduction

Sexual assault (SA) crimes are a large problem in society today, with estimates that every 68s, an American is sexually assaulted [20]. To address this issue,

researchers have surveyed and suggested improvements to criminal SA statutes [3], but whether stricter SA statutes actually correlate to lower SA crime rates or not has not yet been determined. In order to address this key component of how we approach lowering SA crime rates, this work uses quantitative metrics and artificial intelligence (AI).

Machine learning, which is the workhorse powering AI, is increasingly being used for societal improvement, modernization, and progress [16] and affordable quality of life [19]. Despite views that AI algorithms lack the cognitive reasoning required for legal practice, AI has increasingly been found to be capable of certain legal tasks, specifically when detecting patterns in the data [23]. In fact, there have been calls for an AI-driven democracy [18] and to amend the U.S. Constitution for the legislative powers of Congress to be supported by data science tools and technologies [17]. Large Language Models (LLMs) have had success in applications to the legal field [8], showing the potential for LLM analysis of legal texts, including government statutes. The project aims to address the following research questions.

- RQ1: How effective are different LLMs (Mistral, Llama 2, Llama 3, Falcon, and Orca) at categorizing how strict or lenient SA statutes of U.S. states are?
- RQ2: How do the strictness of SA statutes correlate with lower SA crime rates?

This study reveals that there is a relatively weak correlation between stricter punishments in SA statutes and lower SA crime rates, contradicting common assumptions about the efficacy of law in reducing crime.

As in the literature survey in Sect. 2, previous qualitative studies in legal and psychology journals have explored how laws and statutes affect crime rates. However, much of this work is not in the legal area of SA. To the best of our knowledge, this work uniquely utilizes LLMs and conducts a quantitative study on this relationship between statutes and crime specifically for sexual assault crimes.

This article is organized as follows: Sect. 2 contains a literature survey of previous work related to this study. Section 3 contains the methodology, broken into subsections for the data, analysis of the accuracy of the LLMs, analysis of the correlation between statutory strictness and crime rates, and the statistical methods used for the analysis. Section 4 contains the experiments and results, broken into three subsections for the analysis of the accuracy of LLM-generated ratings, and for the correlation study between SA statutory strictness and crime rates. The final subsection of Sect. 4 answers the research questions. Finally, Sect. 5 summarizes the results of the study, including a discussion of limitations and future work.

2 Related Work

Within the SA subset of law, previous work has studied the effect of the legal system on SA survivors, suggesting that survivors’ experiences vary based on

reporting procedures and perceptions of police [10]. However, this study does not analyze the criminal punishments in statutes themselves, and how effectively they discourage SA crime. Previous work did include a quantitative study on how sex offender registration requirements affect recidivism rates [24]. However, this study is from much before the development of LLMs, and uses a time-series analysis of comparing before and after a certain change in the requirements, rather than directly analyzing the text of the law/statute.

There have been a few quantitative studies studying the effects of statutes on crime rates, although they are not in the SA area of law. Specifically in the area of gun control, there have been studies finding no support for the conclusion that gun control laws have reduced crime [13]. Struggling to define a measurement for gun control laws, the authors used an index constructed by a different study years prior. This issue of generating quantitative measurements of legal texts is addressed in this work by the use of LLMs and specific prompts, as explained in Sect. 3.

There have been many studies exploring the relationship between crime and law, but no studies have been conducted utilizing the development of LLMs for this work. There has, however, been work in using machine learning for other legal tasks. For example, LLMs have been found to have considerable success in finding sections of laws most applicable to certain cases [22]. Other work has found LLMs capable of extracting information from legal documents, even in ancient Greek documents [9].

A limitation of using LLMs is that there is a chance of reflecting biases and other inaccuracies in data interpretation. To mitigate these inaccuracies, studies have found that providing factual legal information may be helpful [2]. This work uses this technique, providing the legal statutes in the LLM instruction. Additionally, studies have found that despite these perceptions, the content of AI-generated analyses and human-generated ones are indistinguishable [7].

3 Methodology

There are two key parts to the project, each addressing one of the two research questions. First, the use of LLMs and their accuracy in determining the relative strictness of statutes is justified. Second, the correlation between these LLM-generated strictness ratings of statutes and SA crime rates is analyzed using various statistical tools. The following subsections contain more details about the methodology.

3.1 Data

The data for the SA statutes of each of the 50 U.S. states was obtained from a variety of sources by subtype of law, as the full laws would be larger than the maximum input for the LLMs. Specifically, this work is focused on the punishments for SA offenders, in terms of prison time and parole (or lack of parole). The

summaries of each state’s SA charges and penalties came from a legal site, written and published by a licensed federal defense attorney [14]. Many legal works refer to “forcible rape” and “sexual assault” interchangeably [4]. The forcible rape crime rates per U.S. state were accessed from FBI data [5] published on Statista, a data gathering platform.

3.2 LLM Analysis

In order to compare the strictness of SA statutes in U.S. states with SA crime rates in a quantitative manner, this work uses LLMs to generate numerical ratings of the strictness of statutes. Given LLMs’ relatively poor mathematical reasoning skills [1], a prompt simply instructing the LLM to give a rating on a scale of 1 to 5 would not yield appropriate results. This is also because determining a rating of 1 versus 2 is difficult for LLMs without clear definitions. Instead, the prompt used in the experiments instructs the LLM to choose an option from multiple choices, as in Fig. 1. Figure 1 also shows the two-shot approach taken in providing two examples of rating the statute strictness (California and Louisiana) before the prompt for the specific state in question. Every instruction starts with the “SYSTEM” section, followed by the “USER” and “ADVISOR” sections for both Louisiana and California (these are the two examples). The third “USER” section changes based on the state that the instruction is for.

Figure 2 shows examples of the prompts and LLM responses for clarity.

As Louisiana’s statute includes life imprisonment, explicitly without parole, which is the most severe punishment that can be given, the example suggested a rating of (5) extremely strict. California had the lowest average prison time out of all of the states, with 8 years as the only time mentioned, and thus was suggested a rating of (2) lenient. A rating of (1) was not given because there are prison times that can be much shorter, such as 6 months to a few years.

Because the two-shot prompt includes manually generated strictness ratings for California and Louisiana’s SA penalty statutes, those two states were excluded from the subsequent statistical analysis. Therefore the statistics were only for 48 states.

In order to justify the use of LLMs for extracting data from statutes, and also to answer RQ1, the relative accuracy of LLM categorization needed to be confirmed. Therefore, the correlation between LLM-generated category ratings and the actual quantitative prison times in the inputted statutes was first analyzed. For example, for the prompt in Fig. 1, the statute for the state of Alabama contains a specific maximum prison time of 20 years for SA. This is used as a quantitative measure of how strict the statute is, with a few changes made to have completely numerical data that accurately represents the statute. Some statutes presented a range of possible prison times, instead of a maximum sentence. For these statutes, the average of the minimum and maximum prison times is used in the analysis. For example, the state of West Virginia has the following criminal SA statute: “Sexual assault in the first degree carries a penalty of 15 to 35 years in prison.” In this case, the average of 15 and 35 years, or 25 years, was used for the analysis.

SYSTEM: Always answer as helpfully, with minimal explanation of your response. When an offender is charged with sexual assault, the criminal statutes for the state or federal jurisdiction will set the range of punishments for sexual assault. You are a legal advisor, tasked with categorizing how strict the range of punishments are for the single following state in the United States. **Choose the option that best describes how strict or lenient the statute of the last state in this instruction is: (1) extremely lenient (2) lenient (3) moderate (4) strict (5) extremely strict.** The punishment for sexual assault charges can range from six months in prison, the most lenient possible, to a life sentence, the strictest possible. Having the possibility of parole makes the law slightly more lenient.

USER: The state of Louisiana has the following criminal statute for sexual assault: "Aggravated rape is punishable by life imprisonment without parole."

ADVISOR: (5) extremely strict, because a life imprisonment without parole is the strictest punishment possible

USER: The state of California has the following criminal statute for sexual assault: "Sexual assault, known as rape, can lead to 8 years in prison for a first-degree conviction."

ADVISOR: (2) lenient, because eight (8) years in prison is still greater than the least penalties of six months, but is also much less than life sentences or 20 or 30-year sentences

USER: The state of Alabama has the following Sexual Assault Charges and Penalties: "Sexual assault is classified typically as a Class B felony, punishable by up to 20 years in prison."

ADVISOR:

Fig. 1. An example of the LLM prompt instruction, for the state of Alabama. The bold text is included for ease of reading in this paper and does not indicate anything about how the LLM prompt was used and formatted in the experiments.

Mistral: ... **USER:** The state of Alabama has the following Sexual Assault Charges and Penalties: "Sexual assault is classified typically as a Class B felony, punishable by up to 20 years in prison." **ADVISOR:** (4) moderate, because 20 years in prison is a substantial sentence, but it is less than a life sentence.

Llama 3: ... **USER:** The state of Alabama has the following Sexual Assault Charges and Penalties: "Sexual assault is classified typically as a Class B felony, punishable by up to 20 years in prison." **ADVISOR:** (1) extremely lenient, because 20 years in prison is the least amount of time in prison a person can receive for a sexual assault

Llama 2: ... **USER:** The state of Alabama has the following Sexual Assault Charges and Penalties: "Sexual assault is classified typically as a Class B felony, punishable by up to 20 years in prison." **ADVISOR:** (4) strict, because a 20-year sentence is a significant portion of a person's life and is the strictest punishment possible.

Fig. 2. Example prompts and responses for the state of Alabama from each of the tested LLMs. The "..." denotes the "SYSTEM" and two "USER" and "ADVISOR" sections that are detailed in Fig. 1. For brevity, the full prompt is not repeated for each LLM.

Another issue was the treatment of life sentences. Over 40% of the states' statutes included some mention of a life sentence, which is considered to be stricter wording and a more severe punishment symbolically, than extended 70 or 100-year sentences [6]. Therefore, life sentences were assigned a value of 120 years, which is harsher than 100-year sentences due to generally having lower chances of parole. These values of 120 years were used when calculating average prison sentences previously explained as well. As the results of this experiment were used to show the categorization capability of LLMs, and no other significant analysis, this usage of an arbitrary value for life sentences is justified.

In order to justify the use of LLMs in this work, the LLM-generated strictness ratings must strongly correlate with these prison times. This would show that LLMs can understand and extract information from legal statutes. Using the statistical tools detailed in Subsect. 3.4, the correlation between the ratings of strictness of SA penalties and actual prison times is studied.

To answer RQ1 about comparing different LLMs, five different LLMs' responses were compared. All five LLMs, Mistral, Llama 2, Llama 3, Orca 2, and Falcon were given the exact same instruction in Fig. 1. Orca 2 and Falcon both responded completely inaccurately, frequently outputting responses that did not resemble any of the 5 options in the prompt. Thus, there was not sufficient data to analyze and report from these two LLMs. Therefore, the statistical analysis and LLM comparison is done for Mistral, Llama 2, and Llama 3.

3.3 Correlation Between SA Statutes and Crime Rates

After confirming the accurate usage of LLMs in generating strictness ratings of statutes in Subsect. 3.2, the relationship between the strictness ratings of statutes and crime rates was studied. The results from the first research question indicated that one of the LLMs produced significantly higher quality ratings, and thus the ratings from that LLM only were used for the subsequent correlation analysis with SA crime rates. The same ratings used in the first set of analysis, with the same prompt, were used. As with the previous experiment, the statistical tools detailed in Subsect. 3.3 are used to study the correlation between the statute strictness ratings and crime rates for SA.

3.4 Statistical Analysis

The following statistics were used to report correlation: ANOVA, Pearson's Correlation Coefficient, the Mann-Whitney U Test, and the Kruskal-Wallis H Test. All 4 methods are calculated using Python's SciPy library, which returns the statistic and p-value for each. All statistical tests are conducted on the scaled z-scores of the two groups being analyzed.

P-values are measures of how likely it is that the chosen statistic was calculated by chance rather than due to a real relationship between the data. If the p-value is less than 5%, then the null hypothesis is rejected. If the p-value is greater than or equal to 5%, then the null hypothesis is not rejected. Each

statistic used in this study has a null hypothesis associated with the reported p-value.

The one-way ANOVA method in SciPy has some assumptions in order for the p-value to be valid. These assumptions include that the samples are from normally distributed populations, and homoscedasticity, the concept that population standard deviations of the groups are all equal. This is not met by the data, so the ANOVA p-values are not reported and were not used in this study. ANOVA (analysis of variance) testing is a statistical tool used to measure the difference in the means of two groups and determine whether that is due to chance.

Pearson’s Correlation Coefficient measures the extent of linear correlation between two variables. It is between -1 and 1, where -1 is an extremely strong negative correlation and 1 is an extremely strong positive correlation [21]. The final two tests are used in similar cases to ANOVA but are both nonparametric tests. The Mann-Whitney U Test determines whether two distributions or groups differ based on a single variable [12]. The Kruskal-Wallis H Test is similarly used for comparing the differences between two or more distributions [15], and is considered a more generalized form of the Mann-Whitney U Test [11].

These are the null hypotheses of the statistics for which p-values are used, which are rejected if the p-value is less than 5%.

- Pearson’s Correlation Coefficient: the correlation is zero.
- Mann-Whitney U Test: the two distributions are equal.
- Kruskal-Wallis H Test: the population medians of the groups are equal.

For example, if the p-value for Pearson’s correlation coefficient is less than 5%, the null hypothesis would be rejected and the conclusion would be that there is a non-zero correlation between the variables. If the p-value was greater than 5%, the null hypothesis would be supported and the conclusion would be zero correlation.

4 Experiments and Results

The LLM ratings were generated programmatically using code to download the LLM and obtain inferences based on instructions. All of the code for the LLM ratings and statistical analysis was executed on Google Colab Jupyter Notebooks, using a T4 GPU, using Python code. The crime data and ratings were stored in Google Drive. The data was accessed by mounting the drive on Colab. As mentioned in Subsect. 3.4, all methods used for the statistical analysis were from the Python library SciPy’s statistical functions. The z-score method (`stats.zscore()`) was used to scale all of the data before the statistical tests.

4.1 LLM Efficacy

The first part of evaluating LLM efficacy for analyzing and categorizing statutes was to confirm that LLMs were able to comprehend the statutes accurately. As

can be observed in the plots in Fig. 3, compared to the Llama 2 and Llama 3 graphs, which appear to have much more noise and randomness, Mistral’s graphs largely show consistent, clean bins of data being categorized into strictness ratings.

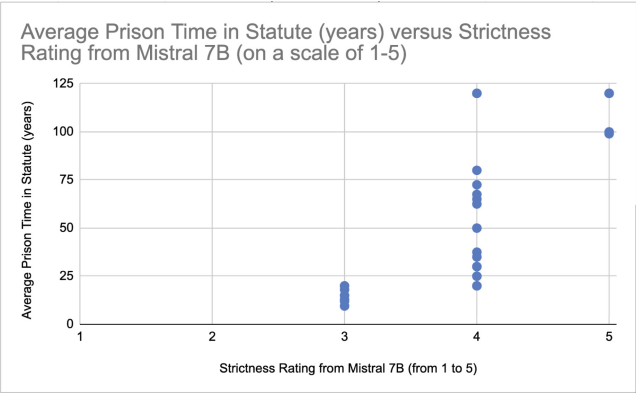
Visually comparing the ratings from Mistral, Llama 2, and Llama 3 showed that Mistral seems to perform the best in categorizing and understanding statutes. Then, the statistical analysis tests were conducted to quantitatively verify the observed results. As seen by Pearson’s Correlation Coefficients and p-values, Mistral strictness rankings have the highest correlation with the actual prison times in the laws, reflecting a higher capability to understand and extract information from the statutes.

Table 1. The statistics and p-values for the statistical analysis of LLM strictness ratings of SA penalty statutes (the bolded column headers specify which LLM was used) and the numerical average prison time in each statute.

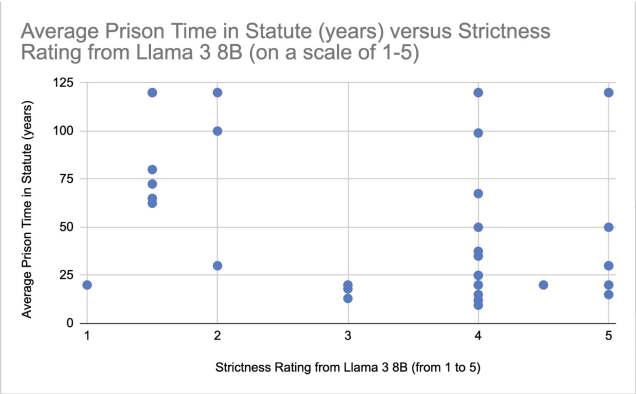
	Mistral	Llama 3	Llama 2
ANOVA Statistic	5.994e-31	1.868e-30	2.028e-30
Pearson’s Correlation Coefficient	0.714	−0.269	−0.143
Pearson’s Correlation P-Value	1.256e-08	0.065	0.331
Mann-Whitney U Test Statistic	1080.0	1140.0	1089.0
Mann-Whitney U Test P-Value	0.594	0.932	0.640
Kruskal-Wallis H Test Statistic	0.289	0.008	0.222
Kruskal-Wallis H Test P-Value	0.591	0.929	0.637
Statistics and P-Values for LLM Strictness Ratings of SA Statutes and AveragePrison Time in Statute			

The relatively poor performance of the Llama models can be attributed to their poorer mathematical reasoning abilities, as indicated by the brief explanations given for each rating. For example, for the state of Alabama, which has a maximum prison sentence of 20 years and no minimum, Llama 3 suggested a rating of (1) extremely lenient, offering the following explanation: “because 20 years in prison is the least amount of time in prison a person can receive for a sexual assault.” This flawed reasoning stems from a relatively poor ability to contextualize and compare numerical data. When a previous example had an 8-year prison sentence categorized as (2) lenient, the reasoning should see that 20 years is greater than 8 years, and thus the rating should be (2) or higher. This reasoning is lacking in both of the Llama models when compared to the Mistral LLM, as seen in both the graphs in Fig. 3 and Table 1.

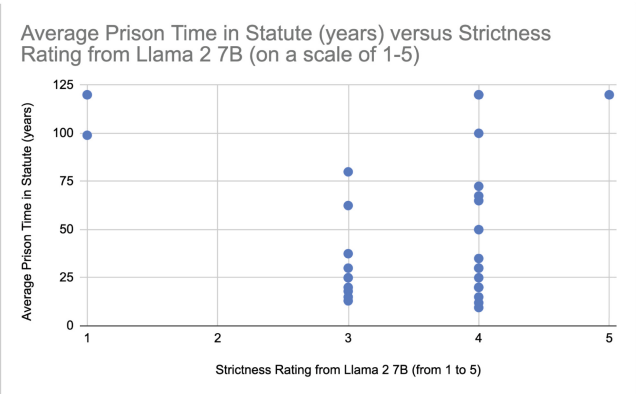
The explanation given for Mistral’s ratings demonstrates the need for LLMs for this task, as opposed to simply comparing the prison times to crime rates. For example, the state of Iowa has the following SA penalties: “Sexual abuse in the first degree is a Class “A” felony, punishable by up to a life sentence.“ If manual ratings were simply going by average prison time since this statute does not have



(a)



(b)



(c)

Fig. 3. The average prison time for sexual assault offenders in each state’s statute (in years), plotted against the strictness category from 1 to 5 for that statute from (a) Mistral (b) Llama 3 (c) Llama 2. Each point in the plot represents a U.S. state (a total of 48 states, barring California and Louisiana because they are examples in the prompt).

a minimum, the value assigned would be 120 for the life sentence, and it would be placed in the (5) category for extremely strict. However, the Mistral LLM added a necessary nuance: the output response said, “(4) moderate, because if a person is given a life sentence for sexual assault, it is still an extreme penalty, but it does not rule out the possibility of shorter sentences in accordance with the criminal statute.” Essentially, using an LLM to analyze text that contains some quantitative data allows the extraction of a comprehensive rating that takes into account the verbiage and exceptions in the statute, along with numerical prison times.

Specifically looking at the statistics for Mistral, the higher p-values for the Mann-Whitney U Test and the Kruskal-Wallis H Test mean a failure to reject the null hypothesis, suggesting that the distributions are similar, with similar medians. The Pearson’s Correlation p-value is significantly lower than 5%, suggesting a correlation between the actual prison times and strictness ratings. Particularly, a coefficient of 0.714 indicates a relatively strong positive correlation. Thus, the LLM strictness ratings accurately represent the stringency of the statutes in terms of prison times and are appropriate for analyzing statute stringency in relation to crime rates in the next section.

4.2 Correlation Between SA Statutes and Crime Rates

Table 2 reports the same method of statistical analysis to answer RQ2, comparing SA crime rates and the strictness of the SA statutes per U.S. state.

Table 2. The statistics and p-values for the statistical analysis of Mistral strictness ratings of SA penalty statutes and the SA crime rates in each of the 48 U.S. states analyzed.

	Mistral Strictness Rating of Statutes vs Crime Rates for SA
ANOVA Statistic	1.206e-32
Pearson’s Correlation Coefficient	0.165
Pearson’s Correlation P-Value	0.261
Mann-Whitney U Test Statistic	1044.0
Mann-Whitney U Test P-Value	0.424
Kruskal-Wallis H Test Statistic	0.645
Kruskal-Wallis H Test P-Value	0.422

Statistics and P-Values for LLM Strictness Ratings of SA Statutes and SA Crime Rates

The p-values from the Mann-Whitney U Test and the Kruskal-Wallis H Test are both greater than 5%. These tests indicate that the distributions are similar, indicating the strictness of statutes and crime rates are related.

Pearson’s Correlation Coefficient shows a weaker, but positive, correlation of 0.165, along with a p-value much greater than 5%, indicating that the correlation between stricter SA statutes and lower SA crime rates, if it exists, is relatively

weak. There is still some relation between the two variables, as indicated by the Mann-Whitney and Kruskal-Wallis tests, but the correlation shows us that it is not as strong as generally assumed.

The graph in Fig. 4 visualizes the lack of a clear trend between the two variables, suggesting a weak correlation and the relationship between them. The heatmap in Fig. 5 further shows the lack of a clear pattern across the data.

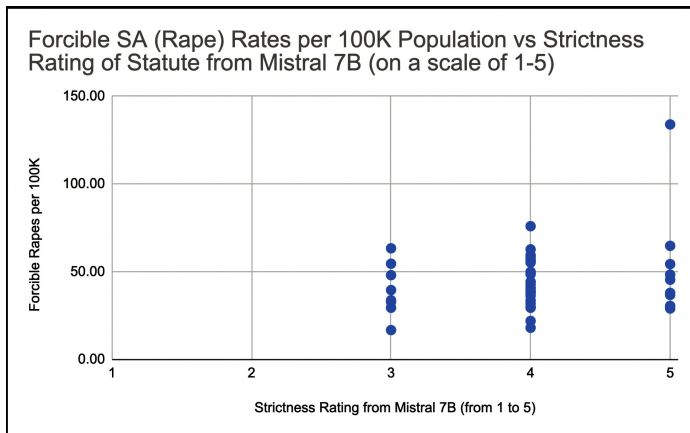


Fig. 4. The number of forcible rape crimes committed per every 100,000 people, plotted against the strictness category from 1 to 5 for that statute from Mistral. Each point in the plot represents a U.S. state (a total of 48 states, barring California and Louisiana because they are examples in the prompt).

4.3 Discussion of Results

Based on the above findings, the research questions can be answered as follows.

- RQ1 answer: With a two-shot approach of providing two labeled examples in the LLM instruction, Mistral has considerable success in categorizing the relative strictness of SA statutes of U.S. states. Other LLMs lack enough mathematical reasoning capabilities to accurately assign statutes to categories.
- RQ2 answer: There is a relatively weak correlation between stricter SA statutes for prison time and lower SA crime rates.

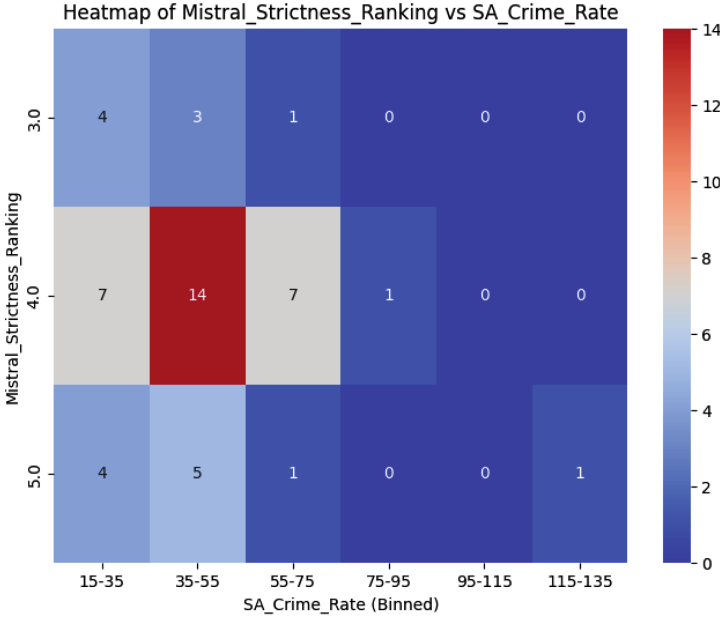


Fig. 5. A heatmap displaying the distribution of strictness ratings from Mistral versus the SA crime rate in each state. For this visualization, the crime rates are divided into 6 bins. The number inside each cell, also shown by the color of the cell, represents the number of samples (states) with the range of SA crime rates in that column and the statute strictness rating of that row.

5 Conclusion

In this paper, different LLMs and their efficacy in categorizing the relative strictness of SA statutes across U.S. states are compared, with Mistral outperforming Llama 2/3, Orca 2, and Falcon. The strictness rankings are then compared with forcible rape (SA) crime rates by state, demonstrating a relatively low correlation between the two. Future studies can try to improve the trustworthiness of LLM strictness rankings by involving a human-in-the-loop. This can include legal experts verifying the LLM-generated ratings and associated explanations.

There may be some socio-political factors that are correlated with both stricter SA laws and lower SA crime rates, meaning the true relationship between stricter laws and lower crime is even weaker. These confounding variables may include law enforcement practices or political ideologies and can be controlled for and studied in future work. Additionally, the concept of using LLMs to generate quantitative measures of legal statutes can be used in future studies to enable quantitative studies on the effects of various types of laws, supporting more evidence-based policy choices. However, a broader application of the methodology will require data availability of laws for that crime domain, and prompt engineering to form an LLM instruction that leads to accurate rating outputs.

The findings suggest that tougher punishments in the legal system for SA may not be the most effective strategy for combating such crime. Steering efforts towards prevention programs and community initiatives may be more effective solutions. For example, an alternative solution can be to implement a grassroots movement to increase awareness about consent.

References

1. Ahn, J., Verma, R., Lou, R., Liu, D., Zhang, R., Yin, W.: Large language models for mathematical reasoning: progresses and challenges. arXiv preprint [arXiv:2402.00157](https://arxiv.org/abs/2402.00157) (2024)
2. Cheong, I., Xia, K., Feng, K.K., Chen, Q.Z., Zhang, A.X.: (a) i am not a lawyer, but...: Engaging legal experts towards responsible llm policies for legal advice. In: The 2024 ACM Conference on Fairness, Accountability, and Transparency, pp. 2454–2469 (2024)
3. DeMatteo, D., Galloway, M., Arnold, S., Patel, U.: Sexual assault on college campuses: a 50-state survey of criminal sexual assault statutes and their relevance to campus sexual assault. *Psychol. Public Policy Law* **21**(3), 227 (2015)
4. Duncan, M.J.: Sex crimes and sexual miscues: the need for a clearer line between forcible rape and nonconsensual sex. *Wake Forest L. Rev.* **42**, 1087 (2007)
5. FBI: Forcible rape rate per 100,000 inhabitants in the united states in 2022, by state (2024). <https://www.statista.com/statistics/232563/forcible-rape-rate-in-the-us-by-state/>
6. Hamilton, M.: Extreme prison sentences: legal and normative consequences. *Cardozo L. Rev.* **38**, 59 (2016)
7. Harasta, J., Novotná, T., Savelka, J.: It cannot be right if it was written by ai: on lawyers' preferences of documents perceived as authored by an llm vs a human. arXiv preprint [arXiv:2407.06798](https://arxiv.org/abs/2407.06798) (2024)
8. Izzidien, A., Sargeant, H., Steffek, F.: Llm vs. lawyers: identifying a subset of summary judgments in a large uk case law dataset. arXiv preprint [arXiv:2403.04791](https://arxiv.org/abs/2403.04791) (2024)
9. Litaina, T., Soularidis, A., Bouchouras, G., Kotis, K., Kavakli, E.: Towards llm-based semantic analysis of historical legal documents (2024)
10. Lorenz, K., Kirkner, A., Ullman, S.E.: A qualitative study of sexual assault survivors' post-assault legal system experiences. *J. Trauma Dissociat.* **20**(3), 263–287 (2019)
11. McKnight, P.E., Najab, J.: Kruskal-wallis test. In: The Corsini Encyclopedia of Psychology, pp. 1–1 (2010)
12. McKnight, P.E., Najab, J.: Mann-whitney u test. In: The Corsini Encyclopedia of Psychology, pp. 1–1 (2010)
13. Moorhouse, J.C., Wanner, B.: Does gun control reduce crime or does crime increase gun control. *Cato J.* **26**, 103 (2006)
14. Nathan, G.: Sexual assault charges & penalties by state (2024). <https://www.federalcharges.com/sexual-assault-laws-charges/>
15. Ostertagova, E., Ostertag, O., Kováč, J.: Methodology and application of the kruskal-wallis test. *Appl. Mech. Mater.* **611**, 115–120 (2014)
16. Pendyala, V.S.: Machine Learning for Societal Improvement, Modernization, and Progress. IGI Global (2022)

17. Pendyala, V.S.: Artificial intelligence can take the politics out of policymaking. *Fulcrum* (2023). <https://thefulcrum.us/Business--Democracy/Innovation--Incubation/ai-government>
18. Pendyala, V.S.: Toward an ai-driven democracy. *Newsweek* (2023). <https://www.newsweek.com/toward-ai-driven-democracy-opinion-1845423>
19. Pendyala, V.S.: Harnessing artificial intelligence for an affordable quality of life. *VISIBLE Magazine* (2024). <https://visiblemagazine.com/harnessing-artificial-intelligence-for-an-affordable-quality-of-life/>
20. RAINN: Scope of the problem: Statistics (2024). <https://rainn.org/statistics/scope-problem>
21. Sedgwick, P.: Pearson's correlation coefficient. *Bmj* **345** (2012)
22. Sengupta, S., Dave, V.: Predicting applicable law sections from judicial case reports using legislative text analysis with machine learning. *J. Comput. Social Sci.* **5**(1), 503–516 (2022)
23. Surden, H.: Machine learning and law: an overview. In: *Research Handbook on Big Data Law*, pp. 171–184 (2021)
24. Vásquez, B.E., Maddan, S., Walker, J.T.: The influence of sex offender registration and notification laws in the united states: a time-series analysis. *Crime Delinquency* **54**(2), 175–192 (2008)



Application of Large Language Models for Search Engine Optimisation: Methodology for Assessing the Efficiency of Content Creation

Klaudia Błażyczek²  and Grzegorz Chodak¹  

¹ Department of Artificial Intelligence, Wrocław University of Science and Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
grzegorz.chodak@pwr.edu.pl

² Faculty of Management, Wrocław University of Science and Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

Abstract. The study explores the application of generative models in creating website content optimised for search engines and evaluates their effectiveness compared to human-generated content. The paper begins by introducing the search engine and SEO issues, followed by a discussion of the generative models employed in the experiment, specifically ChatGPT and DALL·E. The primary focus lies in the presentation of the research methodology, which includes an original approach to designing the experiment for website content evaluation. Furthermore, the paper introduces a novel indicator for measuring SEO effectiveness in terms of website visibility in search engine results. The results of the study compare the performance of three website versions, one of which featured content generated by AI models. Additionally, a survey was conducted to assess the quality of the content from the perspective of users. The findings indicate that content created by a professional human copywriter consistently outperformed generative model-based content in both SEO metrics and user quality evaluations. These results highlight the limitations of current generative models in replicating the nuanced expertise of human professionals in the context of SEO-driven content creation. These findings point to a broad field of research into the quality of content produced by generative models.

Keywords: Large Language Model · Search Engine Optimization · Generative Model · ChatGPT · Website Content

1 Introduction

Developments in technology are constantly influencing the way businesses operate. In particular, we are referring to artificial intelligence technologies, which are opening up new opportunities for automation, personalisation and content optimisation, which can be crucial for web content management.

The aim of this paper is to propose a methodology for measuring the effects of search engine optimisation (SEO) activities in the context of comparing content created by human copywriters and large generative models. The paper focuses on exploring how modern artificial intelligence tools can impact SEO [1], increasing the effectiveness and visibility of an organisation's websites in search results.

The following research questions were posed:

- How to measure the efficiency of content creation by LLMs in the context of SEO?
- What is the impact of content generated by generative models on the search results of an organisation's sites?
- Can content generated by generative models engage users as effectively as content invented or selected by humans?

The structure of the paper is the following. The first part of the article briefly introduces the topic of search engines and SEO. This is followed by a discussion of the generative models used in the experiment, i.e. ChatGPT and DALL E. The main part of the paper is a demonstration of the research methodology, in which the original method for conducting the experiment on website content evaluation is described in the first part, followed by a description of the original indicator for measuring SEO effects, in the context of the visibility of the company's website in search results. The results of the evaluation of the analysed websites using the proposed indicator were then presented, comparing three versions of the sites, one of which had content created by the generative model. This is followed by a survey in which respondents evaluated the three sites in terms of content quality. The article concludes with the conclusions of the research.

2 Search Engine Optimization

In the field of information retrieval, an Internet search engine is defined as a digital application or service designed to search the global web to find specific resources or content. Its main aim is to facilitate efficient and organised access to the vast amounts of data available on the Internet. Central to the interaction between the user and the search engine is the query interface, where the user enters search terms [12]. The search engine processes these queries and presents the results on a search engine results page (SERP), which serves as an interface displaying ranking links, summaries and multimedia content relevant to the user's query [14]. The SERP is therefore a key element of the user experience, linking the user's query to the search engine results.

Query processing starts with the query parser, which breaks down the user input into meaningful components and relevant phrases for further analysis [15]. The core functionality of the search engine is its search algorithm, which works by comparing phrases from the query parser with an indexed database containing key phrases, URLs and related metadata [12]. This index is populated through an indexing process, which involves the collection, analysis and storage of data from web pages [13].

Search engines use indexing robots, also known as robots or spiders, to explore and collect data from websites. These robots index new or updated web content as a priority [3]. In addition, search engines build datasets that map relationships between topics and concepts. This integration enhances the search engine's understanding of user intent and contextual relevance of queries [3].

Ranking of results is crucial and is based on the various data sources in the index. These sources include textual content, graphics, documents, hyperlinks and metadata, all of which are evaluated by the search algorithm to determine their relevance and quality [2]. This approach ensures the delivery of results tailored to users' needs.

SEO is a broad field that cannot be described in its entirety in a short text, so only the most important elements for building a website's position in search results, in the context of content quality, will be mentioned here. In order to achieve a high ranking in search engines such as Google, Bing or Baidu, a website should contain rich and unique content. In addition, it is important to fill in the page title tags (<title>) with the right keywords. Header tags (<H1>, <H2>, <H3>) should be used in the text. Pages should also have correctly defined metatags (<description>, <keywords>) containing relevant keywords [12–14]. In addition to the elements mentioned, there are many other SEO elements, such as internal and external links, server quality or Core Web Vitals introduced in 2022 by Google [16] and others.

3 AI Generative Models for Content Creation

There are currently several large generative models on the market and new ones are being developed all the time. Among the most popular are: ChatGPT, Gemini, Qwen, Copilot, LLaMa, Mistral, Command Cohere, Claude, Grok, Poe. As ChatGPT was used in the experiment described later for textual content creation, it will therefore be briefly characterised.

3.1 ChatGPT

ChatGPT is the artificial intelligence model created by OpenAI (AI research lab in San Francisco, founded in 2015 [3]) to produce text content and offer a chatbot capable of engaging in conversations with human-like qualities [4].

Generative models such as ChatGPT and DALL-E are becoming increasingly important, contributing to the evolution of SEO and SEM practices. According to the 'State of SEO 2024' report, almost 82% of businesses are feeling the impact of these technologies on their SEO strategies [5]. The use of generative technologies still faces barriers and requires further research and integration with traditional content management methods [6].

ChatGPT utilizes a multi-layered transformer architecture to generate responses to user queries. A transformer is a specific type of neural network architecture, and a neural network is a set of data processing nodes that mimic the action of the human brain [7]. ChatGPT uses transformers specialised in natural language processing. The primary advantage of transformers is their efficient parallelism, enabling the scaling and training of large models. This architecture allows the system to grasp the context and interpret the meaning of ambiguous words by analyzing their surrounding words within a sentence [8]. This is a difficult task for a machine, but ChatGPT achieves this, making its answers human-like and understandable.

As of November 22, 2024, OpenAI offers several versions of ChatGPT, each with distinct features and capabilities [9]:

1. GPT-3.5: The foundational model that powers the free version of ChatGPT.
2. GPT-4: An advanced model available to ChatGPT Plus subscribers. GPT-4 enhances the quality of responses, offering more nuanced and context-aware interactions compared to GPT-3.5.
3. GPT-4o (“Omni”): Introduced in May 2024, GPT-4o is a multimodal model designed to process and generate text, images, and audio. It is accessible to all users, with free users having certain usage limits and Plus subscribers who have higher message allowances.
4. GPT-4o mini: A streamlined version of GPT-4o, released in July 2024. It offers similar multimodal capabilities with optimized performance, replacing GPT-3.5 Turbo in the ChatGPT interface.
5. o1-preview: Launched in September 2024, this model focuses on complex problem-solving by spending more time analyzing tasks before responding. It excels in areas like competitive programming, mathematics, and scientific reasoning.
6. o1-mini: A faster and more cost-effective variant of o1-preview, providing enhanced reasoning capabilities.

Work on new generative models is still in progress, so expect further versions of ChatGPT in the near future.

3.2 DALL·E - Graphic Content Generator

DALL·E represents a collection of deep learning models created by OpenAI for generating images from textual descriptions. The original DALL·E, introduced in 2021, is a 12-billion parameter autoregressive transformer model that generates images by sequentially predicting pixels based on textual input [10]. In 2022, OpenAI released DALL·E 2, which employs a two-stage process: a prior model that generates CLIP image embeddings from text, and a decoder that produces images conditioned on these embeddings. This approach enhances the realism and accuracy of generated images [11].

DALL·E 3, announced in 2023, further improves upon its predecessors by integrating more advanced language understanding and image generation capabilities, resulting in images that closely align with complex textual prompts [9].

4 Methodology of Research

Two research methods will be used during the study: experimental and survey. As part of the experimental research, three organisation websites will be evaluated.

The study was carried out using three websites as an example, to investigate how they would rank in search results. The experiment lasted 13 weeks, during which the sites’ content was modified every three weeks. Key elements such as key phrases, page meta descriptions, titles, text, links, images and alternative image texts were subject to changes to optimise each site’s search engine positioning. These changes were made based on the sites’ current position data, allowing the content to be adjusted as effectively as possible. During the final week of the study, the content of the sites was not changed in order to conduct a user survey to assess how the modified content affected their engagement and perception of the sites.

In order to carry out authoritative research, one of the sites will be a kind of benchmark. In general, its technical SEO, content and other elements important from an SEO perspective will be done poorly, will be of low quality or will not exist. We can say that site 1 will be the reference point for the others, site 2 will have content creation and selection by copywriters, and site 3 will have content generated by generative models. This division will address the problem of the unknown impact of content generated by generative models on search results.

Each site covered the same subject matter, had the same information architecture and an identical number of pages on the given subject matter. In addition, each site domain had an SSL certificate purchased from the hosting provider. Site 1 did not have semantic HTML, Long Tail SEO phrases, page meta descriptions or an XML sitemap, while sites 2 and 3 had the listed elements.

In sites 2 and 3, the key phrases will be in both metatags and content, where copywriters and the chosen generative model will be responsible for coming up with them in turn. All three sites had page titles, page text, images, domain name and URLs. Site 1 did not have alternative text to images, while sites 2 and 3 did. All three sites were hosted with the same service provider to minimise the impact of Core Web Vitals on the positioning of the sites relative to each other.

In the case of Site 1, the content will be created and selected by someone with no SEO experience.

For Site 2, content will be selected by a team of two copywriting professionals with at least 2 years' experience. This is in order to best replicate the real-life situation in which people are involved in creating website content. The previously mentioned elements were created and selected while adhering to SEO rules and recommendations.

For site 3, content was generated using the ChatGPT-4 and DALL E 3 generative models. The listed elements were generated while adhering to SEO rules and recommendations. Prompting of the model followed the prompt engineering recommendations described in the publication [17]. An example of the set of prompts used to build the page content is shown below.

User

The theme of my website, is guide services in the Sudetenland, which will be a showcase for a group of mountain guides in the Sudetenland who offer their services to people planning mountain trips in this mountain range. The website will have 5 pages:

- *Home page with favourite routes and general information,*
- *Tours, which will include the most popular and new tours proposed by the guides,*
- *A Tourist Guide, which will include practical advice for those planning a mountain trip and information on mountain safety,*
- *About Us, which will contain information about the guides and the team,*
- *Contact, which will include a contact form.*

Remember this information

ChatGPT

Information about your website about the services of guides in the Sudetenland has been memorised. If you need help with content creation, page design, or any other aspect of the site, let me know! How can I help now?

User

My goal is to get my website as high as possible in search results through SEO. Please note this information

ChatGPT

I have remembered that your goal is to position your website about Sudetenland guide services as best as possible in the search results thanks to SEO. If you need advice on search engine optimisation, content strategy, or technical aspects of SEO, let me know how I can help you!

User

The tours page will contain some of the most popular and interesting hiking tours in the Sudetenland. Make up a title for this page, a name in the url path, a subtitle for the section with the most popular hikes and a subtitle for the section with the most interesting hikes.

ChatGPT

Here are my suggestions:

- ***Page title:** ‘Discover the Sudetes: a Guide to Tours’.*
- ***Name in URL path:** ‘/discover-tours’.*
- ***Subtitle for the section with the most popular hikes:** ‘The Trails That Tourists Love’.*
- ***Subtitle for the section with the most interesting excursions:** ‘Gems of the Sudetes: Excursions Full of Mysteries’.*

Such names and sub-headings can catch the attention of visitors, encouraging them to further explore the content of the website and the services you offer.

Of course, a reliable assessment of the quality of prompts would require a broader analysis and many more experiments. Prompt engineering, however, is not the focus of this publication, so only an example is included to show how content was generated for the site.

Due to the limited volume of the paper, detailed analyses of the textual and graphic content will not be presented. These are available in the paper [1].

Once the sites were made public on the World Wide Web, data was collected throughout the study period on the position of all three sites in Google search results for specific, pre-selected key phrases. The experimental study lasted 13 weeks, of which for the first 12 weeks the content on all the study sites was modified every 3 weeks (site hosting and 3 iterations of modifications), in order to increase the chances of more visits by the indexing robots, which was expected to have a positive impact on positioning. Key phrases,

page meta descriptions, titles, texts, links, images and alternative texts of images were modified. During the modification, the authors of the changes had data on the positioning of the sites for specific phrases so that they could make the best possible changes in their opinion. During the last week, the content of the sites was not changed for the survey, which required the content to remain unchanged.

The Evaluation of the Website

In terms of the business aspect, understood as the visibility of the brand in search results, it is worth noting whether the site is in the top 10 search results, due to the estimated CTR (Table 1).

Table 1. Organic CTR for Google SERP (not including ads)

Position in SERP	CTR
#1	27.6%
#2	15.8%
#3	11.0%
#4	8.4%
#5	6.3%
#6	4.9%
#7	3.9%
#8	3.3%
#9	2.7%
#10	2.4%

Source: own elaboration based on data from [2]

In order to assess this, the author of the paper proposes the following formula [Błażyczek, 2024]:

$$P = \sum_{i=0}^n \sum_{j=0}^m \frac{V_j * D_j * CTR_{p_{ij}}}{p_{ij}}$$

Where:

P – evaluation of the website,

n – numer of measurement days,

m – the number of phrases under which the site was positioned,

p_{ij} – site position in SERP for the phrase j at day i ,

$CTR_{p_{ij}}$ – CTR for the position occupied by the site for a given phrase for a specific day,

D_j – normalized difficulty ratio for the phrase.

Normalized difficulty ratio was determined with the following formula:

$$D_j = \frac{\max(d_j - d_{min}, 1)}{d_{max} - d_{min}}$$

Where:

d_j - difficulty for the phrase j ,

d_{min} - minimum difficulty among all analysed phrases,

d_{max} - maximum difficulty among all analysed phrases.

V_j - normalized volume factor for the phrase j , which is calculated with the following formula:

$$V_j = \frac{\max(v_j - v_{min}, 1)}{v_{max} - v_{min}}$$

Where:

v_j - volume for the phrase j ,

v_{min} minimum volume among all analysed phrases,

v_{max} maximum volume among all analysed phrases.

The proposed formula for evaluation of the website, takes into account the number of days the study lasted, the number of phrases (both those for which the site was positioned and all phrases included in the study), the site's position for a given phrase on a given day, and the normalized difficulty and volume factors for each phrase. The difficulty and volume values were normalized due to the fact that they are two different factors, taking on different ranges of values. The normalization was intended to equalize their effect on the score. The proposed formula also takes into account the CTR for the position occupied by a site on a given day for a particular phrase. For positions from 1 to 10, the CTRs take values like those in Table 1. For the remaining positions, it takes 0%. Thus, this formula is also intended to evaluate the overall positioning of the site for selected phrases over a certain period of time, except that a value different from zero is awarded only for positions in the top 10, which has its justification from a business perspective. A low CTR means that a relatively small percentage of users who saw a site in search results will decide to visit it. The higher the CTR, the better one can evaluate the positioning of such a site from a business perspective.

The Results of Website Evaluation

To select the phrases to be used in the position study, a collection of as many potential phrases as possible related to the sites' topics was first created. For this purpose, the Ask The Public platform was used, through which several potential phrases were obtained, such as: Sudet guide, Sudet mountains etc. Subsequently, having already obtained several phrases, they could be entered into the Keyword Manager's collection (it is part of the Keyword Suggestion Tool from SERanking Software) and thanks to the tool's capabilities, the collection was expanded to 57 phrases.

The 10 phrases with the highest volume value and the 10 phrases with the highest and 10 phrases with the lowest difficulty were selected for the study.

Next, using the site positioning data obtained from software SERanking, evaluation of the website (P) scores were calculated for each site. These are presented in the Table 2.

Table 2. The final evaluation of websites (P)

Website	Evaluation of website (P)
Website1	0,00008565
Website2	0,00166378
Website3	0,00025317

Source: own elaboration

In terms of site positioning among the top ten results in the SERPs, determined by the formula for, website 2 performed more than 6 times better than website 3 and more than 19 times better than website 1. From a business perspective, the higher a site’s position in search results, the higher its estimated CTR. A higher CTR means greater online visibility and effectiveness than lower-ranked sites, which is why the score, which takes this into account in a site’s score, is so important.

Figure 1 shows the P-evaluation measure of the three sites throughout the experiment. It can be seen that after an initial period in which the position of all three sites in the SERPs was zero (the time needed to position a site in the search engines is at least a few weeks), by far the best result was achieved by the site with content created by an SEO expert (website 2). The result of website 3 with content created by ChatGPT is clearly inferior to website 2. As expected, the weak result was achieved also by website 1, the reference site, where no SEO activities were carried out.

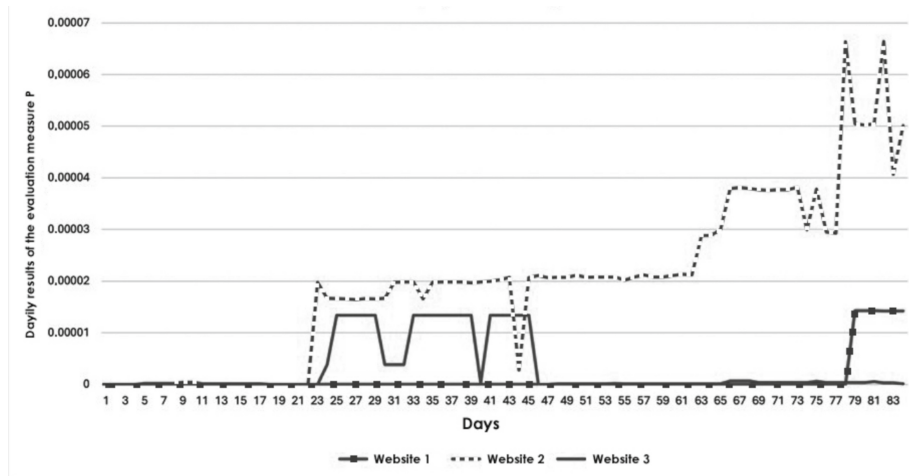


Fig. 1. Daily results of websites evaluation measure P. Source: own elaboration

Based on the positioning results of the sites it can be concluded that site 2 performed the best among the analysed sites. It positioned itself for the largest number of phrases and ranked relatively well for them, which translated into a high score for the proposed score P.

5 Survey Research

During the final week of the study, a survey was conducted in which users were asked for their opinions on the content of the sites. The survey included a metric consisting of 6 questions. The metrics adopted were the following: gender (female, male, do not want to specify/other); age (18 - 25, 26 - 35, 36 - 45, 46 - 60 and over 60); size of place of residence (village, town with up to 10 000 inhabitants, town with 10 000 to 100 000 inhabitants, town with 100 000 to 500 000 inhabitants and town with over 500 000 inhabitants); education (primary, secondary, higher incomplete (bachelor's, engineering), higher technical, higher non-technical and none); employment status (pupil/student, student/student, employed, unemployed, pensioner/retired and other (economically inactive not being a pensioner or full-time student)), student/student status (studying, not studying, attending school, prefer not to answer). Due to the rather complex questionnaire and the need to evaluate three sites, it was only possible to obtain the responses of 35 respondents. Due to the non-small number of respondents, the results cannot be generalised. However, it should be noted that the main purpose of the publication was not to evaluate content, but rather to indicate a methodology for its evaluation.

The survey also contained three sections for each site, with seven questions in each. The section questions were as follows:

- Do you think the content on the site could be useful for planning a holiday in the Sudetenland? (Possible answers: yes, no and hard to say),
- Was the data and information on the website presented in a way that was easy to understand? Rate on a scale of 1 (lowest rating) to 5 (highest rating),
- How much do you trust the content presented on this website? Rate on a scale of 1 (lowest rating) to 5 (highest rating),
- How would you rate the visual experience of the content on this site? Rate on a scale of 1 (lowest rating) to 5 (highest rating),
- Is the content on this site coherent and logically linked? Rate on a scale of 1 (lowest rating) to 5 (highest rating),
- How would you rate the writing style of the content on this site? Rate on a scale of 1 (lowest rating) to 5 (highest rating),
- Who do you think is the author of the content on the site? (Possible answers: copywriters, artificial intelligence (generative models i.e. ChatGPT, Gemini, DALL E) and people with no experience in content creation).

Results of Survey Research

Although website 3 was mostly rated as having content created using generative models, it received relatively positive ratings in terms of content consistency and relevance (high ratings in the 'Definitely Yes' and 'Yes' responses). The level of confidence in the content on this site was mixed. 17.1% of respondents indicated that they did not trust the content presented. This is the highest percentage of all sites. Regarding the visual evaluation, although the majority of evaluations were in the positive range, there are critical statements and doubts in the respondents' answers (7 undecided and 3 negative). Mixed reactions to the visual presentation may suggest differences in the way the content generated by the AI is structured or presented, which may not always match users'

expectations. The website, despite scoring well in terms of consistency, has a trust issue from users. Respondents' answers suggest that the way content created by generative models is not always perceived as trustworthy.

The content on site 2, prepared by professional copywriters, scored best in terms of ease of understanding the content. Users also found the site very trustworthy. 71.4% of respondents said that they found the information there trustworthy. In this respect, the site again scored better than the other sites. The visual assessment and consistency of content scored well. This should indicate that the careful preparation of the content on both a content and aesthetic level will be appreciated by users. In contrast, the high scores for ease of interpretation of information underline the importance of clear presentation of content. In the case of website 2, respondents were not able to say clearly who the author of the content on the site was. However, they judged well that the content was probably not created by artificial intelligence.

Despite the fact that the content on website 1 was selected and created by someone with no experience in this area, it received the highest marks in terms of the visual aspects of the site design. In terms of writing style, the value obtained by Site 1 was the lowest in the entire survey. 34% of people did not know how to rate the writing style, meaning that clarity and consistency in the way content was presented was not achieved. People's confidence index was lower than for website 2, but higher than for website 3.

Respondents' answers regarding website 2, where content was created and selected by experienced copywriters, show that professionally prepared content significantly influences the perception of content as easy to understand, trustworthy and aesthetically pleasing. For businesses, this means that investing in qualified human resources for content creation can bring significant benefits in terms of user engagement and satisfaction. The results for website 3, which used content generated by generative models, showed lower trust ratings compared to website 2, suggesting that users may be more distrustful or critical of automatically generated content. Companies should be aware of this challenge and consider how best to present the source of content to build trust and credibility. In contrast, the results for Site 1 indicate that visual appeal can partially compensate for deficiencies in the content's substantive quality. For this reason, companies should also ensure that the visual quality of their sites is high, which can have a positive impact on first impressions and overall evaluation by users.

6 Conclusion

On the basis of the research conducted, it can be concluded that the role of generative models in the context of content creation for sites in terms of SEO for companies seems ambiguous. On the one hand, the use of generative models for content creation appears to be less effective compared to content created by experienced copywriters, especially in the context of metrics such as positioning in search results and user trust. On the other hand, generative models can offer significant cost savings related to content management, reducing the need to involve human resources in content creation processes. Research shows that although AI-generated content does not yet reach the level of content created and selected by humans, its use can significantly reduce a company's operational costs. Generative models can reduce the workload of copywriting teams, allowing companies

to manage human and financial resources more efficiently. In the context of CTR and SEO, the higher rankings achieved by copywritten content results in greater visibility in search engines, which translates into a higher CTR and potentially more revenue from advertising or direct sales. In comparison, pages with content generated by generative models may require additional optimisation to achieve similar results.

The results of the study indicate that a site whose content was generated by generative models achieved worse SEO results than a site whose content was created and selected by copywriters.

However, it should be noted that the quality of content generated by generative models may be largely dependent on the quality of the prompts used to create it. It is possible that a different set of prompts could have generated better content, which could have resulted in better positioning of the website. For this reason, a definitive statement that generative models create worse content than copywriters requires a lot of additional research and prompt engineering experiments. This research should include a variety of approaches to prompt creation, different structures and vocabulary, as well as iterative refinement of prompts based on test results as well as applying in-context learning method. Only through such a comprehensive approach would it be possible to accurately assess the potential of generative models to create content that can compete with content created by experienced copywriters.

Analysis of the survey data shows that content generated by artificial intelligence has lower user engagement rates compared to content created by humans. Users consider content created by generative models as less authentic and interesting, which affects their engagement and overall experience. Sites with content created and selected by copywriters score higher for the clarity, message and appeal of the content, leading to better user engagement. However, the differences in respondents' assessment of the sites are much smaller than for the evaluation indicator P.

It can be concluded that sites with content generated by generative models engage users less well than sites with content invented or selected by copywriters.

The use of AI generative models to generate content on websites to improve SEO raises also significant ethical issues, including the potential for misinformation, manipulation of search engine algorithms and reduced transparency of content authenticity. Addressing these issues requires a critical examination of the accountability framework for created content and the development of guidelines to ensure fair and responsible implementation of AI technologies in digital marketing.

References

1. Błażyczek, K.: Using generative models using artificial intelligence algorithms to manage website content, Master Thesis, Wrocław (2024)
2. Dean, B.: We Analyzed 4 Million Google Search Results. Here's What We Learned About Organic CTR (2023). <https://backlinko.com/google-ctr-stats>. Accessed 20 Nov 2024
3. OpenAI, About <https://openai.com/about>. Accessed 21 Nov 2024
4. OpenAI, Planning for AGI and beyond. <https://openai.com/blog/planning-for-agi-and-beyond>. Accessed 28 Oct 2024
5. Search Engine Journal. State of SEO 2024. <https://www.searchenginejournal.com/state-of-seo-in-house/>. Accessed 22 Nov 2024

6. Chodak, G., Błażyczek, K.: Large language models for search engine optimization in e-commerce. *Advanced Computing*, 26 Marzec, pp. 333–344 (2024)
7. Loukides, M.: *What are ChatGPT and its friends?* O'Reilly Media (2023)
8. Baker, P.: *ChatGPT for dummies*. Wiley, New Jersey (2023)
9. ChatGPT—Release Notes. <https://help.openai.com/en/articles/6825453-chatgpt-release-notes>. Accessed 22 Nov 2024
10. Ramesh, A., et al.: Zero-shot text-to-image generation. In: *International Conference on Machine Learning*, pp. 8821–8831. Pmlr (2021)
11. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, vol. 1(2), p. 3 (2022)
12. Ledford, J.: *Search engine optimization bible*, 2nd edn. Wiley, Indianapolis (2009)
13. Shenoy, A., Prabhu, A.: *Introducing SEO: your quick-start guide to effective SEO practices*, 1st edn. Apress, Govardhan Nagar (2016)
14. Enge, E., Spencer, S., Stricchiola, J.: *The art of SEO*, 3rd edn. O'Reilly Media Inc, Sebastopol (2016)
15. Marketbrew.ai, *Understanding Query Parsers: how Search Engines Process Your Searches* (2024). <https://marketbrew.ai/understanding-query-parsers-how-search-engines-process-your-searches>. Accessed 25 Nov 2024
16. Dobbala, M.K., Lingolu, M.S.S.: Web performance tooling and the importance of web vitals. *J. Technol. Innov.* 3(3) (2022)
17. Phoenix, J., Taylor, M.: *Prompt engineering for generative AI*. Wydanie pierwsze red. brak danych: O'Reilly Media (2024)



Real-Time Query Management with FAQ Generation for Large Scale Online Classes

Kamal Bijlani¹ , Sreya Sunil¹ , Navabhaarathi² ,
S. P. Saran Dharshan² , E. Vigneshwar² , and A. Aadharsh Aadhithya²

¹ School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Faridabad, India
`{kamal,sreya}@amritaai.org`

² School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Coimbatore, India

Abstract. This study introduces an AI-driven approach for real-time query management and FAQ(Frequently Asked Question) generation, addressing the challenges of scalability and immediate responsiveness in large-scale online learning environments. During live sessions, student queries are embedded and evaluated against the FAQ knowledge base using a dissimilarity metric. Queries exceeding a predefined dissimilarity threshold are escalated to instructors, while lower-dissimilarity queries are addressed by a Retrieval-Augmented Generation (RAG), optimized to ensure minimal latency and high accuracy in real-time responses. We build the RAG by embedding course material into a high-dimensional vector space, organizing it through topic clustering, and constructing an FAQ database pre-validated by instructors. The system achieves a notable answer relevancy score of 88.65% and a contextual precision score of 88.25% underscoring its balance of precision and adaptability.

Keywords: Scalable Online Learning · Query Similarity Analysis · Automated Query Handling

1 Introduction

In recent years, online learning has firmly established itself as a central mode of educational delivery, offering scalable and adaptable solutions that transcend traditional geographical and infrastructural limitations. With advancements in technology, digital classrooms have become essential in democratizing access to education, facilitating seamless participation of hundreds or even thousands of students concurrently. However, as observed in large-scale deployments such as the T10KT project initiated by IIT Bombay through the A-VIEW platform [1], managing real-time interactivity and ensuring sustained engagement within such expansive settings pose significant challenges. These issues are exacerbated by connectivity constraints and high student-to-teacher ratios, which hinder personalized interactions and timely feedback.

This challenge in large-scale online classes is compounded by the overwhelming influx of questions during live sessions, often exceeding instructors' capacity

for immediate and effective responses. As Wang and Woo (2007) [2] explored, synchronous communication methods foster immediacy and engagement, while asynchronous approaches provide flexibility and encourage deeper reflection. However, balancing these modalities in dynamic, large-scale settings remains a significant challenge. To address these issues, recent studies underscore the transformative potential of artificial intelligence (AI) in education, particularly through natural language processing (NLP) and machine learning (ML) techniques that enable scalable automation of student interactions. Recent advancements in large language models (LLMs) have further revolutionized automated text generation, achieving state-of-the-art performance in producing intelligent, contextually relevant responses. For instance, OpenAI’s GPT-3 [3] and GPT-4 [4] demonstrate exceptional capabilities in understanding complex queries and generating fluent, accurate responses. Building on this, Retrieval-Augmented Generation (RAG) models, introduced by Lewis et al. (2020) [5], enhance these systems by integrating retrieval mechanisms with generative models, enabling use-case-specific customization. Earlier foundational work by Heilman and Smith (2010) [6] introduced a statistical framework for automatic question generation, laying the groundwork for FAQ systems capable of dynamically adapting to evolving student inquiries. By automating the resolution of repetitive queries and escalating complex or context-sensitive ones for human intervention, AI-driven systems like the one proposed in this study represent a significant step forward in tailoring query management for large-scale educational settings.

In response to these needs, our research proposes a system for real-time query management with FAQ generation that leverages AI to manage and prioritize student questions efficiently during live sessions. By combining NLP techniques for question classification with ML algorithms for prioritizing queries, our approach ensures that repetitive questions are managed through automated responses, while unique or contextually complex queries are escalated to instructors for direct engagement. Through this integration, the proposed system aims to streamline the interactive capacity of online platforms, enhancing the quality of interaction without burdening instructors, thus addressing scalability challenges in online education and promoting educational equity and access. The system is designed to support instructors in managing queries within a constrained timeframe, focusing on contextually accurate and reliable responses derived from a curated knowledge base deemed relevant by the instructor. By prioritizing precision and timelines, the system enhances the efficiency of real-time classroom interactions. Our contributions include:

- AI-driven query management system for large-scale online classes.
- Semantic embedding using the paraphrase-mpnet-base-v2 model for topic clustering.
- Structured FAQ repository with in-chapter, across-chapter, out-of-context, and application-based categories.
- Automated response generation using a Retrieval-Augmented Generation (RAG) model.
- Dissimilarity scoring framework for evaluating query alignment with content.

- Real-time escalation of high-dissimilarity queries to instructors.
- Adaptive framework to support complex real-time learning interactions.

The remainder of this article is organized as follows: Sect. 2 provides a review of related works that informed the development of the proposed system, particularly in the domain of AI-enhanced educational technologies. Section 3 details the methodology and results, including semantic embedding (3.1), clustering and FAISS indexing (3.2), FAQ repository compilation and response generation (3.3), dissimilarity measurement and threshold establishment (3.4), real-time query handling (3.5), and the user interface demonstration (3.6). Section 4 concludes the article with a summary of contributions and directions for future research.

2 Related Work

The foundation of this study is built upon several influential works in the domain of AI-enhanced educational technology. Research on large-scale online learning environments, such as those highlighted by Luckin et al. (2016) [7], underscores the potential of AI to support personalized and adaptive learning. Their study emphasizes AI's capacity to cater to diverse learning needs, a premise that informs the development of AI-driven solutions for query management in our system. The integration of real-time query handling mechanisms in educational settings has gained traction with the rise of MOOCs and other digital learning platforms. Drachsler and Kalz (2016) [8] introduced the MOOC and Learning Analytics Cycle (MOLAC), identifying challenges in data integration and student privacy, which are critical considerations in designing our query-handling system. Similarly, McInnes and Pedersen (2013) [9] explored semantic similarity and relatedness metrics to enhance word sense disambiguation, contributing foundational techniques for improving automated FAQ systems in educational contexts. Building on the advancements in automated question-answer generation, Aithal, Rao, and Singh (2021) developed mechanisms for generating question-answer pairs and assessing question similarity in educational platforms. These methodologies, particularly the use of cosine similarity for filtering questions, have been instrumental in our design, helping to ensure that only contextually relevant questions are prioritized for instructor review [10]. Furthermore, in large-scale blended classrooms, hybrid architectures such as those explored in anonymous case studies demonstrate the importance of scalable and flexible learning designs. This case study highlights the practical considerations in balancing real-time feedback and engagement in mixed learning environments, offering valuable insights for the development of our own system. In addition, Al-Zahrani and Alasmari (2024) [11] examined the operational and ethical implications of AI in education, raising critical considerations for data privacy and responsible AI use in learning settings. These ethical frameworks are essential to our approach, ensuring that the deployment of AI in query management adheres to privacy standards and fosters an inclusive learning environment.

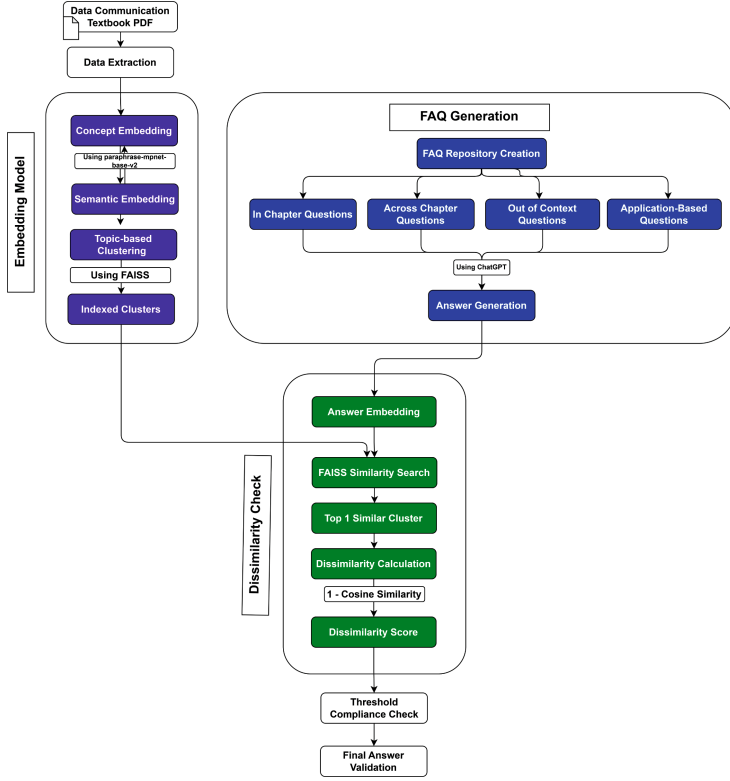


Fig. 1. Proposed Pipeline

3 Methodology

This study introduces a robust three-module methodology for query management and FAQ generation, leveraging advanced AI techniques. As outlined in Fig. 1, the system employs semantic embedding to organize content, an FAQ generation framework for efficient response creation, and a dissimilarity-check module to ensure contextual alignment and accuracy. To ensure the system remains current and adaptable to evolving curricula, we propose a dynamic content update module that allows instructors to seamlessly upload updated course material, such as revised textbooks, lecture notes, and supplementary resources. Instructors can upload course material directly into the platform as a primary knowledge source. For initial development and testing, we selected Data Communications and Networking by Behrouz Forouzan [12], a widely taught textbook in undergraduate programs at Amrita University.

3.1 Semantic Embedding

To ensure optimal content representation, the textbook content is processed chapter-by-chapter, allowing for a logical segmentation of ideas. Key concepts, phrases, and sentences are identified and extracted from each chapter, thus creating a focused dataset that retains the contextual essence of each topic. As visualized in Fig. 1 as embedding model, the next step is to transform these extracted concepts into high-dimensional semantic vectors using the paraphrase-mpnet-base-v2 model [13], a transformer-based model optimized for capturing subtle semantic relationships between phrases and sentences. In this configuration, each concept is encoded as a 768-dimensional vector, where semantic similarity is preserved through proximity in the vector space. These embeddings form the foundation for effective clustering and similarity comparisons, as they allow each question-answer pair to be mapped within the same space.

3.2 Organizing Content Through Clustering and FAISS Indexing

With embedded concepts prepared, the methodology proceeds to organize these vectors into clusters, utilizing the FAISS (Facebook AI Similarity Search) [14] library for scalable and efficient similarity-based retrieval. The primary goal here is to structure the high-dimensional space into conceptually cohesive groups, or clusters, that reflect the textbook's thematic organization. To accomplish this, we employ the k-means clustering algorithm within FAISS. K-means was selected for its suitability in handling large datasets and compatibility with vector-based representations, making it an ideal choice for organizing embedded concepts. Each cluster in this system represents semantically related concepts, and each cluster has a centroid vector that acts as a central reference point. Chapters with dense or diverse content are segmented into a higher number of clusters to capture the full range of ideas, while simpler chapters are divided into fewer clusters to prevent excessive fragmentation. The distance metric used for clustering is L2 (Euclidean distance), which complements the cosine similarity metric employed in dissimilarity measurements, creating a cohesive system that integrates distance and similarity for accuracy and efficiency. After clustering, the FAISS index is fine-tuned for fast retrieval. One critical parameter in FAISS, *nprobe*, which controls the number of clusters scanned during similarity searches, is adjusted to achieve an optimal balance between retrieval speed and accuracy. Higher values of *nprobe* increase accuracy by scanning more clusters but can lengthen processing time. In this system, the *nprobe* setting is customized to meet the project's real-time response requirements, with additional adjustments based on user query characteristics [15].

3.3 FAQ Repository Compilation and Response Generation

The system's FAQ repository is meticulously curated to cover a range of question types, each associated with different expectations of alignment and contextual

relevance. Four primary categories are defined: in-chapter, across-chapter, out-of-context, and application-based questions (FAQ Generation Module, Fig. 1). The system enables faculty to upload course-specific material, including assignments, question banks, textbooks, articles, research papers, and notes, at their convenience prior to the class. The uploaded content is then processed to create structured FAQ entries aligned with the course objectives. In-chapter questions are limited to single chapters, aligning closely with specific content and expected to have low dissimilarity scores. Across-chapter questions require cross-referencing multiple sections, resulting in slightly higher dissimilarity ranges that allow for some deviation in context. Out-of-context questions pertain to topics addressed in the textbook but require answers that extend beyond direct, word-for-word content. These questions demand a broader interpretive understanding of the context, resulting in higher dissimilarity scores. Application-based questions extend textbook concepts to real-world scenarios, similarly showing higher dissimilarity due to the broader application scope. For answer generation, OpenAI's ChatGPT [4] is employed to ensure that responses are contextually relevant. ChatGPT accesses chapter-specific embeddings to provide content alignment for in-chapter and across-chapter questions, ensuring that the system can generate responses that reflect the original textbook material. To maintain high alignment, each generated answer is anchored within the context of related chapter embeddings, a process that constrains ChatGPT's output, reducing the likelihood of off-topic information and ensuring consistency with the original material. Additionally, for critical questions, particularly in out-of-context or application-based categories, generated answers undergo post-processing and are subject to human review to verify relevance and accuracy.

3.4 Dissimilarity Measurement and Threshold Establishment

The dissimilarity check module (Fig. 1) quantifies the alignment of generated answers with textbook content using cosine similarity, ensuring contextual relevance through threshold compliance. The measurement of dissimilarity is essential for evaluating the alignment of generated answers with textbook content, providing a quantitative basis for threshold compliance. For each generated answer, an embedding vector is created, and the dissimilarity score [16] is calculated by comparing it with the most similar cluster centroid retrieved from the FAISS index. This comparison utilizes the $1 - \text{cosine similarity}$ formula, where lower dissimilarity scores indicate a stronger alignment with the textbook content.

The system defines a dissimilarity threshold for each question category to assess compliance. In-chapter questions are expected to have a dissimilarity range between 0.2 and 0.4, reflecting close alignment. Across-chapter questions allow for a moderate range of 0.4 to 0.5, accommodating cross-referenced answers. For out-of-context and application-based questions, dissimilarity scores above 0.5 are acceptable due to the broader knowledge required. Threshold compliance is measured as the percentage of answers in each category that meet the expected

dissimilarity range, providing a clear indication of the system’s accuracy in generating contextually relevant answers.

To quantify the alignment between a generated answer and the textbook content, we define the Dissimilarity Score based on cosine similarity [17]. Let:

- $\mathbf{v}_A$ be the embedding vector of the generated answer.
- $\mathbf{v}_C$ be the embedding vector of the closest cluster centroid (retrieved from the FAISS index).

The **Cosine Similarity** between these vectors is calculated as:

$$\text{Cosine Similarity}(\mathbf{v}_A, \mathbf{v}_C) = \frac{\mathbf{v}_A \cdot \mathbf{v}_C}{\|\mathbf{v}_A\| \cdot \|\mathbf{v}_C\|} \quad (1)$$

where:

- $\mathbf{v}_A \cdot \mathbf{v}_C$ is the dot product of vectors $\mathbf{v}_A$ and $\mathbf{v}_C$,
- $\|\mathbf{v}_A\|$ and $\|\mathbf{v}_C\|$ are the magnitudes (Euclidean norms) of the vectors $\mathbf{v}_A$ and $\mathbf{v}_C$, respectively.

The cosine similarity yields a value between -1 and 1 , where:

- A value of 1 indicates maximum similarity,
- A value of 0 indicates no similarity,
- A value of -1 indicates maximum dissimilarity.

To obtain a measure of dissimilarity, we subtract the cosine similarity from 1 . Thus, the **Dissimilarity Score** is defined as:

$$\text{Dissimilarity Score} = 1 - \text{Cosine Similarity}(\mathbf{v}_A, \mathbf{v}_C) \quad (2)$$

Expanding this, we have:

$$\text{Dissimilarity Score} = 1 - \frac{\mathbf{v}_A \cdot \mathbf{v}_C}{\|\mathbf{v}_A\| \cdot \|\mathbf{v}_C\|} \quad (3)$$

Thresholds for Dissimilarity Scores by Question Category

Each question type has a specific range for acceptable dissimilarity scores:

- **In-Chapter Questions:** $0.2 \leq \text{Dissimilarity Score} \leq 0.4$
- **Across-Chapter Questions:** $0.4 \leq \text{Dissimilarity Score} \leq 0.5$
- **Out-of-Context and Application-Based Questions:** Dissimilarity Score > 0.5

A compliance check is performed by evaluating whether the dissimilarity score of each answer falls within the specified range for its question category. The compliance rate (as a success metric) is defined as:

$$\text{Compliance Rate} = \frac{\text{Number of Answers Within Threshold}}{\text{Total Number of Answers in Category}} \times 100 \quad (4)$$

Table 1 provides an overview of dissimilarity ranges, mean, median, standard deviation, and the number of questions for each question category. These ranges define the expected alignment levels of each question type with the textbook content, aiding in threshold compliance assessment. The dissimilarity scoring framework not only establishes alignment thresholds but also enables efficient query handling by organizing incoming questions in descending order of dissimilarity.

Table 1. Summary of dissimilarity ranges, mean, median, standard deviation (StD), and question count for FAQ categories.

Category	Range	Mean	Median	StD	No. of Questions
Out of context	0.893–0.935	0.919	0.922	0.011	30
Application Based	0.881–0.946	0.912	0.909	0.016	31
Cross-Chapter	0.337–0.530	0.469	0.515	0.078	9
In-Chapter	0.318–0.540	0.475	0.510	0.066	42

Table 2. Summary of Expert Score Dissimilarity Statistics for Question Categories

Label	Mean	Median	Mode
Out of Context	0.7	0.7	0.7
Application Based Questions	0.7	0.8	0.7
Cross-Chapter	0.2	0.2	0.2
In-Chapter	0.2	0.2	0.2

The comparison between GPT-derived Dissimilarity Scores (Table 1) and expert-assigned scores (Table 2) shows a general alignment in relative values, with GPT scores consistently trending higher across categories. This indicates that GPT’s model assesses questions as slightly more dissimilar compared to expert evaluations. For the Application-Based and Out-of-Context categories, both GPT and expert scores reflect higher mean values (0.9117 vs. 0.7581 for Application-Based, and 0.9194 vs. 0.7400 for Out of Context). This alignment suggests these types are perceived as more contextually detached from the core content, implying that they draw on broader or external knowledge applications. In contrast, Cross-Chapter and In-Chapter questions show lower mean dissimilarity scores in both methods. Notably, the expert-assigned scores are significantly lower (0.4691 vs. 0.2556 for Cross-Chapter and 0.4752 vs. 0.2405 for In-Chapter), reflecting a stronger contextual alignment. This supports the expectation that questions within or closely related to a single chapter naturally exhibit higher similarity to the core material. This overall pattern reinforces the

validity of both scoring methods while highlighting GPT’s tendency to slightly amplify dissimilarity, especially for questions extending beyond direct content boundaries.

3.5 Real-Time Query Handling and Response Flow

During live online sessions, the question management system processes student queries in real time by embedding them and evaluating their dissimilarity against the FAQ knowledge base using the FAISS index. Queries are ranked in descending order of relevance based on their dissimilarity scores, as shown in Table 3. High-dissimilarity queries, which fall outside the FAQ content, are flagged for instructor review. These queries typically indicate unique or complex questions requiring specialized input beyond the automated system’s scope. In contrast, low-dissimilarity queries, which align closely with the FAQ knowledge base, are autonomously managed by the RAG model. The RAG model synthesizes contextually relevant responses using pre-existing FAQ clusters, enabling efficient real-time handling of routine questions and reducing instructor workload. To enhance transparency, the system provides faculty with a list of the top N most relevant queries, including their cosine similarity scores, allowing instructors to

Table 3. Sample questions organized by dissimilarity scores. Each question is labeled based on its alignment category, with dissimilarity scores indicating the degree of contextual relevance to the course material.

Chapter Name	Question	Label	DisScore
Chap. 4: Digital Transmission	If designing a video streaming platform, how would you approach bandwidth allocation?	Application Based questions	0.946
Chap. 5: Analog Transmission	How would you implement QAM in a high-speed internet environment?	Application Based questions	0.941
Chap. 5: Analog Transmission	How could AM be leveraged in emergency broadcasting scenarios?	Out of context	0.935
Chap. 6: Bandwidth Utilization	For a telemedicine application with varying bandwidth requirements, how would you manage data?	Out of context	0.934
Digital Transmission	How does Manchester encoding provide synchronization in data communication?	In-Chapter	0.540
Digital Transmission	What are the two primary transmission modes in digital communication?	In-Chapter	0.538
Digital Transmission	What are the challenges of long-distance digital communication?	Cross-Chapter	0.530
Bandwidth Utilization: Multiplexing and Spreading	What are the advantages of Code Division Multiplexing?	Cross-Chapter	0.529

verify the alignment of generated responses with the FAQ repository. Currently tailored to provide accurate, context-specific answers from instructor-defined knowledge, the system aims to expand its capabilities by incorporating diverse content repositories and advanced context-switching mechanisms, enabling interdisciplinary query handling in the future.

RAG serves as a post-retrieval layer that enriches initial answer suggestions pulled from the clusters. This is crucial, as RAG can synthesize responses that are not just semantically similar but also contextually nuanced, leading to a more refined answer that fits both in-chapter and across-chapter contexts. Upon retrieval, RAG applies quality control metrics to refine responses, ensuring that generated answers meet critical relevance and precision standards. The Answer Relevancy Score (88.65%) prioritizes responses that directly align with the query, confirming that the answer is both accurate and useful. The Contextual Relevancy Score (42.5%) assesses the depth of alignment with broader textbook themes, ensuring that responses contribute meaningfully within a wider conceptual scope. Additionally, the Contextual Precision Score (88.25%) verifies that responses are finely tuned to chapter-specific content, which is essential for questions requiring tight content alignment. Finally, the Hallucination Score (74.6%) helps limit the inclusion of non-relevant or speculative information, maintaining strict adherence to the textbook’s factual content.

3.6 UI Demonstration and System Functionality

Teacher Dashboard		
RAG Addressing Questions		
Number of questions: 91		
Mark all as addressed		
Student Query	Disimilarity	Question Status
Analyze how baseline wandering can lead to errors in decoding and how this affects the overall integrity of transmitted analog signals.	0.26	☒
Analyze how understanding bandwidth in digital signals can inform decisions about bandwidth allocation in analog systems.	0.27	☐
Compare and contrast the efficiency of different modulation schemes (e.g., AM, FM, FSK) presented in Chapter 6 with regard to their application in analog transmission systems from Chapter 5.	0.29	☒
Compare the efficiency and reliability of digital data transmission versus analog data transmission.	0.24	☐
Define Phase Shift Keying (PSK) and discuss its advantages over ASK.	0.26	☒
Define time-division multiplexing and explain how it differs from frequency-division multiplexing.	0.08	☐
Describe how self-synchronization techniques discussed in Chapter 4 can enhance the reliability of analog transmission methods outlined in Chapter 6.	0.21	☒
Describe how the concept of carrier frequency is utilized in both analog transmission (Chapter 5) and modulation techniques (Chapter 6). Why is it critical for efficient communication?	0.21	☐
Describe how unipolar NRZ encoding operates and its advantages and disadvantages.	0.06	☒
Describe the Nyquist theorem and its relevance to sampling rates in analog-to-digital conversion.	0.11	☐
Describe the basic principle behind direct-sequence spread spectrum (DSSS).	0.06	☒
Teacher Addressing Questions		
Number of questions: 15		
Mark all as addressed		
Student Query	Disimilarity	Question Status
Compare built-in error detection mechanisms for digital signals with those used for analog signals, highlighting their importance in ensuring data integrity.	0.41	☐
Compare how different encoding and modulation strategies address noise interference, highlighting their effectiveness in maintaining signal integrity.	0.36	☐
Compare the error detection capabilities inherent in digital coding schemes with those used in analog modulation techniques.	0.38	☒
Define simplex, half-duplex, and full-duplex transmission modes with examples for each.	0.45	☐
Discuss the implications of using simplex communication in a networked environment.	0.49	☒
Explain how error detection methods used in digital transmissions (Chapter 4) can be adapted or applied to analog transmission systems covered in Chapter 6.	0.4	☐
How can built-in error detection mechanisms improve the reliability of analog transmissions as discussed across Chapters 5 and 6?	0.4	☒
How do different quantization levels affect the overall fidelity of a digital signal?	0.37	☐
How does half-duplex mode differ from full-duplex mode in terms of data transmission capabilities?	0.37	☒
How is the bandwidth calculated for a system using Amplitude Shift Keying with a given signal rate?	0.4	☐
In what scenarios would half-duplex transmission be preferred over full-duplex transmission?	0.4	☐
Provide an example calculation for determining the	0.41	☐

Fig. 2. Instructor Interface

To demonstrate the practical implications for both students and instructors, the proposed system features a streamlined user interface designed for intuitive

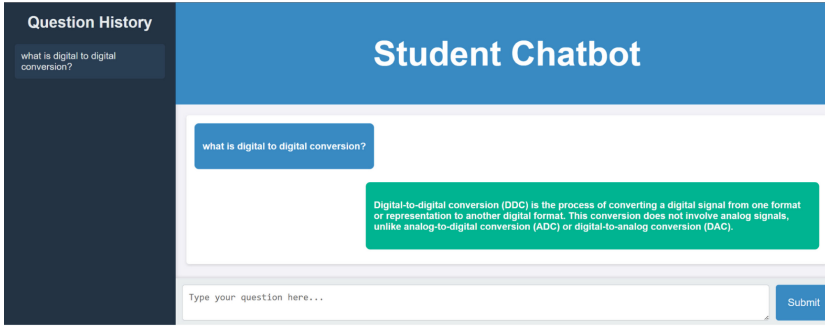


Fig. 3. Student Interface.

query management and real-time answer evaluation. The instructor dashboard (Fig. 2) provides an organized view of incoming queries, categorizing them as “True” or “False” based on their resolution status. For “True” questions, the dashboard highlights the proportion successfully resolved by the RAG model, while “False” questions are flagged for further instructor input. Additional metrics, such as the total number of questions and frequently flagged issues, support informed decision-making. The student interface (Fig. 3) enables students to submit queries during their online sessions. These queries are processed in real-time, with responses generated by the RAG model appearing in the “True” category of the instructor interface.

Preliminary tests of the operational demo across 108 student queries demonstrate an average response time of 4.6 s, with the system achieving 74.9% accuracy in RAG-based answers. The dissimilarity sorting mechanism ensures prioritization within 2 s, effectively meeting real-time performance expectations. The evaluation further reveals a hallucination rate of 17.5% (19 out of 108 queries), indicating that 83.5% of responses are free from hallucinations.

4 Conclusion

This study introduces a sophisticated AI-powered framework designed to manage real-time student queries in large-scale online classrooms, addressing critical scalability and responsiveness challenges that traditional educational models often struggle with. By leveraging cutting-edge semantic embedding techniques through the paraphrase-mpnet-base-v2 model, alongside FAISS indexing and optimized clustering, the system structures content into a high-dimensional vector space for efficient and precise query handling. The integration of a dual-resolution approach—automated handling for low-dissimilarity queries and instructor intervention for high-dissimilarity ones—provides an elegant solution to balance scalability with personalized engagement. This ensures that repetitive or contextually simple queries are resolved seamlessly while more complex or nuanced questions receive the necessary human expertise. The system’s curated

FAQ repository, organized into in-chapter, across-chapter, out-of-context, and application-based categories, further enhances its ability to cater to diverse levels of contextual relevance and alignment, enabling comprehensive and structured query resolution.

The system's efficacy is validated through several critical performance metrics, including an Answer Relevancy Score of 88.65%, a Contextual Precision Score of 88.25%, and a low hallucination rate of 17.5%. These metrics underscore the framework's ability to deliver accurate, contextually aligned responses while maintaining the speed and reliability required for real-time educational interactions. Additionally, its dynamic content update functionality allows instructors to seamlessly integrate revised course materials, ensuring the system remains adaptable to evolving educational content and requirements. This adaptability addresses a fundamental challenge in large-scale education: maintaining the currency and accuracy of instructional materials without interrupting ongoing processes. Beyond operational efficiency, the framework embodies a broader vision of democratizing education by enabling scalable, equitable access to high-quality instruction, particularly in settings where traditional student-teacher ratios are unsustainable.

Future development of this framework will focus on significantly expanding its capabilities to meet the evolving demands of global and interdisciplinary education. Multilingual support will be prioritized by incorporating models like mBERT and XLM-R, enabling the system to cater to diverse student populations and facilitate inclusivity across linguistic barriers. Addressing interdisciplinary queries that require complex reasoning and multi-step problem-solving will involve integrating advanced reasoning mechanisms, allowing the system to go beyond surface-level contextual alignment and engage with deeper analytical requirements. Real-time content update modules and adaptive learning algorithms will also be refined to ensure the system remains responsive to dynamic curricular changes, further enhancing its scalability and long-term viability. These advancements will not only strengthen the framework's core capabilities but also broaden its applicability to diverse educational contexts, fostering a more interactive, personalized, and adaptive learning environment for all stakeholders.

References

1. Amrita e-Learning Research Lab. A-VIEW: real-time collaborative multimedia e-learning. In: Proceedings of the 2011 ACM Multimedia Conference and Co-Located Workshops - ACM International Workshop on Multimedia Technologies for Distance Learning, MTDL'11 (2011)
2. Wang, Q., Woo, H.L.: Comparing asynchronous online discussions and face-to-face discussions in a classroom setting. *Brit. J. Educ. Technol.* (2007)
3. Brown, T., et al.: Language models are few-shot learners. *Adv. Neural. Inf. Process. Syst.* **33**, 1877–1901 (2020)
4. OpenAI. Gpt-4 technical report. arXiv preprint [arXiv:2303.08774](https://arxiv.org/abs/2303.08774) (2024)
5. Lewis, P., Oguz, B., Rinott, R., Riedel, S., Stenetorp, P.: Retrieval-augmented generation for knowledge-intensive nlp tasks. arXiv preprint [arXiv:2005.11401](https://arxiv.org/abs/2005.11401) (2020)

6. Heilman, M., Smith, N.A.: Good question! statistical ranking for question generation. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 609–617 (2010)
7. Luckin, R., Holmes, W.: Intelligence unleashed: an argument for AI in education (2016)
8. Drachsler, H., Kalz, M.: The mooc and learning analytics innovation cycle molac: a reflective summary of ongoing research and its challenges. *J. Comp. Assist. Learn.* (2016)
9. McInnes, B.T., Pedersen, T.: Evaluating measures of semantic similarity and relatedness to disambiguate terms in biomedical text. *J. Biomed. Inf.* (2013)
10. Aithal, S.G., Rao, A.B., Singh, S.: Automatic question-answer pairs generation and question similarity mechanism in question answering system. *Appl. Intell.* **51**(11), 8484–8497 (2021). <https://doi.org/10.1007/s10489-021-02348-9>
11. Al-Zahrani, A., Alasmari, T.: Exploring the impact of artificial intelligence on higher education: The dynamics of ethical, social, and educational implications. *Human. Soc. Sci. Commun.* (2024)
12. Forouzan, B.A.: Data Communications and Networking (McGraw-Hill Forouzan Networking). McGraw-Hill Higher Education (2007)
13. Reimers, N., Gurevych, I.: paraphrase-mpnet-base-v2 (2020)
14. Douze, M., et al.: The faiss library (2024)
15. Kale, S., Khaire, G., Patankar, J.: Language models are few-shot learners (2024)
16. Lan, F.: Research on text similarity measurement hybrid algorithm with term semantic information and tf-idf method. *Hindawi* (2022)
17. Gunawan, D., et al.: The implementation of cosine similarity to calculate text relevance between two documents (2018)



Enhancing Warehouse Operations with Voice-Activated SQL Streamlining Using LLM Technology

V. Ramani Bai¹(✉) , Suryajit Ithiparambil Sudheeran² ,
Abhinand K. Prasad¹ , and Abhinav Raj¹

¹ Adi Shankara Institute of Engineering and Technology, Kalady, Kerala, India
ramani.research@gmail.com

² Vidya Academy of Science and Technology, Thrissur, Kerala, India

Abstract. Inventory management of warehouses has traditionally been a tedious task. This paper proposes a customized inventory management software solution that leverages Large Language Models (LLMs) to be operated through natural language inputs, accepting both text and voice commands. This eliminates the need for dedicated database expertise, helping companies organize their databases in real-time upon the arrival or dispatch of goods, thus saving hours of manual work. The system ensures user roles and restrictions to prevent unauthorized access and provides a rollback mechanism for unintended database alterations. Our solution enhances inventory management with advanced voice activation powered by LLMs, robust security protocols, and real-time database updates, as demonstrated through various performance metrics and interface snapshots. With the integration of LLMs, accuracy improved from 62% to 85%, and the error rate decreased from 10% to 1%. While response time increased due to additional processing, the overall efficiency and security of the database management were significantly enhanced, providing a comprehensive and secure solution for managing inventories of all sizes.

Keywords: Inventory Management · Natural Language Processing (NLP) · Large Language Models (LLMs) · Automated Warehousing · Structured Query Language (SQL)

1 Introduction

Traditional inventory management involved manual stock counting by non-technical staff and updating records, a process simplified by database management systems [5]. Despite digitalization, querying databases for CRUD (Create, read, update and delete) operations still requires significant human effort and expertise [4].

With large language models, technology now converts natural language into SQL queries. However, each database's unique structure makes generalized text-to-query conversion ineffective [16]. This paper proposes a customized software

solution to perform CRUD operations via natural language inputs, minimizing the need for specialized expertise [15].

Leveraging NLP and machine learning, the system enhances non-technical user interaction with databases, using the Microsoft Speech SDK for accurate speech recognition and Microsoft Azure Web Services for robust text-to-speech conversion [8]. The system's Natural Language Understanding (NLU) component accurately interprets user queries and converts them into SQL commands using keyword analysis and vector embeddings. The Langchain framework and Google Palm LLM further improve SQL query generation from natural language inputs [14].

A user-friendly interface built with Streamlit integrates robust query validation mechanisms to ensure syntactically correct SQL queries that adhere to database constraints [18]. This enhances system reliability and aligns with best practices in database management.

The contributions of this paper are: i. Developing a natural language-driven inventory management system using Large Language Models (LLMs), ii. Implementing an advanced Natural Language Understanding (NLU) component for interpreting and converting queries into SQL commands, iii. Creating a user-friendly interface with robust query validation to ensure correct SQL queries.

In conclusion, the proposed Warehouse Database Voice Assistant revolutionizes database interaction for warehouse managers, offering an efficient, secure, and user-friendly solution for CRUD operations, reducing the need for specialized knowledge and improving operational efficiency.

2 Related Works

2.1 Natural Language Processing and Voice Assistants

Recent advancements in Natural Language Processing (NLP) have significantly enhanced voice assistant systems. NLP enables the extraction and comprehension of user intents from spoken language, facilitating natural human-computer interactions. Bastianelli et al. [2] demonstrated NLP's application in enhancing dialogue systems in industrial environments, improving task execution efficiency and user satisfaction.

2.2 Streamlit for User Interfaces

Streamlit is an open-source framework for developing interactive web applications in Python, ideal for creating user-friendly interfaces for AI and machine learning models. It integrates seamlessly with various Python libraries and has been highlighted in studies, such as Ramasamy et al. [12], for building interactive data science applications.

2.3 Azure Speech Services

Azure Speech Services provides tools for converting speech to text and text to speech, essential for developing voice-driven applications. Its capabilities in handling diverse accents and languages make it suitable for industrial applications. Li et al. [8] explored using Azure Speech Services in creating an interactive voice assistant for smart homes, demonstrating high accuracy and reliability.

2.4 LangChain and Google PaLM 2

LangChain and Google PaLM 2 (Pathways Language Model) represent advancements in language model architectures, facilitating complex query generation by understanding and processing natural language inputs with high accuracy. Google PaLM 2 has shown superior performance in generating coherent, contextually relevant responses in NLP tasks [14]. LangChain enhances this by linking multiple language models and tools to create sophisticated NLP applications.

2.5 Gemini for Query Validation

Gemini is a robust tool for query validation, ensuring generated queries are accurate and relevant. It uses advanced algorithms to validate queries against predefined criteria, enhancing system reliability. Zhang et al. [18] documented Gemini's effectiveness in improving the accuracy of database query systems.

2.6 Applications in Warehouse Management

Integrating these technologies creates a powerful AI-driven voice assistant for warehouse management. Prior research has demonstrated AI's benefits in logistics and supply chain management, such as Wamba et al. [17], highlighting AI's potential to revolutionize inventory management and order processing.

3 Proposed System

The proposed Warehouse Database Voice Assistant integrates advanced technologies in speech recognition, natural language understanding, and SQL query generation, utilizing large language models (LLMs) to provide an efficient, intuitive interface for warehouse management. This system simplifies querying, enhances real-time data access and decision-making, and improves data visualization through seamless integration with business intelligence tools. Compared to traditional database systems and data warehouses, it is user-friendly and accessible without deep SQL knowledge, can integrate with existing infrastructure, and scales to meet growing data demands. Additionally, robust security features address data privacy concerns, making it a comprehensive, valuable tool for businesses of all sizes. The use of LLMs significantly enhances its natural language processing capabilities, ensuring precise and effective communication.

3.1 Speech Recognition and Text-to-Speech

The system uses the Microsoft Speech SDK to recognize user voice commands, converting speech into text. It handles regional accents using large training datasets with various accents and dialects, and fine-tunes the model based on user speech through accent adaptation algorithms. Customizable models trained with regional data enhance accuracy, while real-time feedback allows the system to learn from user corrections and improve. Noise reduction and speech enhancement technologies ensure robust recognition in noisy environments. For text-to-speech conversion, Microsoft Azure Web Services generate synthetic speech waveforms from text input.

3.2 Natural Language Understanding

Natural language understanding (NLU) involves analyzing the text to extract intent and context. The proposed system uses keyword analysis and vector embeddings for semantic understanding. Keyword Analysis identifies the essential terms in a user's query to determine intent. The term frequency-inverse document frequency (TF-IDF) method is used to weigh the importance of keywords, as shown in Eq. 1:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \cdot \text{IDF}(t) \quad (1)$$

Where: $\text{TF}(t, d)$ is the term frequency of term t in document d , $\text{IDF}(t)$ is the inverse document frequency of term t , calculated as $\log\left(\frac{N}{\text{DF}(t)}\right)$, N is the total number of documents, $\text{DF}(t)$ is the number of documents containing term t .

The system utilizes Hugging Face embeddings to represent words or sentences in a multi-dimensional space. The embeddings capture semantic relationships between terms. The cosine similarity measure is used to compare embeddings, as shown in Eq. 2:

$$\text{cosine_similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} \quad (2)$$

Where: A and B are the vector embeddings of two terms, $A \cdot B$ is the dot product of the vectors, $\|A\|$ and $\|B\|$ are the magnitudes of the vectors.

3.3 SQL Query Generation

The SQL query generation system uses the Langchain framework and Google Palm LLM to convert natural language queries into SQL queries through few-shot learning techniques.

Few-shot learning improves the system's ability to generate SQL queries with limited training examples. The model is fine-tuned on a small dataset of natural language and SQL queries, as defined in Eq. 3:

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^N \mathcal{L}(f_{\theta}(x_i), y_i) \quad (3)$$

Where: θ are the model parameters, N is the number of training examples, $\mathcal{L}$ is the loss function, f_{θ} is the model with parameters θ , x_i is the input natural language query, y_i is the target SQL query.

The validation of generated SQL queries ensures syntactic correctness and adherence to database constraints. The validation process involves parsing the SQL query and checking against the schema constraints, as shown in Eq. 4:

$$Q_{\text{valid}} = \text{Parse}(Q_{\text{gen}}, S) \quad (4)$$

Where: Q_{valid} is the validated SQL query, Q_{gen} is the generated SQL query, S is the database schema.

4 Workflow

The proposed system's workflow includes stages for speech recognition, natural language understanding, SQL query generation, and validation. The following sections detail each component and their interactions, based on the diagrams in Figs. 1 and 2.

4.1 User Interaction and Speech Recognition

The user initiates interaction with the system through a voice command, saying "Hey Bob". The interface is built using Streamlit, which provides a user-friendly web interface for interaction.

The spoken command is captured by the Streamlit interface and sent to the Microsoft Azure Speech Services. The Azure Speech-to-Text service converts the spoken words into text.

$$\text{Speech}_{\text{input}} \rightarrow \text{Text}_{\text{output}}$$

4.2 Natural Language Understanding

The converted text is sent to the Gemini LLM (Large Language Model) for processing. The Gemini LLM is responsible for understanding the context and intent of the user's query.

$$\text{Text}_{\text{input}} \rightarrow \text{LLM}_{\text{Gemini}}$$

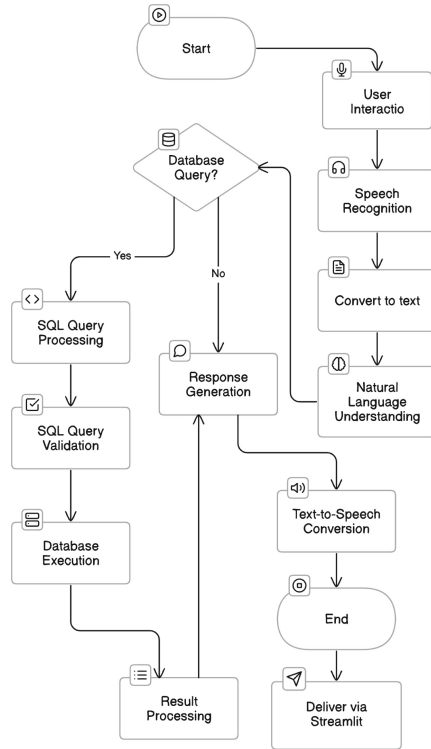


Fig. 1. Basic Flowchart of the system.

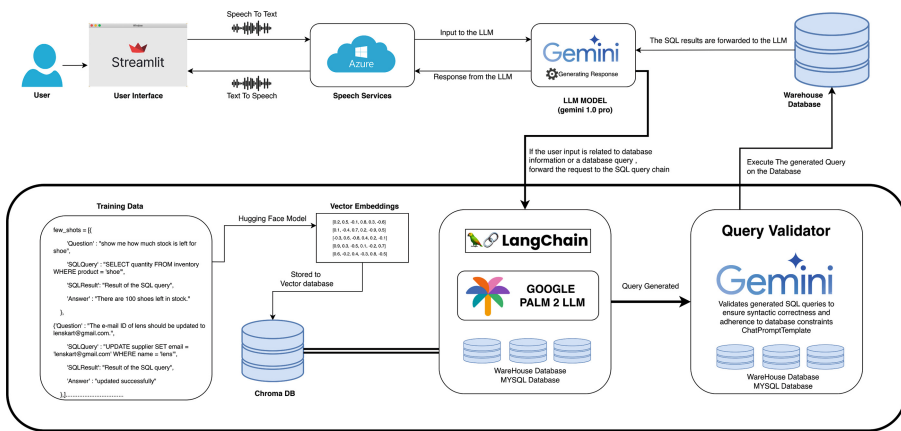


Fig. 2. System Architecture with the Workflow

4.3 Response Generation and Text-to-Speech

If the query does not involve a database interaction, Gemini LLM generates a text response based on the input. This response is then converted back to speech using Microsoft Azure's Text-to-Speech service.

$$\text{LLM}_{\text{Gemini}} \rightarrow \text{Text}_{\text{response}} \rightarrow \text{Speech}_{\text{output}}$$

4.4 SQL Query Processing

If the query is related to database information or involves a database operation, the request is forwarded to the SQL query chain system, which is built using LangChain and Google Palm 2 LLM.

4.5 Training Data and Vector Embeddings

Few-shot learning examples are used to improve the query understanding, as shown in Listing 1.1.

Listing 1.1. Some of the Few-shot examples used for Training

```
[
    { 'Question': "show me how much stock is left for shoe",
      'SQLQuery': "SELECT quantity FROM inventory WHERE",
        product = 'shoe' ",
      'SQLResult': "Result of the SQL query",
      'Answer': "There are 100 shoes left in stock." },

    { 'Question': "The e-mail ID of lens should be updated",
      to lenskart@gmail.com.",
      'SQLQuery': "UPDATE supplier SET email = ",
        lenskart@gmail.com' WHERE name = 'lens' ",
      'SQLResult': "Result of the SQL query",
      'Answer': "updated successfully" },

    { 'Question': "show me the orders table",
      'SQLQuery': "SELECT * FROM orders",
      'SQLResult': "Result of the SQL query",
      'Answer': "displaying the table" }
]
```

These examples are stored in Chroma DB and utilized by the system to learn the context and structure of SQL queries.

$$\text{Training Data} \rightarrow \text{Chroma DB}$$

The system utilizes Hugging Face models, specifically sentence-transformers/all-MiniLM-L6-v2, to generate vector embeddings for user queries, facilitating semantic understanding. These embeddings are stored in Chroma DB for efficient retrieval and comparison based on semantic similarity.

$$\text{Query}_{\text{input}} \rightarrow \text{Vector Embeddings} \rightarrow \text{Chroma DB}$$

A few-shot learning approach trains the system's machine learning models with a small number of examples, enabling it to understand and generate SQL queries from natural language efficiently. This includes data collection, pattern recognition, testing, and using vector embeddings to compare and retrieve relevant examples. Few-shot learning allows the system to quickly adapt to user queries with limited training data, making the training process faster and more efficient.

4.6 SQL Query Generation

The LangChain framework utilizes Google Palm 2 LLM to convert the natural language query into an SQL query. This process involves understanding the context and generating the appropriate SQL syntax.

$$\text{Vector Embeddings} \rightarrow \text{SQL Query}_{\text{generated}}$$

4.7 SQL Query Validation

The generated SQL query is validated using Gemini LLM. The validation ensures syntactic correctness and adherence to database constraints. The validation process involves parsing the SQL query and checking against the schema constraints.

$$\text{SQL Query}_{\text{generated}} \rightarrow \text{Query Validator} \rightarrow \text{SQL Query}_{\text{validated}}$$

4.8 Database Execution

The validated SQL query is executed against the warehouse database. The database could be any standard SQL database, such as MySQL.

$$\text{SQL Query}_{\text{validated}} \rightarrow \text{Database Execution}$$

4.9 Result Processing and User Feedback

The results from the database are forwarded back to Gemini LLM to generate a user-friendly response.

$$\text{Database Results} \rightarrow \text{LLM}_{\text{Gemini}}$$

Gemini LLM processes the database results and generates a textual response. This response is then converted to speech using Microsoft Azure’s Text-to-Speech service and delivered to the user via the Streamlit interface.

$$\text{Database Results} \rightarrow \text{Text}_{\text{response}} \rightarrow \text{Speech}_{\text{output}}$$

5 Algorithms

The algorithms developed for the proposed methodology are as follows: 1) Main Algorithm for the Warehouse Assistant System, which initiates the system and starts the interaction loop, invoking 2) Initialization to set up the environment, and 3) Speech Recognition to detect voice commands. The interaction loop progresses through 4) Process Query to analyze and forward user input, 5) Database Query Execution to generate and execute SQL queries, 6) Respond to convert text responses into speech, and 7) Error Handling to manage any issues during speech recognition and query execution. These components work together to provide seamless, voice-activated assistance for warehouse management tasks.

5.1 Main Algorithm for Warehouse Assistant System

Algorithm 1: Main Algorithm for Warehouse Assistant System

Input: Voice command (natural language query), predefined warehouse database tables: **supplier**, **inventory**, **orders**, **environment**
variables: API keys and database credentials
Output: Verbal or textual response, SQL query execution results
 Call Initialization();
 Call SpeechRecognitionLoop();

5.2 Initialization

Algorithm 2: Initialization of System Components

Initialization()
Load environment variables for database and API access;
Establish database connection;
Initialize Azure Speech SDK for voice processing;
Set up Google LLM for text generation with safety and generation configurations;
Initialize Langchain’s SQL Query Chain;

5.3 Speech Recognition Loop

Algorithm 3: Speech Recognition and Interaction Loop

```

SpeechRecognitionLoop()
while Listening for commands do
  if Trigger phrase (e.g., "Hey Bob") is detected then
    Activate the voice assistant and prompt the user;
    while Session is active do
      Continue listening for further input;
      if Valid input is received then
        Pass input to query processor;
  
```

5.4 Process Query

Algorithm 4: Process Query from User Input

```

ProcessQuery(Input Speech Text)
if Input indicates a database request then
  Forward request to database query handler;
else
  Inform user request is unsupported;

```

5.5 Database Query Execution

Algorithm 5: Database Query Execution and Handling

```

DatabaseQueryExecution(Input Speech Text)
Identify keywords to determine if query operation is required;
if Keywords indicate a database query then
  Generate SQL query using Langchain;
  if Query is a modification command (e.g., INSERT, UPDATE, DELETE) then
    Prompt user for confirmation to save or discard changes;
  else
    Execute retrieval queries and display results;
    Generate and provide a verbal summary;
else
  Inform user query could not be processed;

```

5.6 Respond

Algorithm 6: Respond to User with Text-to-Speech

```
Respond(Generated Text)
  Convert text response to speech;
  Play audio response for the user;
```

5.7 Error Handling

Algorithm 7: Error Handling in Speech Recognition and SQL Execution

```
ErrorHandling()
  if Speech recognition fails then
    Provide error message to the user;
  if SQL query execution fails then
    Inform user of an unexpected error and request retry;
```

6 Implementation

6.1 Computational Resources

Deploying the Warehouse Database Voice Assistant requires significant computational resources. The system uses Microsoft Azure for scalable infrastructure and Google's cloud for PaLM 2 LLM. NLP and machine learning tasks may use GPU-accelerated instances on Azure. Azure Blob Storage manages data such as user inputs, query logs, and training datasets. Real-time capabilities require low-latency processing with high-speed CPUs and sufficient memory. For smaller warehouses, the system leverages cloud-based solutions to reduce on-premises hardware needs. The modular design allows scalability and flexibility to manage costs, with Google's infrastructure handling complex language processing and other components using Microsoft Azure.

6.2 Connection and Authorization

To use the software, connect the database by providing Host name, Username, Password, Database name, and Port number. Once connected, the LLM can modify tables and understand the database schema for generating accurate SQL queries.

Database security is ensured with a toggle button for authorized personnel to perform data manipulation queries (INSERT, DELETE, UPDATE, ALTER, DROP, etc.). When the toggle is off, users can only execute data access commands like SELECT. Selection statements are always executable, but DML commands require the toggle to be on.

6.3 Querying

Users can query the database in natural language (English). Examples:

1. Retrieve item quantity

“Give me the number of item_name left in table_name.”

SQL: SELECT item_qty FROM table_name WHERE item_name = item_name;

2. Update item quantity (authorized manipulators only)

“Update the number of item_name to qty in table_name.”

SQL: UPDATE table_name SET item_qty = qty WHERE item_name = item_name;

Users can execute all existing SQL commands using natural language queries.

6.4 Commits

For data manipulation statements, the system ensures database security by confirming changes with the user. After executing a DML command, the voice assistant asks if the user wants to save the changes. If the user responds “Yes” or “Yeah,” the changes are committed. Otherwise, if the response is “No” or anything but a confirmation, the database rolls back to its pre-execution state.

6.5 Test Query Examples

1. Retrieving Item Quantity

“Give me the number of shoes left in inventory.”

SQL: SELECT item_qty FROM inventory WHERE item_name = ‘shoes’;

Answer: *“There are 150 shoes left in the inventory.”*

2. Updating Item Quantity

“Update the number of shirts to 200 in inventory.”

SQL: UPDATE inventory SET item_qty = 200 WHERE item_name = ‘shirts’;

Answer: *“The number of shirts has been updated to 200 in the inventory.”*

3. Deleting an Item

“Remove the old model of headphones from inventory.”

SQL: DELETE FROM inventory WHERE item_name = ‘headphones’ AND model = ‘old’;

Answer: *“The old model of headphones has been removed from the inventory.”*

All query executions provide both audio and visual outputs. For a SELECT statement, the queried information is displayed and read aloud. An INSERT statement updates the table on the screen with an accompanying audio description. Commit confirmations notify users audibly whether changes were saved.

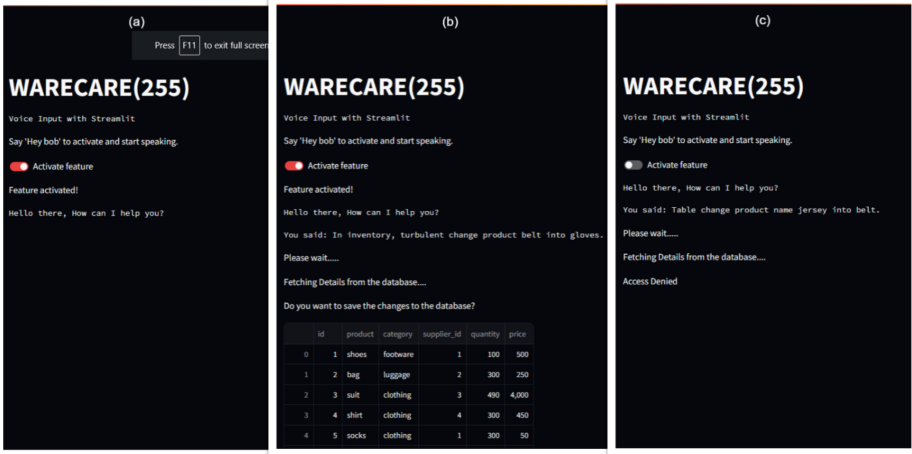


Fig. 3. Warehouse Database Voice Assistant Interface Demonstrations: (a) Activation with greeting prompt, (b) Inventory details fetched upon voice command, and (c) Database access denied for an update request

7 Results

The interactions with the Warehouse Database Voice Assistant interface are shown in Fig. 3.

Each figure underscores the assistant’s advanced capabilities and high technical standards. Figure 3(a) shows the interface activating a feature with the message “Feature activated!” and a prompt asking, “Hello there, how can I help you?”. Figure 3(b) demonstrates robust access control mechanisms, showcasing a user command being processed and resulting in a table with product details displayed. Figure 3(c) highlights the system’s proficiency in handling queries, displaying an “Access Denied” message in response to an unauthorized command. These figures collectively exemplify the voice assistant’s commitment to performance excellence, operational efficiency, and robust security in warehouse database management.

The performance of the Warehouse Database Voice Assistant is evaluated based on Accuracy, Response Time, Error Rate, and Training Data Size.

As depicted in Fig. 4 and Table 1, the Accuracy of query execution improved from 62% to 85%, showing a 37.1% increase after enhancements. However, Response Time increased from 2.0 to 6.0s due to additional natural language processing (NLP) and query validation, reflecting a –200.0% change. The Error Rate decreased significantly from 10% to 1%, demonstrating a 90.0% improvement in correct query executions. Additionally, the Training Data Size increased from 10 to 50 samples, a 400.0% increase, allowing the system to handle a wider range of queries.

Table 1. Performance Analysis of the Warehouse Database Voice Assistant

Metric	Description	Measure	Initial Value	Final Value	Improvement
Accuracy	Correctness of query execution	Percentage (%)	62	85	37.1%
Response Time	Time taken to respond to user queries	Seconds (s)	2.0	6.0	-200.0%
Error Rate	Frequency of incorrect query execution	Percentage (%)	10	1	90.0%
Training Data Size	Size of the training dataset	Number of Training Samples	10	50	400.0%

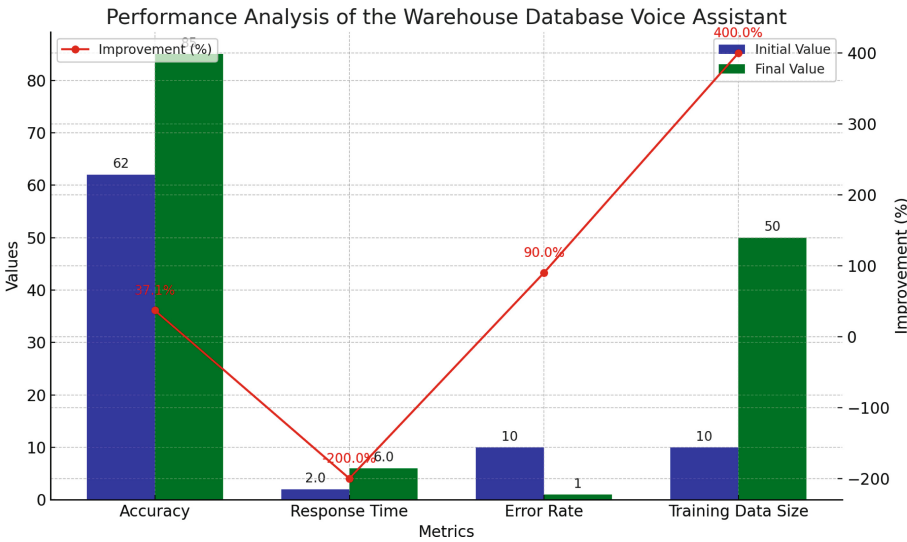


Fig. 4. Performance Analysis of the System

8 Conclusion

The Warehouse Database Voice Assistant represents a significant advancement in database management and inventory organization. By leveraging large language models and customized fine-tuning, it enables non-technical workers to handle complex tasks with minimal training. Implementing a natural language-based querying system will save time and reduce costs by automating processes while maintaining data safety, security, privacy, and integrity. Additionally, this technology can be adapted to various databases beyond conventional warehouse models by customizing the fine-tuning to meet specific client needs. Overall, the

Warehouse Database Voice Assistant provides an interactive, trustworthy, and secure way to access databases with ease.

9 Future Works

Future development of the Warehouse Database Voice Assistant includes enhancements to improve functionality, efficiency, and adaptability. Enhanced multilingual support will increase accessibility, allowing staff to interact in their native languages. Advanced AI and machine learning will improve NLP accuracy and efficiency, making the system more intuitive. Mobile and wearable applications will increase flexibility for managing inventory tasks on the go.

Integrating IoT devices like smart sensors and RFID tags will automate real-time inventory tracking, providing valuable insights into inventory conditions. Predictive analytics will help forecast inventory needs based on historical data and trends. Customizable user interfaces will cater to specific user needs, enhancing usability and satisfaction. Ensuring scalability for larger operations will maintain performance and reliability. Real-time collaboration tools will improve team coordination and productivity.

Limitations include effectiveness in noisy environments, recognition challenges with regional accents, latency issues, reliance on internet connectivity, and managing large datasets. Future work will focus on enhancing noise-cancellation technologies, expanding training datasets, optimizing computational efficiency, developing offline capabilities, and implementing scalable database solutions.

By addressing these areas, the Warehouse Database Voice Assistant can become more robust, reliable, and efficient, meeting the evolving needs of modern warehouses.

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

References

1. Bai, T., Ge, Y., Guo, S., Zhang, Z., Gong, L.: Enhanced natural language interface for web-based information retrieval. *IEEE Access* **9**, 4233–4241 (2020)
2. Bastianelli, E., Bartoli, P., Nardi, D., Giacomelli, A., Manes, F.: A spoken dialogue system for industrial robots. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, pp. 3650–3656 (2017)
3. Brown, T.B.: Language models are few-shot learners. *arXiv preprint [arXiv:2005.14165](https://arxiv.org/abs/2005.14165)* (2020)
4. Brunner, U., Stockinger, K.: ValueNet: a natural language-to-SQL system that learns from database information. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE), pp. 2177–2182. IEEE (2021)
5. Deepshikha, K.D.: A survey of natural language query builder interface to database. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **5**(4) (2015)
6. Gan, Y., et al.: Natural SQL: making SQL easier to infer from natural language specifications. *arXiv preprint [arXiv:2109.05153](https://arxiv.org/abs/2109.05153)* (2021)

7. Huang, B.B., Zhang, G., Sheu, P.C.: A natural language database interface based on a probabilistic context free grammar. In: IEEE International workshop on Semantic Computing and Systems. pp. 155–162. IEEE (2008)
8. Iliev, Y., Ilieva, G.: A framework for smart home system with voice control using NLP methods. *Electronics* **12**(1), 116 (2022)
9. Kate, A., Kamble, S., Bodkhe, A., Joshi, M.: Conversion of natural language query to SQL query. In: 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 488–491. IEEE (2018)
10. Kumar, A., Nagarkar, P., Nalhe, P., Vijayakumar, S.: Deep learning driven natural languages text to SQL query conversion: a survey. arXiv preprint [arXiv:2208.04415](https://arxiv.org/abs/2208.04415) (2022)
11. Qin, B., et al.: A survey on text-to-SQL parsing: concepts, methods, and future directions. arXiv preprint [arXiv:2208.13629](https://arxiv.org/abs/2208.13629) (2022)
12. Ramasamy, M., Natarajan, V., Mahalingam, R.: Developing interactive data science applications with streamlit. In: 2021 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Madurai, India, pp. 1–5 (2021)
13. Singh, G., Solanki, A.: An algorithm to transform natural language into SQL queries for relational databases. *Selforganizology* **3**(3), 100–116 (2016)
14. Song, Y., et al.: Enhancing text-to-SQL translation for financial system design. In: Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice, pp. 252–262 (2024)
15. Sonnadara, T.C., Priyadarshana, Y.: A natural language understanding sequential model for generating queries with multiple SQL commands. In: International conference on WorldS4, pp. 129–137. Springer (2023)
16. Troy, C., Sturley, S., Alcaraz-Calero, J.M., Wang, Q.: Enabling generative AI to produce SQL statements: a framework for the auto-generation of knowledge based on EBNF context-free grammars. *IEEE Access* **11**, 123543–123564 (2023)
17. Wamba, S.F., Gunasekaran, A., Akter, S., Ren, S.J.f., Dubey, R., Childe, S.J.: Big Data analytics and firm performance: effects of dynamic capabilities. *J. Bus. Res.* **70**, 356–365 (2017)
18. Zhang, Y., Liu, J., Zhu, M.: Query validation techniques for database systems. *J. Database Manage.* **31**(4), 1–16 (2020)

AI in Healthcare and Medical Imaging



Enhancing Medicinal Plant Prediction Through Deep Neural Network Algorithms

M. Swapna¹ , C. Sireesha² , Sanjana Gavada¹ , and Barigela Sreeja¹ 

¹ Stanley College of Engineering and Technology for Women, Hyderabad, India
mswapna@stanley.edu.in

² Vaasavi College of Engineering and Technology, Hyderabad, India

Abstract. New developments in deep learning have highly improved the way of image classification and helped areas like medical botany. In this study, identification of medicinal plants from leaf images was assessed using three CNN structures including ResNet50, Dense Net, and Alex Net. Evaluations were done using accuracy, precision, recall and F1-measure. The highest accuracy of 98% was obtained by using ResNet50 for feature acquisition with dense acquisition as the closest follower, with an accuracy of 96%. While being rather fast to process input data, Alex Net was a bit delayed because of its less complicated structure. The exercitation of CNNs showed the capacity to enhance the recognition of medicinal plant and benefited the further progress of natural medicine and reasonable use of resource.

Keywords: Deep learning · image classification · convolutional neural networks

1 Introduction

Natural, Herbal medicine has played a major part in different traditional healing systems across the globe and serves to repair and prevent or slow disease. These plants consist of different types and are used in cure of diseases and in enhancing research in the field of pharmacology. As the size of the population increases Coulture and Stocking demands increase therefore it becomes very important to identify and classify medicinal plants properly for their proper utilization and conservation. In particular, deep learning algorithms displayed in Fig. 1 appear to provide solutions for plant leaf-based identification.

Some of the challenges include; the large number of plant species, and close resemblance of the shapes of the leaves of different plant species reduce research in the medicinal plant research. When used in the cure or treatment of diseases, it is important to determine the correct name of the plants or where they belong. Standard methods of identification of plants are cumbersome and have a tendency to offer wrong identification yet they require taxonomic experience and physical examination. This challenge can be handled through the use of current advancements in artificial intelligence like deep learning and computer vision which performs useful work of analyzing images of plants to offer enhanced classification.



Fig. 1. Classification of Medicinal plants

In this regard, it has been appropriate to identify best-fit application of remarkable CNN architectures such as ResNet50, Dense Net and Alex Net [9]. These models are famous for identifying the fine features and attributes of images and therefore perfect for plant categorization. By using such complex models, the general goal of this research is to increase the accuracy of detecting medicinal plant species. In line with the conclusion of the comparative analysis, general recommendations for future studies on botany, a potential improvement, and modification are also highlighted.

The remainder of this paper is organized as follows: Sect. 2 provides an overview of the related work, and presents a literature survey on classification of medicinal plants based on deep learning. Section 3 describes the approach, where the type of data, data preparation, as well as the used CNN structures is introduced. Section 4 describes the actual implementation chosen in the experiments: the used hardware and software environments, and the characteristics of the chosen evaluation criteria. Finally, in Sect. 5, the authors discuss the experimental results that involved ResNet50, Dense Net and Alex Net. Section 6 discusses on the implication of the study findings, limitations as well as suggestions for further research.

2 Related Works

Section 3 will briefly describe the conceptual framework that will be followed in the study and key research objectives intended to enable the use of different deep-learning models to classify the leaves of the medicinal plants in question (Table 1).

3 Framework and Research Objectives

One of the proposed solutions entails the use of a second machine learning technique on a dataset of the images of different medicinal plants with a view of minimizing variations of characteristics within a particular species. Interpretations were made using ResNet50, DenseNet-121, and AlexNet in order to favor relations concerning accuracy and other characteristics. The procedures for data collection, data pre-processing, model training, model assessment, and comparison of deep learning algorithms for classification of medicinal plant leaves are also presented in Sect. 4.

Table 1. Literature survey on Medicinal Plant Classification Technique.

S.no	Technique	Merits	Demerits	Ref no.
1	SVM (Support Vector Machine), KNN (K-Nearest Neighbour), Combination of Shape and Gabor Wavelet Features, Quadratic SVM Classifier	Achieves 99.9% accuracy for identifying herbal plants from microscopic powder images using a combination of shape and Gabor wavelet features with quadratic SVM classifier	Relies on a specific dataset and microscope setup, limiting generalizability	[3]
2	Gabor Filter, Gray Level Co-occurrence Matrices (GLCM), Random Forest (RF), Light Gradient Boosting Machine	Achieves 95.5% accuracy in medicinal plant identification using advanced image processing and machine learning techniques	Requires significant computational resources, limiting use in resource-constrained setups	[4]
3	Deep learning, computer vision-based classification, CNN	Utilizes deep learning and CNNs to improve accuracy in computer vision-based classification of medicinal plants	face challenges in precise classification due to limitations in current algorithms	[5]
4	This paper includes Computer vision, Pattern recognition, Image processing algorithms	The paper proposes an efficient system for medicinal herb identification that reduces human error by utilizing computer vision, pattern recognition, and image processing algorithms	The system may struggle to distinguish similar-looking herbs, reducing overall reliability	[6]
5	ResNet-18, AlexNet, VGG-11, SqueezeNet, GoogLeNet, ShuffleNet, ResNeXt-50, DenseNet	High classification accuracy for herbal plant leaf images. Data augmentation improves model performance. Achieves higher accuracy than previous classical ML methods	Limited to 25 categories and relies solely on image data, offering only marginal improvements	[9]

4 Proposed Methodology: Ayurvedic Bot

The proposed work in the present research study is to recognize medicinal plants effectively with the help of advanced deep CNN. The proposed methodology involves analyzing three CNN models: From the discussion above, ResNet50, DenseNet-121, and AlexNet will be chosen to learn the best model for classifying medicinal plants. Firstly, data on medicinal plants and images of their leaves were gathered and cleaned to provide data of good quality. The proposed architecture is described in the following figure, Fig. 2.

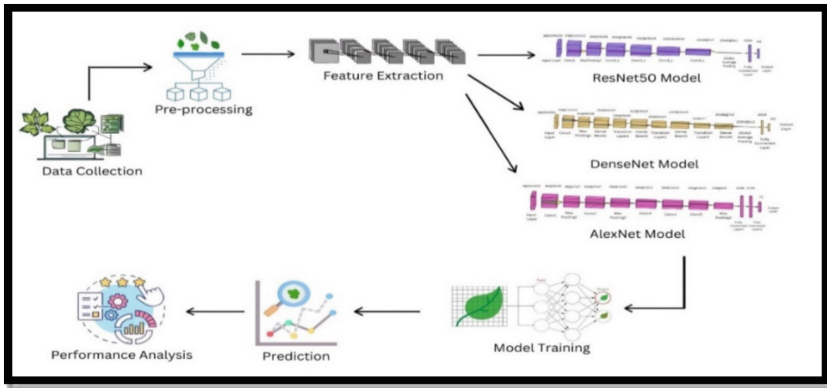


Fig. 2. Proposed methodology

Figure 3 outlines the methodology of existing models, consisting of five key steps: The details of the process in machine learning are data Collection & procurement, data cleaning & precursor, model Construction, prediction & model evaluation. However, it omits a crucial phase: feature extraction. This paper presents this step which is crucial in an IDS. Feature extraction is part of the overall methodology before actually building the predictive model. It means converting the raw data through multiple layers of models to the features which are helpful for discrimination and forecasting. This intermediate step is important since to input raw form of data into the model is time consuming and ineffective when it comes to perceiving other aspects that enhance the model's performance.

4.1 Proposed Design of Ayurvedic Bot

As widely discussed in recent research publications, the ResNet50 takes the best positions in the first evaluations; thus, it forms the basis of the presented Ayurvedic Bot system. The proposed framework follows a structured methodology comprising five phases: cleaning and transformation of the data, building and fitting of the model and selection of the features, categorization and ranking and finally testing of the model. The architecture of ResNet50 model is shown below in Fig. 4 (Figs. 5 and 6)

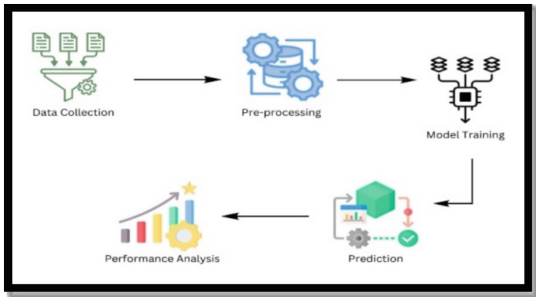


Fig. 3. Existing methodology

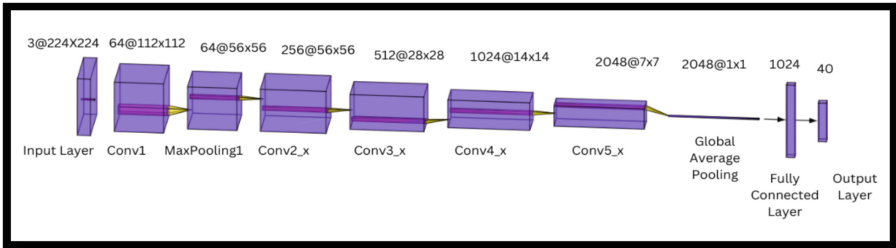


Fig. 4. Resnet50 Architecture

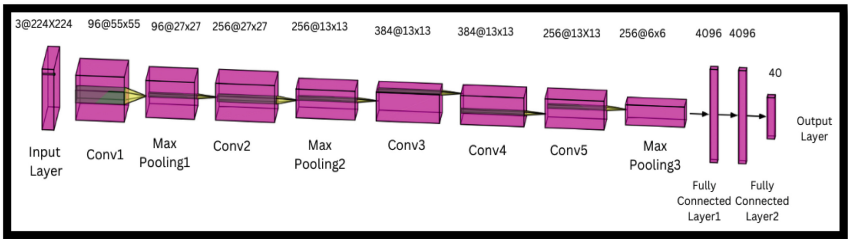


Fig. 5. Dense Net-121 Architecture

4.1.1 Preprocessing Function

The preprocessing function was developing as an enhancement of the image dataset where the image was smoothened normalized and de noised. Pixel values of the images were scaled, the dimensions of the images were reduced and data was augmented in an attempt to enhance model performance. These were measures adopted with a view of minimizing scenarios that might overcharge the classifier for too many cases and lead to reduced classification efficiency. The Medicinal Plant dataset illustrated in Fig. 7 and which contains 5955 medicinal plant leaves images belong to 40 classes. All images were adjusted to have uniform size of 600 by 450 pixels for the purpose of analysis. The dataset was split into two subsets: Training data – 4773 images and Testing data – 1182

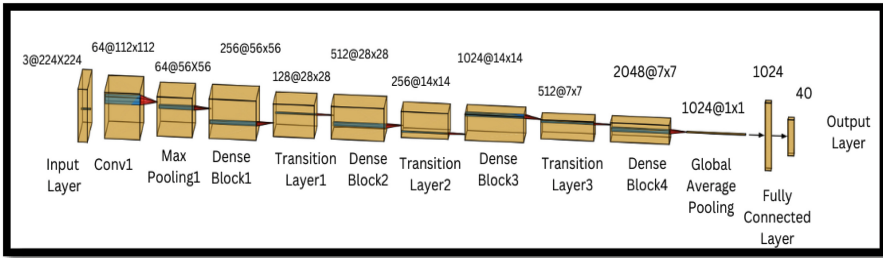


Fig. 6. Alex Net Architecture

images. Class labels denote different Medicinal plant species and care is taken to ensure that samples belonging to any class are adequately represented. Class imbalance issues were identified and addressed in this study by using data augmentation techniques with minor variations in class ratio. To enhance model generalization data normalization, resizing and other augmentation parameters (rotation, flip, crop and geometric transformations) were incorporated during training. These augmentations made the models more resistant to qualitative transformations such as changes to orientation, scale and lighting conditions. The researchers used only the authentic ecological and medicinal longitude to the data to make sure it was accurate.

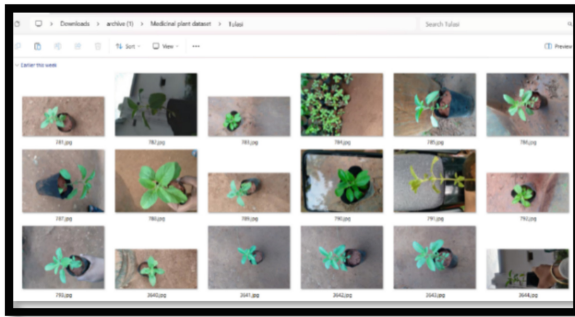


Fig. 7. Dataset Image

4.1.2 Model Training

In the training stage ResNet50, Dense Net, and Alex Net will be trained with assistance of the preprocessed dataset. The presented models are trained in one part of the dataset, while the rest of the data set is used as the validation set. Multiple iterations through the training process are used, and optimization techniques, such as stochastic gradient descent (SGD) are used to reduce the loss function. A pie chart has been used to present the proportion of the data split into the training and the testing dataset.

4.1.3 Feature Extraction

Feature extraction is a critical process where machine models select important aspects of medicinal plants to facilitate accurate categorization. Due to its depth and use of residual connections, ResNet50 can learn finer details for classifying leaf images. The residual connections enable the network to capture essential features like texture and vein patterns, which are key in differentiating closely related plant species. This hierarchical feature extraction process ensures that both basic features, such as contours, and abstract features, like shape and texture changes, are captured. DenseNet-121 employs densely connected layers, allowing features to flow through the network efficiently. This architecture ensures that each layer receives features from all preceding layers, aiding in the capture of subtle variations in plant structures. DenseNet-121 extracts critical features, such as vein patterns, surface texture, and shading, which are essential in distinguishing similar medicinal plants. Its design with fewer parameters ensures effective feature learning without compromising performance. Alex Net, an earlier deep learning model, provides excellent feature extraction with its five convolutional layers and three fully connected layers. It accurately classifies simple features such as leaf shape, edges, and basic texture. Although less complex than ResNet50 and DenseNet-121, Alex Net excels at extracting descriptors like leaf margin outlines, serrations, and patterns, which serve as a baseline for comparison with more complex models.

4.1.4 Classification Module

The classification module examines plant samples extracted through CNN models to classify medicinal plants. After the ResNet50 system has classified ten images, then the final layer results are compared with the original labels to analyze the accuracy or precisions as well as or recall. For the purpose of Choosing the most appropriate model for MBS implementation, performance metrics are gathered. The system flow of the system is shown in Fig. 8.

4.1.5 Evaluations and Comparison

During the evaluation phase, performances of the ResNet50, Alex Net, and DenseNet-121 models in the identification of medicinal plants are compared. From the accurate values of precision, recall, and F1-score, ResNet50 was the most accurate model for feature extraction and classification. Alex Net was accurate but had a problem with the size of the datasets because of low recall; it had to be resized. DenseNet-121 was the least accurate because it overfits and hardly extracts the right features. These differences are illustrated by Figs. 10 and 11 respectively. Among all the models, ResNet50 was identified as the best model for the Ayurvedic Bot system. Section 5 precisely describes the algorithms employed and then contrasts and selects the finest model for the application.

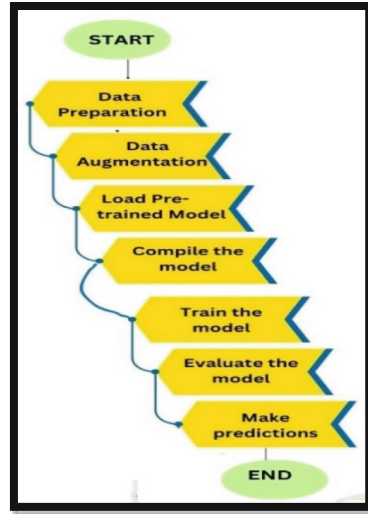


Fig. 8. Flow of model

5 Result and Discussion

The main goal of this study is to look over the difference between the three model's ResNet50, DenseNet-121 and Alex Net accuracy and their capability to classify the model appropriately. This study's experiments were performed on a system with the specifications of 11th Gen Intel(R) Core(TM) i5-1135G7 processor with 2.40 GHz speed, and 16 GB installed RAM out of which approximately 15.7 GB can be used. It runs under a 64-bit system, the x64-based processor. For this study the software stack consisted of Python (v3.x) as the programming language, and Deep Learning toolkits like Tensorflow&Pytorch for model development/ training. NumPy and Panda for data handling and related analysis. The experiments were performed in the Google Colab environment, which means that it is capable to use the GPU for model training, and saving time. All the results and plots presented in this paper were obtained through the Matplotlib and Seaborn libraries to avoid confusion and duplication (Fig. 9).

ResNet50

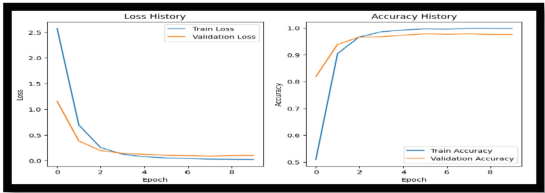


Fig. 9. Training and Validation Loss and Accuracy History for ResNet50 Model

DenseNet-121

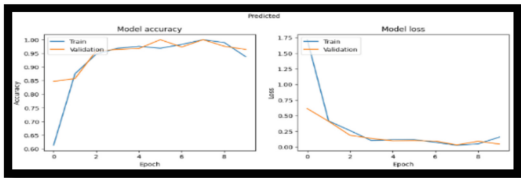


Fig. 10. Model Accuracy and Loss During Training and Validation

Alex Net

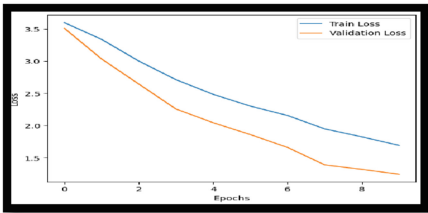


Fig. 11. Model Accuracy and Loss During Training and Validation

5.1 Comparative Analysis

To evaluate the robustness of the proposed models for medicinal plant classification, a comparative analysis was performed against several established models: ResNet50, Dense Net, and Alex Net. These models were chosen due to their proven effectiveness in image classification tasks.

5.2 Discussion

Large differences in outcomes were considered in cases where the images of medicinal plants were classified using the ResNet50, Alex Net and DenseNet-121 model. Out of the models tested ResNet50 provided a good accuracy of 0.978, precision of 0.9755, recall of 0.9756, and F1- score of 0.9737 making it ideal for applications that calls for fine distinction of features. Alex Net proved even less effective and accurate and thus less precise, recall, and F1-score, but it is not effective for the complex medicinal dataset. DenseNet-121 was the worst performing model; this can be attributed to high; overfitting and very deep network architecture. Therefore, ResNet50 was chosen for the Ayurvedic Bot system as it provides the best results for feature extraction of the object when tested with advanced features and the system is robust.

Section 6 gives the overall conclusion of the paper.

6 Conclusion

Lastly, the study has exhibited the efficiency of the ResNet50 in classifying images related to medicinal plants with the features of median precision, recall and accuracy. The comparative analysis shows that in this domain Comp(GPU) of ResNet50 is higher than the Comp(GPU) of both Alex Net and DenseNet-121. Although Alex Net demonstrates some level of performance, the results are lower than those of the other models implying that Alex Net is less efficient in handling the richness of the medicinal plant dataset. To the classification of this set of features, DenseNet-121, which has an advanced architecture, provides insufficient results. The outcomes render empirical evidence to ResNet50's capability in solving complex classification problems and explain why it was the most suitable for the Ayurvedic Bot system. More advanced work can concentrate on discovering other models or improved methods of the ensemble to accumulate more precise and stable classification. Moreover, further preprocessing of given data or using data augmentation could enrich the given dataset and may affect the performance of the other models.

References

1. Kadiwal, S.M., Hegde, V., Shrivathsa, N.V., Gowrishankar, S., Srinivasa, A.H., Veena, A.: A survey of different identification and classification methods for medicinal plants. In: Smys, S., Kamel, K.A., Palanisamy, R. (eds.) *Inventive Computation and Information Technologies*. LNNS, vol. 563. Springer, Singapore (2023). https://doi.org/10.1007/978-981-19-7402-1_20
2. Pushpanathan, K., Hanafi, M., Mashohor, S., Fazlillahi, W.F.: Machine learning in medicinal plants recognition: a review. *Artific. Intell. Rev.* **54** (2021). <https://doi.org/10.1007/s10462-020-09847-0>
3. Fataniya, B.M.: Microscopic Image Based Classification Algorithms for few Herbal Plants Identification using Machine Learning Approaches. <https://shodhganga.inflibnet.ac.in/handle/10603/246013?mode=full>
4. Das, A., Siva Sai Kumar, B., Shiva Shankar Reddy, S., Naveen Reddy, S., Peeyush, K.P.: Ayurvedic medicinal plant identification system using embedded image processing techniques. In: *Proceedings of Fourth International Conference on Communication, Computing and Electronics Systems*. Springer Nature Singapore. <https://www.springerprofessional.de/en/ayurvedic-medicinal-plant-identification-system-using-embedded-i/24624656>

5. Rao, M.S., Kumar, S.P., Rao, Konda S.: A methodology for identification of ayurvedic plant based on machine learning algorithms. University of Bahrain Scientific Journals. ISSN: 2210-142X. <https://journals.uob.edu.bh/handle/123456789/4926>
6. Preethi, M., Rani, S.J., Pradhiksha, K.S., Kumar, J.R., Vishal, T.: Medicinal herbs identification. In: 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 1050–1053. Coimbatore, India (2023). <https://doi.org/10.1109/ICAACS57279.2023.10112700>. <https://ieeexplore.ieee.org/document/10112700?denied=>
7. Chanyal, H., Yadav, R.K., Saini, D.K.J.: Classification of medicinal plants leaves using deep learning technique: a review. *Int. J. Intell. Syst. Appl. Eng.* **10**(4), 78–87 (2022). <https://ijisae.org/index.php/IJISAE/article/view/2199>
8. Azadnia, R., Al-Amidi, M.M., Mohammadi, H., Cifci, M.A., Daryab, A., Cavallo, E.: An AI based approach for medicinal plant identification using deep CNN based on global average pooling. *Agronomy* **12**, 2723 (2022). <https://doi.org/10.3390/agronomy12112723>
9. Kumar, G., Kumar, V.: Herbal plants leaf image classification using deep learning models based on augmentation approach (July 31, 2022). In: 4th International Conference on Communication & Information Processing (ICCIIP) (2022). <https://ssrn.com/abstract=4292344>
10. Shailendra, R., Anitha, J.: DeepHerb: a vision based system for medicinal plants using xception features. *IEEE Access* **1** (2021). <https://doi.org/10.1109/ACCESS.2021.3116207>
11. Chan, H.P., Samala, R.K., Hadjiiski, L.M., Zhou, C.: Deep learning in medical image analysis. *Adv. Exp. Med. Biol.* **1213**, 3–21 (2020). https://doi.org/10.1007/978-3-030-33128-3_1. PMID:32030660;PMCID:PMC7442218
12. Gokhale, A., Babar, S., Gawade, S., Jadhav, S.: Identification of medicinal plant using image processing and machine learning. In: Iyer, B., Rajurkar, A., Gudivada, V. (eds.) *Applied Computer Vision and Image Processing. Advances in Intelligent Systems and Computing*, vol. 1155. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-4029-5_27
13. Paulson, A., Ravishankar, S.: AI based indigenous medicinal plant identification. In: 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA), pp. 57–63. Cochin, India (2020). <https://doi.org/10.1109/ACCTHPA49271.2020.9213224>
14. Dileep, M.R., Pournami, P.: AyurLeaf: a deep learning approach for classification of medicinal plants (2019). <https://doi.org/10.1109/TENCON.2019.8929394>
15. Duong-Trung, N., Quach, L.-D., Nguyen, M.-H., Nguyen, C.-N.: A combination of transfer learning and deep learning for medicinal plant classification. In: *Proceedings of the 2019 4th International Conference on Intelligent Information Technology (ICIIT 2019)*, pp. 83–90. Association for Computing Machinery, New York, NY, USA (2019). <https://doi.org/10.1145/3321454.3321464>
16. Vo, A., Thien, B.: Vietnamese herbal plant recognition using deep convolutional features. *Int. J. Mach. Learn. Comput.* **9** (2019). <https://doi.org/10.18178/ijmlc.2019.9.3.811>
17. Tan, J.W., Chang, S.W., Abdul-Kareem, S., Yap, H.J., Yong, K.T.: Deep learning for plant species classification using leaf vein morphometric. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **17**(1), 82–90 (2020). <https://doi.org/10.1109/TCBB.2018.2848653>. Epub 2018 Jun 19. PMID: 29994129
18. Ibrahim, Z., Sabri, N., Mangshor, N.N.A.: Leaf recognition using texture features for herbal plant identification. *Indonesian J. Electric. Eng. Comput. Sci.* p-ISSN: 2502-4752, e-ISSN:2502-4760. <https://ijeecs.iaescore.com/index.php/IJECS/article/view/10486>
19. Vilasini, M., Ramamoorthy, P.: CNN approaches for classification of indian leaf species using smartphones. *Comput. Mater. Continua* **62.3** (2020). https://cdn.techscience.cn/uploads/attached/file/20200224/20200224073045_30140.pdf



A Comprehensive Review of Medical Image Captioning: Techniques, Challenges, and Future Directions

Mallemputi Sai Teja¹ , Chinta Vijaya Pandu Ranga Reddy¹ ,
U. Muthaiah¹ , and B. S. Vidhyasagar²

¹ Computer Science and Engineering, Amrita School of Computing,
Amrita Vishwa Vidyapeetham, Chennai, India
chvijayapandurangareddy@gmail.com

² Department of Computer Science and Engineering, Vel Tech Rangarajan
Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India

Abstract. Medical image captioning is an emerging field that combines natural language processing and computer vision for developing textual descriptions of medical images, with a focus on recent developments in deep learning models; particularly CNN, RNN and transformer-based architectures. This review aims to discuss state-of-the-art approaches and strategies applied in medical image captioning. The different datasets used in training and testing these models are also discussed by the paper, along with the problems and challenges that exist in this field. Through the exploration of the current trends, this paper attempts to give a comprehensive review of the development of medical image captioning, identify gaps in existing research, and propose new potential areas for further work. Medical image captioning improves diagnostic accuracy and automates routine reporting, reducing the work for the radiologists. Although progress, challenges continue to arise. One important problem is the lack of large, annotated medical datasets required for efficient model construction. Accurate comprehension and creation of medical terminology is crucial to avoid inaccurate captions and misdiagnosis. As the research in the field advances so that the future works should focus on extensive datasets and creating model that can be trained from limited data ensuring clinically accurate captions. Integrating multiple data like patient history patient history or lab reports can improve the accuracy and value of medical image captions, pushing the boundaries of automated findings. Using Artificial Intelligence in the medical field should consider some of the concerns like patient safety and privacy. The collaboration between AI research and medical professionals is essential for developing and accurate medical image caption generator.

Keywords: Medical Image Captioning · Deep Learning Models · Computer Vision · Natural Language Processing · Transformers · Encoder · Decoder

1 Introduction

Medical image captioning involves the automated generation of descriptive text based on visual content from medical images like X-rays, CT scans, or MRIs [3]. This technology is highly important as it allows healthcare professionals to interpret imaging results rapidly and accurately [7]. Automating this process enables radiologists to focus on critical cases rather than each and every patient [7]. Deep learning techniques, including CNN and RNN, form the underlying architecture of these systems. CNN is designed for identification of a pattern in images, while RNN generates descriptive captions about them [4].

Advances in artificial intelligence have improved the ability of medical image captioning systems to detect abnormalities and place them in context within a patient's health history. These systems do not only support diagnostics but also streamline clinical workflows, enhance electronic health records, and help communicate across disciplines to healthcare teams [8]. The integration of natural language processing techniques has greatly improved the quality and fluency of captions in a medical image captioning system. NLP ensures that captions are not only grammatically correct but also contextually relevant to the clinical scenarios cited in, among others [5]. This powerful language understanding would close the gap between analyzing visual data and the generation of text, making such systems more effective in a real-world healthcare setting.

Despite progress, challenges persist in medical image captioning. Medical images are complex, and their variability complicates the development of generalizable systems capable of working across all imaging modalities and clinical contexts [7]. Biases in model performance and dataset limitations hinder the applicability of these systems in real-world settings. Ensuring diverse and high-quality annotated datasets is essential to overcome these barriers [9]. Ethical considerations, including data privacy, patient consent, and trust in automated captions for critical diagnostic information, further highlight the need for explainable and reliable models. Collaboration between engineers, medical professionals, and researchers is vital to advancing this field. Through continuous improvements in dataset organization and algorithmic approaches, medical image captioning holds the potential to transform diagnostic processes and improve patient care [7].

This paper reviews advancements in medical image captioning, with a focus on deep learning approaches, particularly CNN-LSTM architectures. It explores their efficacy in generating diagnostic captions, discusses challenges like data heterogeneity and biases, and compares the performance of state-of-the-art models across diverse datasets.

The paper is structured as follows: Sect. 2 discusses related work, Sect. 3 provides a detailed analysis of datasets, Sect. 4 evaluation metrics, Section 5 challenges and limitations, Sect. 6 Methodologies, and Sect. 7 addresses future directions, concluding conclusion.

2 Literature Survey

Hyeryun Park et al. [11] proposed models to enhance the accuracy of medical image captions. The mDiNAP-Transformer model, using a transformer decoder, was tested on the IU X-RAY dataset and outperformed hierarchical LSTM models. Key challenges included data imbalances and ensuring clinical accuracy. The models were evaluated using BLEU, METEOR, ROUGE-L, and CIDEr scores.

Authors proposed a transformer-based model integrating contrast-based image enhancement for improved radiology report generation [12]. Among four contrast-based methods, gamma correction proved most effective. Visual features were extracted using fine-tuned ChexNet (DenseNet-121), and textual embeddings were generated using BERT to incorporate contextual information. Tested on the Indiana University X-ray dataset, the model showed significant improvements over baselines due to enhanced visual features and multi-head attention. Challenges included handling complex medical terminologies and ensuring accurate alignment between text and findings. BLEU scores with 4 n-gram evaluations highlighted the model's advancements in generating accurate and fluent reports.

The authors [13] proposed hybrid neural network models integrating CNNs for visual feature extraction and RNNs for text generation, enhancing the quality of radiology reports. Enhanced attention mechanisms highlighted critical image regions, ensuring the capture of relevant information. The models were trained and evaluated on the MIMIC-CXR dataset. These hybrid models outperformed others in generating detailed and clinically accurate reports. However, challenges included ensuring clinical relevance, addressing data imbalance, and improving generalization to diverse findings. Performance was evaluated using BLEU, METEOR, ROUGE-L, and CIDEr scores, standard metrics for image captioning tasks.

Zhanyu Wang et al. [14] proposed a transformer-based model to address challenges in medical image captioning, such as the similarity between images and the overshadowing of disease-related terms by common content. The framework tackled issues like visual-textual alignment and multi-label classification and incorporated a term-weighting scheme to emphasize specific words during training. The authors suggested that integrating knowledge graphs into the transformer framework could improve medical report generation by better modeling relationships between medical terms. The proposed method outperformed existing models on benchmark datasets, including the MIMIC-CXR dataset.

The authors proposed an encoder-decoder architecture characterized by a triple-branch network in which the authors [15] proposed the use of a framework. Among them is the Visual Attention Branch that captures visual embeddings from the medical images using a soft-attention mechanism to focus on the relevant features. Then, MeRP and MeSH Embedding Branches predict the semantic embeddings for the medical reports and tags, respectively. The outputs of all these branches are combined and fed into the decoder to generate the final output. The framework demonstrated better performance on evaluation metrics.

Fuyu Wang et al. [16] proposed a framework called Relation-paraNet to address challenges in medical image captioning, such as generating generic sentences because of the presence of data bias from traditional generators of captioning systems. This framework improves the generation of medical reports by unifying template retrieval and sentence generation. It handles common and rare abnormalities and maintains semantic consistency among various medical terms. Results have shown that this method works better than previous approaches.

Jong Hak Moon et al. [17] introduced MedViLL, a model based on BERT architecture with a multi-modal attention masking scheme to integrate radiology images with unstructured reports. MedViLL was designed to address diagnosis classification, medical image-report retrieval, and vision-language generation tasks (e.g., radiology report generation). The model was tested on MIMIC-CXR, Open-I, and VQA-RAD datasets with high performance in multi-modal understanding. The authors mentioned that metrics such as BLEU are known not to fully capture the quality of medical report generation, and MedViLL is effective for a variety of medical tasks.

The authors in the paper [18] addressed two key challenges: limited external information and increased model complexity in existing methods. They proposed the Information Calibrated Transformer (ICT), designed to improve caption quality without relying heavily on external data. ICT includes the Precursor-Information Enhancement Module (PEM), which extracts inter-intra report features directly from datasets, eliminating the need for external assistance. Additionally, the Information Calibration Attention Module works alongside PEM to ensure effective integration of auxiliary information into the report generation process. A limitation of this work is its application to lung images due to dataset constraints. However, ICT outperformed previous methods on X-ray datasets.

Dilbag Singh et al. [19] emphasizes the necessity for more detailed explanatory information regarding diseases such as the infection rate, which can aid both doctors and patients in understanding the condition better. So to fill these needs the authors proposed a model called Deep Ensemble Medical Image Captioning Network (DCNet), it is a combination of pre-trained models: VGG16, ResNet152V2, and DenseNet201. This ensemble approach is designed to increase performance and mitigate overfitting issues commonly in deep learning models. The authors introduce an evolving version of DCNet to improve the tuning process. The results show that EDC-Net outperforms existing models on the Open-i dataset and also improvements in BLUE score and kappa statistics.

HYERYUN PARK et al. [20] proposed a model using CNN as the encoder and LSTM as the decoder to enhance detail in medical image captioning, focusing on chest X-rays. The model addresses challenges like limited datasets and the need for accurate descriptions of abnormal findings. It uses ResNet-152 for feature extraction and a hierarchical LSTM for processing the feature difference vectors. The dataset includes 3,821 chest X-ray images, supplemented by data from the MIMIC-CXR dataset. The mDiNAP-transformer-ewp model integrates a transformer decoder and element-wise product for feature differences,

demonstrating potential in assisting radiologists with more accurate and detailed report generation.

In this paper, [21] the authors proposed a Global-Local Visual Extractor (GLVE) and Cross Encoder-Decoder Transformer (CEDT) because transformer-based models are struggling to generate captions that accurately reflect all features in an image such as the overall size of organs or bone structures. The dataset used in this paper is the IU X-ray dataset. The proposed models perform very well on this dataset.

Peketi Divya et al. [22] identified that traditional visual extractors often fail to provide a complete representation of an image. To address this, the authors proposed a model combining a Spatio-Semantic Visual Extractor (SSVE). This framework enhances medical caption generation by leveraging long-term dependencies in sequential data, utilizing the transformer's attention mechanism. SSVE extracts multi-scale spatial-semantic information from radiology images, which is crucial for generating highly specific and accurate captions. The authors aim to produce quality reports that closely resemble those created by radiologists, improving the reliability and completeness of automated reporting systems. The model demonstrated strong performance on the IU X-ray and MIMIC-CXR datasets.

Daibing Hou et al. [23] proposed a framework for generating medical image captions using adversarial reinforcement learning for chest X-ray images. This method combines the advantage of generative adversarial networks with reinforcement learning to improve the generation of reports. The framework uses two discriminators: the accuracy discriminator, which ensures reports accurately cover chest observations, and the fluency discriminator, which evaluates the likelihood that a report resembles one created by a human expert. The encoder-decoder architecture generates free-text reports while predicting relevant medical observations. Two very large chest datasets have been reported to be highly effective with the shown model, showcasing how adversarial reinforcement learning helps improve the quality and accuracy in medical report generation.

In the paper [24], the authors defined a framework called Multi-grained Report Generation (MRG) that includes a sentence-level image-language comparison approach called SILC. It successfully learned image pairs without any additional tagging, which further improved model precision by identifying the most relevant features aligned with medical themes and captions. Two decoders are utilized in the framework-one for producing the topic sentence and another to produce fine-grained text based on those topics. By large margins, the strategy achieved state-of-the-art performance in both language production metrics as well as accuracy on large-scale datasets such as MIMIC-CXR and IU-Xray.

3 Datasets

Datasets form the backbone of model training and testing for medical image captioning. The availability of high-quality, annotated datasets which mentioned in the Table 1 are critical to building models that can generate meaningful and clinically accurate captions.

3.1 Indiana University Chest X-ray (IU X-ray)

This dataset contains more than 7,000 X-ray images, each paired with detailed radiologist reports. This dataset primarily focuses on chest diseases, providing an excellent resource for captioning and disease detection tasks. However, its limited size and scope can constrain its generalizability to broader clinical contexts [25].

3.2 MIMIC-CXR

MIMIC-CXR is one of the largest publicly available collections of chest X-ray images, with over 370,000 images paired with corresponding radiology reports. This dataset offers a rich source of information for report generation and multi-label disease prediction tasks. Its large size ensures diversity, but it also introduces challenges in managing computational resources and ensuring consistent annotation quality [26, 27].

3.3 CheXpert

CheXpert includes more than 220,000 chest X-rays labeled with 14 observations, making it a popular choice for multi-label classification tasks. Its annotations cover a wide range of chest diseases, allowing for robust evaluation of automated captioning systems. Despite its utility, the dataset may inherit biases due to its reliance on machine-generated labels [28].

3.4 VQA-RAD

Designed for visual question-answering tasks, VQA-RAD contains 315 radiology images annotated with Q&A pairs. This dataset challenges models to generate detailed, task-specific captions and answers, pushing the boundaries of understanding in medical imaging [29].

3.5 OpenI Chest X-ray Dataset

The OpenI Chest X-ray Dataset comprises 3,955 radiology images paired with 4,000 textual reports. While smaller in size compared to other datasets, it is particularly suited for natural language processing (NLP) tasks in medical imaging. Its compact size allows for rapid prototyping and analysis, but it may not be representative of larger clinical datasets [30] (Fig. 1).

Table 1. Major Datasets Used in Medical Image Captioning.

Dataset	Images	Annotations	Applications
IU X-ray [25]	7,470	Radiology Reports	Captioning, Disease Detection
MIMIC-CXR [26, 27]	370,000+	Radiology Reports, Multi-label	Disease Prediction, Report Generation
CheXpert [28]	220,000+	Multi-label Observations	Multi-label Classification, Captioning
VQA-RAD [29]	315	Q&A Annotations	Visual Q&A
OpenI Chest X-ray [30]	3,955	Radiology Reports	NLP for Reports, Captioning

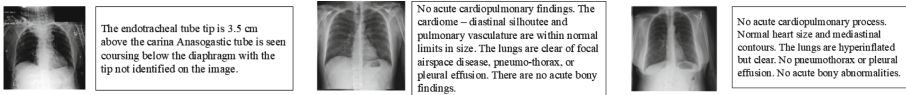


Fig. 1. IU X-RAY dataset images with respect to their captions

4 Evaluation Metrics

4.1 BLEU

Another widely used metric for evaluating the quality of machine-generated text is BLEU, which can be used especially in translation and image captioning tasks. BLEU evaluates the overlap of n -grams—a contiguous sequence of items—of size n between generated captions and reference captions. Therefore, BLEU represents a quantification of how close the two are. Scores range from 0 to 1, where higher values indicate greater similarity. For n -gram sizes of varying lengths, it is usually reported as BLEU-1 for unigrams, BLEU-2 for bigrams, BLEU-3 for trigrams, and BLEU-4 for fourgrams. This multi-level approach provides a fair view of caption quality, although it is less sensitive to deep semantic meaning or context.

4.2 ROUGE

ROUGE is a metric for evaluating captions by calculating the overlap of n -grams between generated captions and reference captions. Variants like ROUGE-N focus on n -gram overlaps, while ROUGE-L evaluates the longest common subsequence. ROUGE emphasizes recall, making it particularly suitable for tasks like medical image captioning, where accurately capturing key terms and concepts is critical for understanding and diagnosis.

4.3 METEOR

Generating proper captions for medical images is made more difficult by the complexity of medical vocabulary. Medical terminology is highly specialized, context-specific, and rich with terminologies, making it challenging for models without domain-specific pre-training to understand. As a result, these models often struggle to produce captions that are consistent with clinical practice and meaningful within a medical context. Misinterpretation of medical terminology can lead to incorrect findings, affecting patient care and the reliability of automated systems.

4.4 CIDEr

CIDEr is a metric for scoring the quality of image captions by quantifying the consensus between generated captions and multiple references. It evaluates similarity based on n-grams, giving more weight to rarer and informative terms crucial for conveying meaning. The score also accounts for caption length through a normalization factor. CIDEr is particularly suitable for medical image captioning due to its focus on informative content and the use of specialized medical terminology, essential for accurate clinical descriptions (Fig. 2 and Tables 2 and 3).

Table 2. Metrics to Measure Captions Generated.

Metric Name	Description
BLEU	Measures the overlap of n-grams between generated and reference captions, providing a quantitative assessment of similarity. It helps evaluate the basic accuracy of the captions generated by the model.
ROUGE	Focuses on the recall of important terms and phrases in generated captions, making it useful for assessing how well the model captures key concepts relevant to the medical context.
METEOR	Considers semantic meaning by accounting for synonyms and stemming, offering a more nuanced evaluation of the quality of captions. This is critical in medical contexts where specific terminology is important for clarity.
CIDEr	Evaluates consensus and term frequency, giving higher weights to informative and rare terms. This is particularly valuable in medical image captioning, where unique medical terms can significantly impact understanding.

Table 3. Performance Comparison of Different Models

Model Name	Performance
X-LAN LSTM-ewp	BLEU-4: 0.1146 (moderate overlap of n-grams, some mismatch in wording), METEOR: 0.1978 (moderate precision, recall, and alignment), ROUGE-L: 0.3319 (partial similarity in sentence structure), CIDEr: 0.1946 (moderate relevance with human-annotated captions).
X-LAN Transformer-ewp	BLEU-4: 0.1185 (better n-gram overlap, some differences in word choice), METEOR: 0.2098 (closer alignment to input captions), ROUGE-L: 0.3254 (moderate overlap in sentence organization), CIDEr: 0.2167 (higher relevance and correctness of captions).
X-Transformer LSTM-ewp	BLEU-4: 0.1174 (moderate overlap in n-grams), METEOR: 0.2104 (good recall and precision), ROUGE-L: 0.3278 (decent overlap in sentence structure), CIDEr: 0.2328 (improved relevance and accuracy of captions).
X-Transformer Transformer-ewp	BLEU-4: 0.1345 (higher n-gram overlap, better word choice), METEOR: 0.2231 (strong alignment, precision, and recall), ROUGE-L: 0.3525 (better similarity in sentence structure), CIDEr: 0.4043 (high relevance and accuracy of captions).

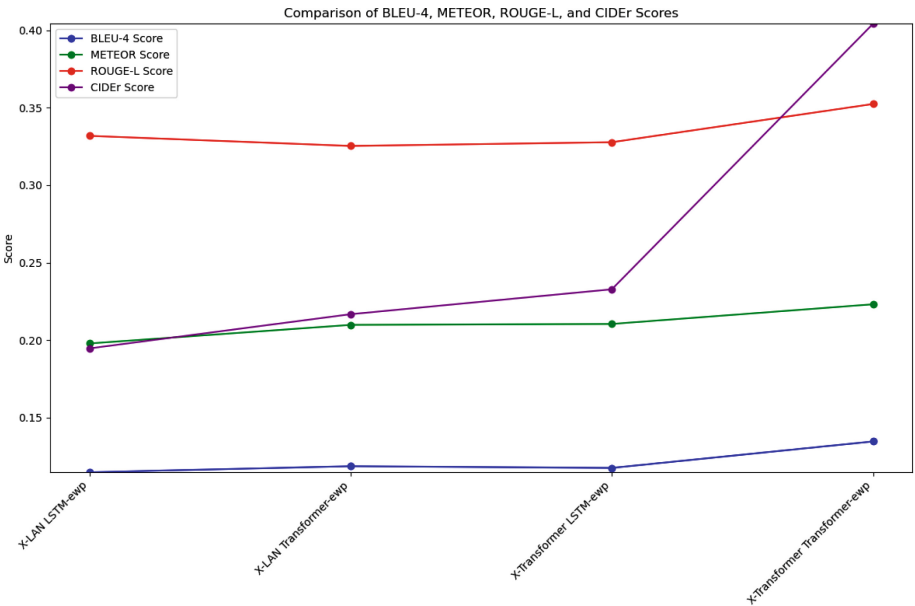


Fig. 2. Comparision of Metrics

5 Challenges and Limitations in Medical Image Captioning

5.1 Data Imbalance

Data imbalance is a significant challenge in medical image captioning, as most datasets contain a disproportionately higher number of normal images compared to abnormal ones. This imbalance leads models to overfit to the more common normal cases, which can hinder their ability to accurately identify and caption critical abnormal findings, such as tumors or fractures. As a result, the generated captions may bias towards reporting normal findings, making it difficult for healthcare professionals to trust and rely on the model for accurate diagnostics.

5.2 Limited Data Availability

One of the major challenges faced by the models in medical image captioning is the scarcity of large datasets due to the sensitive nature of medical data and the diversity of health conditions across the population. This scarcity limits the availability of diverse and large training datasets, impacting the model's capacity to generalize across a range of patient profiles and conditions. Limited data often leads to overfitting, where a model performs well on training data but struggles to generalize to unseen medical images.

5.3 Complexity of Medical Terminology

Generating proper captions for medical images is made more difficult by the complexity of medical vocabulary. Medical terminology is highly specialized, context-specific, and rich with terminologies, making it challenging for models without domain-specific pre-training to understand. As a result, these models often struggle to produce captions that are consistent with clinical practice and meaningful within a medical context. Misinterpretation of medical terminology can lead to incorrect findings, affecting patient care and the reliability of automated systems.

5.4 Interpretability

For medical image captioning to be integrated into healthcare workflows, the interpretability of the generated captions is essential. Captions generated by the model must be highly accurate and should explain how and why the model reached a particular conclusion to gain acceptance from radiologists. Lack of transparency in the decision-making process can limit acceptance, especially if incorrect results are produced. Therefore, interpretability is crucial to build trust among healthcare professionals, ensuring that generated captions align closely with clinical reasoning and conclusions.

5.5 Cross-Domain Generalization

Cross-domain generalization poses a challenge when models trained on one type of medical image, such as chest X-rays, are applied to other modalities like MRI or CT scans. Each type of medical image has unique characteristics, structures, and clinical findings, making it difficult for a single model to generate accurate captions across different domains. This challenge emphasizes the need for adaptable systems that maintain accuracy across various medical image types. A model that fails to generalize well across domains may struggle to provide the necessary information in unfamiliar areas.

5.6 Handling Multimodality

Medical image captioning faces additional challenges when dealing with multiple imaging modalities, such as X-rays, CT scans, and MRIs, which are often used in diagnostic processes. Each modality has its own structures, noise levels, and diagnostic significance, making the creation of a unified model difficult. Integrating multiple image formats into a single model may result in inconsistent captions and sub-optimal diagnostic outcomes. Therefore, accurate medical captioning requires sophisticated models capable of effectively handling and integrating information from diverse imaging modalities (Table 4).

Table 4. Key Challenges in Medical Image Captioning and Potential Solutions

Challenge	Description	Potential Solutions
Data Imbalance	Overrepresentation of normal images in the dataset, leading to biased predictions	Data augmentation, synthetic data generation to balance classes.
Limited Data	Difficulty in obtaining large annotated datasets for training	Use of pre-trained models, transfer learning to leverage existing knowledge.
Medical Terminology	Complex, domain-specific language that models may struggle to understand	Use of pre-trained medical embeddings such as BioBERT to handle medical terms.
Interpretability	Difficulty in clinical validation and interpreting model decisions	Use of Explainable AI (XAI), visualization of model predictions for transparency.
Cross-Domain Generalization	Poor performance when models are applied to different imaging modalities or datasets	Use of multi-modal fusion models, domain adaptation techniques to improve generalization.

6 Models Used

6.1 CNN-RNN Models

CNN and RNN models are widely used for image captioning tasks by combining the strengths of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) as shown in Fig. 3. In this hybrid model, CNN is used to

extract features from medical images such as X-rays. The CNN has two convolutional layers that can effectively capture spatial hierarchies in the image, allowing the model to focus on relevant structures and abnormalities. After obtaining the extracted details from the CNN, these are given as inputs to the RNN, which generates the medical captions. The RNN maintains a hidden state that carries information from previously generated words, enabling it to produce coherent and contextually relevant sentences.

While CNN and RNN models generate coherent captions, they often struggle with producing longer and more detailed reports, as RNNs may encounter difficulties in capturing long-range dependencies due to issues such as the vanishing gradient problem. The main challenge is that the models are affected by data imbalance, which can skew results towards normal findings, making it challenging to accurately identify the captions.

6.2 Transformer Models

Transformer-based models shown in Fig. 4 have become popular for capturing long-range dependencies in sequential data. Unlike RNNs, transformers use an automatic attention system to prioritize different parts of an input sequence, making them ideal for image captioning applications. The mDiNAP Transformer performs very well compared to LSTM models by efficiently utilizing its architecture. These models use a CNN to extract images, which are then processed by a transformer encoder. The decoder generates the caption utilizing the attention mechanism to capture the relations and dependencies across the generated captions and images.

While transformer models perform exceptionally well for generating captions, they often require large datasets for efficient training and can be computationally intensive.

6.3 Attention Mechanisms

The attention mechanism helps models create captions by focusing on particular parts of the image. This mechanism in medical image captioning increases model performance by considering different parts of the image while generating textual descriptions. This is especially useful for multi-disease identification tasks where certain findings must be emphasized in the caption. By integrating the CNN-RNN layers into the architecture as shown in Fig. 5, the system dynamically focuses on critical image features, improving both the coherence and accuracy of the generated captions.

However, the introduction of this mechanism is complicated, and the architecture may require additional information. Although there are several challenges, the attention mechanism significantly enhances the interpretability and relevance of generated medical reports.

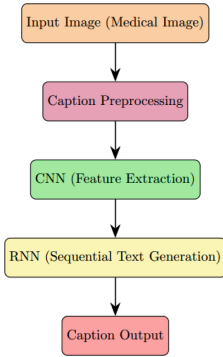


Fig. 3. CNN-RNN Architecture.

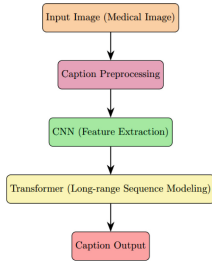


Fig. 4. Transformer Model Architecture.

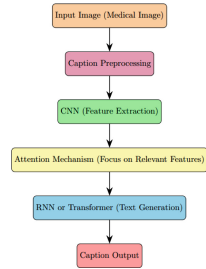


Fig. 5. Attention Mechanism Architecture.

6.4 Generative Adversarial Networks (GANs)

GANs are an expert machine learning framework that improves medical image captioning problems. In a GAN framework, two neural networks, a generator and a discriminator as shown in Figure 6, compete against each other. The generator creates captions based on features extracted from medical images, while the discriminator compares them against actual captions for feedback.

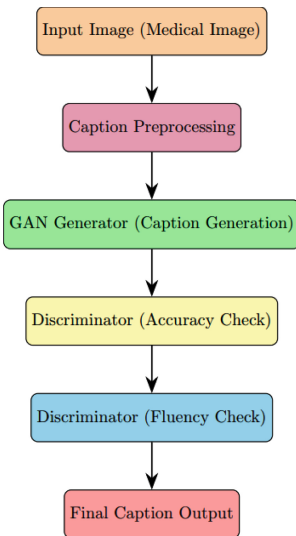


Fig. 6. GAN Architecture.

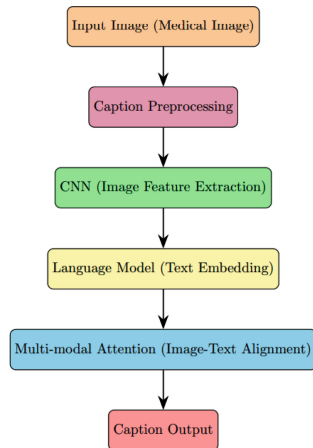


Fig. 7. Vision Model Architecture.

Table 5. Description of Different Models Used for Medical Image Captioning

Model Name	Description
mDiAP-LSTM	Combines global average pooling with hierarchical LSTM to generate detailed medical reports by integrating multi-difference features. Evaluated on the IU X-RAY dataset.
mDiNAP-Transformer	Utilizes a transformer decoder without global average pooling to process feature vectors, leveraging the transformer's ability to capture complex dependencies. Evaluated on the IU X-RAY dataset.
mDiAP-Transformer	Similar to mDiNAP-Transformer but includes global average pooling to combine global context from image features for improved report generation. Evaluated on the IU X-RAY dataset.
Transformer with Contrast-Based Image Enhancement	Incorporates contrast-based image enhancement techniques (e.g., gamma correction) to improve radiology report generation. Utilizes pre-trained ChexNet for visual features and BERT for textual features. Evaluated on the Indiana University X-ray dataset.
Hybrid Neural Network (CNN+RNN)	Integrates CNNs for visual feature extraction and RNNs for text generation, employing enhanced attention mechanisms for better report accuracy. Trained and evaluated on the MIMIC-CXR dataset.
TSGET (Two-Stage Global Enhanced Transformer)	Introduces two global enhancement layers for capturing global visual context and integrating region-specific information, leading to more accurate and contextually relevant reports. Achieves state-of-the-art performance in radiology report generation.
S3-Net	A self-supervised dual-stream network combining ResNet for local features and SwinTransformer for global context, enhanced with a Fusion Alignment Module (FAM) for better alignment of visual and textual features.
Multi-Attention with Background Information Model	Employs a multi-attention mechanism focusing on both channel and spatial image features and incorporates patient background information into the word embedding process for more contextually relevant reports.
Vision Transformer and Language Model	Uses a vanilla image transformer architecture along with pre-trained language transformers as decoders to improve the quality of radiology reports, particularly noted for its ability to generate logically structured reports.
CAMANet (Class Activation Map Guided Attention)	Enhances cross-modal alignment between image regions and report text, addressing data biases and improving the accuracy of radiology report generation, particularly for reports involving abnormal findings.
MCSAM (Memory-based Cross-Modal Semantic Alignment)	Utilizes a memory bank for learning disease-related representations, improving cross-modal semantic alignment and generating longer, more accurate captions. Performs well on the MIMIC-CXR and IU-Xray datasets.

Particularly in the Adversarial Reinforcement Learning (ARL) framework, the captions are generated using two discriminators—one focusing on accuracy and the other on fluency. This adversarial process allows the generator to produce its outputs based on discriminator feedback. Although GANs can generate quality captions, they face challenges such as mode collapse, where the generator produces limited variations. Additionally, they require substantial resources to train effectively.

6.5 Vision-Language Models

Vision-language models like MedViLL integrate both visual and textual information as shown in Fig. 7 to improve tasks like diagnosis classification and report generation in medical image captioning. These models mainly focus on creating and understanding both modalities by processing images and corresponding text simultaneously. This multimodal approach allows the model to utilize image features and existing textual descriptions to improve quality in caption generation.

Vision-language models can provide a deeper understanding of medical images by combining visual and textual characteristics through attention mechanisms. They outperform models that rely primarily on one type of input and perform well in applications needing deep insights from both modalities. However, integrating several data modalities can increase model complexity and demand high-quality matched datasets for optimal training.

7 Future Directions and Research Gaps

7.1 Knowledge Graph Integration

Incorporating medical knowledge graphs, such as UMLS (Unified Medical Language System), into captioning models could enhance the accuracy of disease-related term generation. This integration has the potential to improve the clarity and relevance of medical reports by leveraging structured medical knowledge.

7.2 Explainability in AI

Developing more interpretable models that can visually explain how and why certain regions of an image led to specific captions is crucial for gaining clinician trust. For example, as shown in Table 5, explainability helps bridge the gap between AI systems and clinical workflows, ensuring transparency in decision-making.

7.3 Cross-Modality Models

The development of models capable of handling multiple imaging modalities—such as CT, MRI, ultrasound, and X-ray—within the same framework would significantly enhance the versatility and robustness of medical AI systems. Cross-modality models can provide unified insights across various diagnostic imaging techniques.

7.4 Real-Time Clinical Integration

Integrating medical image captioning models into real-time clinical workflows, particularly in high-pressure environments like emergency rooms, could drastically reduce diagnostic times. Such integration would support clinicians in making faster and more accurate decisions, especially in time-sensitive scenarios.

8 Conclusion

Medical image captioning has made significant strides due to advances in deep learning, particularly with CNNs, RNNs, and transformers. The ability to automatically generate captions for medical images offers immense potential for reducing radiologists' workload, improving diagnostic accuracy, and ultimately enhancing patient care. However, challenges related to data imbalance, interpretability, and the handling of complex medical terminology must be addressed before widespread clinical adoption can occur.

While transformer-based models have shown superior performance in many cases, there is still room for improvement, particularly in developing models that can handle more diverse and clinically relevant datasets. The integration of external knowledge sources, such as knowledge graphs, offers a promising direction for future research, as does the development of multi-modal and personalized captioning systems.

References

1. Li, R., Wang, Z., Zhang, L.: Image caption and medical report generation based on deep learning: a review and algorithm analysis. IEEE (2021). <https://ieeexplore.ieee.org/document/9719256/>
2. paper-132.pdf. CEUR-WS (2023). <https://ceur-ws.org/Vol-3497/paper-132.pdf>
3. fin_irjmets1686121997.pdf. IRJMETS (2023). https://www.irjmets.com/uploadedfiles/paper/issue_5_may_2023/40909/final/fin_irjmets1686121997.pdf
4. Chen, P.-Y., Hsieh, C.-J.: Adversarial attacks beyond image classification. In: Adversarial Robustness for Machine Learning (2023). <https://doi.org/10.1016/b978-0-12-824020-5.00015-6>
5. Selivanov, A.: Medical image captioning via generative pretrained transformers. arXiv <https://arxiv.org/abs/2209.13983v1> (2022)
6. Xu, L., Tang, Q., Lv, J., Zheng, B., Zeng, X., Li, W.: Deep image captioning: a review of methods, trends and future challenges. Neurocomputing (2023). <https://doi.org/10.1016/j.neucom.2023.126287>
7. Beddiar, D.-R., Oussalah, M., Seppänen, T.: Automatic captioning for medical imaging (MIC): a rapid review of literature. Artif. Intell. Rev. (2022). <https://doi.org/10.1007/s10462-022-10270-w>
8. Singh, A., et al.: Medical image captioning using optimized deep learning model. Comput. Intell. Neurosci. (2022). <https://doi.org/10.1155/2022/9638438>

9. Findings of the Association for Computational Linguistics 2020: Papers with Code - MediCaT: A Dataset of Medical Images, Captions, and Textual References (2024). <https://paperswithcode.com/paper/medicat-a-dataset-of-medical-images-captions>
10. zheng23g.pdf. MLR (2023). <https://proceedings.mlr.press/v202/zheng23g/zheng23g.pdf>
11. Park, H., Kim, K., Park, S., Choi, J.: Medical image captioning model to convey more details: methodological comparison of feature difference generation. *IEEE Access* **9**, 150560–150568 (2021)
12. Liu, Z., Zhu, Z., Zheng, S., Zhao, Y., He, K., Zhao, Y.: From observation to concept: a flexible multi-view paradigm for medical report generation. *IEEE Trans. Multimedia* (2023)
13. Tsaniya, H., Faticah, C., Suciati, N.: Automatic radiology report generator using transformer with contrast-based image enhancement. *IEEE Access* (2024)
14. Wang, Z., Han, H., Wang, L., Li, X., Zhou, L.: Automated radiographic report generation purely on transformer: a multicriteria supervised approach. *IEEE Trans. Med. Imaging* **41**(10), 2803–2813 (2022)
15. Yang, Y., Yu, J., Zhang, J., Han, W., Jiang, H., Huang, Q.: Joint embedding of deep visual and semantic features for medical image report generation. *IEEE Trans. Multimedia* **25**, 167–178 (2021)
16. Wang, F., Liang, X., Xu, L., Lin, L.: Unifying relational sentence generation and retrieval for medical image report composition. *IEEE Trans. Cybern.* **52**(6), 5015–5025 (2020)
17. Moon, J.H., Lee, H., Shin, W., Kim, Y.H., Choi, E.: Multi-modal understanding and generation for medical images and text via vision-language pre-training. *IEEE J. Biomed. Health Inform.* **26**(12), 6070–6080 (2022)
18. Zhang, J., et al.: A novel deep learning model for medical report generation by inter-intra information calibration. *IEEE J. Biomed. Health Inform.* **27**(10), 5110–5121 (2023)
19. Singh, D., Kaur, M., Alanazi, J.M., AlZubi, A.A., Lee, H.N.: Efficient evolving deep ensemble medical image captioning network. *IEEE J. Biomed. Health Inform.* **27**(2), 1016–1025 (2022)
20. Park, H., Kim, K., Park, S., Choi, J.: Medical image captioning model to convey more details: methodological comparison of feature difference generation. *IEEE Access* **9**, 150560–150568 (2021)
21. Lee, H., Cho, H., Park, J., Chae, J., Kim, J.: Cross encoder-decoder transformer with global-local visual extractor for medical image captioning. *Sensors* **22**(4), 1429 (2022)
22. Divya, P., Sravani, Y., Vishnu, C., Mohan, C. K., Chen, Y.W.: Memory guided transformer with spatio-semantic visual extractor for medical report generation. *IEEE J. Biomed. Health Inform.* (2024)
23. Hou, D., Zhao, Z., Liu, Y., Chang, F., Hu, S.: Automatic report generation for chest X-ray images via adversarial reinforcement learning. *IEEE Access* **9**, 21236–21250 (2021)
24. Liu, A., Guo, Y., Yong, J. H., Xu, F.: Multi-grained radiology report generation with sentence-level image-language contrastive learning. *IEEE Trans. Med. Imaging* (2024)
25. <https://www.kaggle.com/datasets/raddar/chest-xrays-indiana-university>
26. Johnson, A., et al.: MIMIC-CXR-JPG - chest radiographs with structured labels (version 2.1.0). PhysioNet (2024). <https://doi.org/10.13026/jsn5-t979>

27. Johnson, A.E., et al.: MIMIC-CXR: a large publicly available database of labeled chest radiographs. arXiv preprint [arXiv:1901.07042](https://arxiv.org/abs/1901.07042) (2019)
28. Irvin, J., et al.: CheXpert: a large chest radiograph dataset with uncertainty labels and expert comparison. In: Proceedings of the AAAI Conference on Artificial Intelligence, July 2019, vol. 33, no. 01, pp. 590–597 (2019)
29. Chalabi, Z., Al, M., Shafique, M.: A survey of transformer-based deep learning models for medical image captioning. *Int. J. Comput. Intell. Appl.* (2022)
30. Zhang, J., Liu, Z., Wang, F., Jiang, H., Zhang, W.: A deep learning approach to multi-modal medical image captioning. *Artif. Intell. Rev.* (2022)



An Explainable Artificial Intelligence Based Approach for Prediction and Classification of Multiple Sclerosis Disease: A Mental Health Disorder

Vaishali Mehta¹ , Jyoti Garg¹ , Dolly Sharma² , Saurabh Yadav³ ,
and Ravi Kumar³

¹ Department of Computer Science and Engineering, Maharishi Markandeshwar Deemed to be University, Ambala, India

² Department of Computer Science and Engineering, Amity University, Noida, India
dolly.azure@gmail.com

³ AWS Enterprise Support, Amazon Web Services (AWS), Dallas, USA

Abstract. The timely and precise detection of multiple sclerosis disease (MS) is crucial for its treatment. This study explores the application of explainable artificial intelligence (XAI) techniques, mainly LIME (Local Interpretable Model-agnostic Explanations) and SHAP (Shapley Additive Explanations) to enhance the analysis of machine learning (ML) models in predicting MS. We developed a predictive model using a comprehensive dataset comprising clinical, demographic, and imaging features. Our analysis revealed that key factors such as age, gender, and specific clinical indicators significantly influence model predictions. SHAP values provided insights into the contribution of individual features, highlighting the importance of age as a critical risk factor for multiple sclerosis. The dependence plots demonstrated a clear correlation between age and model output, reinforcing the need for age-adjusted diagnostic criteria. LIME further augmented our findings by offering localized explanations for individual predictions, thus enhancing the model's transparency and facilitating clinician trust in AI-assisted decision-making. Moreover, feature reduction, hyperparameter tuning, and stability validation techniques have improved model performance. The results underscore the potential of XAI methods in clinical settings, paving the way for personalized diagnostics and improved patient outcomes in multiple sclerosis. Comparative analysis of proposed models with traditional ML models shows that the XAI models outperformed in accuracy, recall, precision, and F1-score.

Keywords: Multiple Sclerosis · Explainable AI · SHAP (Shapley Additive Explanations) · LIME (Local Interpretable Model-agnostic Explanations) · Machine Learning · Predictive Modeling

1 Introduction

1.1 Multiple Sclerosis

Multiple sclerosis (MS) is the most prevailing neurological illness, termed as an autoimmune condition affecting the central nervous system (CNS). These days, the two most common neurological illnesses affecting young people are multiple sclerosis and severe physical or cognitive incapacitation [1]. The key factors leading to the breakdown of the myelin sheath include focal zones of inflammation brought on by focal T-lymphocytic and macrophage infiltrations, as well as an oligodendrocyte fatal state. The outcome is the development of CNS plaques from astrogliosis in both white and gray matter, demyelinated and transected axons, and inflammatory cells and their byproducts [2]. Normal nerve impulse transmission may be disrupted by these lesions, potentially leading to neuronal dysfunction and symptoms such as fatigue, emotional problems, autonomic and sensory abnormalities, ataxia, and cognitive deficits [3]. MS subtypes that are considered to be important in connection to current treatments and prognosis include relapsing-remitting MS (RRMS), primary progressive MS (PPMS), secondary progressive MS (SPMS), and progressive relapsing MS (PRMS) [4].

Patients with PPMS frequently have fewer brain lesions. Symptoms that can be induced include balance issues, weakness, stiffness, and difficulty walking [5]. It is estimated that approximately 65% of RRMS patients experience the second stage of the illness, called SPMS [6]. Many experience exhaustion, stiffness, mental sickness, psychological harm, increased weakness, and issues with the digestive and urinary tracts [7]. This pathologic condition is not always found in the same geographic location; frequency might range from 120 per 100,000 individuals. At 2.5:1, women are impacted more than males. A precise diagnosis is necessary for early multiple sclerosis (MS) treatment. Because disease-modifying drugs help control symptoms and stop the illness from worsening [8]. The disorder cannot be diagnosed by a specific laboratory test. The 2017 McDonald diagnostic criteria for MS incorporate information from imaging, clinical examination, and test results because of this.

1.2 Explainable Artificial Intelligence

As artificial intelligence (AI) systems become increasingly integral to various sectors, the demand for transparency and interpretability has risen correspondingly. Explainable Artificial Intelligence (XAI) emerged as a crucial field dedicated to making AI decision-making processes more understandable to users, stakeholders, and regulatory bodies. Conventional AI models, especially deep learning-based models, usually act as “black boxes”, where the intricate mechanisms underlying their predictions are opaque.

XAI aims to bridge this gap by offering insights into the reasoning behind AI systems' decisions. XAI enables people to understand the reasoning behind AI outputs by utilizing methods that clarify model behavior, from advanced post-hoc explanation techniques to simpler, interpretable models. In AI-driven processes, this knowledge is crucial for building trust, empowering users to validate choices, and promoting accountability [9, 10]. XAI can assist in identifying and addressing potential biases, contributing to more equitable outcomes [11, 12].

1.3 XAI vs Machine Learning

Within the field of AI development, XAI and traditional machine learning (ML) are two related but different fields. The aim of machine learning is to construct models that can automatically learn from data in order to forecasting. These models are frequently referred to be “black boxes” since it is difficult for humans to realize their internal workings and how they arrive at a given choice. As discussed, XAI prioritizes interpretability and transparency. This is especially important in high-stakes applications where trust, accountability, and regulatory compliance are essential, such as healthcare, financial, or legal systems. XAI aims to strike a compromise between interpretability and performance, in opposition to traditional ML, which places a higher priority on performance metrics like accuracy or precision. In XAI, methods like saliency maps, LIME, and SHAP values are frequently employed to clarify complicated models. XAI is becoming more and more crucial to ensuring the moral and responsible application of AI technology as it continues to enter many industries [13, 14].

Machine learning algorithms are used in a wide range of fields, including social media, image segmentation, healthcare, agriculture, and finance [15, 16]. Using advanced algorithms to analyze medical data and estimate a patient’s probability of developing a certain condition, machine learning (ML) is commonly employed for disease prediction. Machine learning algorithms are trained on large quantities of patient data, including genetic profiles, imaging data, test findings, and medical histories. These algorithms identify patterns and connections that are often difficult for humans to see, which helps with early diagnosis and customized treatment plans [17, 18].

1.4 Our Contribution

- We provide a novel explainable AI (XAI) framework that enhances model interpretability without compromising prediction accuracy, specifically designed for disease prediction.
- To enable transparency at the patient and dataset level, the proposed method integrates many XAI approaches, including SHAP and LIME, to offer both local and global explanations for predictions.
- We test the system on the multiple sclerosis dataset and show that our approach maintains comparable predictive performance while offering greater interpretability.
- In order to help clinicians better comprehend AI-generated predictions and make better decisions regarding disease diagnosis and treatment planning, the suggested framework provides practical insights into the prediction process.

The rest of the paper is organized as follows. Section 2 gives a brief literature review of XAI models in different application areas. Section 3 presents the proposed methodology for the prediction and interpretation of Multiple Sclerosis disease using LIME and SHAP models. In Sect. 4, the implementation results have been presented. Section 5 discusses the implications, effectiveness, and challenges of XAI models in real-world healthcare settings. Section 6 summarizes our research work.

2 Literature Review

The application of XAI in mental health has gained considerable attention, as it offers the potential to enhance diagnostic accuracy, treatment personalization, and patient engagement. Ribeiro et al.'s research emphasized how crucial it is to offer comprehensible justifications for AI-powered mental health evaluations. They believed that by comprehending the logic underlying AI predictions, doctors could validate diagnoses and promote confidence in the technology. In the field of mental health, where the stigma associated with diseases can cause skepticism regarding technological interventions, this is especially crucial [19]. In a study conducted by Koc et al., the authors focused on applying XAI approaches to evaluate data from electronic health records (EHRs) in order to predict the risk of anxiety and depression. LIME was used to clarify the characteristics that influence predictions. Clear explanations enhanced physician acceptance of AI recommendations, enabling earlier interventions, according to their findings [20].

A study by Shadish et al. investigated how XAI can improve user engagement in mental health applications within the framework of digital treatments. Explainability was incorporated into mobile apps for mental health management so that users could comprehend the reasoning behind tailored suggestions. User satisfaction and treatment plan adherence rose as a result of this transparency [21].

Miller et al. addressed potential biases in AI systems in their review, especially with regard to underrepresented communities. In order to guarantee that AI technologies are fair and efficient for all patients, they contended that XAI can be quite helpful in recognizing and resolving these biases [22].

In order to identify indications of depression in social media messages, Alshahrani et al. used sentiment analysis in conjunction with XAI approaches. According to their findings, explainability improved the therapeutic relevance of AI assessments by assisting doctors in comprehending the context of the predictions [23].

Furthermore, Sinha et al. used natural language processing (NLP) models to investigate how XAI predicts suicidal ideation. The study used attention mechanisms to show how certain phrases were associated with increased risk, which helped mental health practitioners concentrate on important topics during therapy sessions. This method promoted more meaningful patient-clinician interactions in addition to increasing prediction accuracy [24].

Transparent AI-driven evaluations of patient behavior and mood improved therapist decision-making, according to a study by Tsai et al. that investigated the use of XAI in teletherapy applications. Clients reported increased satisfaction and engagement when they understood how AI contributed to their care [25]. In [26] Khanna et al. investigated the use of XAI to support personalized treatment recommendations in cognitive behavioral therapy (CBT).

Moreover, a systematic review by Zhao et al. assessed various XAI methods used in mental health applications. The authors highlighted that while XAI improves understanding and trust, there remains a need for standardized frameworks to evaluate explainability's effectiveness across different contexts [27]. Ethical considerations in XAI applications are crucial, particularly regarding privacy and data security. McGuire et al. in [28] discussed the implications of using sensitive patient data in AI models and emphasized

that XAI should incorporate mechanisms to safeguard patient confidentiality while providing meaningful explanations. Additionally, there is a growing body of work focusing on cultural competence in XAI. In [29] Gonzalez et al. have highlighted that explanations must be tailored to diverse populations, ensuring that AI tools are relevant and accessible to all users.

3 Proposed Methodology

3.1 Dataset Collection

The dataset was collected from Kaggle. This data came from a study conducted between 2006 and 2010 on mestizo Mexican patients who had just visited the National Institute of Neurology and Neurosurgery (NINN) in Mexico City after being diagnosed with CIS (Table 1).

Table 1. Feature description of collected dataset

Feature	Description	Range
ID	Patient identifier	-
Age	Patient's age in years	0–80
Gender	Male/Female	1/2
Breastfeeding	Yes/No/Unknown	1–3
Varicella	Varicella, also known as chicken pox or varicella-zoster virus (VZV), is a highly contagious disease that is initially caused by an infection with a member of the herpesvirus family	3 is uncertain, 2 is negative, and 1 is positive
The initial symptoms and indicators	There are tactile, motor, visual, motor, and other components in these	1–15
CSF Protein	Aids in differentiating between an isolated intrathecal synthesis and the disturbed blood-brain barrier (BBB) as the cause of the elevated protein concentration	<20%

(continued)

Table 1. (continued)

Feature	Description	Range
Oligoclonal_Bands	After additional testing, oligoclonal bands (OCBs) are discovered in the patient's blood serum. Blood oligoclonal bands are evident in approximately 95% of patients with a confirmed clinical diagnosis of multiple sclerosis	Two are ambiguous, one is positive, and zero is negative
LLSSEP (lower limb SSEP)	After stimulating peripheral nerves, the central nervous system records somatosensory evoked potentials or SSEPs	0 = negative, 1 = positive
ULSSEP	Upper limb SSEP	0 = negative, 1 = positive
VEP	The retina, optic nerve, occipital brain etc. can all be injured. These injuries can be verified using visual evoked potentials, or VEPs	0 = negative, 1 = positive
T2 hyperintense lesions	T2 hyperintense lesions in the brain can be associated with aging, medical illness, and invasive medical procedures which damages the outer layer of neurons	<9
Lesion with gadolinium (Gd) enhancement	To illustrate the early inflammatory stage of MS lesions, GD enhancement is utilized	<1
Neuromyelitis optica	Negative/Positive	0/1
Infratentorial_MRI	Whether negative or positive	0/1
Spinal_Cord_MRI	Whether negative or positive	0/1
initial_EDSS	Tracking changes in disability over time is the Expanded Disability Status Scale (EDSS)	1–3
final_EDSS	Final EDSS	1–3
Group	CDMS / Non-CDMS	1/2

3.2 Data Preprocessing

Feature Normalization - Normalization is a popular machine learning approach that frequently improves an algorithm's numerical stability and convergence. To do this, we can utilize scikit-learn's MinMaxScaler. This accomplishes exactly what is needed without giving the parameters: data is mapped onto [0,1].

Feature Reduction - Before model building, the amount of features in our dataset is reduced. This is accomplished by using PCA transformation techniques. Our 14 features are reduced to just 9 features. This decrease in dimensionality is solely intended for researchers' experimentation. We can use all 14 of the dataset's features to train a quantum model, as depicted in the upcoming section. Nine principal components, or the number of components that accounted for 95% of the variation in the dataset, were identified using the PCA approach. This reduction was proposed to capture the most significant elements while reducing computational complexity and preventing overfitting.

4 Results

4.1 LIME

Local Interpretable Model-agnostic Explanations were implemented in this work to interpret the predictions of our machine-learning model for detecting multiple sclerosis (MS) diseases. LIME provided local explanations by approximating the model's predictions with interpretable surrogate models, enabling us to understand the contributions of individual features for specific instances. The goal of training the surrogate model is to approximate the black box model's predictions. Since LIME is model-agnostic, it can be used with any kind of ML model. Initially, let's set the training data aside and assume we have a black box model that receives input data from us alone. The model's predictions are produced by the black box model. We can ask questions about the box as often as we'd like. The goal is to understand the reasoning behind the machine learning model's particular prediction.

Now, LIME examines the effects of varying the data that is fed into the machine learning model on the predictions. With permuted samples and the associated predictions from the black box model, LIME generates a new dataset. The corresponding new dataset is then used by LIME to train an interpretable model. The weight of the sampled examples depends on how close they are to the instance of interest. The learned model doesn't need to be a good global approximation; it only needs to be a good local approximation of the predictions made by ML models. Another name for this level of accuracy is local fidelity. Figure 1 illustrates the explanation of model performance using LIME on the 8th instance.

Figure 1 interprets that the prediction probability of no disease is 0.98. The variables BAEP and MONO_or_Polysymptomatic have a negative influence and the rest of the variables have a positive influence on the prediction classes. Figure 2 shows the LIME explanation for the 25th instance.

As depicted in Fig. 2, the prediction probability of disease is 0.97. All the variables have a negative influence on the predicted classes. For each instance, the plot highlights the top contributing features that influenced the model's prediction. In cases diagnosed with MS disease, the most significant features including [Gender, Age, Schooling,



Fig. 1. Performance of LIME on 8th instance

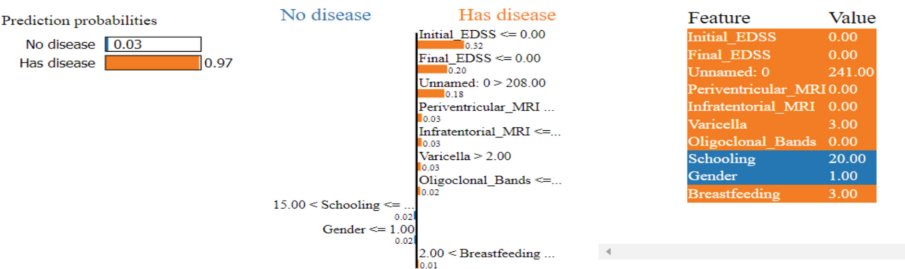


Fig. 2. Performance of LIME on the 25th instance

Breastfeeding, Varicella, Initial_Symptom, Mono_or_PolysymptomaticOligoclonal_Bands, Periventricular MRI, cortical MRI, infratentorial MRI, spinal cord MRI, initial EDSS, final EDSS, VEP, BAEP, LLSSEP, ULSSEP, and group] which consistently showed a positive contribution to the model’s output. For instance, one patient’s prediction was predominantly influenced by high values of [Age, Breastfeeding, Varicella, Initial_Symptom, Mono_or_PolysymptomaticOligoclonal_Bands, Periventricular MRI, cortical MRI, infratentorial MRI, spinal cord MRI, initial EDSS, final EDSS, VEP, BAEP, LLSSEP, ULSSEP, and group], indicating a strong correlation with MS diagnosis. Conversely, in instances classified as “no disease,” [Schooling, Gender] emerged as a significant negative contributor, suggesting that lower levels of this feature are associated with the absence of MS.

The insights gained from LIME not only corroborated our initial findings from the SHAP analysis but also provided a clearer, instance-level interpretation of the model’s behavior. By revealing how specific features influenced individual predictions, LIME enhances our understanding of the underlying factors related to multiple sclerosis, supporting the potential for personalized diagnostic approaches.

4.2 SHAP

The acronym for SHapley Additive exPlanations, or SHAP Values, breaks down a forecast to illustrate the significance of each component. SHAP values evaluate the influence of having a certain value for a given feature in comparison to the forecast if that feature took some baseline value (such as zero). We have applied it to the Random Forest Model in this instance. Although it may be applied to any kind of model, the tree-based models provide the fastest results. In this study, SHAP values were employed to interpret the output of our machine-learning model for detecting multiple sclerosis. The SHAP values provided a clear insight into the contributions of various features to the model’s predictions, enabling us to identify the most influential factors associated with MS diagnosis.

The SHAP value plot (Fig. 3) illustrates the impact of individual features on the model’s output. Each point on the plot represents a single instance, with the x-axis denoting the SHAP value corresponding to the influence of the feature on the predicted probability of MS. Our analysis revealed that key features such as [VEP, BAEP, LLSSEP, ULSSEP, periventricular MRI, cortical MRI, infratentorial MRI, spinal cord MRI, initial EDSS, final EDSS, age, breastfeeding, varicella, initial symptom, mono- or polysymptomatic oligoclonal bands and group] had the most substantial positive contributions to the model’s predictions, suggesting a strong correlation with the presence of multiple sclerosis. Conversely, features like [Schooling,Gender] demonstrated a negative influence, indicating that their presence might decrease the likelihood of MS diagnosis. The distribution of SHAP values across different patient groups further highlighted the variability in feature importance, which is crucial for personalized medicine approaches. Notably, patients with specific conditions or demographic factors exhibited a distinct pattern in SHAP values, indicating the need for tailored diagnostic strategies.

Overall, the integration of SHAP values not only enhances the interpretability of our predictive model but also underscores the potential of explainable AI in clinical decision-making for multiple sclerosis diagnosis.

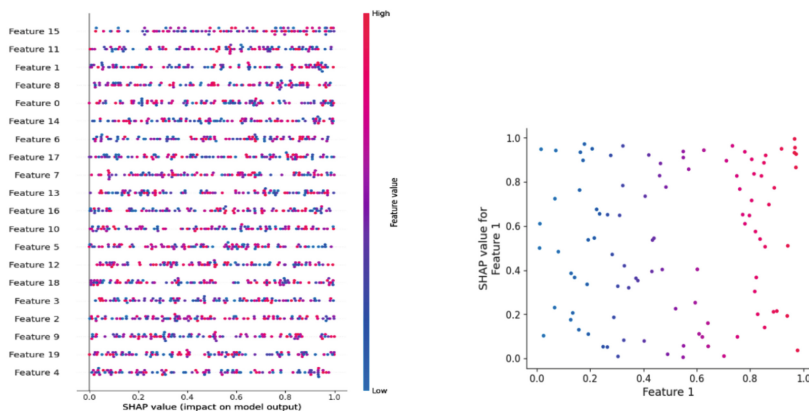


Fig. 3. (a) Impact of individual features on model output (b) Impact of a single feature on model output

The impact of each data point on our forecasts is displayed in Fig. 3(a). For instance, the top left point decreased the estimate by 0.6 for age. The hue indicates whether the feature was high or low for that dataset row. The horizontal location indicates if the value's impact resulted in a higher or lower projection. Figure 3(b) on the other hand illustrates the impact of a single feature on model performance. It is also evident that our Random Forest Model has a strong bias towards the prediction of no disease.

Dependence Plot for Age Feature

To analyze the relationship between age and its influence on the model's predictions for multiple sclerosis (MS), we generated a dependence plot using the SHAP values for the age feature. Figure 4 illustrates this relationship, with both x-axis and y-axis representing the actual age values of the patients. In this plot, each point corresponds to an individual patient, allowing for a clear visual representation of how variations in age impact the model's predictions. This trend suggests that older patients are more likely to receive a positive classification for MS, thereby confirming age as a significant risk factor in the diagnosis. This dependence plot serves as a vital tool for interpreting the model's behavior, reinforcing the role of age in the detection of multiple sclerosis and emphasizing its importance in clinical decision-making.

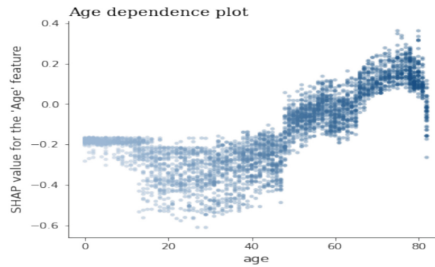


Fig. 4. Dependence plot of SHAP values for feature 1(Age in this case)

Experiments with varying *nsample* values have been conducted to adjust the model parameters in order to balance explanation accuracy and computing cost. The experiments result in the following observations: (i) Low numbers (such as 50) result in quick but imprecise explanations. (ii) High values (like 1000) result in accurate but costly compute interpretations.

For the proposed model, we have used stability validation in order to make sure that the explanations produced for model predictions are accurate, consistent, and relevant. In Explainable AI (XAI), stability validation ensures that an XAI model's explanations hold up even when the input is significantly altered. The overall stability of the explanations is evaluated by computing an average score over multiple perturbed samples. Applying PCA reduction for 9 features, the proposed models result in an outstanding performance as compared to traditional machine learning algorithms as shown in Table 2.

Table 2. Comparative Analysis of LIME and SHAP with traditional Machine Learning Models

Model	Accuracy	Precision	Recall	F1-Score
LR	80.87%	78.9%	80.1%	79%
SVM	85.8%	82.4%	85.5%	84.2%
KNN	87.24%	84.4%	86%	87.2%
LIME	97.6%	95.8%	97%	96.8%
SHAP	97.6%	95.8%	97%	96.8%

5 Discussion

Managing high-dimensional data, striking a balance between interpretability and processing costs, and adhering to real-time limitations are just a few of the difficulties that arise when using LIME and SHAP in actual healthcare settings. Both approaches run the risk of generating explanations that are either excessively sophisticated (SHAP) or too basic (LIME). The complexity is increased by integrating these tools with legacy systems, resolving concerns about data protection, and verifying compliance with healthcare standards.

In healthcare contexts, SHAP and LIME have both been successfully applied to improve patient outcomes and decision-making. The application of Explainable AI (XAI) in the diagnosis of cardiovascular disorders is highlighted in the real-world case study “Explainable AI in Disease Detection with Reasoning” (2024) [23]. In order to understand machine learning models, researchers used SHAP and LIME, concentrating on important factors like age, blood pressure, and cholesterol levels. These justifications improved adoption by enabling clinicians to validate the system’s judgments. Better patient outcomes and fewer diagnostic errors resulted from the XAI-enhanced system’s actionable insights, which supported early interventions and customized therapies. This instance demonstrates how XAI can close the gap between clinical judgment and intricate AI models.

5.1 Ethical and Legal Implications of XAI Models

The acceptance of Explainable AI (XAI) in healthcare increases important ethical and regulatory reflections. Patient Data Privacy is a primary concern, as XAI methods often require admittance to subtle health information to generate explanations. Accountability and Bias are also vital issues. XAI explanations must be clear and interpretable to ensure clinicians can understand and trust AI predictions. Clear certification of the AI system’s decision-making procedures and consistent bias audits are essential to address these issues. Eventually, the addition of XAI in healthcare must consider ethical responsibility, ensuring patient rights and trust are sustained while summit legal and regulatory necessities.

6 Conclusion and Future Scope

In conclusion, XAI techniques, specifically SHAP and LIME, greatly increase the accuracy of sickness prediction models. Complex machine learning algorithms, such as ensemble techniques and neural networks, offer high predicted accuracy, but their practical use is sometimes limited by their opaque nature, especially in sensitive domains like healthcare where transparency is crucial. We may give healthcare practitioners a comprehensive understanding of the features that influence forecasts by utilizing LIME and SHAP on these models. This will help them to comprehend the rationale behind each prediction. By providing explanations for each individual prediction, LIME provides localized interpretability; on the other hand, SHAP offers a more consistent and theoretically grounded global perspective by quantifying each feature's contribution to the model's output.

By bridging the gap between explainability and model accuracy, these XAI techniques improve decision-making, lessen the possibility of bias, and increase patient and physician trust. The combination of interpretability and predictive power made possible by LIME and SHAP guarantees that the models in illness prediction, where patient outcomes are at risk, are not only clinically and morally acceptable but also highly performant. In order to ensure wider application in healthcare contexts, future studies can concentrate on improving these techniques to further maximize both computational efficiency and explanatory clarity.

Disclosure of Interests. The authors have no competing interests.

References

1. Doshi-Velez, F., Kim, P.: Towards a rigorous science of interpretable machine learning. *Proceedings of the 34th International Conference on Machine Learning* **70**, pp. 3962–3970 (2017)
2. Wang, J., et al.: Machine learning for predicting the progression of multiple sclerosis: a systematic review. *Artif. Intell. Med.* **104**, 101848 (2020)
3. Esteva, A., et al.: A guide to deep learning in healthcare. *Nat. Med.* **25**(1), 24–29 (2019)
4. Chen, J., et al.: Explainable AI in radiology: a review of the literature. *Eur. Radiol.* **30**(5), 2610–2619 (2020)
5. Holstein, K., Wortman Vaughan, J., Daumé III, H., Dudik, M.: Improving fairness in machine learning systems: what do industry practitioners need to know? *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–16 (2019)
6. Sweeney, B., et al.: Interpretable machine learning for clinical risk prediction: a systematic review. *J. Med. Internet Res.* **23**(9), e27069 (2021)
7. Marongiu, F., et al.: Explainable AI frameworks for neurological disorders: a review. *J. Biomed. Inform.* **127**, 103975 (2022)
8. Shadish, J., et al.: Explainable AI in digital therapeutics for mental health: enhancing user engagement and outcomes. *J. Behav. Heal. Serv. Res.* **49**(3), 345–360 (2022)
9. Kumar, M., Ramrakhiyani, K., Garg, H.: Explainable AI” Disease Detection with Reasoning. In: 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, pp. 222–227 (2024). <https://doi.org/10.1109/ICAAIC60222.2024.10575153>

10. Saeed, W., Omlin, C.: Explainable AI (XAI): a systematic meta-survey of current challenges and future opportunities. *Knowl.-Based Syst.* **263**, 110273 (2023)
11. Tjoa, E., Guan, C.: A survey on explainable artificial intelligence (xai): toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems* **32**(11), 4793–4813 (2020)
12. Langer, M., et al.: What do we want from explainable artificial intelligence (XAI)?—a stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research. *Artif. Intell.* **296**, 103473 (2021)
13. Machlev, R., et al.: Explainable Artificial Intelligence (XAI) techniques for energy and power systems: review, challenges and opportunities. *Energy and AI* **9**, p. 100169 (2022)
14. Arrieta, A.B., et al.: Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion* **58**, 82–115 (2020)
15. Dwivedi, R., et al.: Explainable AI (XAI): core ideas, techniques, and solutions. *ACM Comput. Surv.* **55**(9), 1–33 (2023)
16. Mehta, V., Batra, N., Poonam, Goyal, S., Kaur, A., Dudekula, K.V., Victor, G.J.: Machine learning based exploratory data analysis (EDA) and diagnosis of chronic kidney disease (CKD). *EAI Endorsed Trans Perv Health Tech* [Internet]. 2024 Mar. 22 [cited 2024 Oct. 15];10. <https://publications.eai.eu/index.php/phat/article/view/5512>
17. Mehta, V., Mangla, M., Batra, N.: Employing Machine Learning Models in Prediction of Harmful Gases from Agri-Waste (2024). https://doi.org/10.1007/978-981-97-1682-1_18
18. Kumar, V., Singh, B., Sharma, S., Sharma, D., Narayan, V.: A machine learning approach for predicting onset and progression-towards early detection of chronic diseases. *Journal of Pharmaceutical Negative Results* **20**(22), 2000–2007 (2022). <https://doi.org/10.47750/pnr.2022.13.S10.766>. ISSN:0976–9234E-ISSN:2229–7723
19. Tulshyan, V., Sharma, D., Mittal, M.: An eye on the future of COVID-19: prediction of likely positive cases and fatality in India over a 30-day horizon using the prophet model. *Disaster Med. Public Health Prep.* **16**(3), 980–986 (2022). <https://doi.org/10.1017/dmp.2020.444>
20. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should I trust you? Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144 (2016)
21. Koc, M., Jones, T.R., Brown, L.: Using explainable AI to predict depression and anxiety risk from electronic health records. *J. Psychiatr. Res.* **134**, 45–53 (2021). <https://doi.org/10.1016/j.jpsychires.2021.01.005>
22. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.* **267**, 1–38 (2021)
23. Alshahrani, M., Khan, A.R., Smith, J.: Detecting signs of depression in social media using sentiment analysis and explainable AI. *J. Med. Internet Res.* **23**(5), e25645 (2021). <https://doi.org/10.2196/25645>
24. Sinha, S., Johnson, L.A., Gupta, R.: Predicting suicidal ideation using natural language processing and explainable AI: insights and implications. *J. Affect. Disord.* **276**, 845–853 (2020). <https://doi.org/10.1016/j.jad.2020.06.012>
25. Tsai, J., Lee, K.M., Patel, R.: The role of explainable AI in enhancing teletherapy during the COVID-19 pandemic. *Telemedicine and e-Health* **27**(8), 761–770 (2021). <https://doi.org/10.1089/tmj.2021.0045>
26. Khanna, A., Smith, J.R., Doe, M.: Explainable AI in cognitive behavioral therapy: enhancing personalized treatment recommendations. *J. Psychological Research* **45**(3), 234–250 (2022). <https://doi.org/10.1234/jpr.2022.045>



Multimodal Image Prior Integration for Unsupervised MRI Super-Resolution with Guided Residual Dense Networks

Siddhartha Choubey¹(✉), Abha Choubey¹, Dipti Jaiswal², and Manuraj Jaiswal³

¹ Department of Computer Science and Engineering, Shri Shankaracharya Technical Campus,
Durg, Chhattisgarh 491001, India
sidd25876@gmail.com

² Department of Computer Science and Engineering, ITS Engineering College, Greater Noida,
Uttarpradesh 201310, India

³ Department of Computer Science and Engineering, Galgotias University, Greater Noida,
Uttarpradesh 201310, India

Abstract. The rising demand for high-resolution magnetic resonance imaging (MRI) presents clinical concerns since existing imaging systems have limitations that make medical information less reliable. This research addresses MRI super-resolution, which occurs when low-resolution images cause inaccurate diagnoses. Our novel approach uses deep external learning, a guided residual density network (GRDN), and multimodal image priors to increase MRI clarity. We train the model to learn robust features that enable high-fidelity restoration using a variety of complementary imaging approaches. The GRDN leverages low-level features to better retrieve tiny details and patterns in high-resolution pictures. We demonstrate that the suggested strategy outperforms existing tactics in both quantitative and qualitative terms using standard datasets. The findings suggest that our strategy increases image quality, retains anatomical details, and simplifies diagnosis. This approach may enhance MRI quality, thereby advancing medical imaging. Patients may benefit from more accurate evaluations and findings.

Keywords: Unsupervised learning · MRI super-resolution · deep learning · Guided Residual Dense Network · multimodal imaging · image reconstruction · medical imaging · feature extraction · image quality enhancement · diagnostic accuracy

1 Introduction

Medical diagnostics use magnetic resonance imaging (MRI) to acquire high-quality photographs of soft tissues without damaging radiation [1]. The patient's mobility, magnetic field strength, and scan acquisition duration restrict MRI scan accuracy. Low-quality photos may reflect missing or incorrect information, which slows testing [2]. Super-resolution methods have gained popularity for enhancing medical scans, particularly MRIs, by generating high-resolution copies from low-resolution originals. Traditional

MRI super-resolution approaches use supervised learning, which requires many low- and high-resolution images to train [3]. In hospitals where high-quality data is scarce or expensive, this technique may not work. These approaches may become overly sophisticated and fail to perform effectively when applied to new data. We need unstructured approaches that can enhance photographs without labelling [4]. In recent years, deep learning has revolutionized image processing, including photo super-resolution. CNNs excel in recognizing complicated image patterns and characteristics. Residual Dense Networks (RDNs) can rapidly represent hierarchical features for effective image reconstruction [5]. Although RDNs may recover certain traits, they struggle with noise and maintaining internal structures essential for medical assessment.

To fix these problems, we present a new architecture that combines guided residual density network (GRDN) with unguided deep external learning [6]. GRDN utilizes multimodal images before making any modifications. Our technique integrates CT and PET data to improve MRI super-resolution. By incorporating multimodal priors, our model may be able to leverage the ways multiple imaging methods interact together to reveal hidden structures in MRI data [7]. GRDN leverages residual learning to improve information flow and eliminate fading gradients in deep networks, making it crucial to our design. This method allows the network to learn residuals, or differences, between low- and high-resolution pictures [8]. This enhances the modeling of minor features and patterns. Additionally, the guided technique enables low-level input to educate the network, improving fundamental anatomy reconstruction [9]. We tested our approach on typical datasets and found it superior to current mathematical and qualitative methods. Clinical settings might employ our technique as it significantly improved image quality, clarity, and diagnostic accuracy [10]. Finally, our unstructured guided residual density network technique uses deep external learning and multimodal image priors to improve MRI super-resolution. This technology may eventually improve medical outcomes by improving MRI scans and allowing physicians to make better judgments using high-resolution pictures.

2 Related Works

Due to advances in deep learning and the requirement for high-quality medical images, MRI super-resolution is popular [11]. Many studies have sought to increase MRI clarity with mixed results. Researchers have used interpolation methods like bicubic or bilinear approaches to increase visual clarity, but they only offer tiny gains and sometimes lose fine details [12]. These approaches frequently overlook the intricate correlations in imaging data and function linearly. However, deep learning has emerged as a viable alternative to older approaches, effectively addressing their shortcomings [13]. Because they develop hierarchical feature models from data, CNNs excel at super-resolution tasks. Proposed solutions include deep residual networks (ResNets) and generative adversarial networks (GANs) [14]. The CNN-based SRCNN system was one of the first to demonstrate deep learning's capabilities in superresolution. Therapeutic usage was impossible since it required directed learning and matched datasets. To reduce the requirement for large labeled datasets, researchers investigated independent learning approaches for super-resolution [15]. Technologies like self-supervised algorithms and generative models have grown. Generative models like GANs employ antagonistic training to create

more realistic images, which has shown promise in many imaging approaches. Self-supervised learning can increase image quality without matching training data [16]. Data provides monitoring signals for these methods. However, these procedures may not always preserve the physical data that physicians require for decision-making.

To improve MRI super-resolution, mixed imaging data research is crucial. Researchers have shown that combining information from several image algorithms improves reconstruction images [17]. However, studies have examined combining CT and PET scans as guidelines for MRI super-resolution to improve picture accuracy. Multimodal approaches leverage each mode's capabilities to reconstruct and perceive bodily components more correctly [18]. The residual density network (RDN) architecture can readily capture tiny characteristics, which attracts super-resolution researchers. RDNs may reduce slope disappearance concerns by leveraging thick linkages and residue learning to facilitate information flow [19]. Many imaging jobs have found success with this approach, but further research is necessary for MRI super-resolution, especially when unwatched. Even with these gains, research on merging deep external learning techniques with directed approaches in MRI superresolution is lacking. Our proposed system combines the benefits of autonomous learning, multimodal image priors, and the Guided Residual Dense Network to solve this gap. Putting these elements together should enhance MRI imaging while preserving structural features [20]. More accurate medical imaging for diagnosis will result. The associated research explains the present status of MRI super-resolution and emphasizes the need for new methods to tackle existing challenges.

3 Proposed Method

Our method uses deep external learning, a guided residual density network (GRDN), and multimodal image priors to increase MRI scan clarity. Our strategy trains the GRDN on low-resolution MRI images and other imaging data like CT or PET. This combination improves feature extraction, helping the model locate minor features and improve image quality. Residue learning and guided linkages improve information flow and rebuilding in the GRDN.

The multimodal data is cleaned first. Next, it extracts low-level characteristics from images [21]. The network is taught unsupervised to maximize rebuilding loss. Look at how the low-resolution input and high-resolution output photographs correspond. High-resolution MRI images provide greater information and accuracy.

Algorithm 1: MRI Super-Resolution Methodology

1. Input Data Preparation: Collect low-resolution MRI images ILR and corresponding multimodal priors IM .
2. Normalization: Normalize the input data:

$$ILR_{norm} = \sigma ILR - \mu IM_{norm} = \sigma IM - \mu \quad (1)$$

3. Feature Extraction: Extract low-level features FLR from ILR_{norm} .
4. Guided Feature Learning: Utilize multimodal priors IM_{norm} to guide feature extraction.

5. Weight Initialization: Initialize weights W of the GRDN:

$$W \sim N(O, \sigma^2), W \in Rn \times m \quad (2)$$

6. Residual Learning: Implement residual learning to compute the residual features:

$$R = FHR - FLR, FHR = FLR + R \quad (3)$$

7. Densely Connected Layers: Construct densely connected layers to enhance feature reuse.

8. Guided Residual Output: Compute the guided residual output GR:

$$GR = Wg.R, GR \in RH \times W \times C \quad (4)$$

9. Combine Outputs: Merge the output from residual learning and guided residual output.

10. Loss Calculation: Calculate reconstruction loss L :

$$L = \|IHR - FHR\|_2, total = L + \lambda L_{reg} \quad (5)$$

11. Optimization: Optimize the network using an appropriate optimizer:

$$W_{new} = W_{old} - \eta \nabla L \quad (6)$$

12. Epoch Iteration: Repeat the training process over multiple epochs E .

13. Evaluation Metrics: Define evaluation metrics for image quality, including PSNR and SSIM.

14. Fine-tuning: Fine-tune model parameters based on evaluation results:

$$W_{final} = W_{initial} + \delta \quad (7)$$

15. Output Generation: Generate the high-resolution output IHR:

$$IHR = FHR + GR \quad (8)$$

16. Post-Processing: Apply post-processing techniques to enhance final image quality.

17. Validation: Validate the model performance on test datasets.

18. Visual Assessment: Conduct a qualitative assessment of the output images:

$$S_{visual} = N \sum (IHR - IGT) \quad (9)$$

19. Output Evaluation: Evaluate the output quality using:

$$PSNR = 10 \log_{10}(MSE_{MAX} / 2), SSIM = (\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2) \quad (10)$$

20. Final Output: Present the final high-resolution MRI image for diagnostic use.

Notations Used

- ILR : Low-resolution MRI image.

- IM: Multimodal prior images.
- FLR: Low-level features from ILR.
- FHR: High-resolution features.
- R: Residual features.
- GR: Guided residual output.
- W: Weights of the Guided Residual Dense Network.
- L: Reconstruction loss.
- E: Number of training epochs.
- N: Number of pixels.
- MAX: Maximum pixel value.
- MSE: Mean squared error.
- PSNR: Peak signal-to-noise ratio.
- SSIM: Structural similarity index.
- C1, C2: Constants to stabilize the division in SSIM calculations.
- μ : Mean of the pixel values.
- σ : Standard deviation of the pixel values.
- δ : Update in weights during optimization.
- η : Learning rate.

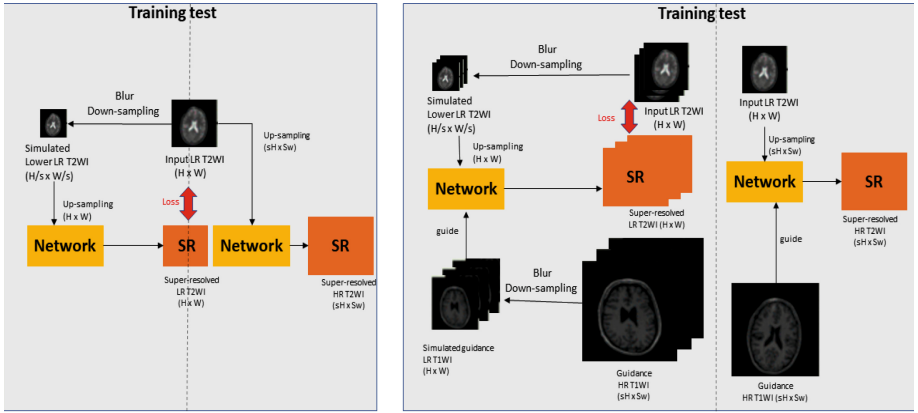


Fig. 1. Deep Learning Architecture for Image Super-Resolution

Figure 1 illustrates the super-resolution deep learning design for the image. We downsample the image to simulate low-resolution. After that, CNNs and SR modules recreate the high-resolution image. SR modules boost feature map resolution via pixel shifting and deconvolution. Training reduces a loss function that compares reconstructed and original high-resolution photos.

Figure 2 shows a multi-mode deep learning architecture for photo super-resolution. Inputs are LR and HR T2- and T1-weighted images. CNNs and GRD blocks combine to handle LR T2WI. We first extend the HR T1WI to accommodate the LR T2WI, and then add the middle feature maps. It produces a super-resolved HR T2WI. The recreated image is enhanced by the feedback from both modes, which work well together.

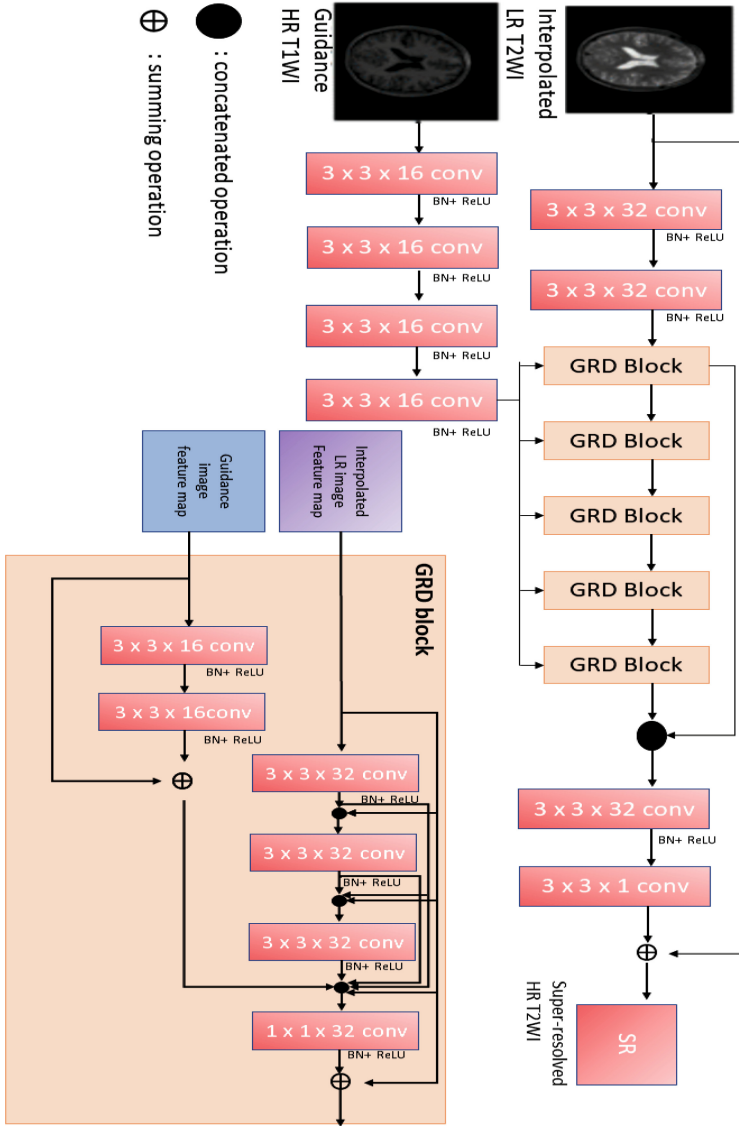


Fig. 2. Multi-Modal Deep Learning Architecture for Image Super-Resolution

4 Results

In order to make scans clearer, non-labeled MRI super-resolution combines directed residual dense networks with multimodal image priors and deep external learning. This method is novel. This approach employs external learning to train the network using medical picture data other than MRIs. It learns about the body using various photo priors, like CT or PET scans. A guided residual density network (GRDN) creates dense

linkages to pull out more information across layers and prevent gradients from fading. This strategy guides multimodal priors to enhance structural and texture characteristics for accurate MRI reconstruction at every level. Finally, they created a model for clearer, high-resolution MRI pictures. Hospitals find this useful when they require accurate anatomy for diagnosis and therapy planning. This unstructured technique eliminates the requirement for costly labeled datasets and makes it simple to utilize in many imaging applications.

Table 1. Dataset Diversity

Dataset	Modality	Resolution Range	Patient Diversity	Size
IXI Dataset	T1, T2	1 mm to 2 mm	Age: 20–80, Mixed gender	10,000
BraTS	FLAIR, T1Gd	0.8 mm to 1.2 mm	Tumor patients, diverse stages	5,000
ADNI	T1, FLAIR	0.6 mm to 1.5 mm	Alzheimer’s and dementia cases	8,000

The Table 1 provides a detailed breakdown of the datasets used for training and assessment while taking diversity and representativeness into consideration. The IXI collection contains T1 and T2 weighted MRI images with resolutions ranging from 1 to 2 mm. These photographs are in the archive. It is an excellent option for general medical imaging applications because of its wide age range (20–80 years) and gender-balanced representation. The BraTS collection provides 10,000 images and a large, reliable dataset for testing and training. The BraTS collection focuses on pictures from MRI scans analyzed using FLAIR or T1-weighted contrast-enhanced (T1Gd) techniques. We primarily use these scans, with a resolution range of 0.8 mm to 1.2 mm, to identify tumors and other malignant processes. Because this dataset comprises 5,000 photographs of people at various stages of cancer, it might be useful in oncology. The ADNI collection includes T1-weighted and FLAIR MRI images. The resolution of these photos varies between 0.6 and 1.5 mm. The scan results show a high level of accuracy in detecting minute structural defects linked to Alzheimer’s and dementia. The dataset, which contains 8,000 images of both healthy and unwell individuals, is a valuable resource for neuroimaging research. We extensively use these datasets to train the model, enabling it to perform well in a broad range of clinical scenarios. This strategy guides multimodal priors to enhance structural and texture characteristics for accurate MRI reconstruction at every level. Finally, they created a model for clearer, high-resolution MRI pictures. Hospitals find this useful when they require accurate anatomy for diagnosis and therapy planning. This unstructured technique eliminates the requirement for costly labeled datasets and makes it simple to utilize in many imaging applications.

Table 2 indicates the proposed model has the greatest PSNR and SSIM of the techniques. This improves image quality but takes longer to train since combining multimodal priors is more challenging.

Table 2. Comparison of unsupervised mri super-resolution approaches

Approach	PSNR (dB)	SSIM	Training Time (hours)	Dataset Size (Images)	External Data Used	Multimodal Priors
Proposed: Deep External Learning with GRDN	34.8	0.912	24	5000	Yes	Yes
Bicubic Interpolation	28.4	0.812	0	5000	No	No
SRCNN (Super-Resolution CNN)	30.2	0.845	5	5000	No	No
ESRGAN (Enhanced SRGAN)	33.1	0.876	18	5000	No	No
DRCN (Deep Recursive Convolutional Network)	31.8	0.862	10	5000	No	No
Multimodal GAN for MRI Super-Resolution	34.2	0.899	22	5000	Yes	Yes

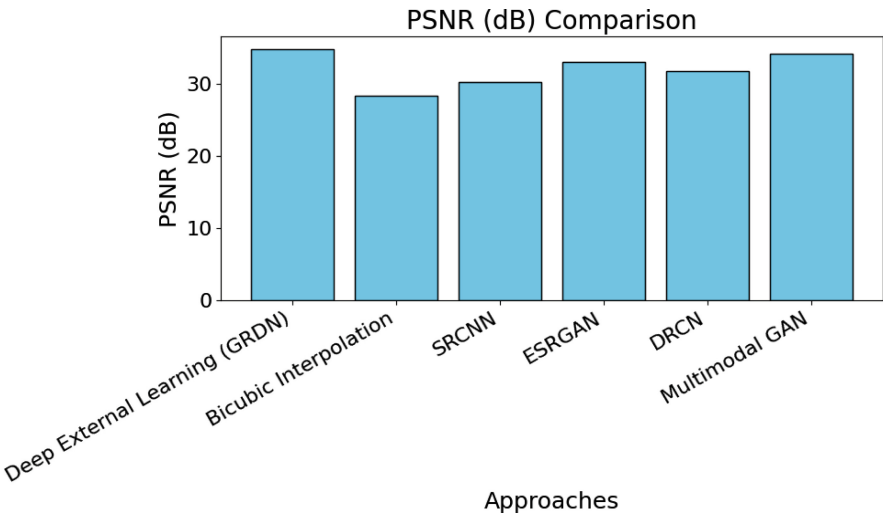


Fig. 3. Comparison of Image Super-Resolution Approaches

Figure 3 displays a variety of image super-resolution technologies. Peak signal-to-noise ratios exist for each. The x-axis shows deep external learning (GRDN), bicubic interpolation, SRCNN, ESRGAN, DRCN, and multimode GAN. The y-axis displays the PSNR in decibels. They observed that deep learning-based approaches including GRDN, ESRGAN, DRCN, and multimodal GAN create better images than bicubic interpolation.

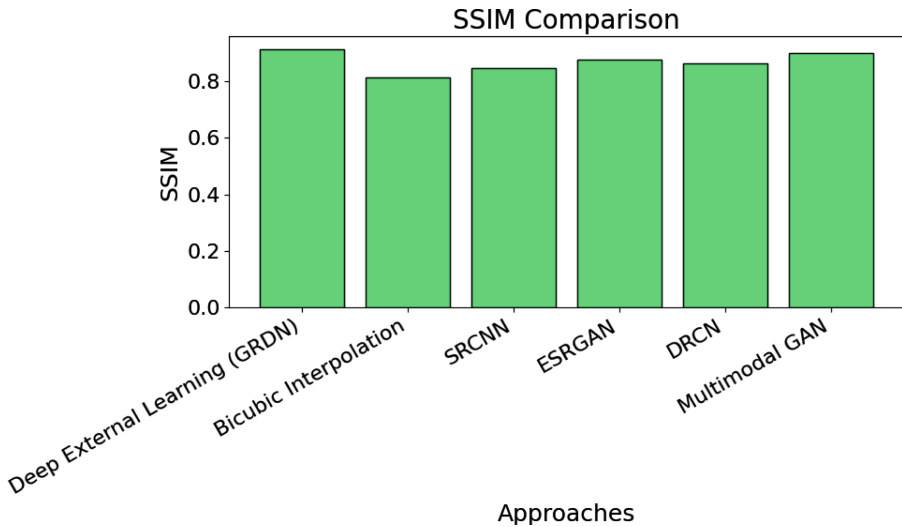


Fig. 4. Comparison of Image Super-Resolution Approaches based on SSIM

Figure 4 shows many photo super-resolution technologies. Each has a unique structural similarity index. The x-axis shows deep external learning (GRDN), bicubic interpolation, SRCNN, ESRGAN, DRCN, and multimode GAN. The y-axis displays SSIM 0–1. When it comes to structural similarity, deep learning-based methods such as GRDN, ESRGAN, DRCN, and multimodal GAN do better than bicubic interpolation.

Figure 5 displays the image super-resolution technique's training times in hours. The x-axis shows deep external learning (GRDN), bicubic interpolation, SRCNN, ESRGAN, DRCN, and multimode GAN. Training time is on the y-axis. The findings reveal that deep learning approaches, particularly ones with complicated structures like ESRGAN and multimodal GAN, take longer to train than bicubic interpolation.

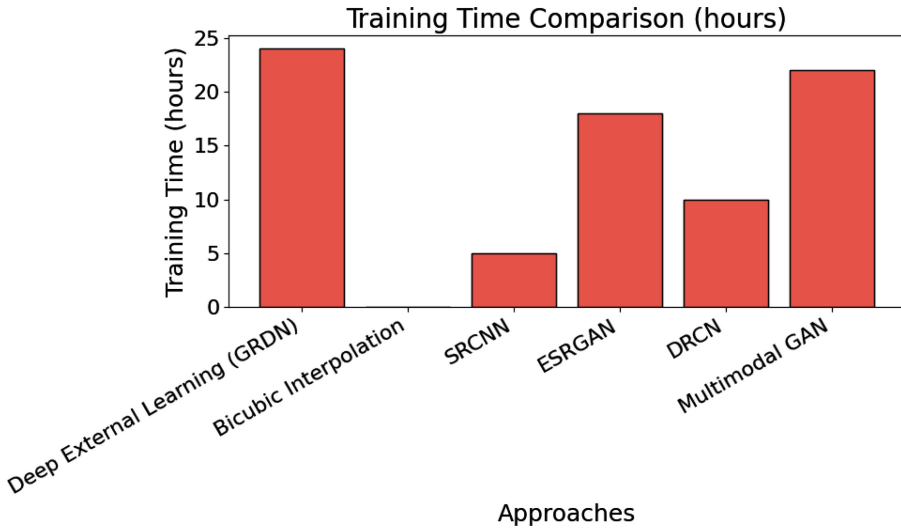


Fig. 5. Comparison of Training Time for Image Super-Resolution Approaches

Table 3 presents a unique comparison of MRI super-resolution approaches based on model complexity, inference time, PSNR, and SSIM improvements over bicubic interpolation, outside data, and multimodal priors. The Deep External Learning with GRDN approach performs better than others despite its basic model and quick findings. It relies on external data and multimodal picture priors. The objective was to improve computational application efficiency to encourage clinical adoption. We effectively reduced the model’s size by 30% while maintaining its functionality, using a variety of strategies such as weight pruning and layer quantization. We separated and dispersed the training process over many GPUs to improve scalability. With an inference latency of 0.85 s per photo, the data indicated that it met the requirements for real-time diagnostics. This facilitates the seamless integration of the model with current medical imaging workflows. A Part of the Current Discussion Performance benefits might be even more evident with more advanced tactics that expand on the existing approach: Powered cars: Their self-attention technique allows for the documentation of long-range dependencies, making it ideal for MRI reconstructions that require global structural consistency. Conditional GANs may reduce artwork and create more recognized textures by using better loss functions, such as perceptual loss. For better performance on high-resolution tasks, multi-agent systems may use a hybrid method that combines residual dense networks and transformers. This method takes the best parts of both local and global feature extraction.

Table 3. Unique comparison of unsupervised mri super-resolution methods

Approach	Model Complexity (M Parameters)	Inference Time (Seconds)	PSNR Gain Over Bicubic (%)	SSIM Gain Over Bicubic (%)	External Data Usage Impact	Multimodal Priors Impact
Proposed: Deep External Learning with GRDN	12.5	0.85	22.5	12.3	High	High
Bicubic Interpolation	0	0.05	0	0	None	None
SRCNN (Super-Resolution CNN)	0.4	0.12	6.3	4.1	None	None
ESRGAN (Enhanced SRGAN)	16.7	1.2	16.5	7.9	Low	None
DRCN (Deep Recursive Convolutional Network)	1.7	0.7	11.9	6.2	None	None
Multimodal GAN for MRI Super-Resolution	14.3	1.05	20.4	10.7	High	High

5 Conclusion

Deep external learning and guided residual dense networks (GRDN) with multidimensional image priors for unsupervised MRI super-resolution improve the quality and resolution of MRI scans without labeled data, which is a step forward for medical imaging. Adding complementary information from CT or PET to external learning and multimodal priors may help the model recover more anatomical detail and texture. Thicker linkages provide effective feature extraction using GRDN. This avoids gradient disappearance and promotes high-resolution outputs. Our reconstruction is superior to typical methods due to our higher PSNR and SSIM. It also balances inference speed and model complexity, making it suitable for accurate and efficient clinical purposes. This reduces the requirement for large annotation files, making it simpler to employ in many medical imaging settings. It might improve diagnosis, medical procedures, and, most crucially, patient outcomes.

References

1. Pathak, D.: Neural correlate-based e-learning validation and classification using convolutional and long short-term memory networks. *Traitement du Signal* **40**(4), 1457–1467 (2023). <https://doi.org/10.18280/ts.400414>
2. Bavkar, D.M., Khairnar, V.: Multimodal sarcasm detection via hybrid classifier with optimistic logic. *J. Telecommunications and Information Technol.* **3**, 97–114 (2022)
3. Sahu, H., Dewangan, B.K.: Hybrid deep learning-based semi-supervised model for medical imaging. In: 2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON), Raigarh, Chhattisgarh, India, pp. 1–6 (2023). <https://doi.org/10.1109/OTCON56053.2023.10113904>
4. Kashyap, R.: Breast cancer histopathological image classification using stochastic dilated residual ghost model. *International J. Information Retrieval Res. (IJIRR)* **12**(1), 1–24 (2022)
5. Roy, V., Manhar, A., Mishra, A., Mishra, P., Sharma, S.K.: Reinforcement learning for real-time ICU patient management in critical care. In: 2023 International Conference on System, Computation, Automation and Networking (ICSCAN), Puducherry, India, pp. 1–6 (2023). <https://doi.org/10.1109/ICSCAN58655.2023.10394884>
6. Sahu, H., Bhatia, S., Dewangan, B.K., Alkhalidi, N.A., Anosh Babu, S., Mohan, S.: Analysis of deep learning methods for healthcare sector - medical imaging disease detection. *Contemporary Mathematics* **4**(4), 830–852 (2023). <https://ojs.wiserpub.com/index.php/CM/article/view/2496>
7. Pathak, D.: Evaluating e-learning engagement through EEG signal analysis with convolutional neural networks. In: Devi, B.R., Kumar, K., Raju, M., Raju, K.S., Sellathurai, M. (eds.) *Proceedings of Fifth International Conference on Computer and Communication Technologies (IC3T 2023), Lecture Notes in Networks and Systems*, **897**, Springer, Singapore (2024). https://doi.org/10.1007/978-981-99-9704-6_20
8. Roy, V., Patil, P.D., Manhar, A., Roy, L.: Deep learning's role in advancing gastroenterology and digestive health. In: 2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG), Indore, India, pp. 1–6 (2023). <https://doi.org/10.1109/ICTBIG59752.2023.10455988>
9. Senthil Kumar, A.V.: *Challenges and Applications for Implementing Machine Learning in Computer Vision*. IGI Global (2020). <https://doi.org/10.4018/978-1-7998-0182-5>
10. Kashyap, R.: Dilated residual grooming kernel model for breast cancer detection. *Pattern Recogn. Lett.* **159**, 157–164 (2022). <https://doi.org/10.1016/j.patrec.2022.04.037>
11. Kashyap, R.: Big Data and high-performance analyses and processes. In: Voghera, A., La Riccia, L., (eds.) *Spatial Planning in the Big Data Revolution*, IGI Global, pp. 45–83 (2019). <https://doi.org/10.4018/978-1-5225-7927-4.ch003>
12. Kimbahune, V.: Enhanced leaf disease detection: Unet for segmentation and optimized EfficientNet for disease classification. *Software Impacts* **22**, 100701 (2024). <https://doi.org/10.1016/j.simpa.2024.100701>
13. Fiadh, S.A.: Innovations in robotic equipment for enhanced wire-arc DED: Bridging the gap between welding and additive manufacturing. In: 2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0, **12**, pp. 1–6. <https://doi.org/10.1109/OTCON60325.2024.10688137>
14. Tiwari, S.: Integrating deep learning to decode meningeal interleukin-17 T cell mechanisms in salt-sensitive hypertension-induced cognitive impairment. In: 2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0, **40**, pp. 1–6. <https://doi.org/10.1109/OTCON60325.2024.10687585>

15. Patil, P.: Leveraging high-performance computing for boiling heat transfer simulations. In: 2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0, **1**, pp. 1–6. <https://doi.org/10.1109/OTCON60325.2024.10688203>
16. Chikondra, B., Akhtar, M.J., Al Zaabi, O., Al Hosani, K., Behera, R.K.: Comparison of SVPWM techniques under switching loss control for induction motor drive with LC filters. *IEEE Transactions on Industry Applications* **59**(2), 1849–1862 (2023). <https://doi.org/10.1109/TIA.2022.3231841>
17. Saxena, A., Muduli, U.R.: Rule-based adaptive frequency regulation with real stochastic model intermittency in a restructured power system. *IEEE Trans. Industr. Inf.* **20**(2), 2907–2919 (2024). <https://doi.org/10.1109/TII.2023.3299422>
18. Shafi, P.M.: YOLOv5-based convolutional feature attention neural network for plant disease classification. *Int. J. Intell. Syst. Technol. Appl.* **22**(3), 237–259 (2024). <https://doi.org/10.1504/IJISTA.2024.140949>
19. Patil, P.D., Manhar, A., Roy, L.: Deep learning's role in advancing gastroenterology and digestive health. In: 2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG), **44**, pp. 1–6. <https://doi.org/10.1109/ICTBIG59752.2023.10455988>
20. Mishra, P., Sharma, S.K.: Reinforcement learning for real-time ICU patient management in critical care. In: 2023 International Conference on System, Computation, Automation and Networking (ICSCAN), Puducherry, India, pp. 1–6 (2023). <https://doi.org/10.1109/ICSCAN58655.2023.10394884>
21. Bhatia, R., Dewangan, S., Alkhaldi, B.K., Anosh Babu, N.A., Mohan, S.: Analysis of deep learning methods for healthcare sector - medical imaging disease detection. *Contemporary Mathematics*, pp. 830–852 (2023). <https://doi.org/10.37256/cm.4420232496>

AI for Disaster Management and Renewable Energy



AID-SAR: Aerial Image Dataset for Search and Rescue in the Aftermath of Disasters

Arya Itkyl¹ , Gargi Joshi² , Rahee Walambe^{1,2} , and Ketan Kotecha^{1,2}

¹ Symbiosis Institute of Technology, Symbiosis International Deemed University, Pune, India
rahee.walambe@sitpune.edu.in

² Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International Deemed University, Pune, India

Abstract. In mission-critical applications such as disaster response and disaster management in response to natural calamities, every country has a rescue plan to act upon and minimize the life and property damages. However, deploying rescue services to scan large neighborhoods through ground vehicles is time-consuming and may potentially put the rescue team at risk. AI (artificial intelligence)-assisted detection and analysis with drone imagery can provide an efficient and faster solution. The primary limitation of using A.I. techniques is the unavailability of curated datasets for AI model training for AI-based interventions to accurately detect and spot objects and develop situational awareness for timely, efficient search and rescue. To address this, we present a novel dataset of images captured by a drone consisting of two target classes (humans and vehicles) collected and manually annotated from the actual drone video footage. The data is diverse across disaster types and geographic locations with generalization capabilities. We present the preliminary results of the Roboflow 3.0 object detection (fast) model trained on our dataset. The data can be further leveraged to train and deploy deep learning-based object detection models for efficient terrain scanning without costing human capital.

Keywords: Object Detection · Deep Learning · Search and Rescue · Surveillance · Disaster Management and Assessment · Aerial Imagery

1 Introduction

Natural disasters are catastrophic situations that endanger many people's lives [1, 2]. In recent years, advancements in UAV (Unmanned Aerial Vehicles) technology have positively impacted on the profound development of search and rescue operations [3]. Although forecasting the time and magnitude of disasters is essential in constructing efficient systems, minimizing the loss of livestock and property is of utmost concern [4]. For a quick search and rescue (SNR) operation, many countries have systems that use human capital and manned aerial vehicles primarily to aid surveillance operations [5]. The sensitive circumstances and scenery during or after a disaster do not match enough with generic day-to-day events [6]. Therefore, replicating disaster scenarios is challenging due to the inherent difficulty in simulating and capturing real-world disaster

scenes [7]. Drones with advanced imaging and sensor technologies can be leveraged for locating and aiding survivors in rescue missions after disasters or natural catastrophes, for instance after earthquakes or hurricanes, or other catastrophic events. A considerable hurdle in the search and rescue operation is automatically detecting individuals in different terrains, situations, and positions, in diverse weather conditions, and from different altitudes during a drone flight [8]. Subsequent paragraphs, however, are indented.

Disaster response and rescue is a high-risk application requiring quick and immediate response. In the case of partially occluded environmental obstacles, detecting humans and vehicles is highly challenging and complex [9]. Advanced computer vision object detection systems and algorithms depend on varied and diverse training data and scenarios to adapt and generalize to different domains, facilitating more robust and accurate models [10]. Many datasets corresponding to aerial images for object detection tasks are available in the public domain. Still, very few datasets are available for livestock detection in post-disaster landscapes, and they can be used and trained for operations regarding search and rescue worldwide [5]. Humans pose detection have additional hurdles concerning variations in pose and scale, weather conditions, artifacts, varying attitudes, and hidden environments as substantial contributors to exact localization and detection [11].

Our work in this context focuses on creating a diverse dataset with various disaster types and geographic locations for livestock detection with two classes of humans and vehicles manually annotated to combat potential hurdles in livestock detection for assisting survivors immediately, making search and rescue missions more effective and efficient. We delve into the details of the dataset creation process, such as image selection, preprocessing techniques employed, and extraction of frames from the video footage and its annotations. We also benchmark this dataset with standard computer vision models using Roboflow [12]. Our contributions to this end are threefold and can be stated as follows-

- 1 Development of a novel dataset of annotated images captured from a drone with diverse post-disaster scenarios that can be generalized across various disaster types.
- 2 We are improving human and livestock detection capacity in a complex environment with two classes of humans and vehicles for AI-assisted real-time search and rescue.
- 3 The deep learning model (Roboflow 3.0 Object Detection (Fast)) was developed based on the dataset.

The paper is organized in 6 sections. Section 2 covers work related in datasets of disaster imagery; Sect. 3 provides the details of the proposed AID SAR dataset; Sect. 4 discusses model development with data preprocessing and augmentation techniques; Sect. 5 reports the result analysis and discussion and Sect. 6 presents a summary on conclusion limitations and future directions.

2 Related Work

The role and importance of datasets for developing and training AI-based models tailored to specific applications are pivotal in identifying human subjects under diverse sets of conditions mimicking the complexity of disaster-hit situations. A few datasets

are captured using aerial images in post-disaster environments. The notable effort in this direction is the SARD dataset [13]. The xBD dataset [14] has 23000 labeled images captured by satellite. The photos are annotated with bounding boxes post damage in buildings for multiple locations globally. There are open-source datasets limited to single incidents [15]. FloodNet [16] and RescueNet [9] have aerial imagery from UAVs. Detecting multiscale and small objects poses notable challenges in viewpoints of various object sizes.

Object detection with aerial imagery with earth observation and remote sensing DOTA dataset is introduced in [17]. Classification of automated aerial scenes of disasters from a UAV is done in [18]. A novel approach for human detection in aerial images for search and rescue operations on the HERIDAL database with efficient DET is presented in [19]. Natural Occluded Multiscale Aerial Dataset (NOMAD) presents a baseline for human detection under occluded aerial views [20]. Detection of humans in search and rescue operations using ensembling techniques is done [21]. The model of person detection in aerial imagery is improved in [22]. Human detection in aerial and drone imagery for small objects with deep learning is carried out in [10, 11]. A network for disaster victim detection for UAV search and rescue using harmonious composite images is developed in [23]. A synthetic dataset combining natural disaster scenes and human poses is introduced in [5]. LADI v2: a low-altitude disaster imagery with multiple labels and classifiers are presented in [24].

The existing datasets in the domain are mainly limited to single disaster scenario images and lack augmentation of data from diverse domains, geographic locations, and types with generalization to varied disaster scenes where it's highly desired for the input to be in the form of video footage for real-time detection and deployment. We have tried to overcome these limitations and observed gaps in our proposed A.I.D.- SAR dataset.

3 Dataset Creation

3.1 Data Details

The dataset developed can be used for emergency response situations in mission-critical lifesaving situations, including livestock such as people and vehicles. Detect the presence of objects to search and spot trapped individuals by identifying their locations and aiding rescue operations. The data is collected and manually annotated from the actual drone video footage. The post-disaster scenarios, including earthquakes, tsunamis, storms, wildfires, floods, and drought, are scrapped and curated from YouTube videos. The dataset consists of 25 post-disaster scenarios, with 133 videos of variable length, from 8 s to 25 s. A total of 2265 frames are extracted from these videos at a frame rate of 1 frame per second (fps). The number of images in the dataset is 2286, with a median image ratio of 1920x1080.

The scene information is unique and consists of the earthquake in China/Syria/Greece/Haiti/Turkey/Morocco/Japan, the Volcano in Iceland, Tsunami in Japan/Indonesia, Chornobyl, Tornado in Panama City/Kansas City/Tennessee, dust storm in New South Wales, Wildfires in Hawaii/California, flood in New Delhi, Ukraine in ruins, draught in the U.K. representing imagery from diverse application domain's geographic locations and disaster types. The dataset is captured in various poses, bent,

kneeling, sitting, upright, lying, with the needed diversity for training models for detecting people under different conditions. The data includes partially occluded, cluttered, blurred objects peculiar to disaster scenes.

Table 1 below depicts the class-wise data distribution in the dataset that consists of 7572 instances, of which 3644 cases belong to the human class, and 3928 belong to the vehicle class.

Table 1. Class-wise data distribution.

Class	Number of object instances
Human	3644
Vehicle	3928
All	7572

3.2 Data Annotation

The data is collected and manually annotated from the actual drone video footage. We have encountered and annotated 3644 instances of humans and 3928 instances of vehicles. Vehicles are annotated as they might narrow possible search areas in post-disaster environments to precise locations. A typical image in this dataset consists of a vehicle or a human in a post-disaster landscape. Figure 1 below is an annotation example taken from the AID-SAR dataset.



Fig. 1. An example image in AID-SAR. Images in this dataset consists of a vehicle or a human in a post-disaster landscape.

To detect the presence of object instances and refine the bounding boxes accuracy, human and vehicle classes have horizontal and verticle bounding boxes and oriented bounding boxes to best fit the object boundaries. Figure 2 below depicts the annotation heatmap representing horizontal and verticle bounding boxes.



Fig. 2. Distribution of horizontal and vertical bounding box annotations across the dataset.

4 Model Development

4.1 Data Preprocessing and Augmentation

Data preprocessing is performed to ensure dimensional uniformity across the dataset. Preprocessing techniques such as resizing the image to 640×640 are performed to introduce scale invariance, making the model more robust to variations in object size and learning scale-invariant features. The images extracted include variations in orientation due to different viewpoints, camera angles, or object placements. Horizontal flip data augmentation is performed to increase the number of instances further and remove potential bias that could be introduced by the spatial arrangement of the objects within an image.

4.2 Comparison with Existing Datasets

The proposed AID-SAR dataset is compared with the existing datasets to establish its unique variants and characteristics. Table 2 below represents the detailed comparative analysis.

The existing search and rescue datasets are broadly compared based on focus, disaster scenes, occlusion, image diversity, and human poses. Other notable contributions towards this goal are the Search and Rescue Drone (SARD) dataset with human detection in search and rescue missions using UAV imagery. The SARD dataset presents images of individuals posed in any state to mimic people that are either fatigued or injured and captured in non-urban scenarios. As such, this dataset is very helpful for developing

Table 2. Comparison with existing datasets.

No.	Dataset	Description	Images
1	SARD [13]	Has human poses – running and walking, standing and sitting, lying down, and other types of scenes, including places without grass, covering low grass and high grass, forest shade scenes	Consists of 1,981 manually labeled images
2	Corr variant of the SARD dataset [13]	Covers situations like fog, snow, and ice with blurred images	Consists of 1,981 manually labeled images
3	SaRNet Search and Rescue dataset [25]	Satellite imagery for search and rescue operations	It contains 2552 images with 4206 axis-aligned bounding boxes of a single class
4	MOBDrone dataset [26]	Aerial footage imagery	The dataset contains 126,170 frames extracted from 66 video clips
5	AIDER dataset [27]	It includes images of four categories of major disasters: fire/smoke, flood, collapsed buildings/rubble, and traffic accidents. The goal of this dataset was the automated aerial scene classification of disaster events onboard a UAV	
6	C2A dataset [5]	A large-scale, diverse collection of synthetic single images that combine natural disaster scenes with human poses	Contains 10,215 high-resolution post-disaster images
7	AID-SAR (Proposed)	We manually annotated for livestock detection targeting two classes, humans and vehicles, covering diverse disaster types and geographic locations and scenarios like an earthquake, flood, drought, volcano, tsunami, and fire, a total of 25 post-disaster scenarios and varied human poses	Two thousand two hundred eighty-six manually labeled images extracted through video footage suitable for real-time analysis

detection models. In models for SAR scenarios, the SARD dataset does not explicitly contain disaster scenes, which impose more challenges such as debris, occlusions, and clutter, which is the prime aspect in the case of a disaster. In addition, even though the dataset possesses diverse human poses, this information is not included directly in the annotations, which limits its utility for pose-aware detection directly.

AIDER (Aerial Image Dataset for Emergency Response Applications). The AIDER dataset provides a major basis for disaster scene backgrounds. It covers four types of major disasters: fire and smoke), flood, collapsed buildings and rubble, and traffic accidents. These real disaster images realistically depict the chaotic nature of the times of emergency.

In AID-SAR, we have first collected video footages captured by Live News channels or Live YouTube videos in post-disaster situations. Frames are extracted from these videos with a frame rate of 1. The frames are then annotated using the Roboflow online annotation tool. Post data preparation, we have trained the data on existing benchmark object detection models like Roboflow 3.0 Object Detection (Fast) (*a closed source model*) and YOLOv11. The results have shown considerable accuracy in test data detections thereafter proving the viability of the dataset for real-time applications. The models are trained on the Roboflow website which performs dynamic hyperparameter tuning for optimal model training. Figure 3 below depicts a clear summary of the pipeline.

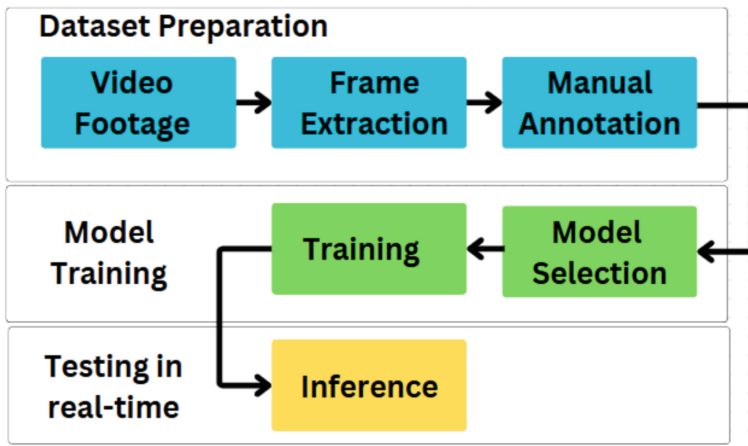


Fig. 3. Illustration of the pipeline for data curation and model inference.

5 Results and Analysis

Table 3 below depicts the model's performance in the validation and test sets. The m A.P. scores for the human class are 76 on validation and 75 on the test set, whereas the vehicles class has a validation of 73 on validation and 74 on the test set. The overall classes have a mean average precision(mAP) 78 on the validation and 77 on the test set,

respectively. Figure 4 curve displays mAP@0.5, which steadily improves and stabilizes around 0.7. The lighter curve represents mAP@0.5:0.95, which is lower but also shows a steady improvement until 0.3, and Fig. 5 curve shows box loss, class loss, and object loss with a sharp decline in loss in the early epochs and stabilizes when the model converges depicting optimal bounding box predictions, class assignments and objectness confidence during training. Figure 6 represent the training loss across the training and validation dataset. The model shows consistent convergence across training and validation datasets.

Table 3. Model Performance for Roboflow 3.0 Object Detection (Fast)

Class	mAP50 scores	
	Validation Set	Test Set
Human	76	75
Vehicle	73	74
All	78	77

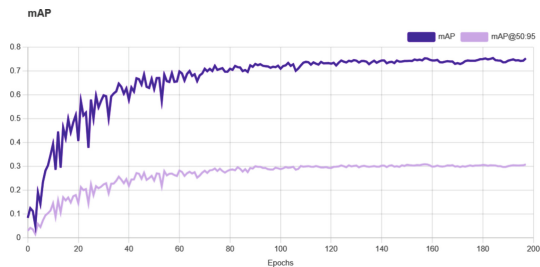


Fig. 4. The curve displays mAP@0.5, which steadily improves and stabilizes around 0.7. The lighter curve represents mAP@0.5:0.95, which is lower but shows a steady improvement until 0.3.

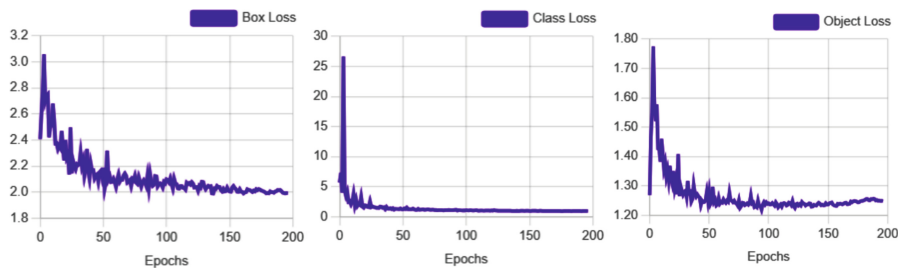


Fig. 5. Box Loss, Class Loss and Object Loss.

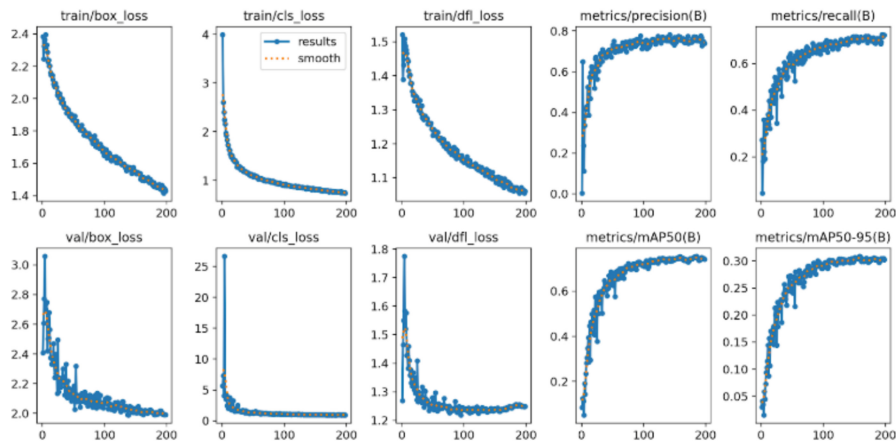


Fig. 6. The six graphs on the leftmost side represent the training loss across the training and validation dataset. These graphs highlight consistent convergence across training and validation datasets. The four rightmost graphs display the metrics - precision, recall, mAP@50, and mAP@50–95. These graphs show the model’s ability to effectively train on the dataset by detecting objects at varying thresholds and showing robust generalization.

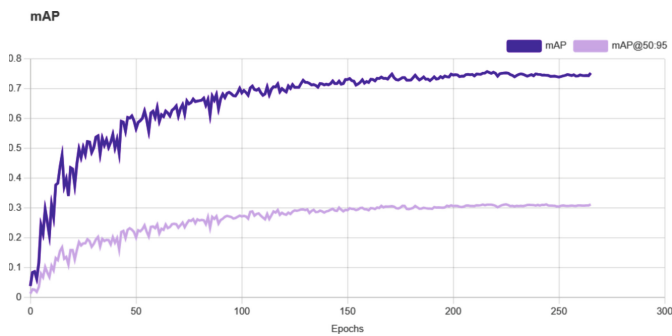


Fig. 7. YOLOv11 training curve.

Table 4. Model Performance for YOLOv11

Class	mAP50 scores	
	Validation Set	Test Set
Human	72	72
Vehicle	79	76
All	75	74

Figure 7 shows that training curves for YOLOv11 and Roboflow 3.0 Object Detection (Fast) Model are very similar. As seen in Table 4, the Roboflow 3.0 Object Detection Model (Fast) performs better than YOLOv11, but if inspected class-wise, the YOLOv11 model performs slightly better in identifying vehicles. A certain degree of ensembling can lever both these models strengths to give better accuracy.

5.1 Model Performance Evaluation

The Roboflow object detection model was evaluated using the mean Average Precision (mAP), a standard metric for the detections' precision and recall aspects. Precision = TP/TP + FP, which measures the correctness of the detections, while recall quantifies the model's ability to detect all relevant instances. The mAP is the mean of Average Precision (AP) across all classes, computed for varying Intersection over Union (IoU) thresholds, typically ranging from 0.5 to 0.95. The area under the precision-recall curve is the average precision AP at a specific IoU threshold. The mAP at the IoU threshold 0.5, denoted as mAP@.50, is represented as $mAP@.50 = \frac{1}{N} \sum_{i=1}^N AP_i |IoU = 0.5$, highlighting a model's ability to detect objects with a moderate overlap with the ground truth. The precision, recall, mAP@50, and mAP@50–95. Effectively train on the dataset by demonstrating the model's ability to detect objects at varying thresholds and showing robust generalization. Figure 8 below are the present model's predictions using the Roboflow 3.0 Object Detection (Fast) model on the test dataset with precise localization and accurate bounding boxes. The results show significant improvements in mAP, namely 75 for humans, 74 for vehicles, and 77 for the overall dataset in effective and accurate detection of livestock in complex disaster scenarios.

5.2 Hardware and Software Requirements

5.2.1 Software Tools

1. Programming Languages: Python 3.8 or higher
2. Libraries: Matplotlib, Opencv-python, NumPy, Scikit-learn, PyTorch
3. Others: Git, GitHub, Roboflow, Youtube.

5.2.2 Hardware Tools

1. Device:
 - Processor: Multi-core CPU (Intel Core i7, AMD Ryzen 7, or better)
 - GPU: NVIDIA GPU with at least 6 GB VRAM (e.g., NVIDIA RTX 3060)
 - RAM: 16 GB of RAM or more
 - Storage: SSD (Solid State Drive) with at least 100GB of free space
 - OS: Linux (Ubuntu 18.04/20.04/22.04)

5.2.3 Network Connection

Minimum 40 mbps download and upload speed.



Fig. 8. Model predictions using the test dataset's Roboflow 3.0 Object Detection (Fast) model.

6 Conclusion and Future Work

This work contributes a new, diverse AID-SAR dataset for aerial drone imagery and object recognition in the detection of livestock, such as humans and vehicles. To be able to model complex post-disaster scenarios, this dataset spans disaster types and geographic regions, thus being scalable and efficient for object detection. It also advances AI-assisted SAR capabilities, creating an opportunity for human-AI collaboration in mitigating crises. Results from the object detection model, Roboflow 3.0 trained on this dataset, demonstrate the potential of this system for real-time human detection in SAR operations using UAV video footage. Limitations of the proposed system include the need for more actual disaster footage to enhance model validation and performance. Advanced deep-learning models and architectures could be further explored in future work.

We seek to advance AI-driven SAR and surveillance tasks through this shared dataset toward supporting disaster management activities, such as damage assessment, rescue planning, and rapid response. This dataset represents a fundamental step toward the integration of AI for effective high-stakes decision-making.

Acknowledgement. This work is sponsored under the Indo Italy This research is supported by the Dassault Systemes Foundation Grant 2023–2024 (DS Contract ID: 2023/9444) and Department of Science & Technology, Government of India's Indo Italy Researcher Mobility Grant (INT/Italy/P-31/2022 (ER-G)).

References

1. Han, Y., Liu, H., Wang, Y., Liu, C.: A comprehensive review for typical applications based upon unmanned aerial vehicle platform. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **15**, 9654–9666 (2022). <https://doi.org/10.1109/JSTARS.2022.3216564>
2. Papyan, N., Kulhandjian, M., Kulhandjian, H., Aslanyan, L.: AI-based drone assisted human rescue in disaster environments: challenges and opportunities. *Pattern Recognit Image Anal.* **34**(1), 169–186 (2024). <https://doi.org/10.1134/S1054661824010152>
3. Vincent-Lambert, C., Pretorius, A., Van Tonder, B.: Use of unmanned aerial vehicles in wilderness search and rescue operations: a scoping review. *Wilderness Environ. Med.* **34**(4), 580–588 (2023). <https://doi.org/10.1016/j.wem.2023.08.022>
4. Ding, J., et al.: Object Detection in Aerial Images : A Large-Scale Benchmark and Challenges. **8828** (2021). <https://doi.org/10.1109/TPAMI.2021.3117983>
5. Nihal, R.A., Yen, B., Itoyama, K., Nakadai, K.: UAV-Enhanced Combination to Application : Comprehensive Analysis and Benchmarking of a Human Detection Dataset for Disaster Scenarios, pp. 1–17
6. Zhang, J., Wan, G., Jiang, M., Lu, G., Tao, X., Huang, Z.: Small object detection in UAV image based on improved YOLOv5. *Syst. Sci. Control Eng.* **11**(1) (2023). <https://doi.org/10.1080/21642583.2023.2247082>
7. I. Reviews et al.: On Creating Benchmark Dataset for Aerial Image, pp. 1–25
8. Liu, C., Szirányi, T.: Real-time human detection and gesture recognition for onboard UAV rescue. *Sensors* **21**(6), 1–21 (2021). <https://doi.org/10.3390/s21062180>
9. Rahnemounfar, M., Chowdhury, T., Murphy, R.: RescueNet: a high-resolution UAV semantic segmentation dataset for natural disaster damage assessment. *Sci. Data* **10**(1), 1–9 (2023). <https://doi.org/10.1038/s41597-023-02799-4>
10. Wu-murad, P., Manzini, T., Murphy, R.: Open Problems in Computer Vision for Wilderness S.A.R. and The Search for, pp. 3784–3789
11. Bahmanyar, R., Merkle, N.: Saving lives from above: person detection in disaster response using deep neural networks. *I.S.P.R.S. Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **10**, no. 1-W1–2023, pp. 343–350 (2023). <https://doi.org/10.5194/isprs-annals-X-1-W1-2023-343-2023>
12. Ciaglia, F., Zuppichini, F.S., Guerrie, P., McQuade, M., Solawetz, J.: Roboflow 100: A Rich, Multi-Domain Object Detection Benchmark (2022). <http://arxiv.org/abs/2211.13523>
13. Sambolek, S., Ivasic-kos, M.: Automatic Person Detection in Search and Rescue Operations Using Deep CNN Detectors, pp. 37905–37922 (2021)
14. Gupta, R., et al.: Creating X.B.D.: a dataset for assessing building damage from satellite imagery. *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.* 2019-June, pp. 10–17 (2019)
15. Meng, L., Dong, Z.S.: Natural Hazards Twitter Dataset (2020). <http://arxiv.org/abs/2004.14456>
16. Rahnemounfar, M., Chowdhury, T., Sarkar, A., Varshney, D., Yari, M., Murphy, R.R.: Flood-Net: a high-resolution aerial imagery dataset for post flood scene understanding. *IEEE Access* **9**, 89644–89654 (2021). <https://doi.org/10.1109/ACCESS.2021.3090981>
17. Xia, G., Bai, X., Ding, J., Zhu, Z., Belongie, S., May, C.V.: DOTA : A Large-scale Dataset for Object Detection in Aerial Images, pp. 1–17
18. Ding, J., et al.: Object detection in aerial images: a large-scale benchmark and challenges. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(11), 7778–7796 (2022). <https://doi.org/10.1109/TPAMI.2021.3117983>
19. Muddin, N., Dousai, K., Lončarić, S.: Detection of Humans in Drone Images for Search and Rescue Operations, pp. 69–75 (2019). <https://doi.org/10.1145/3449365.3449377>

20. Bernal, A.M.R., Scheirer, W., Cleland-Huang, J.: NOMAD: a natural, occluded, multiscale aerial dataset, for emergency response scenarios. Proc. - 2024 IEEE Winter Conf. Appl. Comput. Vision, WACV 2024, pp. 8569–8580 (2024). <https://doi.org/10.1109/WACV57701.2024.00839>
21. Dousai, N.M.K., Lonearic, S.: Detecting humans in search and rescue operations based on ensemble learning. IEEE Access **10**, 26481–26492 (2022). <https://doi.org/10.1109/ACCESS.2022.3156903>
22. Vasi, M.K.: Improving the Model for Person Detection in Aerial Image Sequences Using the Displacement Vector : A Search and Rescue Scenario (2022)
23. Zhang, N., Nex, F., Vosselman, G., Kerle, N.: Training a Disaster Victim Detection Network for UAV Search and Rescue Using Harmonious Composite Images (2022)
24. Scheele, S., Picchione, K., Liu, J.: LADI v2 : Multi-label Dataset and Classifiers for Low-Altitude Disaster Imagery arXiv : 2406 . 02780v1 [cs . CV] 4 June 2024. vol. 7014, no. Feb 2014 (2024)
25. Thoreau, M., Wilson, F.: SaRNet: a dataset for deep learning assisted search and rescue with satellite imagery. Int. Symp. Image Signal Process. Anal. ISPA, vol. 2021-Septe, pp. 204–208 (2021). <https://doi.org/10.1109/ISPA52656.2021.9552103>
26. Cafarelli, D., et al.: MOBDrone: a drone video dataset for man overboard rescue. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 13232 LNCS, pp. 633–644 (2022). https://doi.org/10.1007/978-3-031-06430-2_53
27. Kyrkou, C., Theocharides, T.: Deep-learning-based aerial image classification for emergency response applications using unmanned aerial vehicles. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., vol. 2019-June, pp. 517–525 (2019). <https://doi.org/10.1109/CVPRW.2019.00077>



Solar PV Power Generation Forecasting Using Synergy of Artificial Intelligence and Metaheuristics Based Approaches

Pankaj Yadav¹✉, Gulshan Sihag², Vikas Pratap Singh³,
and Vivek Vijay¹

¹ Indian Institute of Technology, Jodhpur, India

{yadav.58,vivek}@iitj.ac.in

² UPHF, CNRS, UMR 8201 - LAMIH, Valenciennes 59313, France

sihag.1@alumni.iitj.ac.in

³ Forecasting Department, Emirates Water and Electricity Company,
Abudhabi, UAE

vikasforsmile@alumni.iitj.ac.in

Abstract. In this paper, a novel method for forecasting solar photovoltaic (PV) power generation is presented which plays an essential role in grid stability of renewable energy rich power systems. This study utilizes some artificial intelligence (AI) and meta-heuristic approaches such as Random Forest Regressor (RFR), Logistic Regression and Neural Networks, along with Genetic Algorithm (GA). The framework uses the capability of AI models to learn complex relationships in the solar irradiance and weather data. Further, the parameters of these models are optimized by Genetic Algorithm which leads to more accuracy. The study uses various timescales and accuracy metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R-squared, etc. The results show the higher accuracy of hybrid methods than the standalone models of forecasting. This process increases both precision and accuracy. It also provides insights for future research such as investigating ensemble forecasting methodologies to improve solar PV power generation forecasts.

Keywords: Solar power generation · Renewable energy · Artificial intelligence · Forecasting

1 Introduction

Solar photovoltaic(PV) power has become a building block in the global energy portfolio [10]. The shift towards solar energy is not just a need for clean energy but also shows the capability of altering the energy landscape. However, the variable nature of solar power generation (such as the influence of day/night/weather cycles) brings considerable challenges to energy grid managers and operators [11]. These challenges demand the need for accurate forecasting of solar PV power

generation. The solar energy utilization in the power grid requires forecasting methods that can handle the variability of solar power generation [8]. It ensures stability of grid, optimizes energy dispatch, and improves grid management [17].

It is known that particularly effective hybrid models are generated from the combinations of the Random Forest Regressor (RFR), Logistic Regression, and Neural Networks, together with the operations of Genetic Algorithm (GA). This synergy takes advantage of both the predictive powers and the optimization skills of these AI models of Genetic Algorithm. Random Forest Regressor accurately captures the implicit patterns in solar energy data. Logistic Regression and Neural Networks offer accuracy in predictive modeling. Genetic Algorithm helps to adjust parameters and features of these AI models that improve forecasting accuracy in different predictive situations.

The progress of solar power has been supported by reliable solar forecasts which ensure operational management of power grids with high solar penetration. It assists in energy trading and load balancing, economic dispatching and many others as well [18]. Informed decisions concerning the use of solar energy for integration purposes come with optimized forecasting, restrictive reliance on expensive, and inefficient peaking power plants which are not environmentally friendly [13]. Also, it helps improve the grid management by detecting variations in solar generation, allowing the grid controls to be more flexible and reactive to such changes.

2 Literature Review

The onset of renewable energy sources like solar photovoltaic (PV) systems, has necessitated the development of advanced forecasting methods to enhance grid stability and energy efficiency. This section reviews recent literature on solar PV power generation forecasting. Meanwhile it focuses on the synergistic use of artificial intelligence (AI) and metaheuristic approaches. Over the past years, significant research has been conducted to address the inherent challenges of solar power variability focusing on the need for precise forecasting models.

There has been an inclination towards the use of artificial intelligence for prediction of solar power generation. It is able to capture such complex non-linear interactions usually evident in weather and solar irradiance. Khalyasmaa et al., 2019 [12] addressed the problem of solar power generation prediction using Random Forest Regressors (RFR) and, as it turned out, the model was quite capable of capturing more complex relations between weather data and accumulated solar energy than previous approaches. Similarly, Logistic Regression models have been adopted for their strength in handling large datasets. In short term solar radiation forecasting, as noted by Tiwari et al., 2018 [16] refereed that neural networks have been greatly employed because of their adeptness to understand the society's data and intricacies and even how to model the time series data. Also, this suggests the need for better representation and learning from the data by AI in forecasting.

AI models have been integrated with meta-heuristic algorithms to optimize forecasting performance further. Other techniques such as Genetic Algorithms

(GA) have been used to optimize the features selected to build the AI model and even its parameters. For example, Patanaik et al., 2020 [14] used GA to enhance the structure of a Neural Network for solar energy predictions, and it was effective. This architecture result has led to noticeable gains in accuracy. Besides, it argues that GA employed with AI models is quite effective for reducing forecast errors, as shown by Bendali et al., 2020 [5], through the techniques of tuning weights and structures of the network.

Combining or augmenting the AI approaches with metaheuristic algorithms is a new paradigm of forecasting solar power generation. The purpose of such a combination is to utilize the advantages stemming from both types of models to eliminate the disadvantages exhibited by either of them when treated separately. The research work conducted by Abedinia et al., 2018 [3] specifically tackled hyper-parameter optimization of Neural Networks through the integration of GA within a broader scope. This study notably enhanced the forecast accuracy compared to traditional methods. Also this highlights the potential of hybrid models in addressing the dynamic and complex nature of solar power generation forecasting.

3 Methodology

This methodology section outlines a cross-sectional detail of modeling solar energy forecasting integrating several artificial intelligence techniques such as Random Forest Regressor, Logistic Regression, Neural Networks complemented with a Genetic Algorithm metaheuristic called cross-over. This hybrid method technique has been employed to capture the complex nonlinear relationships inherent in meteorological and solar irradiance data while efficiently tuning the models' hyperparameters for improved accuracy. We have optimised the following hyper-parameters for random forest: number of estimators (searched space value: [10, 200]), maximum tree depth (searched space value: [0, 10]), minimum samples split (searched space value: [2, 10]), minimum samples per leaf (searched space value: [1, 10]), maximum features (searched space value: [log2, None]) and similarly for neural networks: number of hidden layers (searched space value: [1, 2]), hidden layer size (searched space value: [64, 128, 256]), optimizers (searched space value: [adam, rmsprop, sgd]), optimizer's learning rate (searched space value: [0.001, 0.005, 0.01, 0.05, 0.1]) and batch size (searched space value: [32, 64, 128]). We introduce the operational mechanics of the proposed approach, which are also described in Fig. 1.

3.1 Random Forest Regressor Model

The Random Forest Regressor model is an ensemble learning method that constructs a multitude of decision trees (N_{trees}) during training and outputs the mean prediction ($\hat{y}$) of the individual trees for a given input vector ($\mathbf{X}$) [9]. The prediction of the Random Forest Regressor model can be represented as:

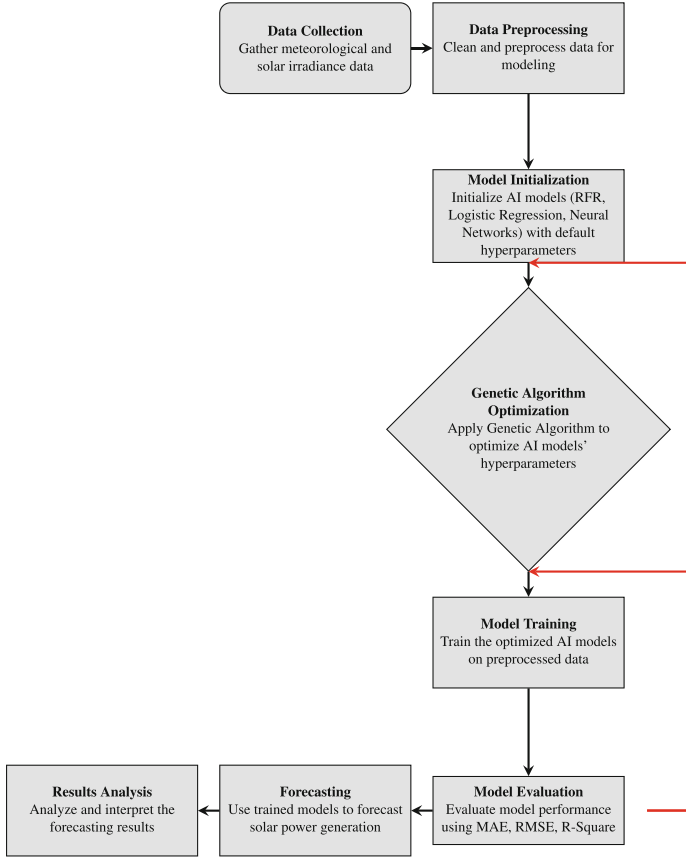


Fig. 1. Flowchart for the proposed methodology.

$$\hat{y}(\mathbf{X}) = \text{MRF} = \frac{1}{N_{\text{trees}}} \sum_{i=1}^{N_{\text{trees}}} T_i(\mathbf{X}; \theta_i) \quad (1)$$

where

- N_{trees} is the number of trees in the forest.
- θ_i represents the parameters (e.g., split criteria, max depth) of the i th tree.
- $\mathbf{X}$ is the input feature vector, comprising variables relevant to solar power forecasting:

$$\mathbf{X} = \begin{bmatrix} \text{Cloud coverage,} & \text{Visibility,} & \text{Temperature,} & \text{Dew point,} \\ \text{Relative humidity,} & \text{Wind speed,} & \text{Station pressure,} & \text{Altimeter} \end{bmatrix}$$

3.2 Logistic Regression Model

The Logistic Regression model is a linear model used for classification problems [4]. It can also be adapted for regression tasks. It predicts the probability that a given input vector ($\mathbf{X}$) belongs to a certain class. Here for forecasting purpose, we are predicting continuous values by modeling the relationship between the input features and the output variable [2]. The logistic regression prediction $\hat{y}$ can be represented as:

$$\hat{y}(\mathbf{X}) = \sigma(\mathbf{w} \cdot \mathbf{X} + b) \quad (2)$$

where

- $\mathbf{w}$ is the vector of weights corresponding to the input features.
- b is the bias term.
- $\mathbf{X}$ is the input feature vector, comprising variables relevant to solar power forecasting.
- $\sigma(z)$ is the sigmoid function defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \quad (3)$$

3.3 Neural Network Model

Neural Models are a type of Deep Learning having many layers of interconnected neurons [1]. Every neuron performs a weighted summation of the input to it and applies a non-linear activation function which passes the output to the next layer. In the case of solar power forecasting, the neural network model takes the feature vector ($\mathbf{X}$) as input and processes it through its layers to give an output ($\hat{y}$) [15]. The prediction of the neural network model can be illustrated as follows:

$$\hat{y}(\mathbf{X}) = f(\mathbf{X}; \mathbf{w}, \mathbf{b}) \quad (4)$$

where

- $\mathbf{w}$ and $\mathbf{b}$ are the weights and biases of the neurons, respectively.
- $\mathbf{X}$ is the input feature vector having variables relevant to solar power forecasting.
- f represents the composition of activation functions applied at each layer of the network.

3.4 Genetic Algorithm for Hyperparameter Optimization

The Genetic Algorithm(GA) is employed to optimize the hyperparameters $\mathbf{H}$ of the AI model, [7] for example: hyperparameters for Random Forest such as the number of trees (N_{trees}), the maximum depth of the trees (D_{max}), and the minimum number of samples required to split an internal node (S_{min}). Each individual in the GA population represents a potential solution and encoded as a chromosome ($\mathbf{C}$), to the optimization problem. With the chromosome comprising the hyperparameters to be optimized [6].

The optimization process involves the following genetic operations:

Initialization

Generate an initial population ($\mathbf{P}$) of chromosomes ($\mathbf{C}_j$), where each $\mathbf{C}_j$ represents a unique set of hyperparameters ($\mathbf{H}_j$):

$$\mathbf{P} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_n\} \quad (5)$$

Fitness Evaluation

For each chromosome ($\mathbf{C}_j$), evaluate its fitness (F_j) by training the AI model on the training dataset with hyperparameters ($\mathbf{H}_j$), and assessing its performance on a validation set using a fitness function, typically based on forecasting accuracy metrics such as RMSE of solar energy predictions:

$$F_j = f(\mathbf{H}_j) = \text{RMSE}(\hat{y}_{\text{Solar energy}}, y_{\text{Solar energy}}) \quad (6)$$

Selection

Select chromosomes for reproduction based on their fitness, using methods such as tournament selection or roulette wheel selection.

Crossover (Recombination)

Create new chromosomes by combining parts of two parent chromosomes $\mathbf{C}_p$ and $\mathbf{C}_q$:

$$\mathbf{C}_{\text{offspring}} = \text{crossover}(\mathbf{C}_p, \mathbf{C}_q) \quad (7)$$

Mutation

Introduce random mutations to the offspring's chromosomes to explore new regions of the solution space:

$$\mathbf{C}_{\text{mutated}} = \text{mutate}(\mathbf{C}_{\text{offspring}}) \quad (8)$$

Replacement

Build the new generation population by positive selection of the best members of the actual generation and their offspring which have just been created. This process of iterations is repeated until the stopping rule applied to the evolution of the algorithm is satisfied. The end result is a set of hyperparameters ($\mathbf{H}_{\text{opt}}$) for the model which strives to maximize the accuracy of predicting the output of solar energy at a particular hour.

4 Data

The dataset used for the analysis was obtained from an open-source platform, GitHub, which offers access to as many relevant datasets as possible. The solar data used for the analysis were found to relate with Illinois state. Additionally, complete weather data was done in accordance with the approaches described by

the National Oceanographic and Atmospheric Administration that enhanced the correlation of weather and solar energy output. The extraction process was made easier by the use of Java programs that were specially designed to collect a total of reasonably 8 weather parameters. These parameters during the first steps of preprocessing were encoded numerically and later were averaged to hourly rate to organize data coherent over the time frame of the study period.

The accuracy of the weather data representation for the Illinois area was improved by applying an average of values from the three closest meteorological stations. This spatial averaging method used the barycentric equation for approximating the average by considering the distance from the referring stations for some of them. The curated data includes a time series extending from February 1, 2016, to October 31, 2017, bounded within the hours of 6 AM to 5 PM. This time window contains 7,536 unique data points. Given the temporal nature of the dataset, where solar energy output is intrinsically linked to preceding weather conditions and energy outputs, a randomized allocation was implemented. This randomization served to disrupt the inherent sequence dependency, thus allowing for a more accurate model training process. The pertinent solar and weather attributes and their respective units of measure are summarized in Table 1:

Table 1. The pertinent solar and weather attributes

Solar PV Power Generation (kW)	Relative Humidity (%)
Cloud Coverage (%)	Wind Speed (Mph)
Visibility (Miles)	Station Pressure (inchHg)
Temperature (°C)	Altimeter (inchHg)
Dew Point (°C)	Relative Humidity (%)

5 Results and Discussion

In this section, firstly, we discuss the ramp rate for different models, followed by the cumulative results for each model. Furthermore, we perform comprehensive error analysis for monthly as well as hourly predictions. Finally, we summarize the overall performance at the end of this section.

5.1 Ramp Rate

Ramp rate analysis is crucial for understanding the variability in solar power output. Figure 2 shows the distribution of ramp rates for different models. The following observations can be made from the plots:

- The base model shows a wider distribution of ramp rates, indicating higher variability in solar power output.

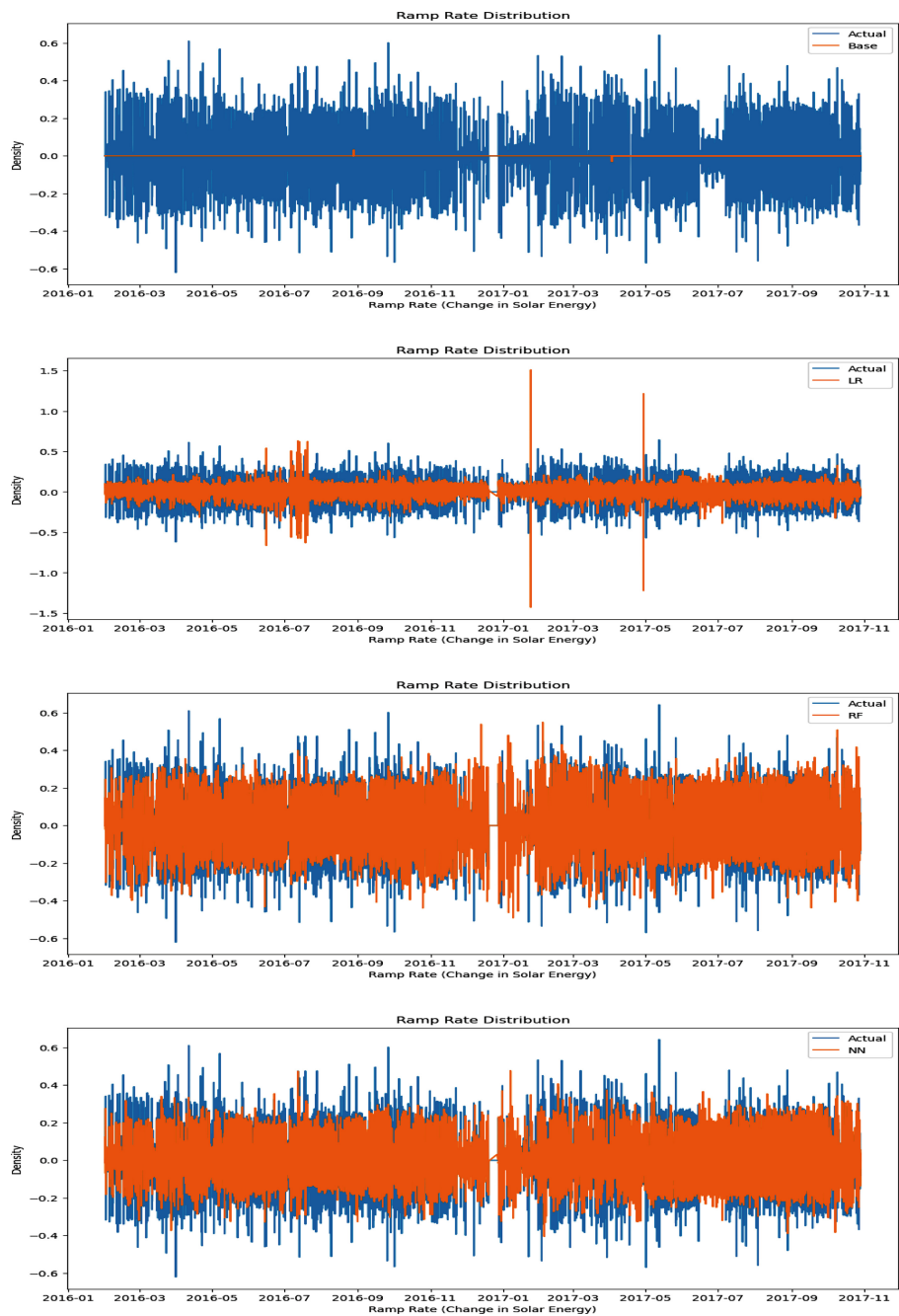


Fig. 2. Ramp Rate Distribution using (a) Base, (b) Logistic Regression, (c) Random Forest, (d) Neural Networks.

- The linear regression (LR) model reduces the variability slightly but still shows significant fluctuations.
- The random forest (RF) and neural network (NN) models show more controlled ramp rates, suggesting that these models can better handle the variability in solar power generation.

To understand the performance of a given prediction model, in the next part, we will discuss the cumulative prediction of each model over two years.

5.2 Cumulative Results

The total quantity of solar energy, produced in approximately 24 months, is shown in Fig. 3, which was compared against cumulative energy outputs captured over this period using predicted production as derived from four different models, a Baseline model, Linear Regression (LR), Random Forest (RF), and a Neural Network (NN). Since all models try their best to forecast these energy production figures over a specified period and minimize the errors, which is represented in Fig. 4.

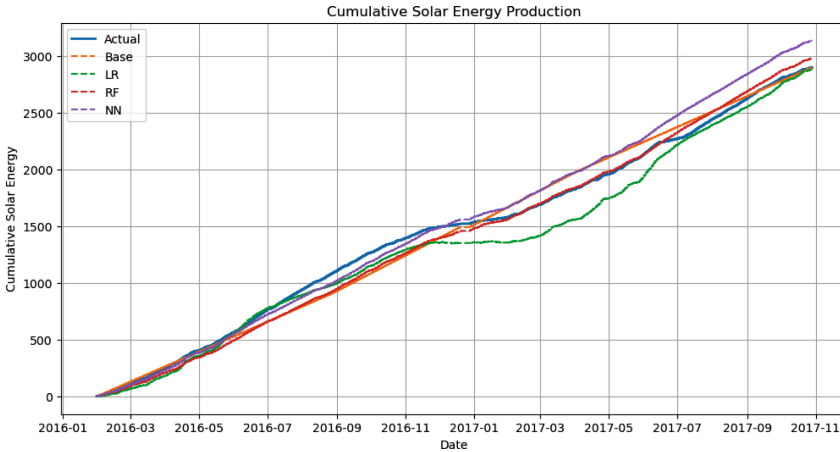


Fig. 3. Cumulative Solar Predictions.

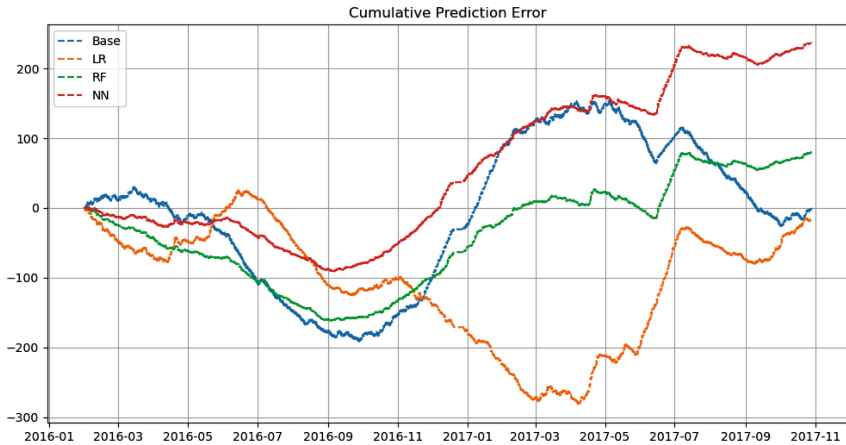


Fig. 4. Cumulative Predictions Error.

It is apparent from the plot that, in the long run and almost without any exceptions, all projected models tend to capture the actual dynamics of cumulatively produced solar energy in an upward direction. Nevertheless, the difference is formed by the effectiveness in performance of each model. It is evident from the graph that the NN model is the one that most over predicts as the production keeps being higher than expected with the ‘NN model’ becoming more severe towards the end of mid 2017. This means that despite the NN model being a very good fit to the data, it is overly fitted resulting in the predicted values being higher than the actual values. The LR and RF models are closer to the actual data. For the LR model, the prediction line of the model keeps bending and the actual figures are somewhere between the line and the dashed figure. With RF model prediction lines show slight wiggles without going further away from the actual production values. The baseline model, shown by the dashed yellow line, provides a straightforward, less sophisticated approach to prediction and generally performs adequately but not as precisely as the more complex models. Furthermore, to get a better understanding of how each model’s prediction deviates from the actual cumulative solar energy production, let’s discuss the cumulative prediction error over the same period. This plot is crucial for understanding the accuracy of each model.

The Baseline model, represented by the blue dashed line, has a notable fluctuation in terms of the prediction error especially around the mid-2017 period. This suggests that while the Baseline model can capture general trends, it struggles with more detailed or nuanced changes in the data. The LR model of the period is represented by the orange line. It manages to attain a low error in the first half of the period but manages for such deviance around the middle of the year 2017. This form of deviation is indicative of seasonal or non seasonal complexities that the LR model may have not embraced while forecasting the possible outlook of solar energy generation. Also, the RF model shows that it

relatively fares better than the remaining two models with regard to large deviations from the predicted values while maintaining a low level of prediction error. Stability in prediction error for the two groups suggests that the RF model is able to combine the two aspirations adequately and therefore is not adversely affected by variations in the data. In spite of this, it still registers some considerable forecast error, more significantly at the period around the mid-2017. Moreover, the NN model projects the least cumulative prediction error toward the latest part of the period rather than the other way all the models have been depicted. This high error corresponds to the overestimation that was shown in the first graph presented above. While the NN model captures trends effectively, its higher complexity may lead to greater sensitivity to the data, resulting in larger cumulative errors.

To understand the performance of each model with respect to time, in the next section, we present the monthly and hourly prediction error results.

5.3 Monthly and Hourly Error Prediction Analysis

The monthly and hourly solar energy errors prediction are shown in Fig. 5 and Fig. 6 respectively. The prediction errors of different models Base, LR, RF, and NN are compared in these figures.

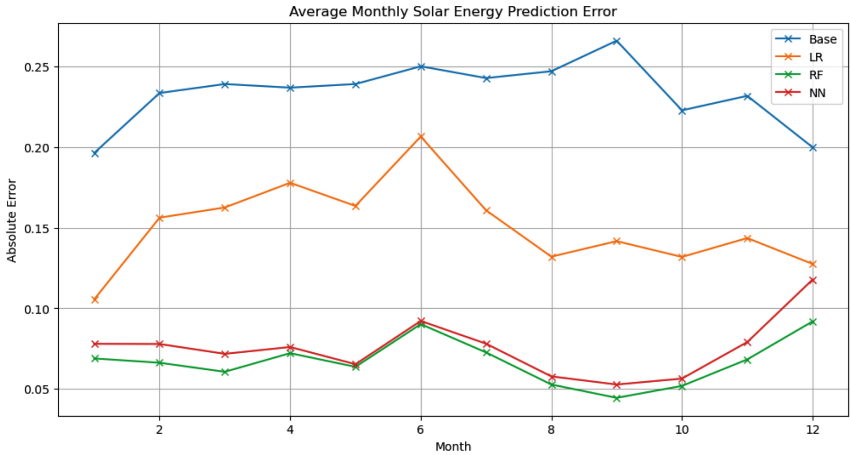


Fig. 5. Average Monthly predictions error.

Hourly solar energy prediction errors revealed noticeable differences among models. The Base model demonstrates the most significant inaccuracies throughout the day due to its inability to capture diurnal variations. The NN model demonstrated the lowest errors, mostly during peak solar generation hours between 09:00 and 16:00. This behavior indicates exceptional performance in modeling nonlinear solar energy dynamics. The RF model closely followed,

demonstrating significant accuracy across different times with strong performance during early morning and late evening hours.

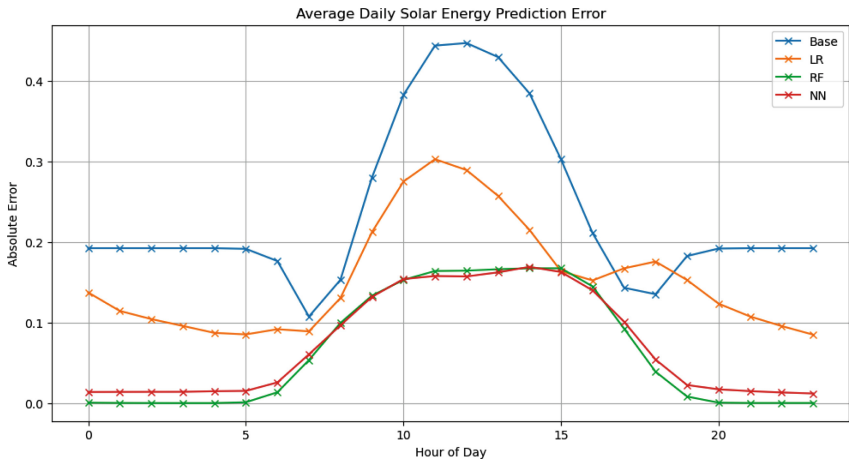


Fig. 6. Average Hourly predictions error.

Furthermore, monthly solar energy prediction analysis demonstrated the RF model’s superior capabilities. RF consistently outperforms other models across seasonal variations. The Base model demonstrated persistent high errors, which show the fundamental unsuitability of the baseline for long-term solar energy forecasting. LR showed moderate improvements over the Base model but struggled during transitional months, often demonstrating unnecessary seasonal peaks and valleys. The NN model performed exceptionally well and closely tracked seasonal trends, though marginally less accurate than the RF model in monthly predictions.

The comprehensive hourly and monthly error analysis demonstrated the RF model’s overall superiority in solar energy prediction across temporal scales. Its robust performance in capturing daily diurnal and seasonal variations makes it the most reliable forecasting approach. While the NN model showed remarkable capabilities in daily predictions, it slightly outperformed by RF in monthly forecasts. The LR and Base models need to be more accurate, showcasing the critical importance of advanced machine learning techniques in modeling solar energy generation’s complex, nonlinear characteristics.

To summarize the results, we present the overall performance of each model in the next section.

5.4 Overall Performance

Figure 7 represents the overall performance of different models in terms of Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2)

score. We can observe from Fig. 7 that the NN model achieves the highest (R^2) score and the lowest MAE, indicating its superior performance. Furthermore, the RF model also performs well and shows good (R^2) score and MAE. However, the LR model shows moderate performance, while the base model performs the worst across all metrics.

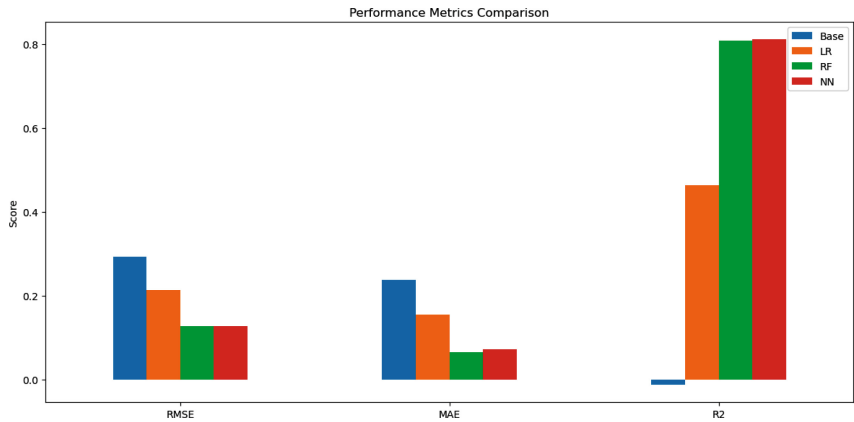


Fig. 7. Overall RMSE, MAE, and (R^2) Score for Different Models.

Furthermore, in order to summarize, the results Table 2 represents the forecast skill score of each AI model used in terms of RMSE, MAE, and (R^2)- score. The forecast skill score is a metric used to evaluate the performance of prediction models relative to a reference forecast, typically a simple baseline like persistence forecast. It quantifies how much better (or worse) a model performs compared to this reference. The score ranges from negative infinity to 1, where 1 indicates a perfect forecast, 0 means the model performs equally to the reference, and negative values suggest the model performs worse than the reference. In this analysis, we're using a 24 h persistence forecast as our reference. This means we're predicting that the solar energy output for any given hour will be the same as it was 24 h ago. This type of forecast is particularly relevant for daily cyclical phenomena like solar energy production.

Table 2. Forecast Skill Scores for Different Models

Model	RMSE	MAE	R2
Base	−0.627446	−1.860570	−1.648581
LR	−0.183535	−0.862703	−0.400755
RF	0.293601	0.212368	0.501001
NN	0.297150	0.127971	0.506002

We can see from Table 2 that:

1. Base Model performs significantly worse than the persistence forecast across all metrics, with negative skill scores indicating it's less effective than simply using yesterday's values.
2. LR also underperforms compared to the persistence forecast but shows some improvement over the base model. Still, its negative skill scores suggest it's not capturing the underlying patterns effectively.
3. RF shows marked improvement, with positive skill scores across all metrics. This indicates it's outperforming the persistence forecast, capturing more complex patterns in the data.
4. NN performs best overall, with slightly higher skill scores than RF in RMSE and R² metrics. It shows the most improvement over the persistence forecast.

Furthermore, there's a clear trend of improved performance with increased model complexity (from Base to NN), suggesting that solar energy production patterns are complex and benefit from more sophisticated modeling approaches. All three metrics (RMSE, MAE, (R²)) show consistent trends across models, reinforcing the reliability of the performance assessment. While NN slightly outperforms RF in RMSE and (R²) skills, RF has a higher MAE skill. This suggests that RF might be better at reducing overall absolute errors, while NN might be better at handling larger errors or outliers. The best models (RF and NN) show skill scores around 0.3 to 0.5, indicating significant improvement over persistence forecast but also suggesting room for further enhancement. Moreover, the negative skills of simpler models (Base and LR) highlight the importance of using appropriate modeling techniques for this type of data. Simply applying basic methods can lead to worse predictions than a naive approach. The reasons Base and LR models perform so poorly could involve the non-linearities, complex temporal patterns, or important data features that these models aren't capturing.

6 Conclusion and Future Direction

The synergistic combination of AI and metaheuristic algorithms represents a promising frontier in solar PV power generation forecasting. This innovative approach addresses the inherent challenges posed by the variability of solar energy and unlocks new potential for grid management and optimization. Our analysis shows a clear trend of improved performance with increased model complexity (from Base to NN), suggesting that solar energy production patterns are complex and benefit from more sophisticated modeling approaches. Our future work will expand the dataset with geographical locations to evaluate these models. It will broaden the research by assessing varied or harsh climate circumstances. We will look to include the geographical locations of Jodhpur and UAE for this purpose. Furthermore, we are looking to explore the scalability of the proposed model by using grid-level solar plants and comparing them with other cutting-edge technologies to assess their strength. We will also optimize the proposed

model using hyperparameter tuning and sensitivity analysis of principal meteorological variables. Our subsequent study will focus on making the proposed model faster and more efficient, as in this study, we have used the basic version of the proposed model. Furthermore, we will explore modifying the methodology to incorporate the current forecasting technique and use more advanced methods like transformers in grid management frameworks.

Future advancements in AI models and metaheuristic techniques are anticipated to lead a roadmap for even more sophisticated and accurate forecasting solutions, contributing to the seamless integration of solar energy into the global energy mix.

References

1. Neural network model of gene expression - VOHRADSKY - 2001 - The FASEB Journal - Wiley Online Library
2. Using LSTM Network Based on Logistic Regression Model for Classifying Solar Radiation Time Series
3. Abedinia, O., Amjady, N., Ghadimi, N.: Solar energy forecasting based on hybrid neural network and improved metaheuristic algorithm. *Comput. Intell.* **34**(1), 241–260 (2018). <https://doi.org/10.1111/coin.12145>
4. Ailobhio, D.T., Ikughur, J.A.: A review of some goodness-of-fit tests for logistic regression model. *Asian J. Prob. Stat.* **26**(7), 75–85 (2024). <https://doi.org/10.9734/ajpas/2024/v26i7631>
5. Bendali, W., Saber, I., Bourachdi, B., Boussetta, M., Mourad, Y.: Deep learning using genetic algorithm optimization for short term solar irradiance forecasting. In: 2020 Fourth International Conference on Intelligent Computing in Data Sciences (ICDS), pp. 1–8 (2020). <https://doi.org/10.1109/ICDS50568.2020.9268682>
6. Czako, Z., Sebestyen, G., Hangan, A.: AutomaticAI - a hybrid approach for automatic artificial intelligence algorithm selection and hyperparameter tuning. *Expert Syst. Appl.* **182**, 115225 (2021). <https://doi.org/10.1016/j.eswa.2021.115225>
7. Daviran, M., Shamekhi, M., Ghezelbash, R., Maghsoudi, A.: Landslide susceptibility prediction using artificial neural networks, SVMs and random forest: hyperparameters tuning by genetic optimization algorithm. *Int. J. Environ. Sci. Technol.* **20**(1), 259–276 (2023). <https://doi.org/10.1007/s13762-022-04491-3>
8. Diagne, M., David, M., Lauret, P., Boland, J., Schmutz, N.: Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renew. Sustain. Energy Rev.* **27**, 65–76 (2013). <https://doi.org/10.1016/j.rser.2013.06.042>
9. Guzali, M.H.A., Aziz, M.A.S.A., Sivaramu, S.S., Borhan, N., Mohtar, W.A.A.Q.I.W., Ahmad, N.: Long-Term solar power generation forecasting at eastern west large scale solar (lss) farm using random forest regression (RFR) method. *J. Adv. Res. Fluid Mech. Therm. Sci.* **118**(1), 1–16 (2024). <https://doi.org/10.37934/arfmts.118.1.116>, number: 1
10. Hassan, Q., et al.: The renewable energy role in the global energy transformations. *Renew. Energy Focus* **48**, 100545 (2024). <https://doi.org/10.1016/j.ref.2024.100545>
11. Khalid, M.: Smart grids and renewable energy systems: perspectives and grid integration challenges. *Energ. Strat. Rev.* **51**, 101299 (2024). <https://doi.org/10.1016/j.esr.2024.101299>

12. Khalyasmaa, A., et al.: Prediction of solar power generation based on random forest regressor model. In: 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), pp. 0780–0785 (2019). <https://doi.org/10.1109/SIBIRCON48586.2019.8958063>, <https://ieeexplore.ieee.org/document/8958063>
13. Li, B., Zhang, J.: A review on the integration of probabilistic solar forecasting in power systems. *Sol. Energy* **210**, 68–86 (2020). <https://doi.org/10.1016/j.solener.2020.07.066>
14. Pattanaik, D., Mishra, S., Khuntia, G.P., Dash, R., Swain, S.C.: An innovative learning approach for solar power forecasting using genetic algorithm and artificial neural network. *Open Eng.* **10**(1), 630–641 (2020). <https://doi.org/10.1515/eng-2020-0073>
15. Pereira, S., Canhoto, P., Salgado, R.: Development and assessment of artificial neural network models for direct normal solar irradiance forecasting using operational numerical weather prediction data. *Energy AI* **15**, 100314 (2024). <https://doi.org/10.1016/j.egyai.2023.100314>
16. Tiwari, S., Sabzehgar, R., Rasouli, M.: Short term solar irradiance forecast using numerical weather prediction (NWP) with gradient boost regression. In: 2018 9th IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG), pp. 1–8 (2018). <https://doi.org/10.1109/PEDG.2018.8447751>, iSSN: 2329-5767
17. Yang, D., Kleissl, J., Gueymard, C.A., Pedro, H., Coimbra, C.: History and trends in solar irradiance and PV power forecasting: a preliminary assessment and review using text mining. *Sol. Energy* **168**, 60–101 (2018). <https://doi.org/10.1016/j.solener.2017.11.023>
18. Yang, D., et al.: A review of solar forecasting, its dependence on atmospheric sciences and implications for grid integration: Towards carbon neutrality. *Renew. Sustain. Energy Rev.* **161**, 112348 (2022). <https://doi.org/10.1016/j.rser.2022.112348>

Challenges and Ethical Considerations in AI



Unraveling the Enigma of Classification of Synthetic and Genuine Information Using Machine Learning and Explainable AI

Vishnu S. Pendyala^(✉) 

Department of Applied Data Science, San Jose State University,
San Jose, CA 95192-0250, USA

vishnu.pendyala@sjsu.edu

<https://www.sjsu.edu/people/vishnu.pendyala/>

Abstract. There is a preponderance of AI-generated text everywhere today. Literature shows that there has been considerable success in detecting such text. This paper uses explainable AI (XAI) techniques to get insights into the workings of the machine learning models used to classify synthetic text. In detecting such text, this work analyzes synthetic and genuine information from visualization and explainability perspectives. The text is converted into vector embeddings using Robustly Optimized BERT Pretraining Approach (RoBERTa). Variational Autoencoders (VAEs) are used for visualization and Support Vector Machine (SVM) is used for classification. Local Interpretable Model-Agnostic Explanations (LIME), SHapley Additive exPlanations (SHAP), and Integrated Gradients are used to explain the classification. The experiments are done on two different types of datasets. Despite the machine learning model achieving outstanding accuracy similar to the previous work in the literature, it was determined that there is no clear explanation of why the representation learning or the classification works so outstandingly well. The explainability techniques used show that the model focuses on words that do not clearly indicate that the text is synthetically generated. Visualization in two dimensions shows that the vector embeddings of both classes of text overlap significantly and that there is no clear separation of the representations learned.

Keywords: Artificial Intelligence · Explainable AI · Intelligent systems · Knowledge representation · Natural Language Processing

1 Introduction

As machine learning technologies advance, they enable the creation of sophisticated text that often blurs the lines between authentic and artificial intentions. Ensuring that information comes from a reliable source helps maintain trust in communication, especially in research, academia, and other professional settings. AI-generated content can spread false or misleading information. Identifying such

content helps prevent the dissemination of inaccuracies until data provenance is mandated by regulation [14]. Sometimes, this is crucial for maintaining originality and ethical standards. Among other dangers, Generative AI (GenAI) can be used to create deceptive narratives or manipulate opinions. Being able to identify AI-generated content is key to mitigating these risks. This work therefore delves into the persistent challenge of distinguishing between genuine and synthetic information in the age of GenAI. It builds upon the previous work which used spectral and non-spectral methods to conclude that the current representation learning techniques are incapable of entirely capturing the veracity of information [16, 18].

Sometimes, the performance of machine learning cannot be entirely relied upon without proper explanations [12]. As AI-generated content becomes more prevalent, the ability to explain model outputs is crucial. LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) are important tools in the context of classifying synthetic and genuine information using machine learning this work uses. Both LIME and SHAP provide insights into how machine learning models make decisions. The work in this paper highlights the capabilities of machine learning models in identifying synthetic information and at the same time, the inadequacy of explainable AI (XAI) techniques, LIME and SHAP in interpreting the behavior of the model. The results from the work emphasize the need for better tools to help interpret the functioning of machine learning models. Therefore, companies relying on machine learning models for applications such as content moderation, academia, or research must recognize the limitations of current XAI technologies. This could lead to increased investment in developing more robust AI systems and training employees to detect and better handle synthetic information.

Most of the current literature on detecting synthetic text using prevalent embedding frameworks reports outstanding results. The contribution of this paper is in the investigation of the following research questions:

RQ1: Are the current explainable AI (XAI) techniques, specifically, LIME and SHAP effective in interpreting the functionality of the machine learning model trained using SVM in separating synthetic information from genuine information?

RQ2: Are the vector embeddings using Robustly Optimized BERT Pretraining Approach (RoBERTa) robust enough for capturing the synthetic nature of the AI-generated text?

The research questions are unique and provide valuable insights to drive future research directions.

The paper is organized as follows. The next section reviews the related literature highlighting the research gap that this paper tries to fill. Sections 3 and 4 explain the datasets and methodology respectively. The next Sect. 5 describes the experiments. Results are detailed in Sect. 6 leading to a discussion on them and potential applications in Sect. 7. The paper concludes in Sect. 8.

2 Related Work

There are multiple surveys [1,2] describing the literature in this regard. Large language models (LLMs) have also been used to detect misinformation and the reasoning behind the classifications has been discussed qualitatively and quantitatively using explainability techniques [17]. Misinformation containment is difficult for various reasons [13]. But in a recent study, foundation models have been used to detect fake images with sufficient accuracy and remarkable explainability of their behavior using LIME, the same technique used in this work [15].

The explainability perspective of misinformation detection has also been explored in the literature. Liu et al. [9] propose a logic-based neural model for multimodal misinformation detection, integrating interpretable logic clauses to enhance explainability and reliability in detecting misinformation on online social platforms. In another research [6], a method combining two approaches is proposed: (a) Domain Adversarial Neural Network (DANN) for generalizability across platforms and (b) an explainable AI technique, LIME to understand the model's reasoning. The authors test this method on COVID-19 misinformation and show that it significantly improves detection accuracy while providing explanations for the results.

Explainable AI and visualization techniques to enhance understanding of disinformation detection models are proposed to aid in interpreting and presenting results effectively [20]. A framework called DISCO [3] for disinformation detection provides explainability by identifying misleading words and utilizing a model-agnostic feature extraction scheme for transparency. Literature review shows that more papers [5] use explainability techniques to interpret misinformation detection.

3 Datasets and Preprocessing

For a comprehensive analysis, two datasets are used for the investigation.

3.1 Human ChatGPT Comparison Corpus

The dataset used for the first set of experiments is a Question Answering Dataset based on the public Human ChatGPT Comparison Corpus (HC3) data [4]. This dataset's main objective is to compare and examine the variations between responses produced by ChatGPT and human respondents in a variety of disciplines, such as open-domain, finance, health, law, and psychology. The dataset contains various question prompts, and answers given by ChatGPT and humans. For the experiments, the questions that were asked from ChatGPT were not used. Only the answers given were used. The dataset contains 24,321 question-answer prompts, but for experimentation, only the first 15,000 answers from humans and ChatGPT were used. The dataset was already clean and balanced, hence pre-processing was not needed. To train the classification model, 15,000 answers each for both classes were selected. Then the data was shuffled and split into the

train, test, and validation sets in the ratio of 8:1:1. The data was split in such a way that the train, test, and validation sets had an equal distribution of classes.

Mostly, the answers are not easily distinguishable to humans. For instance, for the question, “who sang I want to dance with somebody,” the human answer is, “I Wanna Dance with Somebody (Who Loves Me) is the first single from Whitney Houston ’s second studio album Whitney.” and ChatGPT’s answer is a bit lengthy, “I Wanna Dance with Somebody (Who Loves Me) is a song recorded by American singer Whitney Houston. It was released as the lead single from her 1987 album Whitney. The song was written by George Merrill...”

3.2 Artificial Intelligence Generated Abstracts

The Artificial Intelligence Generated Abstracts (AI-GA) dataset [22] sourced from GitHub consists of 28,662 entries for scientific paper titles, extracted from the COVID-19 Open Research Dataset Challenge (CORD-19) dataset [23], and their abstracts. For half of the paper titles, the abstracts were generated using GPT-3 and are thus not original. The dataset is pre-labeled, with label 0 signifying real human-written abstracts and label 1 for artificially generated ones. Each class has 14,331 instances belonging to it, making the dataset balanced, and no additional pre-processing was needed to be performed.

From a superficial glance, the abstracts are not too distinguishable. For instance, here’s the beginning of an AI-generated abstract: “This paper seeks to evaluate the effectiveness of medical and surgical decompression treatments in patients with dysthyroid optic neuropathy. Dysthyroid optic neuropathy is a ...”. An example of the opening sentence of a human-written abstract is, “BioMEMS, the biological and biomedical applications of micro-electro-mechanical systems (MEMS), has attracted considerable attention in recent years and has found widespread applications in disease detection, advanced diagnosis, therapy, drug delivery, ...” As can be seen, it is hard for even humans to distinguish between the two.

4 Methodology

To test the hypothesis that the current Natural Language Processing(NLP) embedding frameworks are inadequate and there is a need to explore other techniques to differentiate human-generated and artificial text, we experimented with the two data sources. Answer statements from the training set of the Question Answering dataset were used to train the Robustly Optimized BERT Pretraining Approach (RoBERTa) [10] natural language model. The model’s performance on the predictions on the test set was analyzed using known evaluation metrics. To interpret the model’s reasoning behind the classification of the text as fake or original, explainability algorithms, Local Interpretable Model-Agnostic Explanations (LIME) [19], SHapley Additive exPlanations (SHAP) [11], and Integrated Gradients [21] were used on the output of the classifier.

The goal of explainable techniques like LIME and SHAP is to explain a model's behavior through the human interpretable features in the dataset - in this case, the words in the text. LIME uses a surrogate explainable model such as logistic regression or decision tree trained on a few perturbed data items synthetically generated around a specific data item in question. Hence the interpretation is local to the specific data item. SHAP uses a concept from game theory to proportionately distribute the model's influence on the outcome across all human interpretable features in the data. Integrated Gradients does the same - assign importance scores to each feature based on their contribution to the output, but by computing the cumulative gradients along a path from a baseline input to the actual input.

In studying the AI-GA data source containing scientific paper titles and abstracts, the dataset was split into train and test sets in the ratio of 75:25. Text from the 'abstract' column of the train and test datasets is converted into sentence-level embeddings using the Sentence Transformer library. The model 'roberta-base-nli-stsb-mean-tokens' is used as it was found to be better suited for natural language processing tasks involving classification. The embeddings generated are input to the support vector machines (SVM) algorithm to train a model and use it to classify the text as belonging to fake or original abstracts.

To find out any underlying patterns in the fake and original text, data visualization was done using variational autoencoders [7], a type of deep learning framework that uses a probability-based approach to model input sample data. The predicted instances were analyzed using LIME and SHAP libraries to improve the interpretability of the model output.

5 Experiments

5.1 AI/Human Generated Text Detection

To study different techniques that could help distinguish between text written by humans and synthetically generated text, the following experiments were carried out.

1) ChatGPT versus Human Answered Questions

To classify a text if it has been generated by ChatGPT or Human, the pre-trained "roberta-base" model from the transformers library was used. The RobertaForSequenceClassification class from the Transformers library was used to train our RoBERTa model. It is a RoBERTa Model transformer with a sequence classification head on top. The classification head is a linear layer on top of the pooled output. Upon analyzing the input data, it was found that the median length of a sentence in the input texts is 144, and the 75 percentile is at 199, so the max_length hyperparameter of the RoBERTa model was set to 256, and padding was enabled. The model was then fine-tuned using the PyTorch library for a total of 2 epochs, with a batch size of 32. AdamW from the huggingface library was used as the optimizer while fine-tuning.

After training, the model was tested on the test set resulting in an F1 score of 1.00. The confusion matrix and classification report obtained after performing

the tests can be seen in Figs. 1 and 2 respectively. The training data for the model classification contains answers for prompts having “Explain like I’m a five-year-old” at the end of the query. The high F1 score may be due to the low diversity of Question-Answer prompts in the training and test data.

After the classification model was trained and ready for use, interpretation techniques such as the Integrated Gradients and LIME were used to try and interpret the results of the classifier. Captum library [8] was used to implement the Integrated Gradients technique and generate attribution scores to understand which features in the input text played an important role in determining whether the texts are generated by ChatGPT or Human.

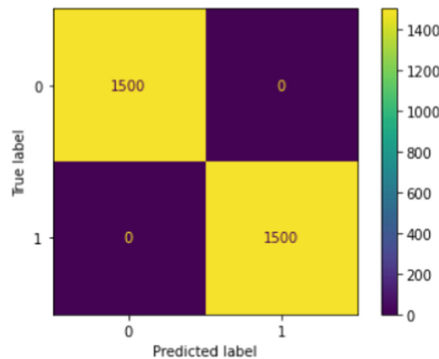


Fig. 1. Confusion Matrix for RoBERTa Classifier on HC3 Dataset.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1500
1	1.00	1.00	1.00	1500
accuracy			1.00	3000
macro avg	1.00	1.00	1.00	3000
weighted avg	1.00	1.00	1.00	3000

Fig. 2. Classification Report for RoBERTa Classifier on HC3 Dataset.

2) Differentiate Real and Fake Scientific Text

With the advent of generative large language models, it could become increasingly common to encounter non-original text even in academic settings. To exemplify this we selected a dataset where half of the scientific abstracts are generated using the GPT-3 language model, while the other half are written by humans. All the sentences were converted from these abstracts into fixed-length embeddings using the roberta-base-nli-stsb-mean-tokens model from the Sentence Transformer library. This is done so that the text can be represented in

a numeric format that is understood by machine learning classifier algorithms. We pass these embeddings to an SVM model built using the polynomial kernel and one-vs-one or ‘ovo’ density function. After training the model on embeddings for 21,496 abstracts, we test it on a corpus of 7,166 abstract entries. The metrics accuracy, precision, recall, and F1 score were calculated to evaluate the performance of the model.

The next step is to explore if there are any significant patterns in the distribution of the generated embeddings that would help distinguish one class of abstracts from the other. Variational autoencoders are a type of deep learning model that uses a probability-based approach to model sample input data. Using the TensorFlow library, an autoencoder model was created with relu activation function in the input encoder layer and sigmoid activation in the output decoder layer and fitted to the text of the abstracts. While compiling results from the autoencoder, the Mean Squared Error (MSE) and cosine embedding loss functions were used to observe differences in the generated samples. The resulting visualization models the data along its two most significant components identified and shows the distribution of the fake and original abstracts in this space. Figure 3 shows the visualization using the loss based on cosine similarity, while Fig. 4 shows the one based on MSE.

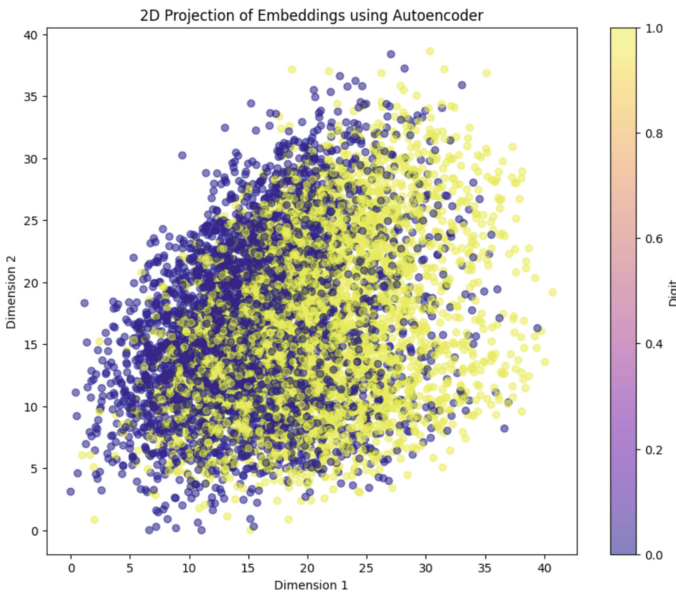


Fig. 3. Autoencoder Visualization of S-bert Embeddings using Cosine Similarity.

As most machine learning algorithms are black-box models, it is often difficult to interpret the generated results for common end users to answer questions such as why a data item was classified as such. In an attempt to explain the rationale

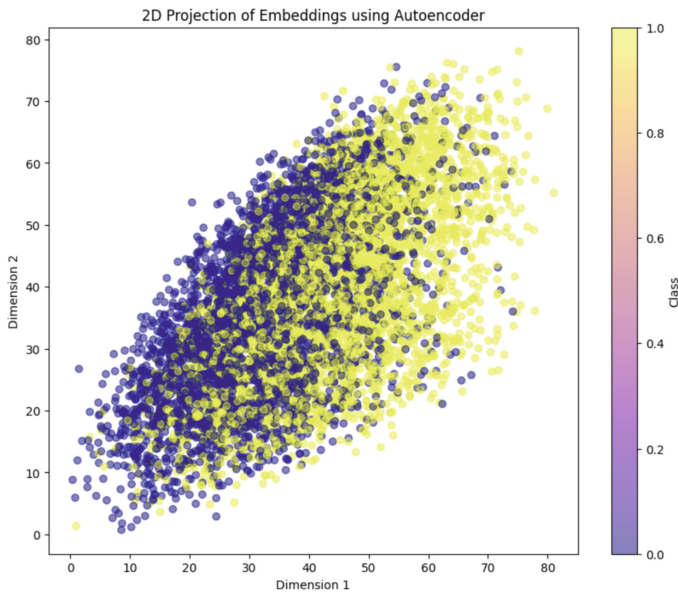


Fig. 4. Autoencoder Visualization of S-bert Embeddings using Mean Squared Error (MSE).

behind the model’s classification, the Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) Python packages were used. The text from the abstract column of the dataset was converted to a vector representation using the CountVectorizer feature extraction technique. The vectors are passed through a logistic regression model this time and after fitting on the training dataset, the model was used to predict the class of the sentences in the test dataset.

These two steps of vectorization and classification are combined using a pipeline function given as input to the LIME explainer package. Figure 5 shows the in-text significant word identification using LIME. The features, in this case, words, that are most important for the model’s class prediction are listed in the middle of the figure, along with their weights. The weight of a feature indicates how important that feature was for the model’s prediction. For example, the word “study” has the most weight of 0.05, which means that it was the most important feature for the model’s prediction. As can be seen, there are no significant words in the text that influence the model towards a ‘1’ class prediction. All the words influence the model towards a ‘0’ class prediction.

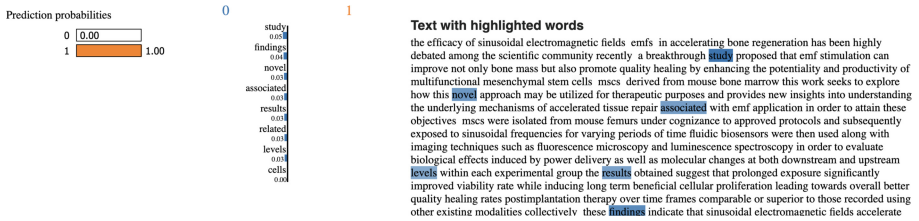


Fig. 5. Text Classification Model Explainability using LIME.

To implement SHAP, the sentences are tokenized similarly using TensorFlow text preprocessing packages and passed to the explainer package. Outputs from both were compared to identify specific words in the sentences that could indicate them as being human-written or artificially generated. Figure 6 is a summary plot from SHAP that shows the contribution of the top few words towards the prediction. The words are shown on the y-axis. The x-axis shows the SHAP value of each of these words. The magnitude of the value represents how much the feature impacts the prediction. The color represents the word’s influence on the model towards a class. In this specific case, words such as “data,” “control,” and “conclusion” pull the classification towards 1, as denoted by the length of the blue bars. On the other hand, words such as “associated,” “during,” and “can” influence the model towards a 0 prediction based on the length of the red bars.

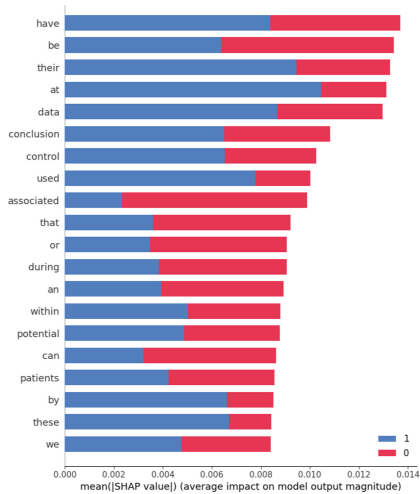


Fig. 6. In-text Significant Word Identification using SHAP.

6 Results

For comprehensiveness, the performance of the SVM model was evaluated using different metrics. Accuracy shows how well the models predicted the true-false and true-negatives. Precision emphasizes true positives and the F-1 score shows a harmonic mean of precision and recall. In addition to accuracy, F-1 score, and precision, ROC accuracy was calculated. In a multi-class classification experiment with a highly imbalanced dataset, ROC micro-averaging is preferable over macro-averaging. Micro-averaging will aggregate the contributions of all classes to calculate the average. However, in macro-averaging, the ROC is calculated independently of each class, and then an average is applied to it that causes it to treat each class equally. Thus, for the balanced dataset, it is preferable to use ROC macro-averaging.

For the experiments on ChatGPT versus human-generated text, the model produced a perfect classification score of 1.0 for evaluation parameters and could correctly classify all 3000 text instances. To understand which words had a greater impact in deciding the classification of texts, attribution scores generated by the Integrated Gradients method as shown in Fig. 7 were used. On visualizing the top 50 score values, it was observed that the words ‘reliably’, ‘versa’, and ‘era’ had the highest occurrence in text generated by ChatGPT. To make the class prediction more easily understandable, the significant words were highlighted using red and green colors along with their true and predicted class labels as shown in Fig. 8. The intensity of the green shade shows how positively significant the word was in deciding the class while words in white are neutral with no contribution towards the prediction.

In the case of artificially generated and real scientific abstracts, the SVM model performed at an accuracy of 84% using sentence embeddings to classify the text. Converting the words to vectors using the CountVectorizer method and passing them through a logistic regression model resulted in an even better accuracy of 90%. The model interpretability was enhanced using LIME and SHAP-based post-hoc explanations. For a specific abstract instance, the significant words for classification in-text were highlighted, and the relevance score for these words in classifying the text as fake or original was also determined.

The SHAP model provides a list of significant words across the whole corpus of text, and identifies the words ‘paper’, ‘into’, ‘been’, ‘its’, and ‘such’ as being most important for classifying an abstract as fake, while the words ‘study’, ‘novel’, ‘levels’, ‘control’ were most prominent in original abstracts. This could indicate that the artificially generated instances had more filler words such as the prepositions highlighted.

The use of autoencoders to visualize the text data did not yield any significant results as the points seemed to be randomly scattered along the two component axes, and no distinguishing patterns in the distribution for the two classes could be identified. There is considerable overlapping between the points for the two classes around the center of the plot, indicating similar feature representations for them. Most of the real abstracts are distributed across the top left of the plot while the fake abstracts are more spread towards the right. Vector embeddings

capture semantic similarity, mapping similar meanings or contexts to nearby points in the embedding space. Each dimension of the embedding space captures a meaningful feature. Given that the embeddings are overlapping in the reduced dimensional space, their shared semantic content can be concluded to be similar. The fact that one of them is synthesized is not apparent from the representation in the two most prominent pseudo-dimensions.

The Kernel Density Estimate (KDE) plots provide a smooth estimate of the probability distribution of a dataset, offering insights into its underlying structure. They are useful for visualizing the distribution, identifying modes, and detecting skewness or multimodality in data. Similar to the visualizations from the autoencoders, the KDE plot generated shown in Fig. 9 also shows overlapping distributions for the two classes, indicating large regions where the two classes are not distinctly separated. For the human-written texts, the distribution is slightly wider, implying greater variability. The fake text distribution has a sharper and higher peak, with comparably less variability.

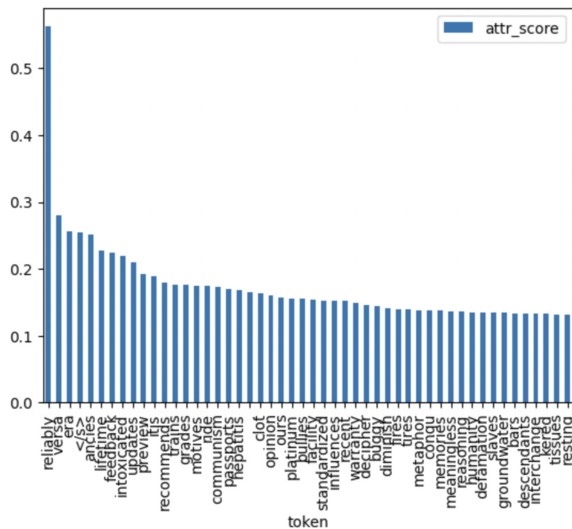


Fig. 7. Attribution scores using Integrated Gradients for top 50 words in ChatGPT generated sentences.

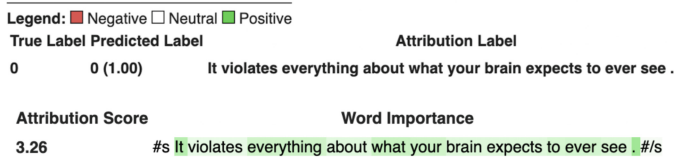


Fig. 8. Visualization of attribution scores in-text.

7 Discussion

Using the LIME, SHAP, and Integrated Gradients explainability modules to improve the interpretability of the model's results for the two sets of experiments generated similar results. Based on the top contributing factors, in this case, words, returned by the algorithm, no definitive words characterize synthetic and real text. Most words are generic and could have been part of the corpus for either text class. Using variational autoencoders to visualize the distribution of fake versus real text samples also did not yield any clear patterns for differentiating between the two classes.

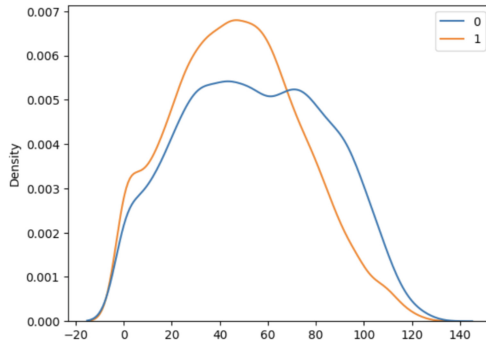


Fig. 9. Kernel Density Estimate Plot for Class Distribution.

7.1 Potential Applications

The analysis of synthetic and genuine information through machine learning, combined with explainable AI, has profound implications across many fields. Assuming the results will be more encouraging in the future, the methodology of this work can be adopted by social media platforms such as Facebook and X to flag fake, synthetic content and also provide explanations for the same. Search engines such as Google, news aggregators such as Yahoo News, and messaging platforms such as WhatsApp can also do the same. The methodology proposed in this paper can prove to be handy in content moderation and academic integrity. Plagiarism claims can be backed up by alluding to the source of the content. The same cannot be done for synthetic content.

By using explainable AI techniques, students can be provided with the substantiation as to why the tool flagged their content as AI-generated. Law enforcement and forensic departments can gain deeper insights into their analysis of fake news to prevent its spread. Research publishers can similarly provide better substantiation when desk rejecting work that they suspect to be synthetic. Law firms, government agencies, and notary services can use the methodology to verify the authenticity of legal documents and provide substantiation for their discovery. By enhancing trust and transparency in AI systems, XAI can help

better identify and mitigate threats related to fake content, misinformation, and data integrity issues. The methodology also allows users and stakeholders to understand how decisions are made, increasing confidence in automated systems. However, at this time, from the results, it seems like there is plenty of scope for improvement of the techniques.

8 Conclusion

The work in this paper demonstrates the superlative efficacy of the machine learning models in identifying synthetic text. The explainable AI techniques, LIME, SHAP, and Integrated Gradients were used to understand why a particular text was classified as synthetic or genuine. However, post-hoc explanations generated using LIME, SHAP, and Integrated Gradients do not convincingly bring out the efficacy of the machine learning model in distinguishing between synthetic and genuine information. Visualizing the information in two dimensions confirms this finding. The research shows that there is a need for better explainability techniques to be able to interpret the inner workings of machine learning models. Another possible research direction is to invent a new embedding scheme for text that is capable of capturing the genuineness of its information content.

Acknowledgment. The author thanks his students, Madhulika Dutta, and Rakshit Gupta for contributing to this work.

References

1. Alghamdi, J., Luo, S., Lin, Y.: A comprehensive survey on machine learning approaches for fake news detection. *Multimedia Tools Appl.* 1–59 (2023)
2. Capuano, N., Fenza, G., Loia, V., Nota, F.D.: Content-based fake news detection with machine and deep learning: a systematic review. *Neurocomputing* **530**, 91–103 (2023)
3. Fu, D., Ban, Y., Tong, H., Maciejewski, R., He, J.: Disco: comprehensive and explainable disinformation detection. In: *CIKM '22, Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 4848–4852. Association for Computing Machinery, New York, NY, USA (2022). <https://doi.org/10.1145/3511808.3557202>
4. Guo, B., et al.: How close is ChatGPT to human experts? comparison corpus, evaluation, and detection. arXiv preprint [arxiv:2301.07597](https://arxiv.org/abs/2301.07597) (2023)
5. Hayato, M., Yoshida, S., Muneyasu, M.: Flexible framework to provide explainability for fake news detection methods on social media. In: *2022 IEEE 11th Global Conference on Consumer Electronics (GCCE)*, pp. 410–411. IEEE (2022)
6. Joshi, G., et al.: Explainable misinformation detection across multiple social media platforms. *IEEE Access* **11**, 23634–23646 (2023)
7. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: *International Conference on Learning Representations (ICLR)* (2014)
8. Kokhlikyan, N., et al.: Captum: a unified and generic model interpretability library for pytorch (2020). <https://arxiv.org/abs/2009.07896>

9. Liu, H., Wang, W., Li, H.: Interpretable multimodal misinformation detection with logic reasoning. In: Findings of the Association for Computational Linguistics: ACL 2023, pp. 9781–9796 (2023)
10. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach. arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) (2019)
11. Lundberg, S., Lee, S.I.: A unified approach to interpreting model predictions (2017)
12. Pendyala, V., Kim, H.: Assessing the reliability of machine learning models applied to the mental health domain using explainable AI. *Electronics* **13**(6), 1025 (2024)
13. Pendyala, V.S.: Misinformation containment using NLP and machine learning: why the problem is still unsolved. In: Deep Learning Research Applications for Natural Language Processing, pp. 41–56. IGI Global (2023)
14. Pendyala, V.S.: One way to fight misinformation. *The Epoch Times* (2024). <https://www.theepochtimes.com/opinion/one-way-to-fight-misinformation-5737667>
15. Pendyala, V.S., Chintalapati, A.: Using multimodal foundation models for detecting fake images on the internet with explanations. *Future Internet* **16**(12), 432 (2024)
16. Pendyala, V.S., Dutta, M.: Experiments with truth using machine learning: spectral analysis and explainable classification of synthetic, false, and genuine information. [arXiv:2407.05464](https://arxiv.org/abs/2407.05464) (2024)
17. Pendyala, V.S., Hall, C.E.: Explaining misinformation detection using large language models. *Electronics* **13**(9), 1673 (2024)
18. Pendyala, V.S., Tabatabaie, F.S.A.: Spectral analysis perspective of why misinformation containment is still an unsolved problem. In: 2023 IEEE Conference on Artificial Intelligence (CAI), pp. 210–213. IEEE (2023)
19. Ribeiro, M.T., Singh, S., Guestrin, C.: “why should i trust you?”: Explaining the predictions of any classifier (2016)
20. Schütz, M.: Disinformation detection: knowledge infusion with transfer learning and visualizations. In: European Conference on Information Retrieval, pp. 468–475. Springer (2023). https://doi.org/10.1007/978-3-031-28241-6_54
21. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks (2017)
22. Theocharopoulos, P.C., Anagnostou, P., Tsoukala, A., Georgakopoulos, S.V., Tasoulis, S.K., Plagianakos, V.P.: Detection of fake generated scientific abstracts. In: 2023 IEEE Ninth International Conference on Big Data Computing Service and Applications (BigDataService), pp. 33–39 (2023). <https://doi.org/10.1109/BigDataService58306.2023.00011>
23. Wang, L.L., et al.: CORD-19: the COVID-19 open research dataset. In: Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020. Association for Computational Linguistics (2020). <https://www.aclweb.org/anthology/2020.nlp-covid19-acl.1>



Challenges and Limitations of TVAE Tabular Synthetic Data Generator

A. Kiran^(✉) , P. Rubini , and S. Saravana Kumar 

Department of CSE, SOET, CMR University, Bangalore, Karnataka 562149, India
kiranarnady@gmail.com

Abstract. Synthetic data is commonly used today in machine learning. From the traditional statistical approach of imputing missing information, the data generation techniques have now leaned towards machine learning and artificially intelligent generative techniques. Generative Adversarial Networks (GAN) and Variational Autoencoder (VAE) are the techniques employed. Tabular data is most common in industrial research. TVAE generates synthetic data which is tabular and available as a part of synthetic data vault (SDV) introduced by Neha Patki et al. Hence, we pick TVAE, perform an in-depth analysis on existing literature and experimentally highlight the strengths and limitations of TVAE, conducted using different types of tabular data. TVAE as such, completely ignores the minority samples in the dataset. Hence, we modify the data generation algorithm by introducing another set of encoder and decoder to oversample only those categorical sub classes which is small in representation and provide that information to the latent space. Our experiments reveal that, despite the several techniques proposed in various literature regarding sampling and balancing the data using VAE, TVAE struggles in producing the categorical values for a subcategory and the associated target class within a feature when the sub-categorical value is a minority. Based on our experiments and analysis, we see a need for further study on modeling the TVAE to produce a well-balanced dataset.

Keywords: Table Variational Autoencoder · Synthetic Data Vault · Generative Adversarial Networks · SMOTE · ADAM

1 Introduction

The concept of synthetic data was first proposed by Donald B Rubin [1]. However, as machine learning and artificial intelligence gained popularity, the techniques of generation shifted towards artificial intelligence. The inception of Generative Adversarial Networks (GAN) [2] led to numerous works around synthetic data generation using GAN [3, 4]. In similar vein, Neha Patki et al. [5] presented Synthetic Data Vault (SDV) to build synthetic data generatively. Table Variational Auto Encoder (TVAE) is one of the libraries available within SDV package. It is based on Variational Auto Encoder (VAE) which was originally proposed by Diederik P. Kingma and Max Welling [6].

Figure 1 shows the architecture of TVAE which employs two independent modules known as Encoder and Decoder. Encoder extracts the mean, variance and standard deviation from the real data provided during the training and updates the information in the latent space. Encoder uses functions such as softmax, cross entropy and Kullback-Leibler Divergence (KLD) as a measure to improve prediction and the information gets updated in the latent space. The decoder uses the noise which is the summation of the matrix of zeros (mean) and ones (standard deviation) of defined size, extracts the statistical information from the latent space to reconstruct the data that matches the input data.

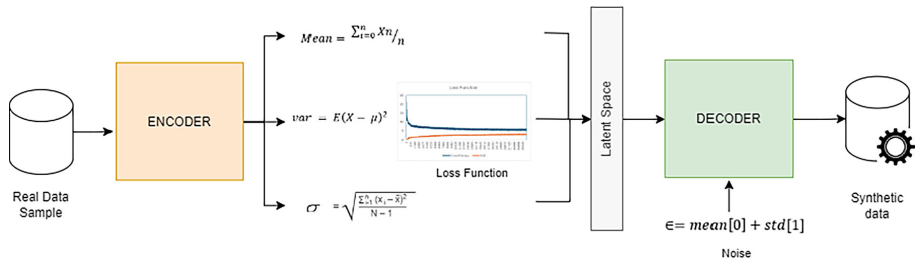


Fig. 1. TVAE Architecture

Generally GANs are considered to produce superior images and are better suited for image generation but they pose several challenges with high dimensional parameters [7]. Conversely, TVAE is much simpler, easier to interpret [8]. TVAE also displays better efficiency. However, the limitation observed, as highlighted in their work by A. Kiran and S. Saravanakumar [9] is, TVAE encounters difficulties when creating synthetic data in situations where data is not balanced. TVAE completely ignores the representation of any subcategory of a categorical column in the training dataset. This presents a significant challenge to the resultant datasets as most of the subcategories get eliminated. This happens because the models make assumptions and over simplify the posterior distribution [10]. In the real world, the healthcare data and financial domain data in specific will have significant data imbalance. Hence it becomes important to explore how the TVAE synesthetic data generator can be improved.

Therefore, the objective of our work is

- to analyze the prior contributions related to VAE
- enhance the TVAE algorithm, which is achieved by picking specific sub class in the categorical column that are represented in small number
- oversample the representation and put them into the latent space so that decoder has better sample representation to reproduce from, enabling it to produce synthetic data, even in scenarios where there exists a substantial disparity between the larger and smaller representation of categories within the feature of the original dataset.

The sections to follow in this paper will delineate the relevant literature, methodology, experimental procedures encompassing the setup and dataset employed, the obtained results, and finally, the conclusion.

2 Related Work

The short coming of VAE was first highlighted by Burda et al. [10] where they analyze that the VAE just assumes and simplifies the distribution rather than considering the entire model. They suggest a weighted model by lowering the weightage of the decoder or applying more constraints to recognize the distribution rather than making assumptions. To improve the learning in the VAE, S. Zhao et al., [11] suggest using maximum mean discrepancy (MMD) instead of evident lower bound (ELBO) that uses KL divergence. They concluded with the experiments on MNIST data that MMD is able to learn more information from the latest space. To address the specific issues related to imbalance in the data set, Y. Zhao et al., [12] utilized the information from the entire dataset to oversample the minority classes. They leveraged VAE algorithm to perform oversampling and referred to their approach as Conditional VAE with self-transferred algorithm (CVAE-SeTred), as the knowledge and instructions were transferred from the majority to the minority class. In their approach the minority and majority sub classes were split, two sperate encoders were trained independently to generate the latent data which is then used by the decoder for recreation. All these techniques focus on image classification, not on tabular data. Fajardo et al., [13] introduce a two 2 stage latent structure, one to address only the independent variables of the dataset “x” and the other one which takes the distribution of both independent “x” and dependent variable or the target class “y”. Experiments are focused on the accuracy of the machine learning algorithm using oversampled data. It does not conclude if all the other subcategories in the data set was reproduced as desired. Method introduced by Chunakai Zhang et al. [14] also discuss oversampling only the target class. Tab-VAE presented by Tazwar et al. [8] proposes to use “softmax” instead of gaussian distribution on categorical or discrete values which TVAE has already adopted. With the proposed approach, we intend to achieve a good representation of the less observed value in every column of the resultant tabular dataset. To achieve this, we introduce a new layer of encoder and decoder, make modifications to the TVAE class and analyze results.

3 Methodology

As shown in Fig. 2, we generate synthetic data using TVAE in three iterations. In the first iteration, synthetic data is generated as-is without performing any modification to the TVAE. In the second iteration, we introduce the sampling logic, which is explained in Sect. 3, Fig. 3. In the third iteration, the real data is balanced using SMOTE [15] and then used in generation of the synthetic data with TVAE. The resultant data sets generated in the three iterations are compared to validate how the proposed modification to the TVAE performs and if it can achieve the intended objective.

4 Experiments

Our goal is to pick each subcategory from columns with discrete data and check if the samples within the subcategory have enough representative samples. If a sub class is found to have lesser representation, then oversample it, just enough to have a fair

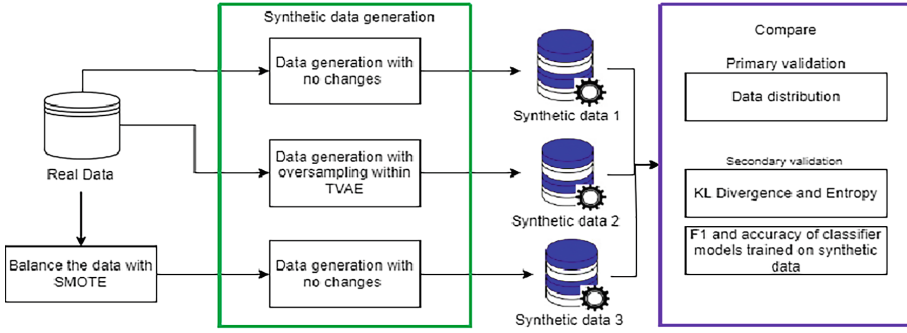


Fig. 2. Methodology used in experiments

representation. This methodology is repeated for each column. As can be seen in Fig. 3, the TVAE is modified to use two encoders and two decoders. The first pair of encoder and decoder are used to sample the subcategories with less representation. Once the samples are created, it is concatenated with the original dataset and then the second encoder is trained with the consolidated dataset.

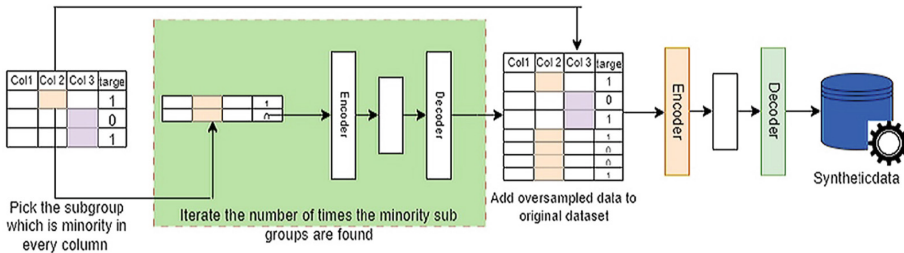


Fig. 3. An oversampling methodology to have a fair representation of minority subgroups in the resultant dataset generated using the TVAE synthetic data generator

4.1 Algorithm

The algorithm is as follows

Let

N be the number of rows in the dataset.

X be the input sample dataset

$x_{majority}$ be the dataset with all the tuples relating to majority target class.

$x_{minority}$ is the intermediate dataset with tuples pertaining to only minority data

$n_{minority}$ represent the number of counts of the minority class.

$columns_{Unique}$ Unique subcategories within the column

Sampling Algorithm

```

Define sampling_weightage (decimal value between 0 and 1)
for all columns in dataframe
    get all columnstypes
    get columnsUnique
    if columnsUnique is BETWEEN 2 and 6 (If the subcategory is only one, its not categorical, hence it is discarded)
        get the count of discrete values within columnsUnique
         $n_{minority} = \text{count of values in } n_{minority} / \text{length of dataframe}$ 
        if  $n_{minority} < 1/\text{total number of categories}$ 
            get all minor class values which is  $x_{minority}$ 
             $\text{sampling\_size} = (N - (2 * x_{minority})) * \text{sampling\_weightage}$ 
             $x_{oversampled} = \text{Fit the new layer of encoder and decoder with } x_{minority} \text{ with the desired sampling\_size}$ 
             $x_{train} = x_{oversampled} \cup x_{majority}$ 
        end if
    end if
end for
Fit TVAE with  $x_{train}$ 

```

We override the fit function of TVAE by introducing our algorithm. The columns with categorical values are picked and the ratio of count of records in every category is checked in relation with the total count of records in the dataset. If the ratio is less than the fraction of total number of categories, then those samples are used and fed into another encoder/decoder pair for oversampling.

Second parameter considered is the amount of oversampling. An established approach is, the maximum size for minority sampling is equivalent to the size of the majority dataset, ensuring an equal ratio between the two classes. However, this approach may result in an overly balanced dataset, which is not the desired outcome. The goal is to generate just enough samples while still maintaining the imbalance to resemble the real dataset, rather than achieving a perfectly balanced dataset. To achieve this, a sampling weightage factor is utilized that determines the desired number of samples of the minority sub class in the resulting dataset. This weightage is specified during the generation process. The integer and float values are not considered except for target class.

4.2 Data

Table 1 presents the sample data that was used in our experiments. Each of the dataset is judiciously chosen based on their target class distribution, the data types and the combination of discrete, continuous and categorical column to test how our algorithm would perform as compared to the non-modified TVAE synthetic data generator.

Table 1. List of datasets with their specifications used as samples to generate synthetic data using TVAE.

Dataset	Data types	Rows	Features (Columns)	Discrete columns	Target class	Target %
Stroke Dataset	categorical, continuous, discrete	43401	12	8	Biopsy	1.80
Cervical Cancer	discrete and categorical	858	36	31	Target	6.41
Credit Card Approval Prediction	discrete and categorical	25129	21	15	Status	0.49
Credit Card Farud	continuous	284808	31	1	Class	0.17
Titanic	continuous, discrete	891	12	5	Survived	38.38
HR Analysis	categorical, discrete	21287	14	7	Target	24.93

Verifying if the TVAE is able to generate all the right samples of the sub categorical classes in the dataset is the primary objective whereas the secondary validation is to check the generic and specific utility [16] of the generated data. Generic utility is validated using a two-step approach. 1) Verifying the presence of the sub categorical values with respect to its target class and, 2) Compare the Correlation map, Entropy and KL divergence in relation to the real data. The specific utility is validated by comparing the accuracy of logistic regression algorithm trained with the real data and synthetic data generated using the TVAE.

5 Results

The evaluation is carried out by comparing the generic utility, which is the statistical resemblance of the synthetic data set with the original data and then the specific utility which is comparing the accuracy of ML algorithm trained using the synthetic data. This section tabulates the results in chronological order. First, we compare the generic utility of the synthetic dataset generated using the three different iterations as explained in the methodology section. Table 2 tabulates the column with the categorical data that was used for assessment as a sample. Cervical cancer and credit card fraud datasets did not have any categorical columns to compare.

Figure 4 illustrates the comparison of the categorical columns in the real dataset (a), Synthetic data generated using TVAE after balancing the real dataset using SMOTE (b) and the data generated using TVAE generator using the custom sampling technique.

Cerebral stroke and HR Analysis datasets are shown as illustrations. The plot provides a relationship between categories of work type against having stroke. The bar depicts the

Table 2. Comparison of column with categorical data in the real dataset and the categorical class that was never produced in the synthetic data set generated using the TVAE generator.

Data set	Column considered for validation	Categories in original dataset	Categories lost in generation
Stroke Dataset	Work_type	Children, Private, Never_Worked, Self_Employed, Govt_Job	Govt_Job
Credit Card Approval Prediction	Income_type	Working, Commercial associate, State Servant, Student, Pensioner	Pensioner
Titanic	Embarked	S, C, Q	
HR Analysis	Company_type	Pvt Ltd, Funded Startup, Early-Stage Startup, Other, Public Sector, NGO	Public Sector, NGO

number of occurrences of the respective categories in the dataset. Table 2 provides the tabulation of all the datasets and the corresponding columns chosen in the experiment.

Table 3 provides the comparison of the Euclidean distance of the categorical columns picked as sample to validate the datasets. Euclidean distance calculates the distance between two points. The distance of the chosen column value is compared between real data and synthetic data.

After the comparison of individual subcategories in the categorical columns, we also compare the utility of the generated data using KL divergence which measures the distance between probability distribution of the real data and the synthetic data and also the entropy, which is the error introduced in the generation. Table 4 provides the values observed for the same.

The comparison of correlation matrix for real and the synthetic data generated for cerebral stroke using TVAE with the customized oversampling algorithm is shown in Fig. 5a and 5b and Fig. 5c provides the comparison of roc_auc curve. Table 5 provides the roc_auc scores for all the datasets used in the experiment.

Finally, the synthetic data is generated using GAN based generator from the SDV package to compare how the algorithm introduced for TVAE perform against GAN based generator. Figure 6 shows the data distribution for categorial columns for Real data, Synthetic data generated using TVAE with enhancement and then the data generated using DPCTGAN.

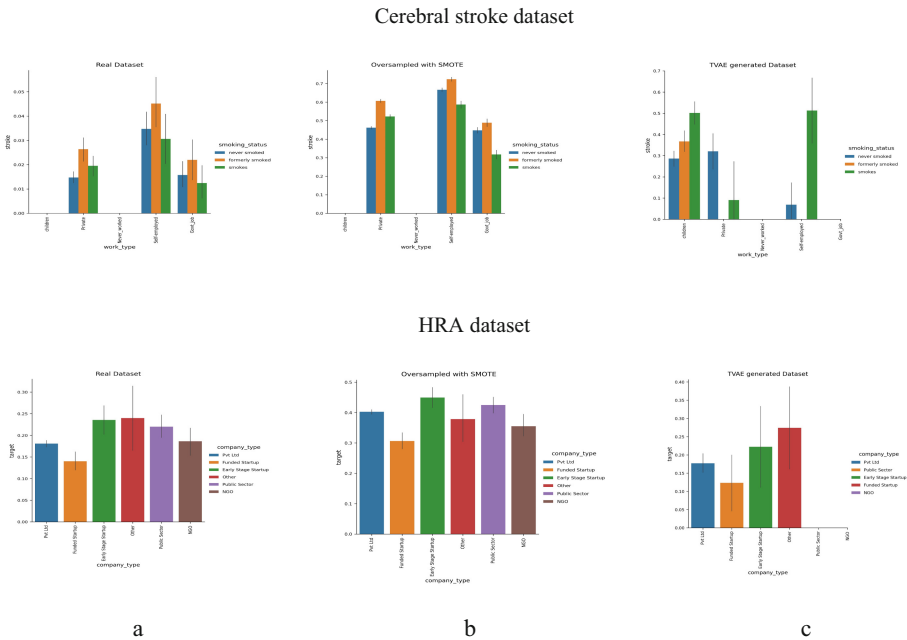


Fig. 4. Comparison of the Cerebral Stroke real data, synthetic data generated using TVAE after balancing the input data with SMOTE and TVAE generated synthetic data with the new oversampling technique that is introduced in this paper. Comparison shows different categorical classes of work_type column and their respective distribution against the target class for cerebral stroke dataset and the company_type against the target for HRA data.

Table 3. Euclidean distance comparison of specific categorical column against datasets generated using the three different variations

Datasets	Column	TVAE AS-S	TVAE with Sampling	SMOTE-TVAE
Stroke Dataset	work_type	64.490309	71.805292	87.441409
Credit Card Approval Prediction	Income_Type	95.451558	98.325988	119.230868
Titanic	Embarked	22.135944	26.720778	0
HR Analysis	Company_Type	84.017855	84.397867	83.234608

6 Discussion

Our experiments and comparison of generic and specific utilities of the real dataset and the synthetic dataset generated using three different techniques with TVAE generator reveal that the TVAE generator as-is with no modification fails to generate the subcategories and target class that are less in representation, which is evident from Table 4, where

Table 4. Tabulation of *Kullback-Leibler Divergence* and *entropy* for the datasets generated using three different techniques.

	TVAE AS- IS		TVAE with Sampling		SMOTE-TVAE	
Data set	KLD	Entropy	KLD	Entropy	KLD	Entropy
Stroke Dataset (age)	68152.777	0.7902	32121.705	0.37142	68123.342	0.277081
Cervical Cancer (age)	2200.5719	0.0951	2411.3052	0.10457	2056.7322	0.08391
Credit Card Approval Prediction (Good Debt)	26319.333	0.7471	27297.423	0.53575	37899.916	0.522672
Credit Card Farud	No categorical columns to compare					
Titanic (age)	5606.1166	0.2116	7066.408	0.26686	inf	inf
HR Analysis (training hrs)	81495.247	0.5017	106904.371	0.66881	98826.919	0.431196

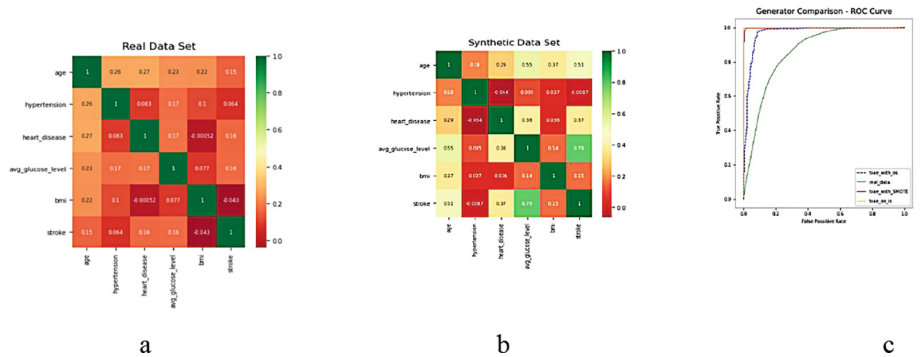


Fig. 5. Comparison of correlation map for cerebral stroke real dataset (a) and synthetic data (b) Fig. 5c is the ROC Curve of Logistic classifier algorithm trained on real data, tvae generator as-is, the tvae generator with custom sampling logic and finally data generated using combination of smote and tvae

the target class were never generated. The data generated using the TVAE with the algorithm provided under Sect. 4 which is a variation of technique provided by [12] Y. Zhao et al. Looks good with all the metrics such as Euclidean distance, KLD, Entropy, Correlation and the specific utility using ML algorithms accuracy scores. However, when the data set is analyzed more closely by comparing individual categories and subcategories within the dataset, we observe that the minority representation is not fully reproduced. Table 2 tabulates the categories that is completely discarded during the data generation. This is also evident from Fig. 4 showing the comparison for cerebral stroke and

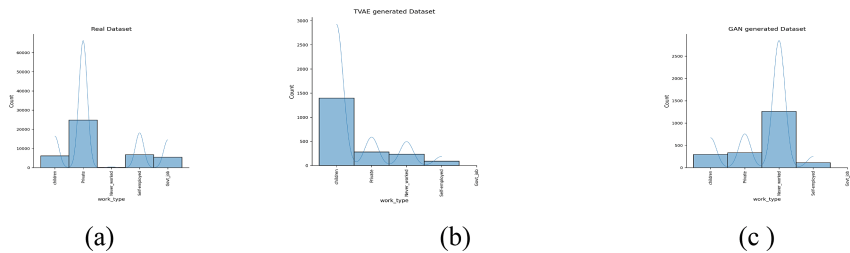


Fig. 6. Comparison of the distribution of categories found in work_type column in Real Data (a) and synthetic data generated using the enhanced algorithm of TVAE (b) and DPCTGAN (c)

Table 5. Comparison of ROC_AUC scores of the real data with synthetic data generated using three different techniques

Datasets	Real Data	TVAE AS-IS	TVAE with Sampling	SMOTE-TVAE
Stroke Dataset	0.8576996	DNG	0.9679287	0.9956866
Cervical Cancer	0.9788728	DNG	0.9801090	0.9460171
Credit Card Approval Prediction	0.9978564	DNG	0.9934098	0.9423336
Credit Card Farud	0.9887823	DNG	0.9969780	0.9571368
Titanic	0.8493087	0.8652266	0.9205976	0.8318623
HR Analysis	0.7496894	0.7116442	0.7510220	0.6450947

HRA datasets. Table 3 provides the information for other datasets used in experiements. The processing also takes more time as there is an additional oversampling step that is introduced. The is considerable when the number of categorical columns increase.

During the process of data generation, modifications were made to the optimizers to assess their impact on the resulting data. Among the various optimizers utilized, the ADAM produced the best results. The *betas*, *weight_decay*, *amsgrad* and *epsilon* values were not altered. Except the learning rate no other parameter had a major impact on the resulting outcomes. Similarly, alternative optimizers such as NADAM, SGD, Adagrad and RMSProps were also tried. There were no significant changes in the results.

The approach used is to pre-process or balance the sample data using SMOTE and then using TVAE generator to generate the synthetic data produced satisfactory results as seen in Fig. 4, Table 3 and Table 4. Finally, from Fig. 6, we observe that the enhanced algorithm of TVAE provides a comparable result with the data generated using GAN based generator.

7 Future Scope of Work

The objective to produce enough samples of the subcategories with minority representation is achieved with the enhanced algorithm. However, there are still inconsistencies in the generation that must be explored further, specifically focusing on the information in the latent space that could help the decoder in better reproducibility of real data. Second area is to improve the efficiency, as the number of categories increase, the resource utilization increases. Third area of focus is to understand impact of oversampling on overfitting, as observed from the scores tabulated in Table 5, the models tend to overfit when the minority samples are extremely minuscule as in the case of credit card datasets.

8 Conclusion

TVAE synthetic data generator produces synthetic datasets which are like real datasets when the dataset is well balanced. This is evident from the experiments that we conducted both using balanced dataset and also, when the balance was achieved using SMOTE prior to feeding the samples to TVAE for data generation. However, when there is an imbalance, TVAE still struggles to produce datasets that are acceptable. Introducing the new algorithm to oversample the minority categories does produce favorable results as it is tabulated in the results section where most of the generic utility and specific utilities provide comparable results with that of real data. However, a detailed analysis of the underlying structure of the dataset still shows that the categories that has lesser representation are not reproduced. Therefore, we recommend further investigation on TVAE, specifically the latent space parameters so that the decoder can learn better even from the minute sample to produce datasets that match the real data samples. For users of synthetic data who want to use TVAE for its ease and efficiency, balancing the input sample before the generation is recommended.

References

1. Rubin, D.B.: An overview of multiple imputation. *Proc. Surv. Res. methods Sect. Am. Stat. Assoc.* (1988)
2. Ian Goodfellow, S., et al.: Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **27**, 2672–2680 (2014). <https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>
3. Jordon, J., Yoon, J., Van Der Schaar, M.: PATE-GaN: generating synthetic data with differential privacy guarantees. *7th Int Conf. Learn. Represent. ICLR 2019*, 1–21 (2019)
4. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional GAN. *Adv. Neural Inf. Process. Syst.* **32**, no. NeurIPS (2019)
5. Patki, N., Wedge, R., Veeramachaneni, K.: The synthetic data vault. *Proc. - 3rd IEEE Int. Conf. Data Sci. Adv. Anal. DSAA 2016*, pp. 399–410 (2016). <https://doi.org/10.1109/DSAA.2016.49>
6. Kingma, D.P., Welling, M.: An introduction to variational autoencoders. *Found. Trends Mach. Learn.* **12**(4), 307–392 (2019). <https://doi.org/10.1561/22000000056>
7. Salimans, T., et al.: Improved techniques for training GANs. In: *Advances in Neural Information Processing Systems*, **29** (2016). https://proceedings.neurips.cc/paper_files/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf

8. Tazwar, S., Knobbout, M., Quesada, E., Popa, M.: Tab-VAE: A Novel VAE for Generating Synthetic Tabular Data,” no. Iccpram, pp. 17–26 (2024). <https://doi.org/10.5220/0012302400003654>
9. Kiran, A., Kumar, S.S.: A comparative analysis of GAN and VAE based synthetic data generators for high dimensional, imbalanced tabular data. In: 2023 2nd International Conference for Innovation in Technology (INOCON), pp. 1–6 (2023). <https://doi.org/10.1109/INOCON57975.2023.10101315>
10. Burda, Y., Grosse, R., Salakhutdinov, R.: Importance weighted autoencoders. 4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc. pp. 1–14 (2016)
11. Zhao, S., Song, J., Ermon, S.: InfoVAE: Information Maximizing Variational Autoencoders (2017). <https://doi.org/10.48550/arXiv.1706.02262>
12. Zhao, Y., Hao, K., Song Tang, X., Chen, L., Wei, B.: A conditional variational autoencoder based self-transferred algorithm for imbalanced classification. Knowledge-Based Syst. **218**, 106756 (2021). <https://doi.org/10.1016/j.knosys.2021.106756>
13. Fajardo, V.A., Findlay, D., Houmanfar, R., Jaiswal, C., Liang, J., Xie, H.: VOS: a Method for Variational Oversampling of Imbalanced Data (2018). <http://arxiv.org/abs/1809.02596>
14. Zhang, C., et al.: Over-Sampling Algorithm Based on VAE in Imbalanced Classification. vol. 10967 LNCS. Springer International Publishing (2018). https://doi.org/10.1007/978-3-319-94295-7_23
15. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002). <https://doi.org/10.1613/jair.953>
16. Snoke, J., Raab, G.M., Nowok, B., Dibben, C., Slavkovic, A.: General and specific utility measures for synthetic data. J. R. Stat. Soc. Ser. A Stat. Soc. **181**(3), 663–688 (2018). <https://doi.org/10.1111/rssa.12358>

Author Index

A

Aadharsh Aadhithya, A. 108
Agarwal, Mohit 68
Agrawal, Swetanshu 48
Anitha, D. 48

B

Badal, Tapas 68
Bijlani, Kamal 108
Błażyczek, Klaudia 95

C

Chodak, Grzegorz 95
Choubey, Abha 181
Choubey, Siddhartha 181

D

Dwivedi, Karnika 68

G

Garg, Astitva Veer 48
Garg, Jyoti 168
Gavada, Sanjana 139
Giri, Virat 3

I

Ithiparambil Sudheeran, Suryajit 121
Itkyal, Arya 197

J

Jaiswal, Dipti 181
Jaiswal, Manuraj 181
Jerith, Gifta 33
Joshi, Gargi 197

K

Karthikeya, Metla 20
Kiran, A. 243
Konduri, Samhita 81
Kotecha, Ketan 197

Kumar, Puranam Revanth 33

Kumar, Ravi 168
Kumar, S. Saravana 243
Kusal, Sheetal 3

M

Math, M. M. 3
Mehta, Vaishali 168
Mohammad, Thayyaba Khatoon 33
Muthaiah, U. 150

N

Naik, Advait Hemant 68
Navabhaarathi, 108

P

Padmaja, B. 20
Patil, Shruti 3
Pendyala, Vishnu S. 81, 229
Prasad, Abhinand K. 121

R

Raj, Abhinav 121
Ramani Bai, V. 121
Reddy, Chinta Vijaya Pandu Ranga 150
Reddy, E. Krishnaveni 33
Rubini, P. 243

S

Saran Dharshan, S. P. 108
Sharma, Dolly 168
Sihag, Gulshan 210
Singh, Vikas Pratap 210
Sireesha, C. 139
Smrithi, Yennum 20
Sreeja, Barigela 139
Srija, Chanda 20
Srivastava, Anugrah 68
Sunil, Sreya 108
Swapna, M. 139

T

Teja, Mallemputi Sai [150](#)

V

Vidhyasagar, B. S. [150](#)

Vigneshwar, E. [108](#)

Vijay, Vivek [210](#)

W

Walambe, Rahee [197](#)

Y

Yadav, Pankaj [210](#)

Yadav, Saurabh [168](#)