

Lakhmi C. Jain
Maria Virvou
Vincenzo Piuri
Valentina E. Balas *Editors*

Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals

Proceedings of GUCON 2019

Advances in Intelligent Systems and Computing

Volume 1064

Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences,
Warsaw, Poland

Advisory Editors

Nikhil R. Pal, Indian Statistical Institute, Kolkata, India

Rafael Bello Perez, Faculty of Mathematics, Physics and Computing,
Universidad Central de Las Villas, Santa Clara, Cuba

Emilio S. Corchado, University of Salamanca, Salamanca, Spain

Hani Hagras, School of Computer Science and Electronic Engineering,
University of Essex, Colchester, UK

László T. Kóczy, Department of Automation, Széchenyi István University,
Gyor, Hungary


Vladik Kreinovich, Department of Computer Science, University of Texas
at El Paso, El Paso, TX, USA

Chin-Teng Lin, Department of Electrical Engineering, National Chiao
Tung University, Hsinchu, Taiwan

Jie Lu, Faculty of Engineering and Information Technology,
University of Technology Sydney, Sydney, NSW, Australia

Patricia Melin, Graduate Program of Computer Science, Tijuana Institute
of Technology, Tijuana, Mexico

Nadia Nedjah, Department of Electronics Engineering, University of Rio de Janeiro,
Rio de Janeiro, Brazil

Ngoc Thanh Nguyen , Faculty of Computer Science and Management,
Wrocław University of Technology, Wrocław, Poland

Jun Wang, Department of Mechanical and Automation Engineering,
The Chinese University of Hong Kong, Shatin, Hong Kong

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within “Advances in Intelligent Systems and Computing” are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

**** Indexing: The books of this series are submitted to ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink ****

More information about this series at <http://www.springer.com/series/11156>

Lakhmi C. Jain · Maria Virvou ·
Vincenzo Piuri · Valentina E. Balas
Editors

Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals

Proceedings of GUCON 2019

 Springer

Editors

Lakhmi C. Jain
Faculty of Engineering and IT
University of Technology Sydney
Broadway, NSW, Australia

Maria Virvou
Department of Informatics
University of Piraeus
Piraeus, Greece

Vincenzo Piuri
Department of Computer Science
University of Milan
Milan, Italy

Valentina E. Balas
Department of Automatics
and Applied Software
University of Arad
Arad, Romania

ISSN 2194-5357

ISSN 2194-5365 (electronic)

Advances in Intelligent Systems and Computing

ISBN 978-981-15-0338-2

ISBN 978-981-15-0339-9 (eBook)

<https://doi.org/10.1007/978-981-15-0339-9>

© Springer Nature Singapore Pte Ltd. 2020

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

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

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

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd. The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

GUCON 2019 Organization

Chief Patrons

Mr. Suneel Galgotia, Chancellor, Galgotias University, India
Mr. Dhruv Galgotia, CEO, Galgotias University, India

Patron

Prof. Renu Luthra, Vice-Chancellor, Galgotias University, India

General Chair

Mr. Deepak Mathur, Director-Elect, IEEE Region 10 (Asia and Pacific)

Conference Chair and Chairman, Oversight Committee

Prof. Rabindra Nath Shaw, Galgotias University, India

Conference Secretary

Prof. Priyabrata Adhikary, NHEC, India
Prof. D. Saravanan, Galgotias University, India

Technical Chairs

Prof. Yen-Wei Chen, Ritsumeikan University, Japan

Prof. Maria Virvou, HOD, Department of Informatics, University of Piraeus, Greece

Publication Chairs

Prof. George A. Tsihrintzis, University of Piraeus, Greece

Prof. Valentina E. Balas, University of Arad, Romania

Prof. Dilip Kumar Sharma, GLA University, Mathura

Honorary Chairs

Prof. Vincenzo Piuri, University of Milan, Italy

Prof. Georges Zissis, President, IEEE IAS

Prof. Lakhmi C. Jain, University of Technology Sydney

Dr. Tamas Ruzsanyi, Ganz-Skoda Electric Ltd., Hungary

Honorary Co-chairs

Prof. C. Boccaletti, Sapienza University, Italy

Prof. Mukhopadhyay, Ex-VC, Lingaya's University, India

Dr. Nishad Mendis, Det Norske Veritas, Australia

Dr. Akshay Kumar, Concordia University, Canada

Springer/GUCON Liaison

Dr. Aninda Bose, Senior Editor, Springer Nature

International Advisory Board

Prof. Valentina E. Balas, University of Arad, Romania

Prof. N. R. Pal, President, IEEE CIS

Prof. George T., University of Piraeus, Greece
Prof. Yen-Wei Chen, Ritsumeikan University, Japan
Prof. Milan Simic, RMIT University, Australia
Prof. Marcin Paprzycki, Polish Academy of Sciences
Prof. Maria Virvou, University of Piraeus, Greece
Prof. Vincenzo Piuri, University of Milan, Italy
Prof. D. P. Kothari, Ex-Director, IIT Delhi, India
Prof. S. N. Singh, VC, MMMUT, Gorakhpur, India
Prof. B. K. Panigrahi, IIT Delhi, India
Prof. R. K. Pandey, DG, NPTI, India

Technical Program Committee and Reviewers

Dr. A. R. Abhyankar
Dr. Aditi Sharan
Dr. Ajay Mittal
Dr. Sudhir Kumar Sharma
Dr. Ajai Jain
Dr. Alok Kushwaha
Dr. Amit Agarwal
Dr. Amalendu Patnaik
Dr. Anil K. Ahlawat
Dr. Anil K. Singh
Dr. Anuradha
Dr. Arun Kumar Verma
Dr. Aseem Chandel
Dr. Asheesh K. Singh
Dr. Ashutosh Dixit
Dr. Asif Ekbal
Dr. B. Dushmanta Kumar Patro
Dr. Baij Nath Kaushik
Dr. Bhaskar Biswas
Dr. Bharat Singh Rajpurohit
Dr. C. Patvardhan
Dr. C. Rama Krishna
Dr. C. K. Nagpal
Dr. Chandra Sekaran
Dr. Chiranjeev Kumar
Dr. Chittaranjan Hota
Dr. D. Bhagwan Das
Dr. D. A. Mehta
Dr. D. S. Kushwaha

Dr. D. S. Yadav
Dr. Desh Deepak Sharma
Dr. Dhram Singh
Dr. Dimple J. Gupta
Dr. Diwakar Bhardwaj
Dr. Girish Patnaik
Dr. Jai Govind Singh
Dr. Joy Deep Mitra
Dr. K. V. Arya
Dr. Kiran Kumar Pattanaik
Dr. Kishor K. Bhoyar
Dr. Komal Kumar Bhatia
Dr. Lalit Kumar Awasthi
Dr. M. K. Dutta
Dr. M. P. Singh
Dr. Madhavi Sinha
Dr. Manisha Sharma
Dr. Mohd. Rihan
Dr. Mayank Pandey
Dr. Munesh C. Trivedi
Dr. N. Badal, KNIT
Dr. Nanhay Singh
Dr. Narendra Kohli
Dr. Naresh Chauhan
Dr. Naveen Kumar
Dr. Neelam Duhan
Dr. Neeraj Tyagi
Dr. O. P. Verma
Dr. Pooja Jain
Dr. Pooja Pathak
Dr. Prabhat Ranjan
Dr. Prabhakar Tiwari
Dr. Prabin Panigrahi
Dr. Pragya Dwivedi
Dr. Pradeep Sharma
Dr. Pramod Kumar
Dr. Pramod Kumar Singh
Dr. Punam Bedi
Dr. R. K. Singh
Dr. R. S. Yadav
Dr. R. S. Rao
Dr. Rahul Rishi
Dr. Rajesh Prasad
Dr. Reena Dadhich
Dr. Ruchika Malhotra

Dr. S. P. Tripathi
Dr. Sapna Gambhir
Dr. Suneeta Agarwal
Dr. Sujoy Das
Dr. Sukomal Pal
Dr. Sunil Kumar Khatri
Dr. Tanveer Siddiqui
Dr. Tarun Shrimali
Dr. Vasudha Bhatnagar
Dr. Vishal Bhatnagar
Dr. Yashpal Singh
Prof. Herbert H. C. Lu
Dr. Senthilrajan Agni
Dr. Abhineet Anand
Dr. Anurag Baghel
Dr. Balamurugan Balusamy
Dr. Priti Bansal
Dr. Sonia Bansal
Dr. Annappa Basava
Dr. Rohit Beniwal
Dr. Vandana Bhasin
Dr. Rodrigo Bortoletto
Dr. John Moses Cyril
Dr. Pinaki Chakraborty
Dr. Sansar Chauhan
Dr. Rahul Chaurasiya
Dr. Surya Deo Choudhary
Dr. Anurag Dixit
Dr. Nripendra Narayan Das
Dr. Indrani Das
Dr. Aparna Datt
Dr. Parneeta Dhaliwal
Dr. Chandrakant Divate
Dr. Rajesh Dubey
Dr. Arman Faridi
Dr. Ankush Ghosh
Dr. Utkarsh Goel
Dr. Pallavi Goel
Dr. Amit Goel
Dr. Priyanka Goyal
Dr. Deepak Gupta
Dr. Suneet Gupta
Dr. Raza Haidri
Dr. Syed Shabih Hasan
Dr. Manas Hati

Dr. Brijesh Iyer
Dr. Manisha Jailia
Dr. Prashant Johri
Dr. Jegathesh Amalraj Joseph
Dr. Sandeep K. Singh
Dr. Vinay Kumar
Dr. Amita Kapoor
Dr. Sandhya Katiyar
Dr. Anvesha Katti
Dr. Ruqaiya Khanam
Dr. Aanchal Khatri
Dr. Shrawan Kumar
Dr. Devendra Kumar
Dr. Avneesh Kumar
Dr. Arun Kumar
Dr. Sanjeev Kumar
Dr. Vipin Kumar
Dr. Sanjay Kumar
Dr. Bhavnesh Kumar
Dr. Sandeep Kumar
Dr. Neetesh Kumar
Dr. M. Mohanraj
Dr. Ramakrishnan Malaichamy
Dr. Manas Kumar Mishra
Dr. Baibaswata Mohapatra
Dr. Thiyagarajan Muthunatesan
Dr. Rashid Mahmood
Dr. Yogendra Meena
Dr. Gitanjali Mehta
Dr. A. K. Mishra
Dr. Keshav Niranjana
Dr. Manoj Panda
Dr. Sanjeev Pippal
Dr. V. A. Sankar Ponnappalli
Dr. Shiv Prakash
Dr. Sheetla Prasad
Dr. Mohammed Abdul Qadeer
Dr. R. Gunasundari Ranganathan
Dr. Ranjeet Kumar Ranjan
Dr. Rohit Raja
Dr. Bharti Rana
Dr. Mukesh Rawat
Dr. Navaid Zafar Rizvi
Dr. S. Pravinth Raja
Dr. Anil Kumar Sagar

Dr. Rajeev Sharma
Dr. Birendra Kumar Sharma
Dr. Shreddha Sagar
Dr. Jyoti Sahni
Dr. Mohd. Saifuzzaman
Dr. Kavita Saini
Dr. Kamalesh Sethuramalingam
Dr. Priestly Shan
Dr. Gavaskar Shanmugam
Dr. Dilip Kumar Sharma
Dr. R. P. Sharma
Dr. Mayank Sharma
Dr. Sudhir Sharma
Dr. Lokesh Kumar Sharma
Dr. Vishnusharma
Dr. Jitendra Singh
Dr. Girish Singh
Dr. Karan Singh
Dr. Harikesh Singh
Dr. Prashant Singh
Dr. Neetay Singh
Dr. Ajay Shanker Singh
Dr. Arun Solanki
Dr. Subhranil Som
Dr. Ritesh Srivastava
Dr. Vijayalakshmi Subramanian
Dr. Hardeo Kumar Thakur
Dr. Pradeep Tomar
Dr. Shashi Kant Verma
Dr. Sohan Kumar Yadav
Dr. Vinod Yadav
Dr. Dileep Yadav
Dr. Chandra Yadav
Dr. Emre Yay
Dr. Aasim Zafar
Dr. Usha Chauhan
Dr. Chetna Dabas
Dr. Sanjoy Das
Dr. Sumithra Gavaskar
Dr. Vimal Kumar

Preface

The book at hand contains selected high-quality papers presented at the **International Conference on Computing, Power, and Communication Technologies 2019 (GUCON2019)** which was organized on September 27–28, 2019 by Galgotias University, India.

The scope of GUCON2019 was to become an international forum for the presentation of the latest research results in electrical, computer, and electronics engineering and to stir interest and discussions on related issues and technologies.

The selected GUCON2019 papers are organized into three sections, namely (1) electronic devices, circuits, and systems; (2) signal processing and bioinformatics; and (3) multimedia and machine learning. We believe that the selected papers constitute an interesting compilation for researchers, engineers, and students.

We are thankful to all the authors who have submitted high-quality papers for presentation at GUCON2019. We also acknowledge with thanks the invaluable help from the members of the international program committee, the various chairs, the special session organizers, and the reviewers. Thanks to the efforts of all of them, we were able to collect many papers on interesting topics and fill the conference with interesting presentations and stimulating discussions.

Special thanks are due to Prof. Janusz Kacprzyk for agreeing to include this volume in the Springer series on *Advances in Intelligent Systems and Computing* that he edits. Special thanks are also due to Dr. Thomas Ditzinger (Editorial Director, Interdisciplinary Applied Sciences, Springer), for the excellent collaboration, patience, and help during the evolution of this volume.

We hope that the volume will provide useful information to professors, researchers, and graduate students in the areas of electronic devices, circuits, and

systems; signal processing and bioinformatics; and multimedia and machine learning and that they will find this collection of papers inspiring, informative, and useful.

Finally, we also hope to see you at a future GUCON event.

Broadway, Australia
Piraeus, Greece
Milan, Italy
Arad, Romania

Lakhmi C. Jain
Maria Virvou
Vincenzo Piuri
Valentina E. Balas

Contents

Design of Low Power Montgomery Multiplier Using Clock Technique	1
Ruqaiya Khanam, Ramsha Khan and Pragati Parashar	
Performance Study of PMN-PT/Ni-Based Multiferroic Nanomagnet for Straintronics Device Applications	15
Amlan Sen, Sampreet Sarkar and Ankush Ghosh	
An Enhanced Recycling Folded Cascode Amplifier	25
Praveen Soni and Shweta Gautam	
A New Approach for Bias–Variance Analysis Using Regularized Linear Regression	35
M. Rajasekhar Reddy, B. Nithish Kumar, N. Madhusudana Rao and B. Karthikeyan	
A Heuristic Algorithm for Diameter Calculation	47
Farheen Siddiqui and Nida Raza	
Arbitrary Waveform Generator for Radar Simulator	55
Abhishek Gupta, Anurag Mahajan and K. Aurobindo	
Comparative Analysis of Digital Image Watermarking Based on Lifting Wavelet Transform and Singular Value Decomposition	65
Sameena Naaz, Ehtesham Sana and Iffat Rehman Ansari	
Analysis of Capacitance–Voltage Characteristics for Ultrathin Si/SiGe/Si Hetero-Layered MOS Structure	83
Rudra Sankar Dhar, Lalthanpuii Khiangte, Parvin Sultana and Ankit Kumar	

Diabetic Retinopathy Detection Using Twin Support Vector Machines	91
Manisha Singla, Samyak Soni, Prateek Saini, Abhishek Chaudhary and K. K. Shukla	
Semantic-Based System for Exercise Programming and Dietary Advice	105
Givi-Giorgi Mamatsashvili, Konrad Ponichtera, Mikołaj Małkiński, Maria Ganzha and Marcin Paprzycki	
Offline Handwritten Gurumukhi Character Recognition System Using Deep Learning	121
Udit Jindal, Sheifali Gupta, Vishal Jain and Marcin Paprzycki	
A Hybrid Self-constrained Genetic Algorithm (HSGA) for Digital Image Denoising Based on PSNR Improvement	135
Divya Verma, Virendra Prasad Vishwakarma and Sahil Dalal	
Deep Convolutional Neural Network with Transfer Learning for Detecting Pneumonia on Chest X-Rays	155
Prateek Chhikara, Prabhjot Singh, Prakhar Gupta and Tarunpreet Bhatia	
A Review on Video Analytics Its Challenges and Applications	169
Shaharyar Alam Ansari and Aasim Zafar	
Digital-Controlled Dual-Mode Switching Mode Power Supply for Low-Power Applications	183
K. Shubha Rao and Veena S. Chakravarthi	
Depth Analysis of Different Medical Image Segmentation Techniques for Brain Tumor Detection	197
Kapil Kumar Gupta, Namrata Dhanda and Upendra Kumar	
Comparative Analysis of Classification Techniques for Diagnosis of Diabetes	215
Paramjot Kaur and Ramanpreet Kaur	
Multimedia Tampering Detection: A Comprehensive Review of Available Techniques and Solutions	223
Gautami Tripathi, Mohd. Abdul Ahad and Zeeshan Ali Haq	
Optimising Fractal Encoding Using Cuttlefish Optimisation Algorithm	237
Smriti Sehgal, Laxmi Ahuja and M. Hima Bindu	
Chaos Theory on Generative Adversarial Networks for Encryption and Decryption of Data	251
Juhi Purswani, Rajesh Rajagopal, Riya Khandelwal and Anuraj Singh	
Author Index	261

About the Editors

Lakhmi C. Jain, Ph.D., M.E., B.E. (Hons), Fellow (Engineers Australia), is with University of Technology Sydney, Australia, Liverpool Hope University, UK and KES International, UK. Professor Jain founded the KES International for providing a professional community the opportunities for publications, knowledge exchange, cooperation and teaming. Involving around 5,000 researchers drawn from universities and companies worldwide, KES facilitates international cooperation and generates synergy in teaching and research. KES regularly provides networking opportunities for professional community through one of the largest conferences of its kind in the area of KES. His interests focus on the artificial intelligence paradigms and their applications in complex systems, security, e-education, e-healthcare, unmanned air vehicles and intelligent agents.

Maria Virvou was born in Athens, Greece. She received a B.Sc. degree in Mathematics from the National and Kapodistrian University of Athens, Greece, a M.Sc. degree in Computer Science from the University College London, U.K., and a Ph.D. degree in Computer Science and Artificial Intelligence from the University of Sussex, U.K.

She is a Full Professor, director of postgraduate studies, director of the software engineering lab and former department head in the department of Informatics, University of Piraeus, Greece. She is author/co-author of over 350 research papers published in international journals, books and conference proceedings and of many books and monographs in Computer Science published by Springer and other publishers. She has been the Editor of over 20 collections of papers in conference proceedings or books, published by major academic publishers, such as IEEE, Springer and IOS Press. She has also been the Editor-in-Chief of the SpringerPlus Journal (Springer) for the whole area of Computer Science. Additionally, she has been an Associate Editor of the Knowledge and Information Systems (KAIS) Journal (Springer) and member of the editorial board of many other International Journals. She has been the general co-chair of the yearly conference series of International Conference on Information, Intelligence, Systems and Applications (IISA 2018, 2017, 2016, 2015, 2014, 2013), technically sponsored by IEEE, which

aims at promoting research in the areas of interactive multimedia and major applications such as e-learning and m-learning. She has been the general chair/programme chair of over twenty (20) international conferences. She has been the principal investigator or co-investigator of numerous national/international research projects. She has supervised 12 Ph.D. theses which have been completed successfully, and many of her former Ph.D. students hold academic positions in universities. She is currently supervising 5 Ph.D. students and 5 postdoctoral researchers. Many journal articles of her have been ranked among the most cited/downloaded papers of the respective journals where they have been published. She has been a recipient of many best paper awards in international conferences. She has been an invited keynote speaker for many international conferences. According to Microsoft Academic Search, she has been ranked as 52 in the top 100 authors out of 58000 authors worldwide in the area of computers and education. Her research interests are in the areas of computers and education, artificial intelligence in education, user and student modeling, e-learning and m-learning, knowledge-based software engineering and human–computer interaction.

Vincenzo Piuri is a Professor at the University of Milan, Italy (since 2000). He was an Associate Professor at Politecnico di Milano, Italy, a Visiting Professor at the University of Texas at Austin, USA, and a Visiting Researcher at George Mason University, USA. He has founded a start-up company in the area of intelligent systems for industrial applications and is active in industrial research projects. He received his M.S. and Ph.D. in Computer Engineering from Politecnico di Milano, Italy. His main research and industrial application interests are artificial intelligence, intelligent systems, computational intelligence, pattern analysis and recognition, machine learning, signal and image processing, biometrics, intelligent measurement systems, industrial applications, distributed processing systems, Internet-of-things, cloud computing, fault tolerance, application-specific digital processing architectures and arithmetic architectures. He published over 400 papers in international journals, international conference proceedings and books. He is Fellow of the IEEE and Distinguished Scientist of ACM. He has been IEEE Vice President for Technical Activities (2015), member of the IEEE Board of Directors (2010–2012, 2015), and President of the IEEE Computational Intelligence Society (2006–2007). He is the Editor-in-Chief of the IEEE Systems Journal (2013–2019).

Valentina E. Balas is currently a Full Professor in the Department of Automatics and Applied Software at the Faculty of Engineering, “Aurel Vlaicu” University of Arad, Romania. She holds a Ph.D. in Applied Electronics and Telecommunications from Polytechnic University of Timisoara. Dr. Balas is author of more than 250 research papers in refereed journals and international conferences. Her research interests are in Intelligent Systems, Fuzzy Control, Soft Computing, Smart Sensors, Information Fusion, Modelling and Simulation. She is the Editor-in-Chief to International Journal of Advanced Intelligence Paradigms (IJAIP) and to International Journal of Computational Systems Engineering (IJCSysE); a member in editorial board member of several national and international journals; and

director of Intelligent Systems Research Centre in Aurel Vlaicu University of Arad. She is a member of EUSFLAT, SIAM, and a senior member IEEE, member in TC – Fuzzy Systems (IEEE CIS), member in TC – Emergent Technologies (IEEE CIS), member in TC – Soft Computing (IEEE SMCS).

Design of Low Power Montgomery Multiplier Using Clock Technique



Ruqaiya Khanam, Ramsha Khan and Pragati Parashar

Abstract Looking at the current scenario, digital systems are more beneficial than the analog system. The power consumption and speed are the major performance criteria for any digital system. It is very important to have a high speed and less power consumption in all devices. The priority of having a high speed or less power consumption depends on the need of the system. This paper presents the implementation of high-performance Montgomery modular multiplier which is also low power consuming. It works upon a simple and efficient Montgomery multiplication algorithm. Hardware as well as software fields find applications of Montgomery modular multiplication. Hardware implementations use direct data path, making them faster in contrast to software implementations. Some real-time applications of software implementations are not fast enough; the reason being their flexibility. Such kind of implementations have an additive advantage that they are capable of being customized as per new algorithms. The power consumption of the proposed design is 14mW and has been improved by 59.1% from the existing design.

Keywords Montgomery multiplication · Cryptography · Low-power modular multiplication

1 Introduction

Public Key Cryptography (PKC) algorithms use modular multiplication as the basic operation. Division operation is implemented in modular reduction. Because of the prolonged time of operation, division is often not desirable. Thus, a new representation is proposed by Montgomery which is known as Montgomery residue. According

R. Khanam (✉) · R. Khan · P. Parashar
Galgotias University, SEECE, Greater Noida, UP, India
e-mail: dr.kruqaiya@gmail.com

R. Khan
e-mail: ramsha.khan01_2017@galgotiasuniversity.edu.in

P. Parashar
e-mail: paras.pragati@gmail.com

© Springer Nature Singapore Pte Ltd. 2020

L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_1

to this algorithm, an integer X is represented as $X \cdot R \bmod M$. $R = 2r$ is a radix and M is modulo. Also, R is co-prime to M . It can be seen here that instead of division by M , multiplication by R is performed.

The proposed multiplier, in this case, implements binary representation to receive and output the data. At each addition operation, single-level carry-save adder (CSA) is used. RSA public-key cryptosystem requires modular exponentiation in which Montgomery multiplication is the basic operation. Other cryptographic and cryptanalytic transformations along with methods of factors like ECM, $p - 1$, Pollard's "rho" method, and Elliptic Curve Cryptosystems also use the method of modular multiplication. Modular multiplication has been a subject of constant interest. Adder and shifter are the main components used in the Montgomery multiplier. Among the considerations, power consumption must be minimum for efficient results. Clock gating, pipelining, etc., are the techniques used to reduce the power of the devices.

Cryptography is done for secure communication. It is of two types: symmetric key cryptography and asymmetric cryptography. For secure communication, encryption is done at the sender's end. Here, data in plain text is converted into ciphertext. While at the receiver's end, decryption is carried out. It means converting data from cipher text to plain text. If we talk about the symmetric key algorithms, encryption and decryption of text make use of the same key. It is the secret key. It enables the sharing of secure information between two or more than two parties. Asymmetric cryptography uses pairs of keys. This is called Public Key Cryptography (PKC) [1]. PKC makes use of two keys, namely "public key" and "private key" in order to encrypt and decrypt messages as well as to transmit them. The concept of PKC was developed by Diffie, Hellman, and Merkle.

2 Methodology Related Work

2.1 Algorithm for Public Key Cryptography

1. A1 and A2 are in agreement with two large integers p and q such that $1 < p < q$.
2. A1 then selects a random number i . Then it calculates $I = P^i \bmod q$. A1 forwards I to A2.
3. A2 then selects another random number j . Then it calculates $J = p^j \bmod q$. A2 forwards J to A1.
4. k_1 is computed by A1 as $k_1 = J^i \bmod q$. k_2 is computed by A2 as $k_2 = I^j \bmod q$.
5. Thus, $k_1 = k_2 = p^{(ij)} \bmod q$.
6. Hence, the secret keys for secure transmission are k_1 and k_2 .

2.2 Rivest–Shamir–Adleman (RSA) Algorithm

Since secure PKC distribution is allowed by DHM algorithm, the digital signature is not implemented through DHM algorithm. At MIT, a group of three researchers, namely Ronald Rivest–Shamir–Adleman (RSA) searched a practical mathematical function for the implementation of a complete PKC approach. It is known as the RSA algorithm. This algorithm was based on the product of two prime numbers that are able to fulfill all the requirements of PKC [1].

RSA algorithm can be explained as follows:

$$\begin{aligned}
 \text{Ciphertext} &= (\text{plaintext})^d \bmod (m) \\
 \text{Plaintext} &= (\text{ciphertext})^e \bmod (m) \\
 \text{Private key} &= \{d, m\} \\
 \text{Public key} &= \{e, m\}
 \end{aligned} \tag{1}$$

The functionality of elliptic curve cryptography is similar to RSA [1]. It consumes less power than RSA and it can be implemented on small devices only. For the purpose of key exchange and digital signature processes, E1 GAMA 1 is the mainly used algorithm. It uses logarithmic methods for calculation. The DSA algorithm is used for signing for data but not for the encryption. It was developed by government of the United States for digital signatures. DSA and RSA have got differences in their uses. If comparisons are to be made between RSA and DSA, then it is faster to validate the digital signature rather than creating it in case of RSA. Whereas it is faster to create a digital signature than to validate it in case of DSA. The repeated modular multiplication operation and division operation make the RSA algorithm slower in process.

To perform fast multiplication, a method known as modular multiplication was developed by Montgomery. It is used to compute $x*y \bmod N$ in computers where the values of x , y , and N are large [2].

2.3 Montgomery Algorithm

Rules:

1. Choose multiplicand and multiplier such that both are be smaller than N .
2. Modulus N should be an odd number.
3. Find $n = \lceil \log_2 N \rceil + 1$
4. Modulus N should be relatively prime to the radix.

Steps:

1. X and Y are two integers.
2. Compute $X' = X * R \pmod{N}$. This is done to transform to the Montgomery domain. R is the smaller power of the base that is greater than the modulus. And $Y' = Y * R \pmod{N}$.
3. Compute $X' * Y'$. We get $(X * R) (Y * R) \pmod{N}$, i.e., $X * Y * R * R \pmod{N}$.
4. $X * Y * R \pmod{N}$ is the desired result in the Montgomery domain as such.
5. Instead of dividing by the R , the second approach is $R' = R^2 \pmod{N}$ and then converting a number X to $X * R \pmod{N}$ by simply computing the Montgomery reduction of X and R' since $\text{Montgomery}(X, R') = \text{Montgomery}(X, R^2) = (X * R * R) / R \pmod{N} = X * R \pmod{N}$. To convert back a number $X' = X * R \pmod{N}$, simply compute the Montgomery reduction of X' and 1 since $\text{Montgomery}(X', 1) = \text{Montgomery}(XR, 1) = (XR) / R \pmod{N} = X \pmod{N}$.

3 Modular Multiplication Algorithm

Radix-2 Montgomery modular multiplication also known as MM algorithm as given below is the algorithm for simple Montgomery multiplication. The range of S lies in between 0 to $2n$. One more subtraction is needed at the end of the process.

Inputs: A, B, N (modulus)

Output: $S[k]$

1. $S[0] = 0$;
2. For $i = 0$ to $k - 1$ {
3. $q_i = (S[i]_0 + A_i * B_0) \pmod{2}$;
4. $S[i + 1] = (S[i] + A_i * B + q_i * N) / 2$; }
5. If $(S[k] \geq N)$ $S[k] = S[k] - N$;
6. Return $S[k]$; (2)

Modular exponentiation performs modular multiplication repeatedly and then the final multiplication can be reduced. When the MM algorithm is applied to modular exponentiation, then the following equation arises.

$$q^i = (S^{(i)} + A * B(i)) \pmod{2}$$

$$S^{(i+1)} = (S^i + A * B[i] + q^i * N) / 2 \quad (3)$$

In order to improve the scalable hardware architecture, the MM algorithm needs a word-oriented algorithm. Thus, we choose an algorithm that performs bit-level computation and returns word-level output. Since this algorithm gives efficient hardware implementation by scanning the multiplier bit by bit and multiplicand word

by word, it is called Multiple Word Radix-2 Montgomery Multiplication Algorithm (MWR2MM).

$$\begin{aligned}
 M &= M^{(e-1)}, \dots, M^{(1)}, M^{(0)}, \\
 Y &= (Y^{(e-1)}, \dots, Y^{(1)}, Y^{(0)}), \\
 X &= (x_{m-1}, \dots, x_1, x_0)
 \end{aligned} \tag{4}$$

The MWR2MM Algorithm [3]

$S = 0$ all words of S are initialized

$$\begin{aligned}
 &\text{For } i = 0 \text{ to } m - 1 \\
 &\quad (C, S^{(0)}) := x_i Y^{(0)} + S^{(0)} \\
 &\quad \quad \text{If } S_0^{(0)} = 1 \text{ then} \\
 &\quad \quad (C, S^{(0)}) := (C, S^{(0)}) + M^{(0)} \\
 &\quad \quad \text{For } j = 1 \text{ to } e - 1 \\
 &\quad \quad (C, S^{(j)}) := C + x_i Y^{(j)} + M^{(j)} + S^{(j)} \\
 &\quad \quad \quad S^{(j-1)} := (S_0^{(j)}, S^{(j-1)}_{w-1 \dots 1})
 \end{aligned} \tag{5}$$

3.1 Montgomery Multiplier Architecture

1. Depending on bit a_0 , the first multiplexer MUX21 passes 0 or content of register B . Depending on r_0 ,
2. MUX22 passes 0 or contents register of M .
3. ADDER1 delivers the sum $R + a_i * B$ while ADDER2 gives $R + N$.
4. SHIFT REGISTER1 provides bit a_i and is right-shifted for each I so that $a_0 = z a_i$.
5. The shifting and loading operations of shift registers are synchronized by a controller (Fig. 1).

Data dependency graph shows the parallelism of the algorithm among different loops. Data dependency of MWR2MM algorithm is given below (Fig. 2).

Here, task A is computed up to i th iteration and task B is computed up to j th iteration. It has $i + 1$ tasks per column. Every column is computed by a separate PE.

The figure shows the pipeline stages of the processing element. One column represents one processing element and the pipelining is used to connect the previous processing element to the next processing element. Processing element (PE) is used to compute the column of the graph and the data is pipelined to the next stage of PE [4] (Figs. 3 and 4).

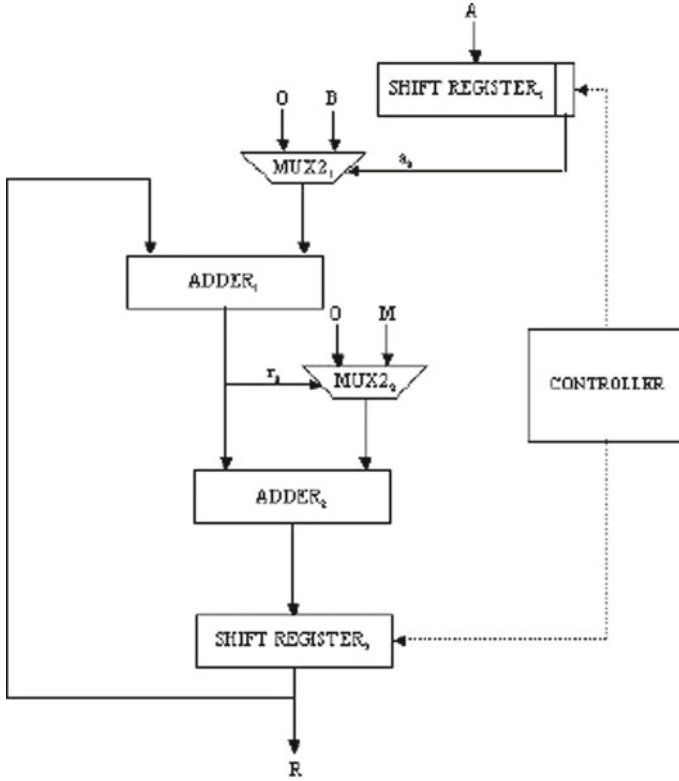


Fig. 1 Architecture of the Montgomery multiplier [5]

Data path and control unit are included in the processing element. It takes $S(J)$, $M(J)$, and $Y(J)$ as inputs from the previous pipeline stage and generates the new output $S(J-1)$. The signal is passed from the interstage register and reaches to the next PE one clock cycle later. It means, when the present PE is working on j th iteration, the next PE is working on $j-2$ iteration. Given below is the processing element data path (Figs. 5 and 6).

In Fig. 7, the pipeline is named KERNEL and it has got a number of processing elements. Other blocks are used as data conversion, memory, and control unit. The previous processing element in the pipeline stage received a word to the next processing element. To store data, RAM is used. To ensure flexibility, the memory of the system is designed in the form of a queue. The maximum length of the queue (q_{\max}) depends on the number of stages (p) in the pipeline and max number of words that can be stored in the memory (e_{\max}). This length is determined as

$$Q_{\max} = \begin{cases} e_{\max} + 2 - L_p & \text{if } (e_{\max} + 2) > L_p \\ 1 & \text{otherwise} \end{cases} \quad (6)$$

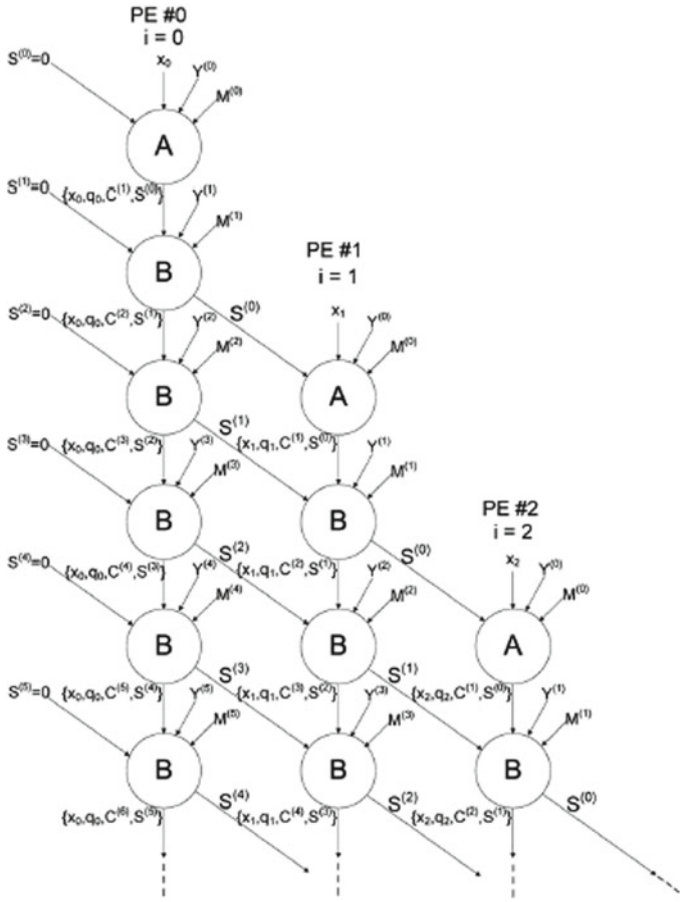


Fig. 2 Graph of data dependency graph (MWR2MM)

4 Experimental Results

Simulation results are shown in Figs. 8 and 9 of the multiplier and Montgomery cell, respectively. Internal architecture of the Montgomery multiplier in the RTL view is shown in Fig. 10 as below.

Table 1 shows the power calculation of the design for Montgomery multiplication; the power is compared with this work: in one case, it was reduced to 65.17%. On comparing to other sources, it was reduced to 78.53, 76.47, and 59.1%.

Modular operation is a fundamental operation in many popular Public Key Cryptography (PKC) algorithms. As the division operation in modular reduction is time-consuming, Montgomery proposes a new algorithm which avoids the division operation. The input is given in binary form and output will also be in binary representation. An integer X is represented as $X * R \text{ mod } (N)$ where N is the modulo and $R = 2^k$ is a

Fig. 3 Example of the pipeline stage

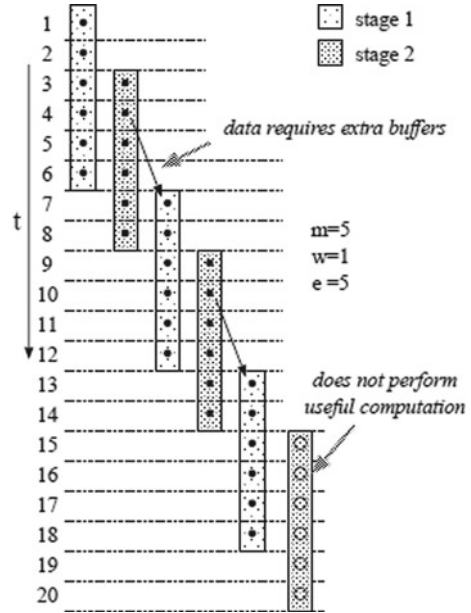
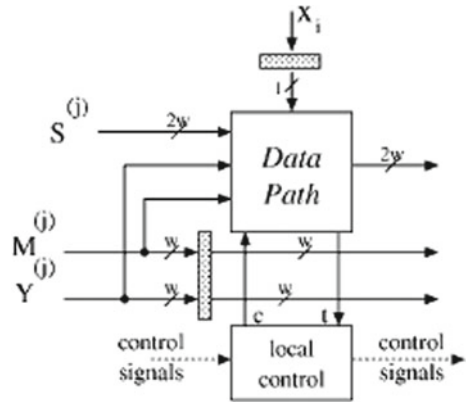


Fig. 4 Block diagram of PE



radix and co-prime to N . A high-performance and low-power design of Montgomery multiplier is presented in this work.

Figure 11 illustrates different design techniques that were used to reduce the power consumption for low-power applications.

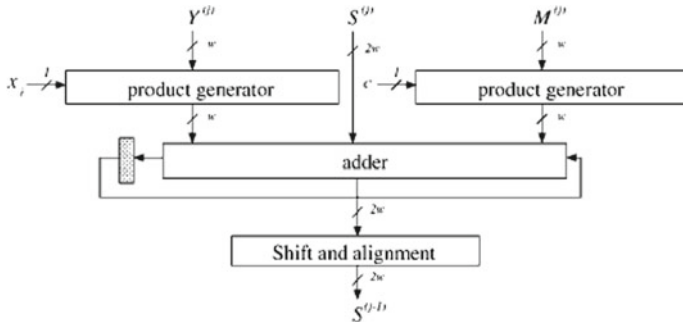


Fig. 5 Block diagram of the data path

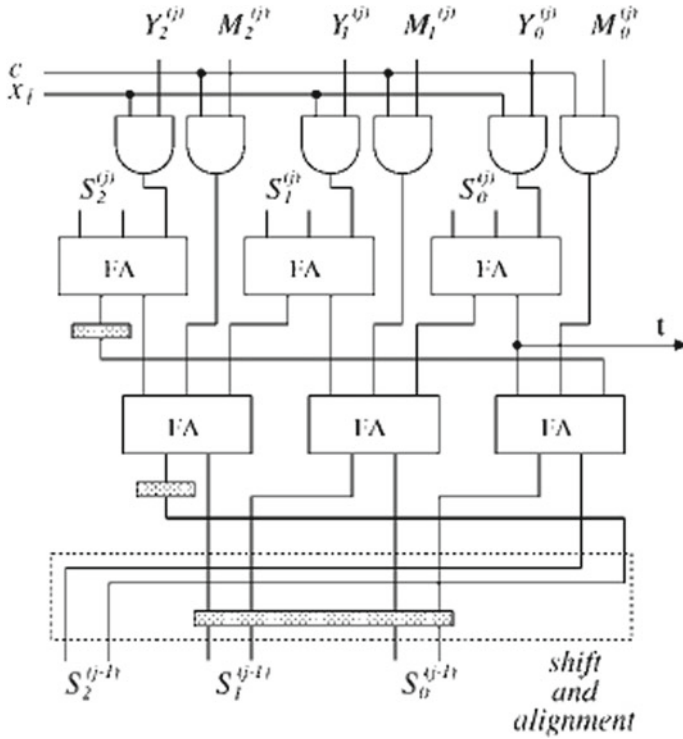


Fig. 6 Logic diagram for $w = 3$

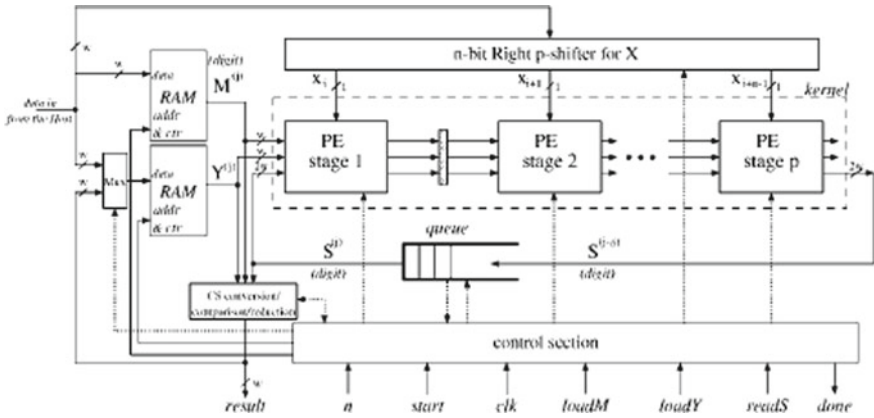


Fig. 7 Pipelined organization

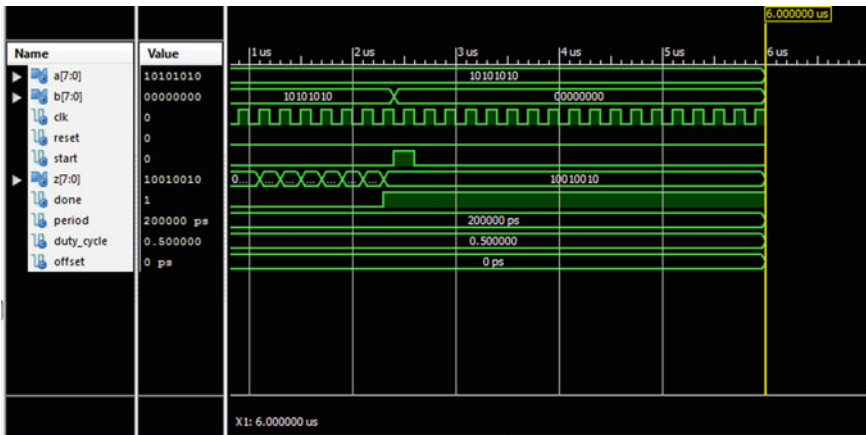


Fig. 8 Simulation results of the multiplier

Finally, we get a multiplier which consumes less power by the clocking approach. We can also reduce the power by reducing the supply voltage but this is a much better approach because in that case, throughput is much affected on reducing the supply voltage. This design is much suitable for low-power and high-performance applications.

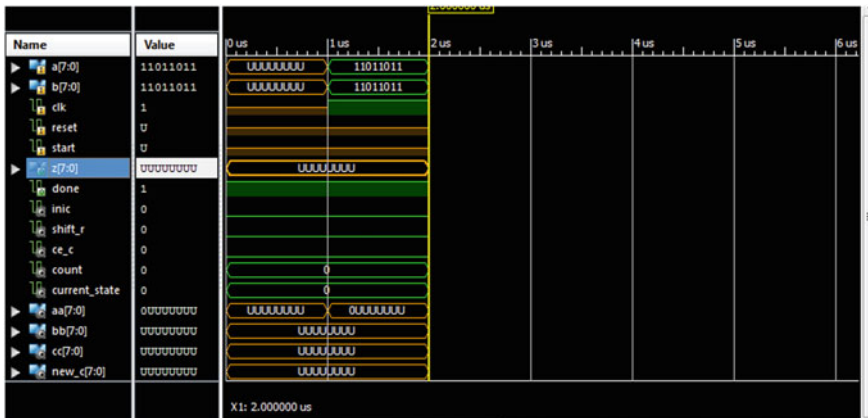


Fig. 9 Simulation results for the Montgomery cell

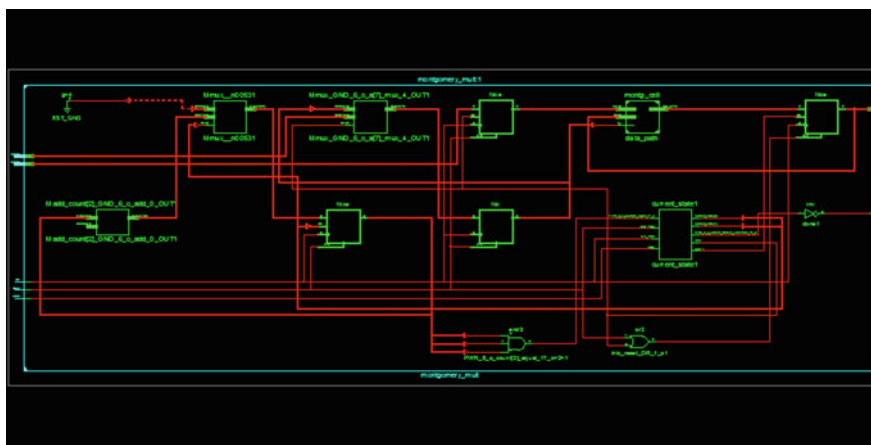


Fig. 10 Internal architecture of a Montgomery multiplier (RTL view)

Table 1 Power calculation of the design for Montgomery multiplications

S. No.	Design	Power (mW)	Power reduction (%)
1.	High speed and low latency [16]	40.20	–
2.	Scalable Arch. MM [6]	65.23	65.17
3.	RSA Cryptography [15]	59.52	78.53
4.	Low latency and low bandwidth [7]	34.23	76.47
5.	Proposed work	14	59.1

$$\text{Reduction \%} = (\text{Power} - \text{Proposed work}) / \text{Power} \times 100$$

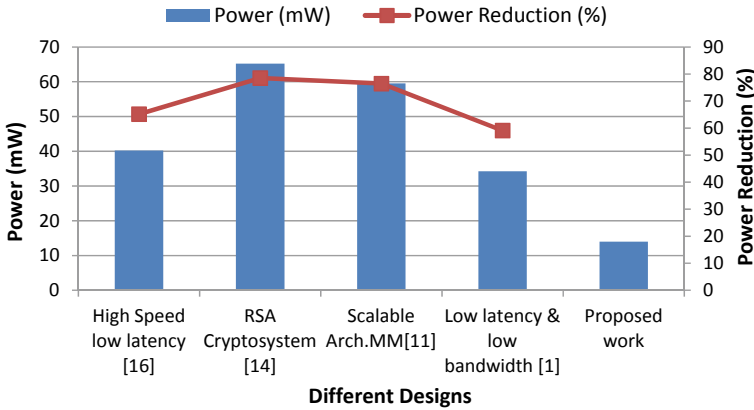


Fig. 11 Different techniques to reduce power consumption

5 Conclusion

The power analyzer tool XPower Analyzer is used to calculate the power of the Montgomery modular multiplier. The estimated power of the multiplier is reduced by 14 mW by reducing the clock period 20 ns–10 ns with the same duty cycle. Although we can further reduce the power of the multiplier, the speed of the multiplier will also be affected by reducing the clock period. The methodology for low-power design can be applied from system level to device level. The advantage of this technique is that, without changing the hardware or replacing any circuit and component and increasing area density which also may be more expensive, we can reduce the power of the multiplier.

References

1. A. Miyamoto, N. Homma, T. Aoki, A. Satoh, Systematic design of RSA processors based on high-radix montgomery multipliers. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **19**(7), 1136–1146 (2011)
2. Y. Gong, S. Li, High-throughput FPGA implementation of 256-bit montgomery modular multiplier, in *Second International Workshop on Education Technology and Second International Workshop on Education Technology and Computer Science* (2010), pp. 173–177
3. A. Ibrahim, F. Gebali, H. El-Simary, A. Nassar, High-performance, low-power architecture for scalable radix 2 montgomery modular multiplication algorithm. *Can. J. Elect. Comput. Eng.* **34**(4), (Fall 2009)
4. G. Gaubatz, Versatile Montgomery Multiplier Architectures, in April, 2002
5. M.M. Sandoval, C.F. Uribe, R. Cumplido, I.A. Badillo, An area/performance trade-off analysis of a GF(2 m) multiplier architecture for elliptic curve cryptography. *Elsevier-Comput. & Electr. Eng.* **35**(1), 54–58 (2007)
6. C. McIvor, M. Mc Loone, J.V Mc Canny, High speed, low latency RSA decryption silicon core, *IEEE*, 133–136 (2003)

7. W.-C. Lin, J.-H. Ye, M.-D. Shieh, Scalable montgomery modular multiplication architecture with low-latency and low-memory bandwidth requirement. *IEEE Trans. Comput.* **63**(2), 475–483 (2014)
8. A. Ibrahim, F. Gebali, H. Elsimary, New and improved word-based unified and scalable architecture for radix 2 montgomery modular multiplication algorithm. *IEEE* (2013)
9. H. Nozaki, M. Motoyama, A. Shimbo, S. Kawamura, Implementation of RSA Algorithm Based on RNS Montgomery Multiplication, (Corporate Research and Development Center, Toshiba Corporation 1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki, Japan, 2001), pp. 364-376
10. X. Wang, P. Noel, T. Kwasniewski, Low power design techniques for a montgomery modular multiplier, in *Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems* (2005), pp. 449–452
11. C. McIvor, M. Mc Loone, J.V Mc Canny, High-radix systolic modular multiplication on reconfigurable hardware, in *IEEE*, (Fall 2005); A.F. Tenca, C.K. Koc, A Scalable architecture for modular multiplication based on montgomery's algorithm. *IEEE Trans. Comput.* **52**(9), (2003)

Performance Study of PMN-PT/Ni-Based Multiferroic Nanomagnet for Straintronics Device Applications



Amlan Sen, Sampreet Sarkar and Ankush Ghosh

Abstract Recent developments in the field of spintronics/straintronics create possibilities to replace current digital logic by nanomagnetic memory logic. This would be a promising candidate for highly dense and low-power applications. Magnetization orientations of nanomagnets are used to encode the digital logic bits. Nonvolatile and non-leakage features of nanomagnets make the system energy efficient. In the present work, the authors studied the performance of a PMN-PT-based multiferroic nanomagnet using the micromagnetic simulation software OOMMF. This work would pave way for potential applications of nanomagnet-based straintronics in modern devices.

Keywords Spintronics · Straintronics · Multiferroic nanomagnet · OOMMF

1 Introduction

While the general method in CMOS-based VLSI fabrication used till date to shrink complex circuitry on to smaller pieces of silicon, is likely to hit the bedrock pretty soon, due to the effects of quantum phenomena and other effects. Low power is also an essential criterion for developing electronic devices. On the other hand, the heat generated by the latest SOC circuits is unmanageable with available heat sink technology. As a consequence, in order to continue scaling devices down, we have to look beyond the traditional approach and rely on some entirely new technologies that can resolve all of these issues simultaneously. Spintronics or straintronics [1–3] is one of them where physical movement of electrons is not required, thereby generating lesser heat and requiring lower power to be operated. Nanomagnets are good option for making straintronic devices [4]. The underlying physics behind these devices is that spins are stored in a ferromagnetic body through interaction and then transferred the angular momentum through an electric but nonmagnetic conductor into another

A. Sen · S. Sarkar · A. Ghosh (✉)

Robotics Engineering Department, The Neotia University, Sarisha 743368, West Bengal, India
e-mail: ankushghosh@gmail.com

© Springer Nature Singapore Pte Ltd. 2020

L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_2

ferromagnet. The nanomagnets are nonvolatile and have no leakage. Hence, the nanomagnetic system used to encode logic bit is an energy-efficient system.

Therefore, straintronics-based nanomagnetic structures may be used for making logic devices. The benefits of nanomagnets are not just structural, but can also be seen in the evaluation and execution of logic [4]. For a single domain nanomagnet having information carriers we can effectively make use of the exchange interactions between the electrons and holes, and treat it as a single spin element, thus providing us with one degree of freedom. Taking into account the static error probability (p) of random switching for sudden variation in thermal conditions, we have a minimum energy dissipation of $K\ln(1/p)$ for nonadiabatic switching [4]. For a CMOS transistor, this expression goes up to $NK\ln(1/p)^2$ for switching [5]. This goes on to show that nanomagnetic logic-based transistors are extremely energy efficient when compared to CMOS logic switches. A “clock” is used to make the logic operations synchronized to the magnetization orientations of the magnets to produce desired output in response to one or more inputs. Various types of logic gates can be implemented with different dipole interactions between nearby magnets and can be connected through logical wire of magnet arrays. In this way any types of circuits can be implemented.

To make it more energy efficient, highly energy-efficient method of switching magnets has been put forward. Here, a two-phase multiferroics nanomagnet, which is made of a piezoelectric layer and is magnetoelastically coupled with the magnetostrictive layers, can be switched by introducing a very small potential difference to the piezoelectric layer [4]. This voltage would generate a strain on the piezoelectric layer along one axis and carried on to the magnetostrictive layer through magnetoelastic coupling. This transfer of voltage can be achieved in two ways, either by the d_{31} or the d_{33} coupling coefficient of the piezoelectric. In d_{31} coupling, we restrict the piezoelectric to expand or contract perpendicular to the applied field by mechanically clamping the multiferroics. As a result, these nanomagnets with uniaxial anisotropy can encode two-state logic levels and, thereby, can be realized for any digital applications. This method reports the energy dissipation in the order of 100 KT with 1 ns delay [6] which is much lower than any modern-day technology.

Straintronics-based logic gate is gaining significant interest as it can process and store information simultaneously. It provides flexible designing of computer architecture by eliminating the need to refresh clock cycles, ultimately reducing energy dissipation. Due to its non-volatility it can improve the system reliability and eliminate the delay of a computer. However, the digital logic gate should require satisfying all the requirements. The proposed structure is based on principle of straintronic nanomagnets, which are very energy efficient as well as fast and hence can provide more functionality on a chip area.

In this paper, the authors studied the performance of a PMN-PT-based multiferroic nanomagnet dimensions of $\sim 100 \text{ nm} \times 100 \text{ nm}$ using nanomagnetic simulation software OOMMF [7]. This analytical study provides important insight view of magnetization direction and property which leads the way for potential development in the field of straintronics nanomagnetic logic technologies.

2 Theory

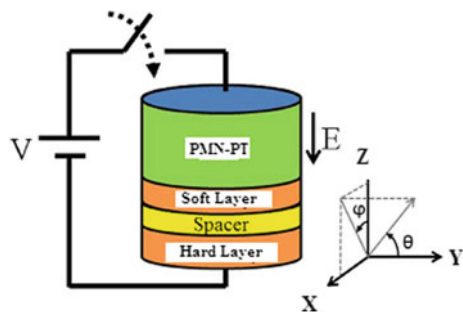
Our proposed model [8] is a two-stage system of multiferroic nanomagnet. It comprises a piezoelectric layer and a magnetostrictive layer as depicted in Fig. 1. The magnetostrictive layer is generally thinner compared to the piezoelectric layer. In such kinds of nanomagnets, the magnetoelectric properties are observed due to the amalgamation of the electric and magnetic effects. If a small potential difference is applied across the magnet, it generates a deformation, which is completely transferred over to the thin magnetostrictive layer. This results in a change in the direction of magnetization.

For spintronic devices, multiferroic materials are used since they exhibit multiple ferroic orders. Therefore, high-density memory and multifunctional devices can be implemented with this. The composite structures consisting of magnetic and piezoelectric materials supersede the single-phase multiferroics because of high magnetoelectric effect. The multiferroic materials studied here is a two-stage system consisting of a piezoelectric layer in upon a thin ferromagnetic layer. In this system, the electric coupling and magnetic interaction make magnetoelectric effect. When an electrostatic potential is applied to the piezoelectric layer, the strain is generated due to change in shape, which is transferred to the magnetostrictive layer that change its magnetization orientation. The magnetization of an isolated magnet can be rotated by applying strain in its hard axis because it moves the energy minimum of the magnet from the easy to the hard axis due to Villari effect. The magnetization will continue to rotate even if the strain is withdrawn and the final magnetization will settle up to the nearest easy axis. This completes rotation by an angle of 180° for two state magnets and 90° for four state magnets.

For our proposed model, we have chosen single-crystal PMN-PT (Lead Magnesium Niobate-Lead Titanate) as our piezoelectric layer as depicted in Fig. 1 [9].

Piezoelectric materials are increasingly used as a means for electromechanical coupling in small-scale systems. Single-crystal PMN-PT is used due to its better electromechanical coupling coefficients. Using modified Bridgman technique, single-crystal PMN-PT can be grown. This material exhibits better piezoelectric properties (e.g., $d_{33} = 3500$ pC/N, $d_{31} = -1200$ pC/N, $k_{33} = 0.95$, etc.) which

Fig. 1 Basic structure of a straintronic device with multiferroic nanomagnet



exceed the PZT material by a factor of 4–5. Therefore, the integration is possible in the miniaturized devices not only in the harvesting stage but also with the traditional electronic circuit.

Aside from being compatible with Silicon micromachining process, the single crystal has higher S/N ratio, better sensitivity compared to its microcrystal counterpart. It becomes very easy to miniaturize the devices using traditional silicon fabrication technology. The magnetostrictive layer has three unique axes of magnetization direction—the easy axis along the $\langle 1\ 1\ 1 \rangle$ direction, the medium axis along the $\langle 1\ 1\ 0 \rangle$ direction, and the hard axis along the $\langle 1\ 0\ 0 \rangle$ direction. If we dissect the crystal along the x-y plane of the nanomagnet, we find the direction of the axes as shown in Fig. 1. Now, if we refer to the four stable directions using a two-bit combination [00, 01, 10, 11], these are also the four ground states of the multiferroic nanomagnet. When we apply a stress to the nanomagnet, the Cartesian coordinate axis is rotated by an angle of $(\theta - \pi/4)$ degrees, about the normal axis. After the removal of the stress, the system settles in the new ground state.

We have used the shape anisotropy technique for creating multiple easy axes. Recent studies show that there is a major and predictable change in the magnetization direction of concave nanomagnets for a minor variation in parameters like width and radius [10].

In our proposed model, we have studied cylindrical nanomagnets, with their symmetric anisotropy as well as different micromagnetic switching modes. The micromagnetic modeling is carried out with respect to the energy equation in Eq. 1, and we have utilized the Object Oriented MicroMagnetic Framework (OOMMF) [11] to simulate the nanomagnet. Using the aforementioned software, we have visualized the interactions between the magnetic moments and the switching of magnetization direction in the cylindrical nanomagnet. In order to study the behavior of the nanomagnets, we are considering the relevant energy expressions such as exchange energy, magnetostatic anisotropy, stress anisotropy, and external magnetic field.

The magnetization M given as the combination of magnetic moments by magnetic structure:

$$\vec{M} = \frac{\sum m}{V} \quad (1)$$

where V is the volume of any nanomagnet (only that of the magnetostrictive layer) and m is magnetic moment and the magnetization (M) in the SI units is A/m.

The magnetic susceptibility χ , can be introduced as the magnetization divided by the absolute value of the magnetic field:

$$\chi = \frac{|\vec{M}|}{|\vec{H}|} \quad (2)$$

The magnetic permeability μ defined as

$$\mu = \frac{|\vec{B}|}{|\vec{H}|} \quad (3)$$

where H is the magnetic field and is measured in Henry per meter (SI) and B is the magnetic induction or magnetic flux density. It depends on magnetic field intensity H and the magnetization M , and is given by

$$\vec{B} = \mu_0(\vec{H} + \vec{M}) \quad (4)$$

where, $\mu_0 = 4\pi \times 10^{-7} (H/m)$ is the permeability in vacuum.

The total energy of the nanomagnet can be expressed as [9]

$$\begin{aligned} E_{total}(\theta_2) = & \frac{\mu_0}{4\pi R^3} [M_s^2 \Omega^2] [-2 \cos \theta_2 (\cos \theta_3 + \cos \theta_1)] \\ & + \sin \theta_2 (\sin \theta_3 + \sin \theta_1)] \\ & + \frac{K_1 \Omega}{4} \cos^2(2\theta_2) - \frac{3}{2} \lambda_{100} \sigma \Omega \cos^2(\theta_2 - \frac{\pi}{4}) \\ & + \frac{\mu_0}{4\pi} [M_s \Omega] H_{applied} \sin \theta_2 \end{aligned} \quad (5)$$

Here θ_2 is the angle of output magnetization vector with the positive x axis, θ_1 and θ_3 are the input magnetization vector angles with positive x axis. In the energy equation mentioned above, the first term is the dipole interaction energy of the output magnet among the adjacent input magnets. The second term is a representation of the energy in magnetocrystalline anisotropy, where K_1 is constant for magnetocrystalline anisotropy. The third term indicates stress anisotropy energy for applied stress along the [100] direction. The stress applied is σ , in the direction being λ_{100} . The final term is the energy of the external static bias magnetic field $B_{applied}$ pointing in the [110] direction where μ_0 is the permeability of free space and M_s is the saturation magnetization. The stress factor σ will be positive for strain and negative for compression.

3 Simulation and Result Discussion

For our proposed model, we have studied the performance analysis by magnetic hysteresis and magnetization vector patterns. We have studied the effects of magnetization for applied strain upon magnetic field and the switching characteristics of the nanomagnet. Figures 2, 3, 4, and 5 depict the magnetization patterns for our proposed piezoelectric-magnetostrictive heterostructures.

Fig. 2 Cross-sectional view of the magnetization patterns for the proposed the piezoelectric-magnetostrictive heterostructures

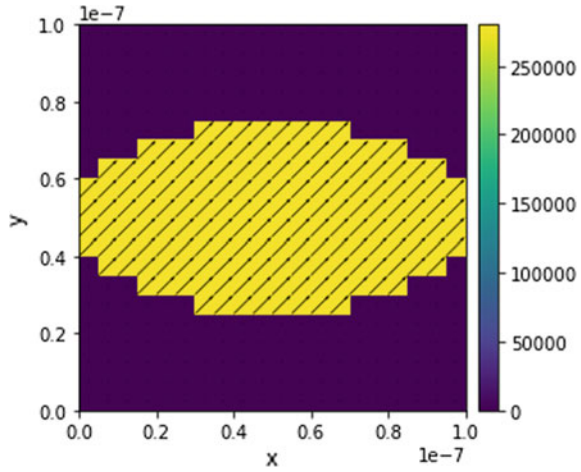
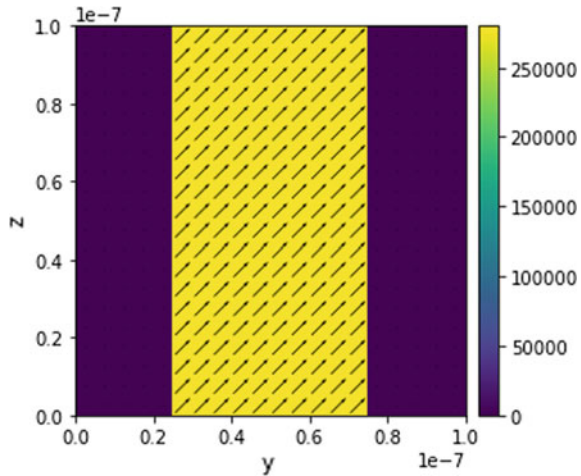


Fig. 3 Sectional view of the magnetization patterns for our proposed the piezoelectric-magnetostrictive heterostructures



In particular, we have closely observed the magnetization hysteresis (anisotropy field), switching coherence, and magnetization dynamics. This knowledge will help us reach an optimal shape which is the best for nanomagnet-based computing.

We have performed micromagnetic simulations in OOMMF [7] to study when the magnetization exchange occurs in nanomagnets. For this purpose, we had to verify the magnetization hysteresis for our proposed model. The results are shown in Fig. 6, provide a clear picture of the normalized hysteresis loop for the multiferroic structure when a magnetic field is applied along the x direction.

Fig. 4 Cross-sectional view of the magnetization patterns with applied potential for the proposed the piezoelectric-magnetostrictive heterostructures

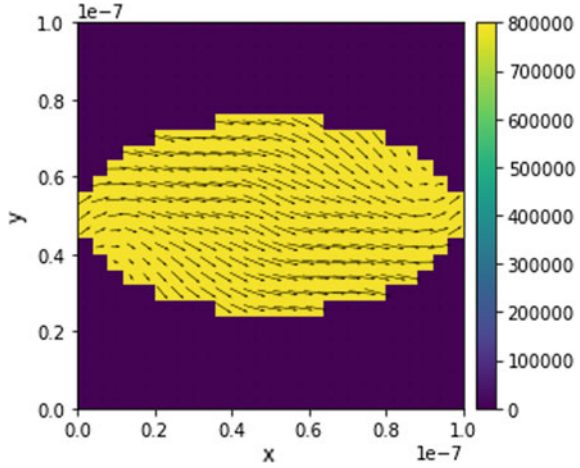
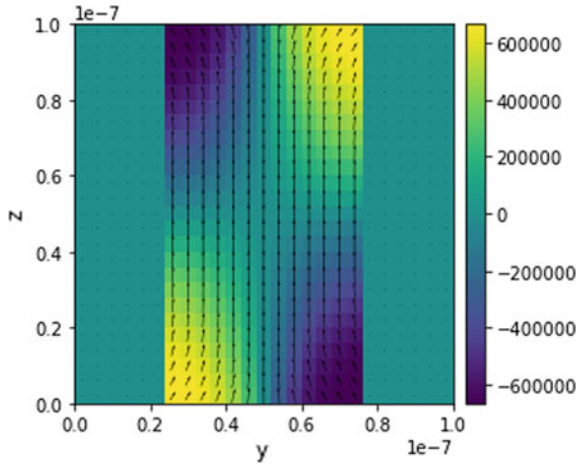


Fig. 5 Sectional view of the magnetization patterns with applied potential for the proposed the piezoelectric-magnetostrictive heterostructures



4 Energy Dissipation

The energy dissipation for a single flip for our proposed PMN-PT nanomagnet has been presented here. The value of d_{31} constant of PMN-PT piezoelectric is 1300 pm/v [12], which would require an electric field of 4 kV/cm. Considering maximum strain at PMN-PT layer as 500 ppm, it can transfer maximum stress of 100 MPa in the nanomagnet. The relative permittivity of PMN-PT layer is 1000 [12], so for 100 nm diameter capacitance of the PMN-PT layer is $\cong 1.75$ fF. A voltage pulse of ± 16 mv will be required to generate successive tension and compression stress cycle for a 100 nm magnet (Fig. 7).

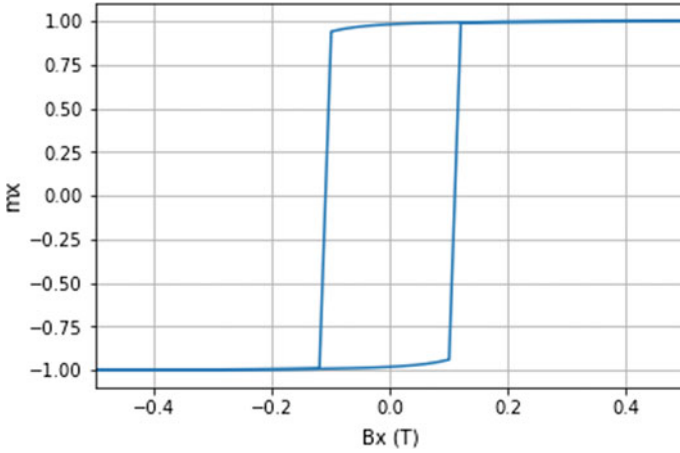


Fig. 6 Magnetization hysteresis (m-B) curves for our proposed the piezoelectric-magnetostrictive heterostructures

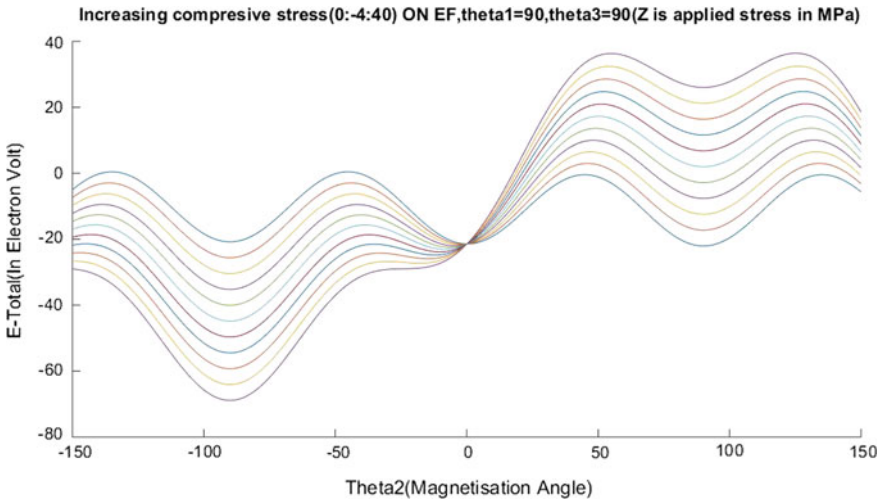


Fig. 7 Energy plots of the output nanomagnet with magnetization angle θ_2

Therefore, the energy dissipated in charging the capacitance abruptly with square wave pulse is $\frac{1}{2} CV^2$ for an event. So for four-state event, total energy of $2CV^2$ will be dissipated per gate operation which is $\cong 2150$ kT which is considerably lower.

There is some additional internal energy dissipation in the magnet due to Gilbert damping constant. But it is comparatively negligible as long as switching frequency belongs to 100 MHz to 1 GHz [12]. Internal dissipation with higher switching frequency can be dealt with elsewhere.

5 Conclusion

In this paper, we have studied the performance analysis by realizing the magnetic hysteresis and magnetization vector patterns for our model with PMN-Pt and Ni using the nanomagnetic simulation software OOMMF and its performance on reliability, magnetization, and switching coherence. This study has potential to realize four-state nanomagnet with PZT piezoelectric and multiferroics nanomagnet (Ni) with biaxial anisotropy. By using PMN-PT piezoelectric, energy dissipation will reduce drastically. This type of nanomagnetic logic increases the device density by fourfold over the conventional devices. It can also be implemented for real time applications such as image recognition and processing.

This work would pave the way for potential applications of straintronics in modern devices as well as can face the challenges for miniaturization of devices according to Moore's Law.

However, there are some challenges associated to build up straintronics switching. These are fabrication challenges, difficulties in retaining piezoelectricity in small scales, generation of uniaxial stress, and to transfer strain accordingly. These problems are intriguing, and hopefully would be resolved by further research and study on this topic.

References

1. M.S. Fashamin, K. Roy, J. Atulasimha, S. Bandyopadhyay, *Nanotechnology* **22**, 155201 (2011)
2. M.S. Fashami, J. Atulasimha, S. Bandyopadhyay, *Nanotechnology* **23**, 105201 (2012)
3. A. Imre, G.H. Bernstein, W. Pord, V. Metlushko, *IEEE Trans. Nanotech.* **1**, 209 (2002)
4. N. D'Souza, J. Atulasimha, S. Bandyopadhyay, *IEEE Trans. Nanotech.* **11**, 896 (2012)
5. S. Bandyopadhyay, B. Das, A.E. Miller, *Nanotechnology* **5**, 113 (1994)
6. S. Bandyopadhyay, M. Cahay, Electron spin for classical information processing: a brief survey of spin-based logic devices, gates and circuits. *Nanotechnology* **20**, 412001 (2009)
7. M.J. Donahue, D.G. Porter, OOMMF User's Guide, Version 1.0, Interagency Report NISTIR 6376 (1999)
8. A. Sarkar, S. Halim, A. Ghosh, S.K. Sarkar, J. *Nanoelectron. Optoelectron.* **12**, 1–6 (2017)
9. A. Sarkar, P.K. Dutta, A. Ghosh, S. Roy, S.K. Sarkar, Implementation of universal gates (NAND) based on nanomagnetic logic using multiferroics. *Quantum Matter* **5**, 505–509 (2016)
10. I.A. Ivan, J. Agnus, M. Rakotondrabe, P. Lutz, N. Chaillet, PMN-PT piezoelectric material and related applications in silicon integrated devices like microactuators and energy harvesters, in *IEEE International Semiconductor Conference, CASII*, Sinaoia, Romania (2011)
11. M.S. Fashami, N. D'Souza, J. Magn. Mater. **438**, 76–84 (2017)
12. N. D'Souza, A. Jayasimha, S. Bandyopadhyay, *J. Phys. D Appl. Phys.* **44**, 265001 (2011)

An Enhanced Recycling Folded Cascode Amplifier



Praveen Soni and Shweta Gautam

Abstract A recycling folded cascode amplifier comprising a local common-mode feedback technique with NMOS, working in a triode region is presented. The proposed amplifier shows enhanced performance as compared to the recycling folded cascode amplifier and OTA based on the local common-mode feedback structure with increased gain, GBW without additional power dissipation. The proposed amplifier performance is verified by the LTspice simulator tool with TSMC 180 nm CMOS process parameters. Performance of this amplifier when compared with CFRFC amplifier shows 202.9% GBW, 15.1 dB dc gain improvement with 30 pF load. The proposed amplifier has low fabrication cost and less verification time.

Keywords Operational transconductance amplifier · Common-mode feedback recycling folded cascode (CFRFC) · Transconductance · GBW improvement · Verification · Optimization

1 Introduction

The advanced research in CMOS technologies has made the portable electronic device's market grow due to the single-chip integration of digital and analog building blocks. So the silicon area and power consumption are important parameters of a design. The operational transconductance amplifiers (OTA) are important analog blocks in switched capacitor circuits with required precise final value and low settling time [1]. Hence, large gain bandwidth, large slew rate, and high dc gain are the main characteristics of efficient OTA.

P. Soni (✉) · S. Gautam

Department of Electronics and Communication Engineering, Netaji Subhas University of Technology, Sector-3, Dwarka 110078, New Delhi, India
e-mail: praveen3773@gmail.com

S. Gautam

e-mail: shwetauiet11@gmail.com

© Springer Nature Singapore Pte Ltd. 2020

L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_3

The PMOS input folded cascode (FC) has lower input common-mode level, higher nondominant poles, and lower flicker noise, so it is preferred over NMOS folded cascode [2]. The recycling folded cascode amplifier (RFC) is preferred over conventional folded cascode OTAs due to its high GBW, gain, and slew rate [3, 4].

In the previous work, multi-path schemes have been used to improve the performance of CFC [5, 6]. To enhance the slew rate and output impedance, another multi-path technique was applied to the OTA of three-current-mirror [7]. To use class AB operation, another scheme is used [8]. However, for high-speed applications these schemes were not suitable because the OTA's transfer function comprises many pole-zero pairs of low frequency.

Double recycling, positive feedback, and current shunt techniques have also been used to increase the slew rate and transconductance of RFC as mentioned in [9–11]. In double recycling, the input signal is recycled two times, so the slew rate and gain bandwidth improve [9]. In the positive feedback technique, current mirror loads using positive feedback are formed to increase the recycling structure's gain [10]. With current shunt technique, ac and dc paths of recycling arrangement are separated by cross-coupled current mirrors to increase the performance of RFC [11].

In [12], a zero is generated by the high-speed current mirror to cancel the first nondominant pole, so the phase margin of RFC is improved. Although In local common-mode feedback technique using a resistor in recycling folded cascode amplifier(CFRFC) the dc gain, GBW and slew rate is increased but the active area consumption is high and verification is also not easy [13–16].

In the proposed enhanced recycling folded cascode amplifier (ERFC) amplifier with common-mode feedback technique using NMOS (for common-mode feedback) the dc gain and GBW is increased by using an optimum value of aspect ratio of NMOS working in triode region. Due to the use of NMOS instead of resistor, the active area consumption will be less. The verification will be also easy because verification corners will be less, so this saved time may be used for the optimization. Since in the fabrication, no extra mask is required, so the fabrication cost will be less because the cost depends on the number of masks [17–21].

This paper is arranged as follows: Conventional folded cascode (CFC), Recycling folded cascode (RFC), and OTA based on the local common-mode feedback structure (CFRFC) are discussed in Sect. 2. The proposed enhanced recycling folded cascode amplifier (ERFC) is discussed in Sect. 3. The simulation results of these OTAs are discussed in Sect. 4. The proposed ERFC amplifier is concluded in Sect. 5.

2 CFC, RFC, and CFRFC Amplifier

In the CFC amplifier shown in Fig. 1, most of the current is conducted by transistors M3 and M4 so these transistors show the highest transconductance. M3 and M4 provide a folding node to the small current generated by input drivers M1 and M2. In CFC, M3 and M4 work as driving transistors [3, 4].

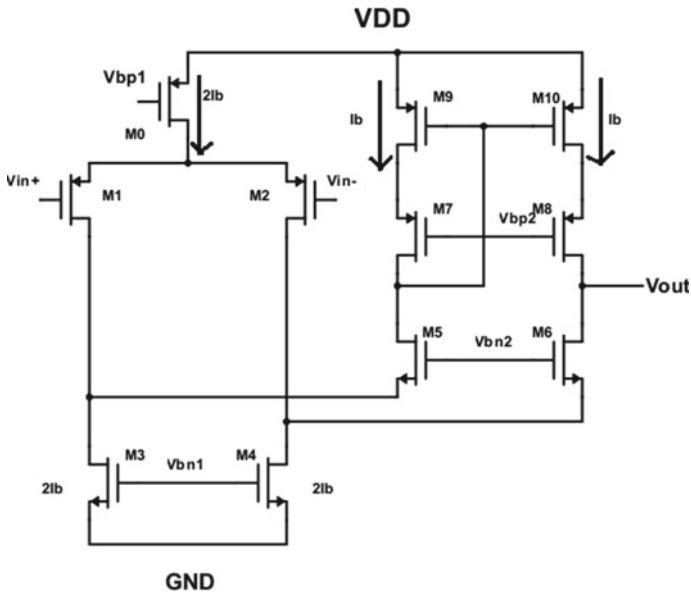


Fig. 1 The conventional folded cascode (CFC) amplifier

The driver transistors M1 and M2 are split to make transistors M1a(b) and M2a(b) in Fig. 2. Now the fixed and equal currents ($I_b/2$) flow through these transistors. Apart from this, transistors M3a(b) and M4a(b) form current mirrors of the ratio of K:1 by

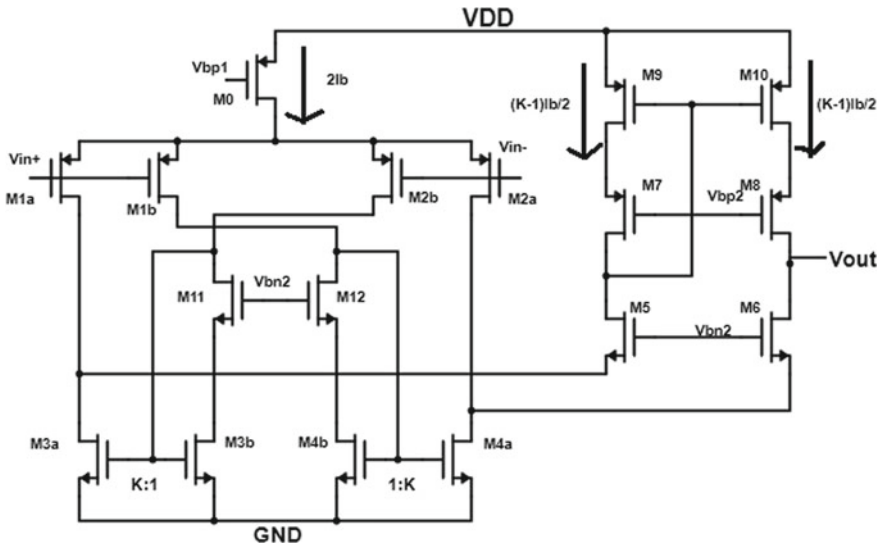


Fig. 2 The recycling folded cascode (RFC) amplifier

splitting M3 and M4. The cross-over connection is made by diode-connected pair M3b and M4b and drives M3a and M4a in such a way that the sources of M5 and M6 get the current in phase [3, 4].

W/L ratio of M11 and M12 are equal to M5 and M6 and M11 and M12 are added to maintain the voltage at the drain of M3a(b) and M4a(b) equal.

In the W/L ratio for M3, M3a, and M4, M4a is 4:3 due to this current being added correctly. So in Fig. 2, the value of factor K for the current mirror is 3. The transistors M1a(b) and M2a(b), M3a and M4a, M3b and M4b, M5 and M6, M7 and M8, M9 and M10 are matched.

In RFC, all transistors are working in the saturation region, so the drain current is

$$I_D = \frac{1}{2} \mu C_{ox} \frac{W}{L} (V_{GS} - V_T)^2 \tag{1}$$

The equivalent transconductance of RFC is given by [3, 4]

$$G_{mRFC} = g_{mM1a}(1 + K) \tag{2}$$

Since $g_{mM1} = 2g_{mM1a}$, for $K = 3$ transconductance of RFC is double than that of the CFC for the same current consumption.

In CFRFC shown in Fig. 3, local feedback structure substitutes the current mirror load of input pairs M1(2)b which include resistors R_1 and R_2 . Here these two resistors are matched [13].

So $R_1 = R_2 = R$.

The feedback common-mode voltage from the drain terminal of transistors M3(4)b is applied to transistors (M3b and M4b) gate. No current flows in R_1 and R_2 under quiescent condition and node voltage of X, Y, and Z will be the same, which is equal to

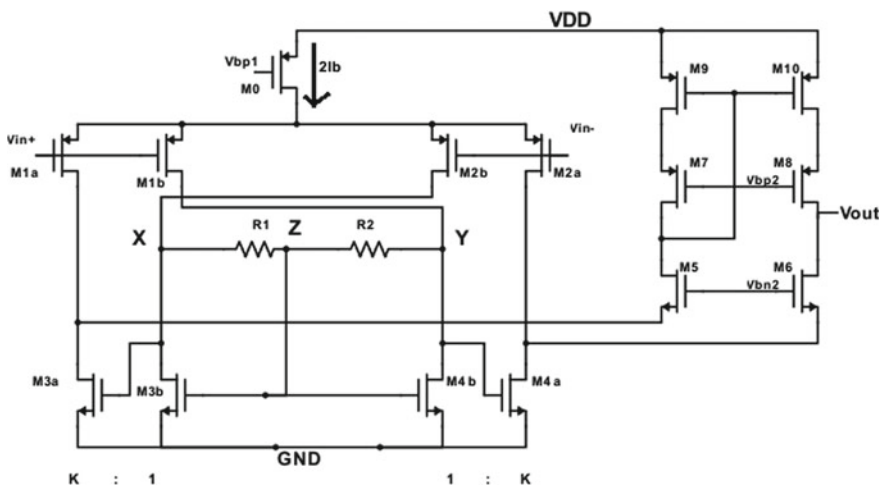


Fig. 3 The common-mode feedback recycling folded cascode (CFRFC) amplifier

$$V_X = V_Y = V_Z = V_{THN} + \sqrt{\frac{I_b}{\beta_{M3(4)b}}} \tag{3}$$

where V_{THN} is the threshold voltage of transistor M3(4)b and $\beta_{M3(4)b}$ is the transconductance factor of transistor M3(4)b. The quiescent current in transistors M3(4)a and M3(4)b is given by

$$\frac{I_{QM3(4)a}}{I_{QM3(4)b}} = \frac{\left(\frac{W}{L}\right)_{M3(4)a}}{\left(\frac{W}{L}\right)_{M3(4)b}} \tag{4}$$

If the value of this ratio is $K = 3$, CFRFC and RFC will consume the same power [13–16].

3 Proposed ERFC Amplifier

Figure 4 shows a diagram of the proposed ERFC amplifier including local common-mode feedback technique comprising the NMOS transistors (M11 and M12) working in triode region instead of resistance as used in CFRFC of Fig. 3. Here M11 and M12 are matched. Due to NMOS working in the triode region, active area consumption will be less and the verification will also be easy. It will be of low cost in technology terms because the number of masks required in fabrication will be less [17].

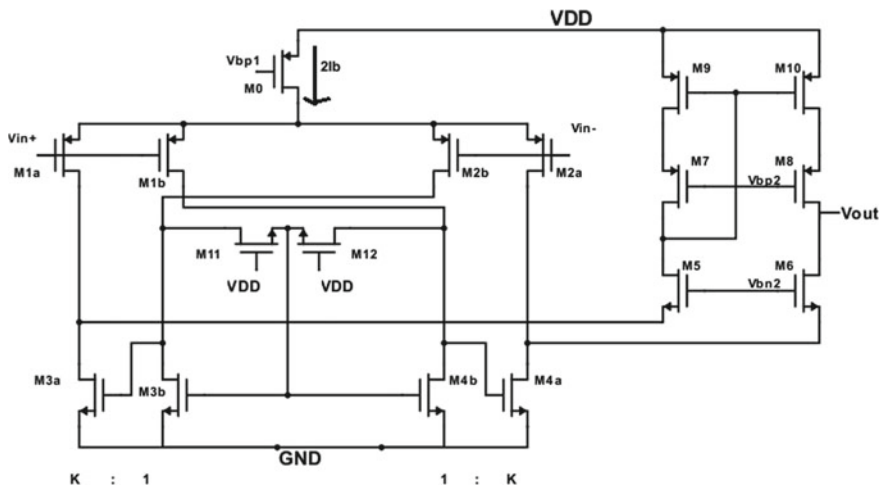


Fig. 4 The enhanced recycling folded cascode (ERFC) amplifier

The condition for the NMOS transistors M11 and M12 working in triode region is

$$V_{DS} < V_{GS} - V_{TH} \quad (5)$$

In triode region, the drain current for NMOS is

$$I_D = K'_n \frac{W}{L} \left[(V_{GS} - V_T) V_{DS} - \frac{1}{2} V_{DS}^2 \right] \quad (6)$$

where $K'_n = \mu_n C_{ox}$ is the process transconductance parameter and W/L , V_{GS} , V_{DS} , and V_T have their usual meaning.

For small values of V_{DS} , the approximation of Eq. (6) is

$$I_D \cong K'_n \frac{W}{L} [(V_{GS} - V_T) V_{DS}] \quad (7)$$

$$r_{DS} = \left[K'_n \frac{W}{L} (V_{GS} - V_T) \right]^{-1} \quad (8)$$

when V_{DS} is small.

To keep the transistors M11 and M12 in the triode region, gate of these transistors are connected to V_{DD} .

3.1 Small Signal Transconductance

In the proposed ERFC OTA, for small signal, ac current flows through transistors M11 and M12 and node Z works as a virtual ground. So the expressions for G_m is given by

$$G_{mERFC} \approx [1 + g_{m4a}(r_{ds1b} || r_{ds4b} || r_{ds12})] g_{m2a} \quad (9)$$

$$G_{mERFC} \approx [1 + g_{m4a} r_{ds12}] g_{m2a}$$

where r_{ds} is drain-to-source resistance of transistors. For the same active area as in CFRFC, with optimum value of aspect ratio for transistors M11 and M12 which work in triode region, we can achieve larger value of r_{ds12} as compared to R ; so the value of $g_{m4a} r_{ds12}$ will be more than K , due to this considerable increment that occurred in G_{mERFC} as compared to that of CFRFC.

3.2 Gain Bandwidth

The increased gain bandwidth of ERFC amplifier due to increment in transconductance is given by

$$G_{BW} \cong \frac{(1 + g_{m4a}r_{ds12})g_{m2a}}{C_L} \quad (10)$$

where C_L is the load capacitance.

3.3 DC Gain

The equivalent output resistance of the proposed ERFC is given by

$$R_{oERFC} \cong [g_{m8}r_{ds8}r_{ds10}] || [g_{m6}r_{ds6}(r_{ds4a} || r_{ds2a})] \quad (11)$$

Since equivalent output resistance of ERFC amplifier is equal to the output resistance of CFRFC amplifier, the dc open-loop gain is given by

$$A_{dc} \approx (1 + g_{m4a}r_{ds12})g_{m2a}R_{oERFC} \quad (12)$$

So due to increment in G_{mERFC} , the ERFC shows enhanced dc gain [18–21].

4 Simulation and Results

To compare the proposed ERFC with CFRFC, these two OTAs (CFRFC and ERFC) are simulated on the LTspice simulator tool in TSMC 180 nm CMOS technology. The supply voltage is 1 V and the bias current I_b is 10 μ A. The load capacitor C_L is 30 pF. For both OTAs, similar transistors have the same W/L ratio.

Figures 5 and 6 show the open-loop ac response of CFRFC and ERFC; thus, the dc gain of CFRFC and ERFC amplifiers is 80.3 dB and 95.4 dB, respectively. The GBW achieved by CFRFC and ERFC is 10.2 MHz and 30.9 MHz, respectively; so by using the optimum value of W/L for M11 and M12 for given active area, large transconductance is achieved.

The ERFC shows 15.1 dB higher dc gain and 202.9% higher GBW than that of CFRFC for the same power dissipation and without any stability penalty. The characterization results of these two amplifiers are summarized in Table 1.

To know the proposed ERFC OTA's performance in this paper, Table 2 summarizes the performance parameter of ERFC with prior current recycling OTAs like double recycling, positive feedback and current shunt mentioned in the previous papers.

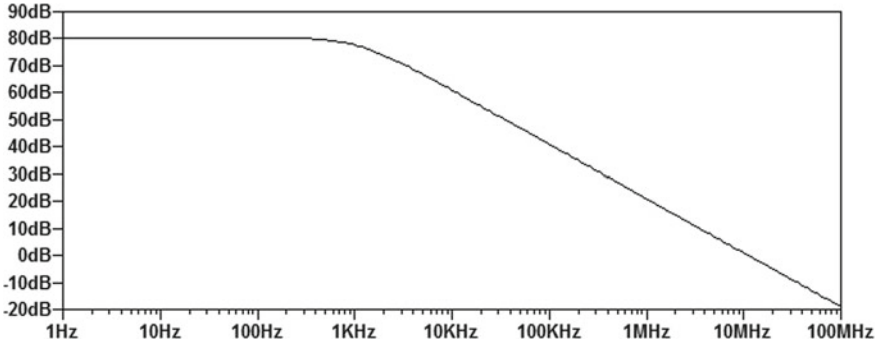


Fig. 5 Open-loop AC response (CFRFC amplifier)

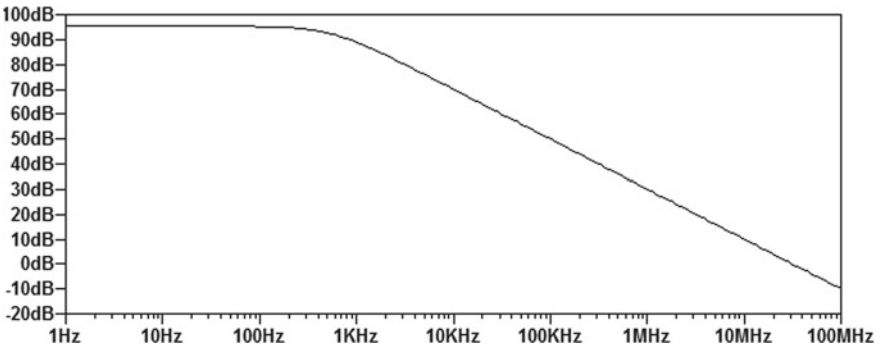


Fig. 6 Open-loop AC response (ERFC amplifier)

Table 1 Amplifier characterization results

Parameter	CFRFC	ERFC
Bias current (μA)	40	40
Supply (V)	1	1
Capacitive load (pF)	30	30
DC gain (dB)	80.3	95.4
GBW (MHz)	10.2	30.9
FoM ₁ (MHzpF/mA)	7650	23,175
R ₁ , R ₂ (Ω)	50 K	–

Since these OTAs are designed for different bias current, supply voltage, capacitive load, and the simulation were done with different processes parameters. So to know the performance figure of merit (FoM) is used, which is given by

$$FoM_1 = GBW * C_L / I_{Qtotal}$$

Table 2 Comparison between different OTAs

Parameter	[4]	[9]	[10]	[11]	ERFC
Bias current (μA)	800	800	110	260	40
Supply (V)	1.8	1	1.2	1.2	1
Technology (nm)	180	65	65	130	180
Capacitive load C_L (pF)	5.6	10	8	7	30
GBW (MHz)	134.2	203.2	63.7	83	30.9
DC gain (dB)	60.9	54.5	87.8	70.2	95.4
FoM ₁ (MHzpF/mA)	939.4	2540	4633	2235	23,175

As per the comparison in Table 2, it is observed that RFC's energy efficiency is least. The double recycling [9], positive feedback [10], and current shunt [11] show improved efficiency as compared to RFC. The proposed ERFC OTA shows the improved dc gain, GBW, and FoM1 than that of other OTAs. That proves the performance enhancement of ERFC OTA.

5 Conclusion

An ERFC OTA is proposed in this paper to enhance the transconductance by using MOS transistor working in triode region instead of resistance as used in CFRFC. By using the optimum value of aspect ratio for NMOS, large value of transconductance is achieved for the given active area. Due to this, the ERFC improves 15.1 dB dc gain and 202.9% GBW with the same power dissipation and without any stability penalty as compared to the CFRFC amplifier. The optimization of ERFC amplifier will be high and its fabrication cost will also be low.

References

1. P.Y. Wu, V.S.-L. Cheung, H.C. Luong, A 1-V 100-MS/s 8-bit CMOS switched-opamp pipelined ADC using loading-free architecture. *IEEE J. Solid-State Circuits* **42**(4), 730–738 (2007)
2. K.S. Lee, S. Kwon, F. Maloberti, A power-efficient two-channel time-interleaved $\Sigma\Delta$ modulator for broadband applications. *IEEE J. Solid-State Circuits* **42**(6), 1206–1215 (2007)
3. R. Assaad, J. Silva-Martinez, Enhancing general performance of folded cascode amplifier by recycling current. *IET Electron. Lett.* **43**(23) (2007)
4. R.S. Assaad, J. Silva-Martinez, The recycling folded cascode: a general enhancement of the folded cascode amplifier. *IEEE J. Solid-State Circuits* **44**(9), 2535–2542 (2009)
5. K. Nakamura, L.R. Carley, An enhanced fully differential folded-cascode op amp. *IEEE J. Solid-State Circuits* **27**(4), 563–568 (1992)
6. J. Adut, J. Silva-Martinez, M. Rocha-Perez, A 10.7 MHz sixth-order SC ladder filter in 0.35 μm CMOS technology. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **53**(8), 1625–1635 (2006)

7. J. Roh, High-gain class-AB OTA with low quiescent current. *Springer J. Analog. Integr. Circuits Signal Process.* **47**(2), 225–228 (2006)
8. L. Yao, M. Steyaert, W. Sansen, A 1-V 140 μ W 88-dB audio sigma-delta modulator in 90-nm CMOS. *IEEE J. Solid-State Circuits* **39**(11), 1809–1818 (2004)
9. Z. Yan, M. Pui-In, R.P. Martins, Double recycling technique for folded-cascode OTA. *Springer Analog Integr. Circ. Signal Process.* **71**, 137–141 (2012)
10. X. Zhao, H. Fang, J. Xu, A transconductance enhanced recycling structure for folded cascode amplifier. *Springer Analog Integr. Circ. Signal Process.* **72**, 259–263 (2012)
11. Y.L. Li, K.F. Han, X. Tan, N. Yan, H. Min, Transconductance enhancement method for operational transconductance amplifiers. *IET Electron. Lett.* **46**(19) (2010)
12. X. Zhao, H. Fang, J. Xu, Phase margin enhancement technique for recycling folded cascode structure. *Springer Analog Integr. Circ. Signal Process.* **74**, 479–483 (2013)
13. X. Zhao, Q. Zhang, Y. Wang, M. Deng, Transconductance and slew rate improvement technique for current recycling folded cascode amplifier. *Elsevier Int. J. Electron. Commun.* 1–5 (2015)
14. S. Baswa, A.J. Lopez-Martin, J. Ramirez-Angulo, R.G. Carvajal, Low-voltage micro power super class AB CMOS OTA. *IEE Electron. Lett.* **40**(4) (2004)
15. A.J. Lopez-Martin, S. Baswa, J. Ramirez-Angulo, R.G. Carvajal, Low-voltage super class AB CMOS OTA cells with very high slew rate and power efficiency. *IEEE J. Solid-State Circuits* **40**(5), 1068–1077 (2005)
16. J.A. Galan, A.J. Lopez-Martin, R.G. Carvajal, J. Ramirez-Angulo, C. Rubia-Marcos, Super class-AB OTAs with adaptive biasing and dynamic output current scaling. *IEEE Trans. Circuits Syst.-I: Regul. Pap.* **54**(3), 449–457 (2007)
17. B. Razavi, *Design of Analog CMOS Integrated Circuits* (Tata McGraw-Hill, 2002)
18. P.E. Allen, D.R. Holberg, *CMOS Analog Circuit Design*, 2nd edn. (Oxford University Press, Oxford, 2012)
19. K.S. Mo, Y. Leblebici, *CMOS Digital Integrated Circuits* (Tata McGraw-Hill, 2002), pp. 48–82
20. M.P. Garde, A. Lopez-Martin, R.G. Carvajal, J. Ramirez-Angulo, Super class-AB recycling folded cascode OTA. *IEEE J. Solid-State Circuits* 1–6 (2018)
21. M. Akbari, S. Biabanifard, S. Asadi, M.C.E. Yagoub, Design and analysis of dc gain and transconductance boosted recycling folded cascode OTA. *Elsevier Int. J. Electron. Commun.* 1–6 (2014)

A New Approach for Bias–Variance Analysis Using Regularized Linear Regression



M. Rajasekhar Reddy, B. Nithish Kumar, N. Madhusudana Rao
and B. Karthikeyan

Abstract Bias–variance is one of the tools to learn the performance of any machine learning algorithm. Various bias–variance models can be observed using regularized linear regression. In this paper, we implement regression on a sample dataset along with some optimization technique. Here, we use a technique called polynomial linear regression on the same dataset to increase the fit and the results will be normalized. Earlier studies show that though the polynomial linear regression is applied, the plot is very complex and it will drop-off at extremes at most of the situations. Regularization is one of the techniques used to optimize and narrower the gaps. These properties vary from one dataset to another and we find an optimistic value of parameter lambda. The error of overfitting from the plotted curve is significantly reduced. Again it is plotted between error obtained and the values of lambda. The main goal of this paper is to avoid the overfitting problem. Cross-validation set is taken and helped in estimation of error and deciding the parameters work best in the model.

Keywords Linear · Regression · Overfitting · Bias–variance · Regularized linear regression · Bias–variance

M. R. Reddy (✉) · B. N. Kumar · B. Karthikeyan
School of Computing, SASTRA Deemed to Be University, Thanjavur, India
e-mail: rajasekharmanyan04@gmail.com

B. N. Kumar
e-mail: bommireddynithishkumar@gmail.com

B. Karthikeyan
e-mail: mbalakarathi@gmail.com

N. M. Rao
Department of Mathematics, School of Engineering, Amrita Viswa Vidyapeetham, Coimbatore,
India
e-mail: madhu031083@gmail.com

1 Introduction

Regularization is the optimization technique used in reducing the error and prevent the outliers. Earlier studies show that the estimation of minimax through mean-square criterion and examined [1, 2]. We append a regularization parameter at the end of the cost function. A keen study about regularization in the interface perspectives was done before [3]. There was an amazing development in the results of the study on regularization from the last ten years [4]. Many supervised learning techniques like regression are used in detection process [5]. A learning algorithm is trained and bias–variance properties of the algorithm are analyzed through using learning curves [6].

Validation process is very important on how much the error which has produced abnormal results [7]. Bias–variance is the major error component in a learning algorithm. Training with high amount of data would not be better if the algorithm has high bias [8], while in the case of variance more training for the algorithm will become useful. Cost function is applied for training as well as cross-validation sets. Underfitting (bias) is defined where $J_{\text{train}}(\theta)$ is higher and $J_{\text{cv}}(\theta)$ can be approximately equal with $J_{\text{train}}(\theta)$ and overfitting is where $J_{\text{train}}(\theta)$ is lower and $J_{\text{cv}}(\theta)$ can be higher than $J_{\text{train}}(\theta)$ [9]. The process of training or testing is done by learning parameter θ from training data and test errors are computed. Test set is used for calculating error after training the algorithm with training set. The index of the hypothesis function ($h(x)$) is varied and error is calculated at every instance with various values of “ m ” (m = number of training examples). There is tremendous increase in the prediction of many areas like liver disease analysis, cancer prediction, etc. [10].

2 Methods Used and Models

2.1 Linear Regression

The output of a regression problem will be continuous. Discrete values are obtained with a classification problem. We utilize the univariate linear regression since we are going to predict real value output. A hypothesis function (two parameters) is used for the calculation of cost function J .

$$(h_{\theta}(p^{(i)})) + q^{(i)})^2 \quad (1)$$

$$\frac{\partial y}{\partial x} \frac{1}{2m} \sum_{i=0}^m (h_{\theta}(p^{(i)}) - q)^2 \quad (2)$$

We calculate two variables θ_0 and θ_1 using gradient descent function.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta} J(\theta_0, \theta_1) \quad (\text{for } j = 0 \text{ and } j = 1) \quad (3)$$

Suitable θ_0 and θ_1 values are calculated with a minimum cost function.

2.2 Polynomial Regression

The hypothesis function does not fit with the data at the beginning. The behavior of the plotted curve with hypothesis function varies with order. The strength between independent and dependent parameters is matched with this polynomial regression.

$$h_\theta(p) = \theta_0 + \theta_1 p_1 \quad (4)$$

Additional features are produced based on p_1 to get the quadratic function or the cubic function.

$$h_\theta(p) = \theta_0 + \theta_1 p_1^0 + \theta_2 p_1^2 + \theta_3 p_1^3 \quad (5)$$

2.3 Overfitting and Bias–Variance

$$J(\theta) = \frac{1}{2m} \sum_{i=0}^m (h_\theta(p^{(i)}) - q^{(i)})^2 \approx 0 \quad (6)$$

Training the algorithm with more data can be overfitting due to irrelevancy. Underfitting is defined as an algorithm which will not be able to model the training data and then generalize to new data.

Bias. An algorithm studies to model on training with the dataset and the goal is estimating the target functions. The predicted error can be either high bias/variance. If there are only few assumptions about the target function model then it is called high bias whereas in the case of low bias lot of assumptions occur about the form of model.

Variance. This is a problem which occurs when our learning algorithm is not good at predicting data that was not in the training set. It means if the data slightly deviates from existing dataset it cannot predict.

To address the overfitting there are only two ways. One is to reduce the number of features and the other is to use regularization.

2.4 Dataset

In this paper, we employed the dataset with the composition of samples between the flow of water from the dam and the respective level. Samples are taken randomly in this experimental dataset. It is spitted in three sets, i.e., training, cross-validation, and testing sets.

The training set will have attributes of p , q , and the cross-validation set p_{val} and q_{val} . Testing set attributes are p_{set} and q_{set} .

3 Proposed Technique: Regularized Linear Regression

Regularization is an optimization technique used for avoiding overfitting problem. The parameter θ_0 is removed from the rest of the parameters in the gradient descent function since θ_0 is not penalized.

$$\begin{aligned}
 &\text{Repeat} \\
 &\{ \\
 &\quad \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m ((h_0(p^{(i)})) - q^{(i)}) x_0^{(i)} \\
 &\quad \theta_j := \theta_j - \alpha \left[\left(\frac{1}{m} \sum_{i=1}^m ((h_0(p^{(i)})) - q^{(i)}) p_0^{(i)} \right) + \frac{\lambda}{m} \theta_j \right] \\
 &\quad j \in \{1, 2, 3, \dots, n\} \\
 &\}
 \end{aligned} \tag{7}$$

This term $\frac{\lambda}{m} \theta_j$ implements optimization.

A high biased model will not be complex and remains underfit and a model with high variance overfit the training data.

4 Proposed Technique: Regularized Linear Regression

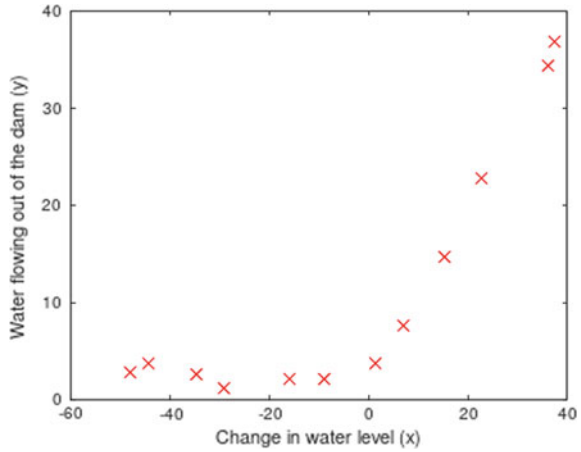
We load and visualize the data and plotted before training the algorithm.

The plot of the given dataset produces Fig. 1.

$$J = (1/(2 * m)) * \text{sum}(\text{power}((a * \text{theta} - b), 2)) + (\text{lambda}/(2 * m)) *$$

Regularization term is not applied here ($\text{lambda} = 0$). Regularized linear regression gradient is calculated for setting the best fit of parameters.

Fig. 1 Plot of the dataset

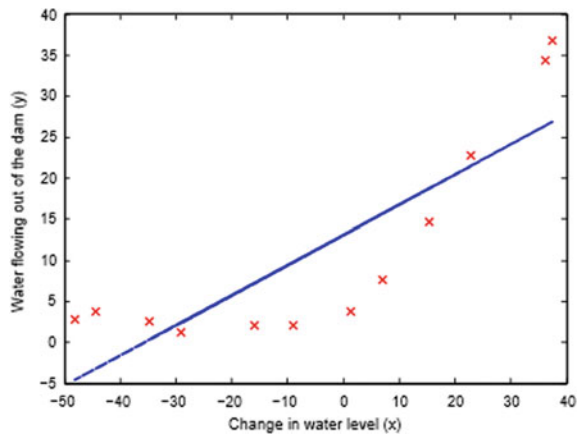


$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(p^{(i)}) - q^{(i)}) p_j^{(i)} \quad \text{for } j = 0 \tag{8}$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m} \sum_{i=1}^m (h_{\theta}(p^{(i)}) - q^{(i)}) p_j^{(i)} \right) + \frac{\lambda}{m} \theta_j \quad \text{for } j \geq 1. \tag{9}$$

The regularization parameter— λ is set to zero initially. As we are implementing regression to fit for a 2-d θ , regularization does not help much for this θ with lower dimension. Later, polynomial regression is executed with regularization. The plotted line (best fit) clearly portrays that the model is not able to make a better fit since the data has a nonlinear pattern. Best fit is visualized to debug the learning algorithm (Fig. 2).

Fig. 2 Linear regression



We need a tool to debug a learning and therefore we use learning curves and generated here. Learning curve will be determined by the function of training set size and the curves will be plotted as training and cross-validation errors. Those errors are returned in the form of a vector for both training and cross-validation sets. Those curves are plotted on taking the training and cross-validation sets with various sizes. Suppose a training set is taken with size—p; the top p samples are taken a(1:p), b(1:p). The learning parameters θ are used to calculate errors on both sets. The training error is calculated by

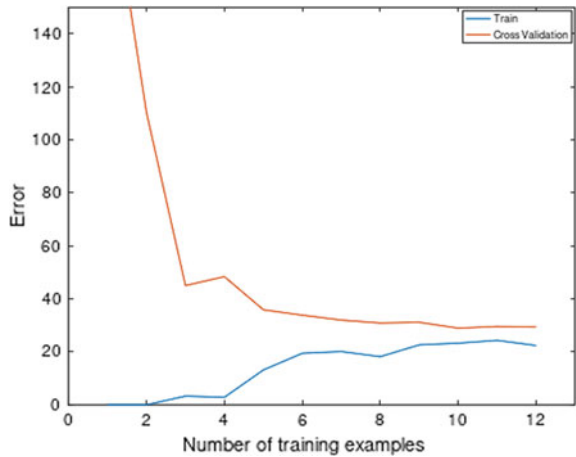
$$J_{train}(\theta) = \frac{1}{2m} \left[\sum_{i=0}^m (h_{\theta}(a^{(i)})) - b^{(i)} \right]^2 \tag{10}$$

Remember that the training error here is calculated without optimization. Cost function with lambda value set to zero is used for the computation of training and cross-validation error. There will be no partitions done in the cross-validation set and the error is calculated. Those errors are stored in vectors error train and error Val. The graph of the learning curves is shown in Fig. 3.

From the observations, both the training and cross-validation error increases with increase in training examples. It shows there exists a high bias for the given model and cannot fit with the dataset well. Thus, polynomial regression helps to pass through the data points better than previous fit. There is a clear observation that the model taken has high bias, i.e., underfitting. So by adding more features and hypothesis function is calculated. The hypothesis function in terms of polynomial regression is given below.

$$\begin{aligned} h_{\theta}(x) &= \theta_0 + \theta_1 * (\text{Level}_1) + \theta_2 * (\text{Level}_2) + \dots + \theta_p * (\text{Level}_p) \\ &= \theta_0 + \theta_1 a_1 + \theta_2 a_2 + \dots + \theta_p a_p. \end{aligned} \tag{11}$$

Fig. 3 Learning curves for linear regression



Here, $a_1 = (\text{Level}_1)$,
 $a_2 = (\text{Level}_2), \dots, a_p = (\text{Level}_p)$.

Higher indices of the real value (level) are added from the dataset.

The linear regression cost functions used in training the algorithm with polynomial regression. The terms with higher degree are converted to features (here polynomial degree = 8) and feature normalization is applied for scaling the features better in the projected data. The learning curves are plotted in Figs. 4 and 5. From Fig. 4, the polynomial plot is able to pass all the inputs in a better manner with low training error. While plotting data points, the curve may drop-off at extremes since the polynomial fit is too complex. Still there is no generalization in an efficient way. The learning curves show that though the training error is low, the cross-validation error remains high. The gap between the two curves in plot shows that there is a heavy variance

Fig. 4 Polynomial regression fit (Lambda = 0.0000)

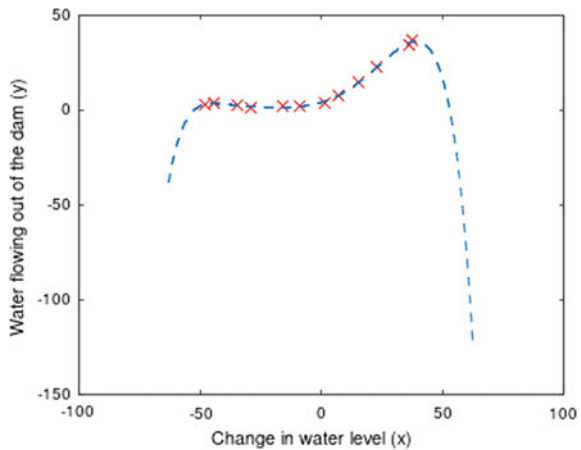


Fig. 5 Polynomial regression learning curve (Lambda = 0.0000)

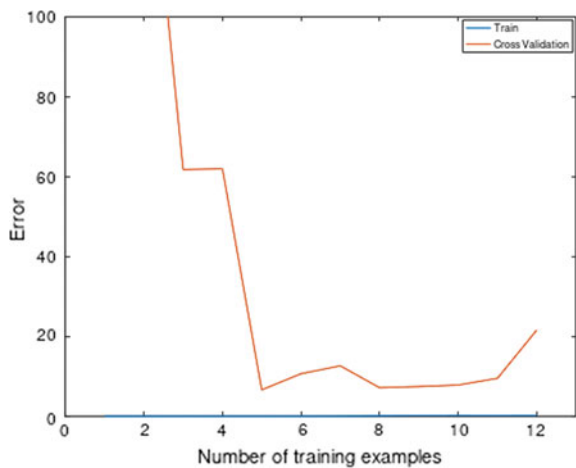


Table 1 Error in the training set and cross-validation set

No of training examples	Train set error	Cross-validation set error
1	00	205.1210960
2	00	110.3003660
3	3.2865950	45.0102310
4	2.8426780	48.3689110
5	13.1540490	35.8651650
6	19.4439630	33.8299620
7	20.0985220	31.9709860
8	18.1728590	30.8624460
9	22.6094050	31.1359980
10	23.2614620	28.9362070
11	24.317250	29.5514320
12	22.3739060	29.4338180

in the model. Different sample sizes were taken and the error in the training set and cross-validation set are calculated and tabulated (Table 1).

The entire dataset is normalized for better scaling of features. Standardization is a method useful in avoiding numerical issues. Normal least squares are invariant while some methods like lasso and ridge regression do not and in this case, there is no need for standardization but sometimes standardization is probably useful.

Now the algorithm is optimized by adding the regularization term to the existing model. At the beginning the same polynomial regression is applied with $\lambda = 0$. There is a simultaneous decrease in both errors and resulted in a better fit (Tables 2 and 3).

The regularization parameter affects the bias–variance properties of the model in the regularized polynomial regression. The parameter lambda is varied between 1 and 100 and the script will generate learning curve to these values. For $\lambda = 1$, the polynomial curve passes along the data points more likely (Fig. 6) and the learning

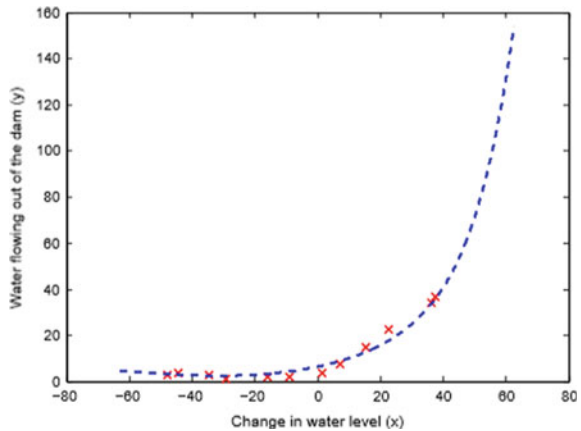
Table 2 Normalized values

Normalized values:
01
−0.3621410
−0.7550870
0.1822260
−0.706190
0.3066180
−0.5908780
0.3445160
−0.5084810

Table 3 Error in the training set and cross-validation set

Training examples	Train set error	Cross-validation set error
1	00	160.72190
2	00	160.121510
3	00	61.7548250
4	00	61.9288950
5	00	6.5988470
6	00	10.6348430
7	0.0183030	12.6052540
8	0.0694230	7.1255250
9	0.1449210	7.418038
10	0.1600040	7.800762
11	0.1468310	9.489085
12	0.1736230	21.578808

Fig. 6 Polynomial regression (Lambda = 1.0000)



curve (Fig. 7). Thus, both the cross-validation and training errors will be converted to a comparatively lower value. We have achieved a trade-off between high bias–variance. The polynomial fit (Fig. 8) does not fit the data points well with $\lambda = 100$. High level of regularization on a model cannot fit the training data is the reason for the case.

On increasing the training samples, the rate of error is constantly decreased from the cross-validation set. Also, there is a great decrease in the training set and the distance between the two learning curves decreased indicating the decrease in variance gently. Clearly, this happens due to increase in the degree of hypothesis function. Though the plot passes all the data points exactly, a difference is seen at the extremes of the graph and does not fit the model. Hence, regularization is applied with varying lambda until the model fits exactly.

Fig. 7 Polynomial regression learning curve (Lambda = 1.0000)

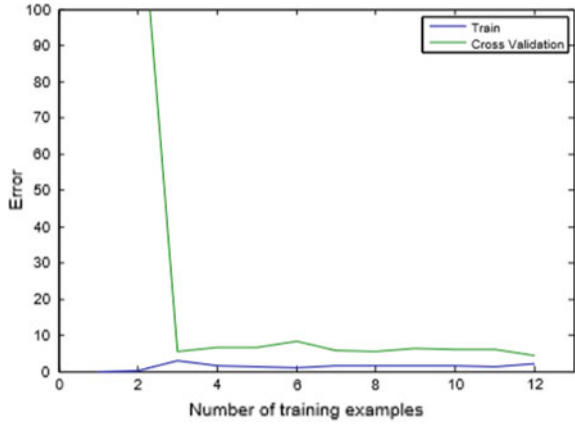
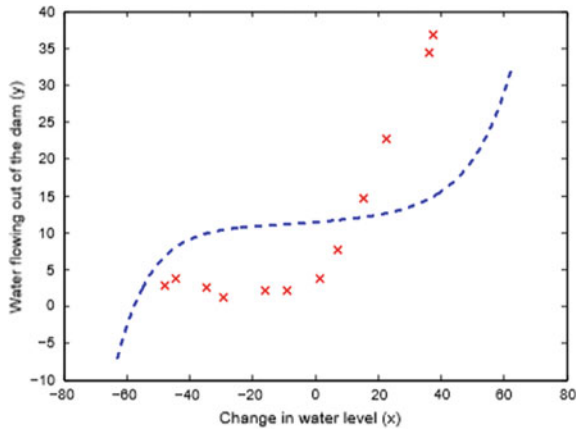


Fig. 8 Polynomial regression (Lambda = 100.000)



From the previous experiments different values for λ gives varying results of regularized polynomial regression. Here, the model is perfectly generalized optimally whereas a non-regularized model fails. In converse high regularization ($\lambda = 100$) may or may not fit the training set. The performance of the original uncovered data after selecting λ is evaluated based on the estimation of the test set. Various lambda values are used in training the model with both training set. Here, the variables are taken in the range [00, 0.0010, 0.0030, 0.010, 0.030, 0.10, 0.30, 01, 03, 10]. When a graph is plotted between error in cross-validation set and λ , it shows that the desired value of lambda is around 03. The error caused from the cross-validation set can be low compared to error from training set which happens due to the variation of the percentage of splits in the sets. Remember that too much regularization can also make disaster (Fig. 9).

The lambda value is varied between the range 1 and 10 and the errors in the training and cross-validation sets vary. The training error raises with lambda from

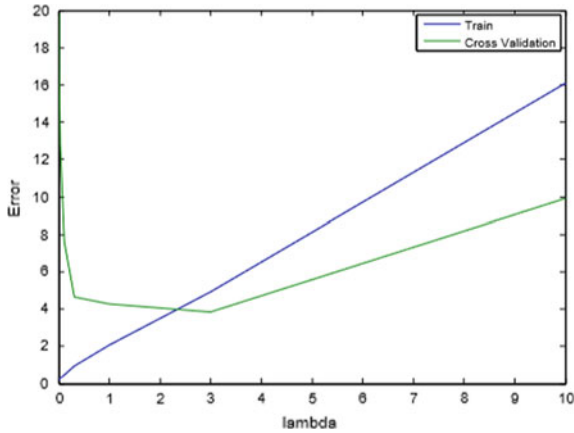


Fig. 9 Lambda with cross-validation set

Table 4 Error in the training set and validation set

Lamda	Train set error	Validation set error
0.0	0.1736230	21.5788080
0.0010	0.127460	11.2456260
0.0030	0.1712660	17.5339750
0.010	0.2215190	16.9508010
0.030	0.2818440	12.8305540
0.10	0.4593180	7.5870140
0.30	0.921760	4.6368330
01	2.0761880	4.2606260
03	4.9013510	3.8229070
10	16.0922130	9.9455080

0.30000 to 1.0000. The validation errors are in descending error and vice versa to training error (Table 4 and Fig. 10).

5 Conclusion

In the paper, there is a lot of improvements in the enactment of different learning algorithms with their respective characteristics. The graphs and tabular columns given above show a clear variation among the algorithms. The accuracy of any dataset is predicted and improved. The bias–variance decomposition on the dataset set is implemented. The net variance which is independent of the kernelis reduced

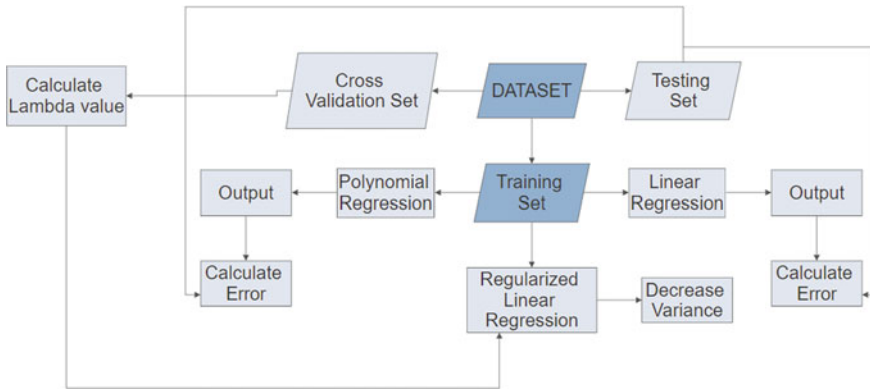


Fig. 10 Workflow diagram

highly because of optimization techniques and polynomial regression. We can highly improve the prediction of learning algorithm with this experimental analysis.

References

1. M.J.A. Eugster, T. Hothorn, F. Leisch, Exploratory and inferential analysis of benchmark experiments (2008)
2. A.J. Miller, *Subset Selection in Regression* (Chapman and Hall/CRC, 2002)
3. A. Rosenberg, Linear regression with regularization (2009). <http://eniac.cs.qc.cuny.edu/andrew/gcml/lecture5.pdf>
4. A. Jovanovich, A. Lazar, Comparison of optimization methods for l1-regularized logistic regression. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.159.2562>
5. A.F. Schmidt, R.H. Groenwold, M.J. Knol, A.W. Hoes, M. Nielen, K.C. Roes, A. De Boer, O.H. Klungel, Exploring interaction effects in small samples increases rates of false-positive and false-negative findings: results from a systematic review and simulation study. *J. Clin. Epidemiol.* **67**(7), 821–829 (2014)
6. P.C. Austin, A. Laupacis, A tutorial on methods to estimating clinically and policy-meaningful measures of treatment effects in prospective observational studies: a review. *Int. J. Biostat.* **7**(1), 1–32 (2011)
7. S. Chari, J.R. Rao, P. Rohatgi, Template attacks, in *International Workshop on Cryptographic Hardware and Embedded Systems*, 13 August 2002 (Springer, Berlin, Heidelberg), pp. 13–28
8. Y. Miyazaki, M. Terakado, K. Ozaki, H. Nozaki, Robust regression for developing software estimation models. *J. Syst. Softw.* **27**(1), 3–16 (1994)
9. T. Foss, E. Stensrud, B. Kitchenham, I. Myrtveit, A simulation study of the model evaluation criterion MMRE. *IEEE Trans. Software Eng.* **29**(11), 985–995 (2003)
10. I. Myrtveit, E. Stensrud, M. Shepperd, Reliability and validity in comparative studies of software prediction models. *IEEE Trans. Software Eng.* **31**(5), 380–391 (2005)

A Heuristic Algorithm for Diameter Calculation



Farheen Siddiqui and Nida Raza

Abstract The world is always considered as to be shrinking and the major contributor for this is the exponentially increasing web of network referred to as “network graph” that is connecting every person on the globe to every other person. This decade particularly has been a witness to a massive boon in the Internet technology. It has led to an increased competition among the market players to provide better services to the masses. Hence, a great deal of research is being done to exploit the graph’s attributes. One such attribute is the graph diameter which is widely been used. Effective computation of the graph diameter can play a great role in improving the quality of services over the internet. This paper proposes a heuristic algorithm that would be helpful in fast and easy computation of the graph diameter.

Keywords Diameter · Heuristic · Effective diameter · Average degree · Mean shortest path length · Eccentricity

1 Introduction

Giant graphs come on their own in several applications. With the social media becoming an inevitable part of our lives, the social network graphs or the World Wide Web graphs have in turn become of great significance [1, 2]. Since the network has many applications, almost in every domain and strata of life so the study of its metrics become even more significant [3].

Eccentricity and diameter are being the basic metric in the field of networks and graphs especially while dealing with the real-world networks. They are difficult to compute but are of great significance when we analyze any network in terms of speed of data flow or evolution of its structure [4, 5]. The diameter exists for all graphs irrespective of their type. It is often brought in use for the evaluation of graph models

F. Siddiqui · N. Raza (✉)

Department of Computer Science, Jamia Hamdard, New Delhi 110062, India

e-mail: nida.raza2010@gmail.com

F. Siddiqui

e-mail: Fatima.mehtab@gmail.com

© Springer Nature Singapore Pte Ltd. 2020

L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,

https://doi.org/10.1007/978-981-15-0339-9_5

and its study tends to be of great help in understanding its relationship with the other metrics of graphs [6–8].

The graph diameter can be said to be as the longest distance in the graph. It is redoubtable if not the most but at least one of the most basic constituent of the graph framework [9]. Any connected graph represented as $G(V, E)$ where V and E represent the vertices and edges of the graph, respectively, then the maximum distance between any two vertices $d(u, v)$ is the graph's eccentricity “ e ” whereas the minimum and maximum eccentricity of the graph are its radius “ r ” and diameter “ d ”, respectively [10].

The graph diameter is equal to the maximum eccentricity of any of the vertex in the graph [11]. A lot of studies have been performed to find the algorithmic complexity of radius and diameter problems. There are algorithms being made for finding the diameter and radius metrics of special sets of graphs.

Graphs comprising on n vertices and m edges with unit length we can find the distances from a single source using breadth-first-search and for the distances that are paired Single-source shortest path can be used [12].

In a graph with n vertices and m edges of unit length, all distances from a single vertex (single-source shortest paths, SSSP) can be computed by breadth-first-search (BFS) in $O(n + m)$ time, and all pairwise distances (all-pairs shortest paths, APSP) can, therefore, be obtained in $O(nm)$ time by solving n times SSSP. When we deal with the random graph data sets, the choice is nevertheless confined to all-pair shortest path algorithm (APSP) that in turn reply on the breadth-first-search approach for finding the distance between all pair of nodes [13, 14]. Algorithms working on different principal are very few.

Many algorithms have been used for better result. They have been basically grouped into three categories, viz., approximate/accurate, static/dynamic, and directed/undirected [15]. Each of them finds edge over the other in one way or the other. A comparison has been drawn between some of the standard algorithms for diameter computation in Table 1.

Table 1 Comparison of algorithms for diameter computation

Algorithm	Size	Type	Weight	Result
All pair shortest path	Small	Static	Unweighted	Exact
Matrix multiplication	Small	Static	Both	Exact
Bounding diameter	Large	Incremental	Unweighted	Exact
Fringe	Large	Static	Both	Exact
iFub	Large	Incremental	Both	Exact
iDiameter	Large	Static	Unweighted	Exact
Filtering	Large	incremental	Unweighted	Exact
Cayley	Variable	Static	Both	Exact
Dijkstra's	Large	Static	Unweighted	Exact

The study of the standard algorithms makes us draw a conclusion that no single algorithm is suitable for all types of graphs and have got limitations.

The algorithms mentioned in table are all exact algorithms. There is another set of algorithms known as the approximation algorithms, which require less intensive computations compared to the exact diameter algorithms. Although the results obtained from these algorithms can be wildly accurate [16], the cost incurred in their computation is significantly lower.

The problem of finding the diameter of a graph diameter can be assumed as to just find the distances between the vertices that are far off an ignoring the near ones but there is much more to this. The graph diameter can also be studied in terms of the average distance of the graph, i.e., the mean distance of all the $(n/2)$ vertices of the graph [17].

2 Rationale for Choosing Heuristic Approach for Graph Diameter

Watts and Strogatz stated that most of the real-world data the average path length is quite small. Since the BFS would cover a majority of the graph in limited steps; hence, it would turn out to be a costly affair for large graphs [18]. It would be quiet impractical to run a BFS or other exact algorithms for a graph due to high time complexity of the order $(O(n^2 + nm))$ required [19].

The heuristic approach never guarantees the best result but it aims at fulfilling the immediate goals. Computations are based on approximation instead of accuracy. Researchers have been bringing about many methods for diameter computation based on approximations of different ratios [15, 20, 21] for quite some time.

In algorithms inspired by the heuristics, the focus on future outcomes is minimalistic rather the attention is on attaining the nearest optimum solution. Since the approach does not focus much on the details deviation from the basic theory being given by researchers can be observed. The focus here is to get a fast result and to reduce the computation cost of the diameter.

3 Heuristics Involved in the Algorithm

Observation of the real-time data for social networks helped to draw a relation between the 90-percentile effective diameter, the average path length, and the graph diameter. For most of the graphs taken into consideration, the 90-percentile effective diameter exceeded the average path length by nearly a value of 0.5 and the diameter turned out to be nearly thrice of the 90-percentile effective diameter.

Table 2 Data set for analysis

Network	Size (in vertices)	Mean shortest path length (in edges)	90-percentile effective diameter	Diameter
Advagato	6,541	3.29	3.82	9
Caster friendship	149,700	2.65	2.94	10
Bright kite	58,228	4.86	5.76	18
Digg friends	279,630	4.41	4.96	18
Facebook NIPS	1,888	3.98	5.52	9
Facebook friendship	63,392	4.31	4.97	15
Filmtipset	104,890	3.75	4.31	9
Flickr links	1,715,255	5.19	6.36	24
Google +	23,628	3.95	4.58	8
Hamsterster full	2,426	3.67	4.80	10
Live journal	4,847,571	5.48	6.19	20
Orkut	3,072,441	4.27	4.81	10
Twitter lists	23,370	6.19	7.56	15
Youtube	3,23,589	5.29	6.64	31
Pokec	1,632,803	4.66	5.11	14

We get a single integer as output in calculating the graph diameter, so investing $\Omega(n^2)$ of time seems unreasonable [14]. Hence, if the average path length is known that can be computed using any standard means, the diameter can be computed easily.

4 Dataset and Analysis

Data that was being analyzed was the Koblenz Network Collection (KONECT) that was collected by Institute of Web Science and Technologies at the University of Koblenz–Landau (Table 2).

5 Proposed Algorithm

Input: The graph dataset for which diameter is to be computed

Output: The graph diameter.

Step 1: Input the values for distance between the nodes $d(u, v)$.

Step 2: Compute the sum of the distances between all nodes.

Step 3: Find the value for the no of distinct nodes.

Step 4: Using the values obtained in Step 3 and 4, calculate the average distance.

$$\text{Average Distance} = (\text{Sum of distances}/\text{Number of distinct nodes})$$

Step 5: Calculate 90-percentile effective diameter “ED”

$$\text{ED} = 0.5 + \text{Average Distance}$$

Step 6: Calculate diameter D

$$D = \text{ED} * 3$$

6 Implementation and Result

The heuristic has been implemented using R programming. The Igraph package has been the biggest help in making the implementation possible. It uses data structures that are time as well as space-efficient. Its functions are carefully curated to make the fastest possible implementations [22].

In most of the real-world graphs the edges grow linearly with the increasing number of nodes, and the average distance between nodes becomes smaller with time [23]. So it would not be wrong to say that the diameter would become smaller too.

The heuristics that have been followed may diverge a bit when compared to the accurate results but this can be neglected by taking into consideration the time, cost, and ease of computation. The diversion from the accurate result varies from graph to graph as well as the algorithm used for the particular data set. The algorithms play a pivotal role in affecting one aspect of the computation or the other. This tends to influence the computation cost of the eccentricities and the graph diameters. Figure 1 shows the variation of results on the wall time usage, user time usage, cpu percentage used, and the maximum RAM usage with some of the common and well-known algorithms.

The algorithm gives an approximate result but the result obtained by the algorithm would be much faster. The variance plot between the heuristic and actual result is shown in Fig. 2.

We also find that every algorithm shows a deflection in finding the eccentricity. The iDiameter algorithm also found a similar difference which can be understood better from Fig. 3.

Not only with the above two depicted algorithm but also this difference could be found in a majority of graphs.

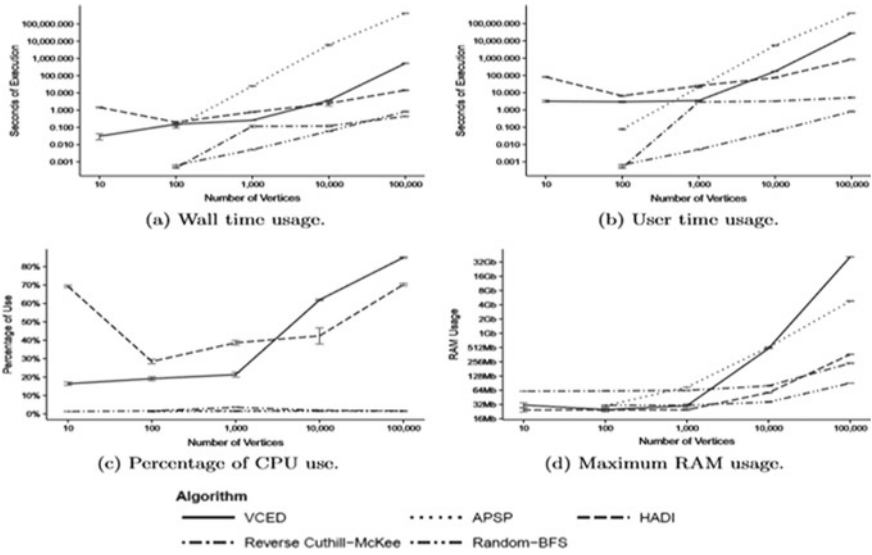


Fig. 1 Erdos-Renyi Graphs ($p = 32\%$) depicting the algorithm performance [4]

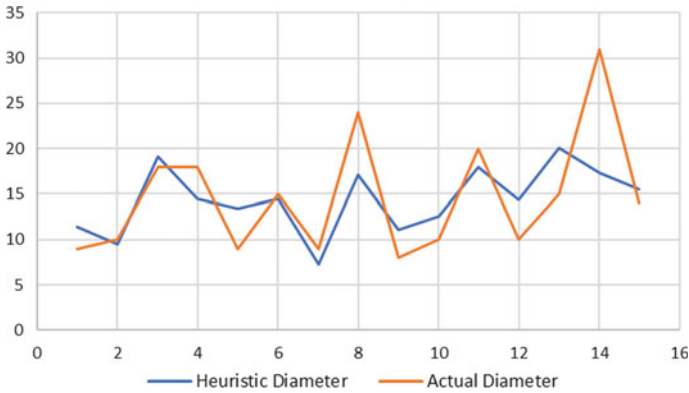


Fig. 2 Comparison of heuristic results

7 Conclusion and Future Work

The result obtained from the heuristics is good estimate of actual diameter. The diameter calculation algorithm discussed in this paper is efficient and scalable. The proposed algorithm can be used for research or commercial purpose if the factor of multiplication is increased by 1 for larger and decreased for smaller one's better results can be achieved. The proposed heuristic algorithm would bring down the operational cost, hence, changing the overall way the network operators perform.

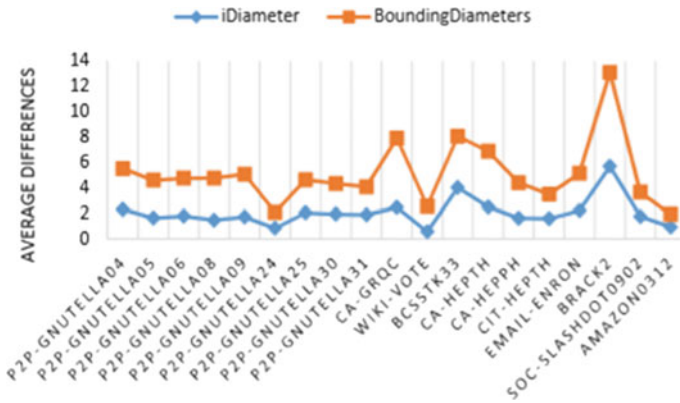


Fig. 3 Eccentricity difference between iDiameter and bounding diameter [5]

This approach can be significant and be useful for organizations irrespective of their size. Its computation is easy and is based on the real-time data of companies and hence, the performance and results are reliable.

Taking the implementation point of view, this algorithm is easy to implement and would require simple tools as the computation involved in minimalist. Finding the path length alone would do the needful to the greatest extent. Organizations with their own chat servers can make use this algorithm profusely as it would not add too much to their overhead expense and at the same time tend to be quite helpful for them.

It is been observed that the radius of the real-world graphs is often closer to the minimum value that is possible when taking the diameter as reference. However, the count of set of nodes from the farthest of all nodes is relatively lower [24]. This can be used to extend our algorithm to the real-world graphs too.

References

1. U. Brandes, T. Erlebach (eds.), Network Analysis, Methodological Foundations
2. P. Dankelmann, W. Goddard, M.A. Henning, H.C. Swart, Generalised Eccentricity, radius and diameter in graphs networks. *Int. J.* **34**(4) (Special Issue: Centrality Concepts in Network Location, 1999), pp. 312–319 (Pietermaritzburg)
3. B. Bollobás, O. Riordan, The diameter of a scale-free random graph. *Combinatorica* **24**(1), 5–34 (2004)
4. C. Pennycuff, T. Weninger, Fast, exact graph diameter computation with vertex programming, in *1st High Performance Graph Mining Workshop* (2015)
5. G.H. Dal, W.A. Kusters, F.W. Takes, Fast Diameter computation of large sparse graphs using GPUs, in *22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing* (2014)
6. E. Chow, A Graph search heuristic for shortest distance paths

7. M. Saghariichian, M.A. Langouri, H. Naderi, Calculating Exact diameter metric of large static graphs. *J. Univers. Comput. Sci.* **22**(3), 302–318 (2016) (submitted: 31/8/15, accepted: 23/1/16, appeared: 1/3/16 © J.UCS)
8. B. Mohar, Eigen values, diameter, and mean distance in graphs. *Graphs Comb.* **7**(1), 53–64 (1991)
9. U. Kang, C. Tsourakakis, A.P. Appel, C. Faloutsos, J. Leskovec, HADI: fast diameter estimation and mining in massive graphs with hadoop, in *CMU-ML-08-117* (2008)
10. S. Chechik, D.H. Larkin, L. Roditty, G. Schoenebeck, R.E. Tarjan, V.V. Williams, Better approximation algorithms for the graph diameter, in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms* (2014)
11. P. Crescenzi, R. Grossi, M. Habib, L. LANZI, A. Marino, On computing the diameter of real-world undirected graphs. *Theor. Comput. Sci.* **514**, 84–95 (2013)
12. A.H. Dekker, B.D. Colbert, Network robustness and graph topology, in *Proceedings of the 27th Australasian conference on Computer Science*, Australian Computer Society, Inc. vol. 26 (2004), pp. 359–368
13. D.G. Corneil, F.F. Dragan, E. Köhler, On the power of BFS to determine a graph’s diameter. in *Proceedings of the LATIN 2002*, ed. by S. Rajsbaum, LNCS 2286 (Springer, Berlin, 2002), pp. 209–223
14. P. Damaschke, Computing giant graph diameters, in *International Workshop on Combinatorial Algorithms IWOCA* (2016), pp. 373–384
15. L. Roditty, V.V. Williams, Fast Approximation algorithms for the diameter and radius of sparse graphs, in *STOC 13 Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, pp. 515–524
16. J. Shun, An evaluation of parallel eccentricity estimation algorithms on undirected real-world graphs, in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1095–1104
17. F.R.K. Chung, Diameters of graphs: old problems and new results. *Congr. Numer.* **60**, 295–317 (1987)
18. J. Leskovec, J. Kleinberg, C. Faloutsos, Graphs over time: densification laws, shrinking diameters and possible explanations, in *KDD’05’ Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 177–187
19. G. Csardi, T. Nepusz, The Igraph software package for complex network research. *Int. J. Complex Syst.* (1695)
20. M.E.J. Newman, The structure and function of complex networks. *SIAM Rev.* **45**(2), 167–256
21. S. Chechik, D.H. Larkin, L. Roditty, G. Schoenebeck, R.E. Tarjan, V. Williams, Better approximation algorithms for the graph diameter, in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms* (2014)
22. D.G. Corneil, F.F. Dragan, M. Habib, C. Paul, Diameter determination on restricted graph families. *Discret. Appl. Math.* **113**(2–3), 143–166 (2001)
23. T. Kitasuka, M. Iida, A Heuristic method of generating diameter 3 graphs for order/degree problem, in *Proceedings of 10th IEEE/ACM International Symposium on Networks-on-Chip* (Nara, Japan, 2016)
24. U. Zwick, Exact and approximate distances in graphs—a survey, in *Algorithms—ESA 2001* (Springer, 2001), pp. 33–48
25. D. Ajwani, A. Beckmann, U. Meyer, D. Veith, I/O-efficient approximation of graph diameters by parallel cluster growing—a first experimental study, in *ARCS Workshops (ARCS)* (2012)
26. R. Floyd, Algorithm 97: shortest path. *Commun. ACM* **5**(6), 345 (1962)
27. S. Warshall, A theorem on Boolean matrices. *J. ACM* **9**(1), 11–12 (1962)
28. R. Williams, Faster all-pairs shortest paths via circuit complexity, in *STOC* (2014), pp. 664–673
29. P. Gawrychowski, H. Kaplan, S. Mozes, M. Sharir, O. Weimann, “Voronoi diagrams on planar graphs, and computing the diameter in deterministic $\tilde{O}(n^{5/3})$ time, [arXiv:1704.02793](https://arxiv.org/abs/1704.02793), Feb 18
30. A. Abboud, V.V. Williams, H. Yu, Matching triangles and basing hardness on an extremely popular conjecture. in *STOC* (2015), pp. 41–50

Arbitrary Waveform Generator for Radar Simulator



Abhishek Gupta, Anurag Mahajan and K. Aurobindo

Abstract The research presents design and performance verification of a high-speed programmable pulse generator based on an integrated Direct Digital Synthesizer (DDS). The generator can produce pulses with an Arbitrary Waveform Modulation of instantaneous frequency up to 11 MHz. Its primary purpose is to generate various radar waveforms with user-defined pulse widths, PRTs, modulation bandwidths (if using modulation schemes within the pulse duration), and amplitude shaping. The waveforms are required to be generated in synchronization with the radar system's timing signals. It is desirable to generate pulsed waveforms at higher Ifs; the upper limit is determined by the FPGA's master clock. The waveform generator can be used as a transmit unit as well as a radar return signal simulator. Multiple channels on the board can provide simultaneous modes.

Keywords Arbitrary Waveform Generator (AWG) · Direct digital synthesis · CORDIC algorithm · Spartan-6 FPGA · VHDL

1 Introduction

The radar system is simple in concept although often complex in function and implementation. Information from the radar echoes can tell a great deal about a target, such as its distance from the radar transmits called as range resolution. Radar waveform has traditionally been produced by means of a baseband signal generator and an RF/microwave modulator which increases size and cost. Several approaches are available for generating the waveform but among those, the Arbitrary Waveform Generator structure (AWG) is a system that generates several output signals, with different frequencies. It is based on the principle that an analog waveform can be generated by using its sample. The AWG technique is partial by the Nyquist criteria

A. Gupta (✉) · A. Mahajan
Electronic & Telecommunication, Engineering, Symbiosis Institute of Technology, Pune, India
e-mail: gupta.abhishek@sitpune.edu.in

K. Aurobindo
Scientist E, Atmospheric & Microwave System Division, SAMEER, IIT, Powai, India

© Springer Nature Singapore Pte Ltd. 2020

L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_6

and the sampling theorem. Through the generation method, it is possible to change the phase frequency and amplitude characteristics of the generator output signal [1, 2].

Several methods are suggested for the implementation of AWG. Direct Digital Synthesizer (DDS) offers effective technique providing low device noise and low power consumption [3]. A numerically controlled oscillator (NCO) [4] is a generating required frequency; it basically uses a counter and provides rectangular waveform output. And direct digital frequency synthesizer (DDFS) [5] is a DSP technique. It may use NCO, other digital techniques to create, modify, and finally uses a DAC to convert it to analog form. The frequency of the generated waveform should not exceed $1/20$ to get a low harmonic output after DAC. Comparing the methodologies, it can be observed that DDS performs better than NCO and DDFS. Therefore, in this paper, we have implemented AWG using DDS in Field-Programmable Logic Array (FPGA).

2 Arbitrary Waveform

The arbitrary waveform used for transmission from the radar waveform source can be classified as, for example, continuous wave and pulse wave. The correct choice of waveform describes the characteristics of the radar as per the performance of the target detection, determination (angular, range, and frequency), and measurement (range, velocity, and error) [5, 9]. Range resolution is the ability of the radar to discriminate or the resolutions nearby together target in the range. The degree of range determination depends on the transmitted pulse which is the main factor in range resolution. Range reply at the output of the matched filter has auto-correction for the communicated signal. To consume “thin” in series (time) area, for waveform necessity consume “wide” band in frequency area. Expected at unmodulated pulse, there exists a coupling between document a de-coupling between choice resolve and waveform energy. Some of the radar using intra-pulse modulation which allows very high resolution are obtained with long pulses and high average power [1]. Following waveform is:

2.1 Continuous Waveform

The wave which simultaneously transmits and receives from the antenna is called a continuous wave [6]. It requires separate transmit and receives antennas. In a continuous wave, isolation requirements limit transmit power. One of the advantages is that radar has no blind ranges.

2.2 Pulse Waveforms

Wave interleaving for transmitting and receive periods is called pulse. Pulse wave used in radar transmission has a similar antenna used to communicate and obtain signals. Time multiplexing reduces separation necessities to let high switch. Radar consumes various varieties due to hiding during transmit actions [8]. When we use waveforms in wireless communication/digital communication, signal modification and control are necessary which give rise to the term modulation. Radar signals also use modulation and demodulation schemes.

2.3 Linear Frequency Modulated (LFM) Waveforms

An LFM signal is a kind of signal in which the frequency of the transmitted signal varies between pulse duration of T_p . This variation of the frequency from low to high or high to low is known as “chirping” [1]. The method of applying a dissimilar chirp rate aimed at respective pulse is known as “chirp diversity”. Chirps have been deeply caused trendy radar applications then by way of a result; progressive sources for show and coordinated filters for the response of linear chirps are obtainable. LFM phase is quadratic. Rapid frequency is clear as the period derivative of the phase. Prompt frequency is linear in LFM signals.

3 Proposed Block Diagram of AWG

AWG comprises Phase accumulator, Phase offset, Amplitude shaping, Digital to Analog Conversion (DAC) and Low pass filter (LPF) shown in Fig. 1. Each component involved in the AWG is explained below.

3.1 Phase Accumulator

The phase accumulator is related toward phase wheel. Trendy constant time area the indication takes the angular stage in series beginning $0-4\pi$, trendy the numerical situation the phase Accumulator everything trendy the similar method by 1 one-bit calculators besides lists. A constant sine wave remains able to be appreciated by way of the opposites move about the phase wheel [1] In the phase wheel, single revolt about extremes of the wheel equivalents toward every rotation of the sine-cosine wave.

For an i -bit phase accumulator, there are 2^i possible phase wheels. The frequency tuning word in the delta phase register, N , represents the amount that the phase

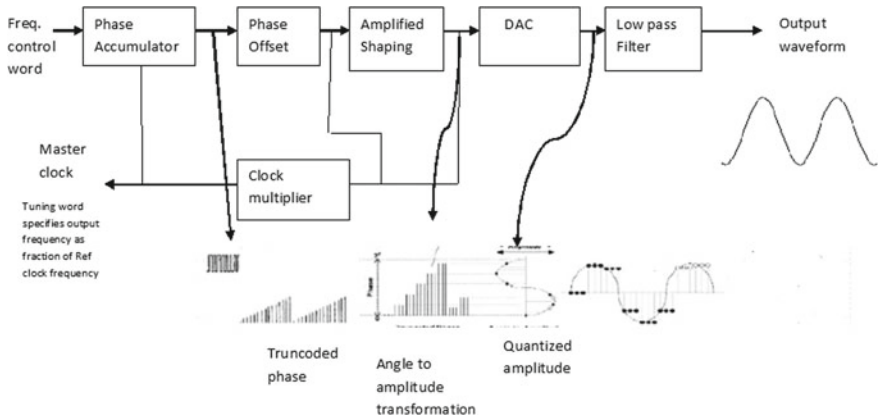


Fig. 1 Proposed block diagram of AWG

accumulator is incremented in each clock cycle. If F_{clk} is the clock frequency, then the frequency of the output sine wave is equal to F_{out} .

$$F_{out} = N * F_{clk} / 2^i \tag{1}$$

where

F_{out} = Output frequency of the digital synthesis

N = Frequency tuning word

F_{clk} = Internal reference clock frequency (system clock)

i = Length of the phase accumulator.

Therefore, respectively pole trendy the point wheel parallels to a particular every point in the sine waveform. If around is i bit digital phase accumulator next to that point near are $2i$ poles similarly spread out a round of phase wheel. On behalf of a 16 while bit phase accumulator $i = 16$.

However, the input N is 1000000000000000 at that opinion the point wheel has 216 extremes, consequently lacking 216 clock series completed at one period cycle. However, if N is "0111111111111111" needs objective 2 timepiece series to whole single retro and this is the minimum as recognized by using Nyquist. Therefore F_{out} Eq. (1).

$$\text{Resolution} = (\text{Clock Frequency}) / 2^i \tag{2}$$

3.2 Phase to Amplitude Conversion

Phase to amplitude conversion is one of the parts of direct digital synthesis. When the phase accumulator output, the phase angle should be converted into amplitude which provides sinusoidal waveform. Toward the design phase to amplitude converter, some algorithms are used like look-up tables, ROM compression, and Coordinate Rotation Digital Computer (CORDIC) [7]. CORDIC vector diagram is presented in Fig. 2.

where, $X = r \cos \theta$, $Y = r \sin \theta$.

As shown in the figure, the Coordinate Rotation Digital Computer algorithm transforms a vector x, y into a new vector x', y' . The basic iteration functions of CORDIC algorithm can be seen in

$$\begin{aligned} x' &= x \cos \theta - y \sin \theta \\ y' &= y \cos \theta + x \sin \theta \end{aligned} \tag{3}$$

where

$$\begin{aligned} x' &= \cos \theta [x - y \tan \theta] \\ y' &= \cos \theta [y + x \tan \theta] \end{aligned} \tag{4}$$

Form Formula (3), Formula (4) can be derived.

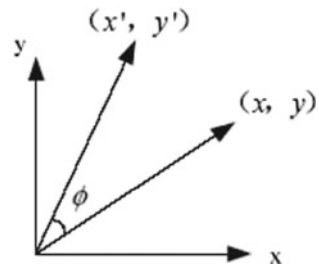
The Coordinate Rotation Digital Computer (CORDIC) uses two modes: vectoring or rotating mode. In our design, the basic equation of CORDIC for iteration in a circular coordinate system is obtained using the equation.

$$X_{i+1} = [X_i - d_i * Y_i * 2^i] * K_i \quad 0 < i \leq 32 \tag{5}$$

$$Y_{i+1} = [Y_i + d_i * X_i * 2^i] * K_i \tag{6}$$

$$Z_{i+1} = Z_i - d_i * \arctan(2^{-i}) \quad \text{where } d_i = \pm 1 \tag{7}$$

Fig. 2 CORDIC vector diagram



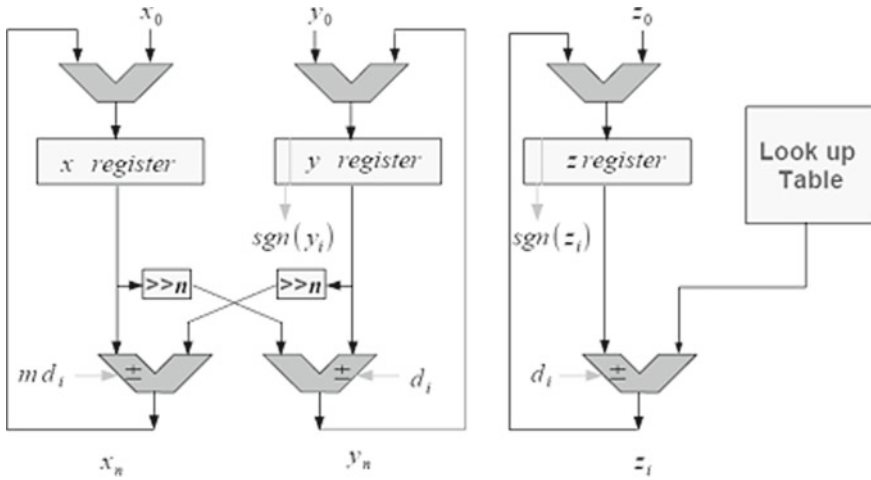


Fig. 3 Proposed bit-pipelined structure of CORDIC

d_i is the direction of rotation, where whether the rotation is clockwise or anticlockwise k_i is the scaling factor for the iteration stage. Based on the derived equations, the proposed bit-pipelined architecture is as shown in Fig. 3.

3.3 Phase Offset Rotation

The phase is the position of a point in period (an instant) on a waveform cycle. A complete cycle is distinct as the intermission essential for the waveform to return to its arbitrary final value.

3.4 Digital to Analog Converter (DAC)

A device that changes digital data into an analog signal remains the Digital to Analog Converter. Giving toward the Nyquist–Shannon sampling theorem, approximately tested data container remains reconstructed perfectly by means of bandwidth than the Nyquist criteria. A DAC can rebuild sampled data into an analog signal by accuracy. The digital information might be shaped of a phase offset, Field Programmable Gate Array (FPGA), and then originally the data needs the change to an analog signal in order to relay with the real world.

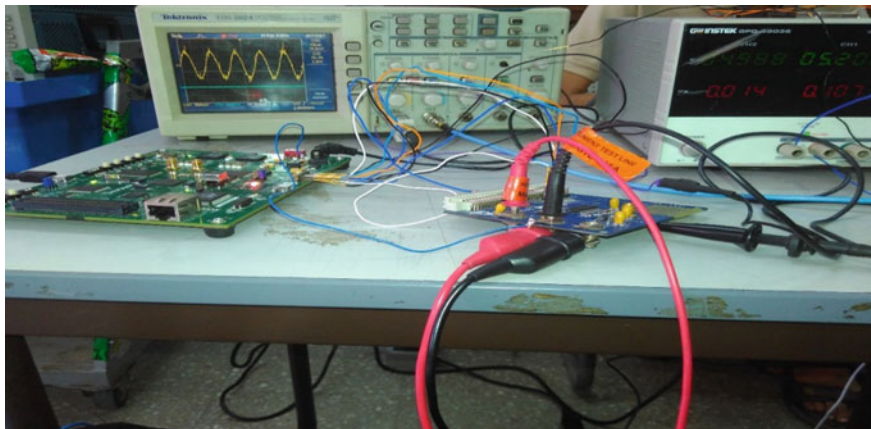


Fig. 4 Experiment platform

4 Experimental Setup

This section discusses all the hardware components used during the entire research and analysis. The research utilized the modules like “Spartan-6 FPGA SP601” as the evaluation kit board and “HARRIS HI5721” as the DAC. JTAG connection cables were used to establish the connectivity between the computing system and the FPGA board with the objective to upload the VHDL code. User I/O header from the FPGA board was used to connect the DAC which converts the digital code to analog. The DAC is connected to Digital Storage Oscilloscope (DSO) to observe the analog outputs generated (Fig. 4).

5 Results

Following are the different arbitrary waveforms observed in DSO. A sine wave can be described as a continuous wave for the FTW value provided. Pulsed wave is a nonperiodic waveform. The F_{out} values for the corresponding input FTW values along with their waveforms are presented in Figs. 5 and 6.

The FTW values were selected and tried upon considering the phase wheel. Over the iterations for the FTW values, the output F_{out} values were found to be ranging between 100 and 11 MHz.

We have also compared the existing CORDIC based on the selected critical parameters between the “Direct Digital Synthesizer (DDS)” [1] to check the performance of the proposed DDS design at higher frequencies and better frequency resolution. The values for the DDS system and the proposed system are presented in Table 1.

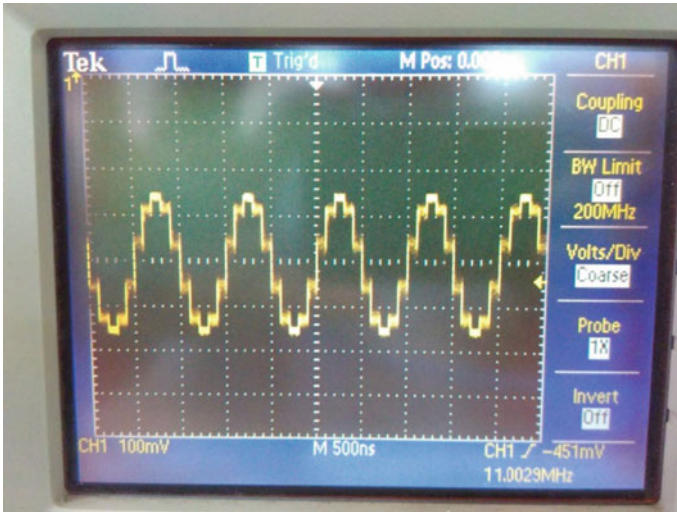


Fig. 5 Sine wave (FTW = 65536, F_{out} = 11.0029 MHz)

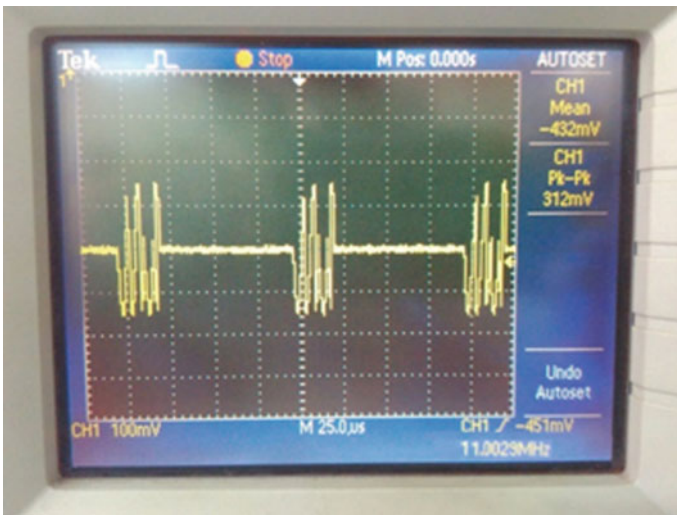


Fig. 6 Pulsed wave (FTW = 65536, F_{out} = 11.0029 MHz)

Table 1 Comparison between DDS and the proposed system

CORDIC based DDS	Anjali Powar [1]	Proposed system
No. of input bits	45	16
Frequency resolution	85 Hz	167.8 Hz
Maximum output frequency	118 kHz	11 MHz

Table 2 Device utilization summary of the proposed system

Logic utilization	Used	Available	Utilization (%)
Number of slice register	373	18224	2
Number of slice LUT	1514	9112	16
Number of fully used LUT-FF pairs	120	1767	6
Number of bonded IOBs	30	232	12
Number of block RAM/FIFO	7	32	21

From Table 1, it is clear that the proposed structure generates higher frequency as compared to the structure of [1] with less number of input bits. The number of input bits required by the proposed structure is less; therefore, the proposed structure required less hardware than the existing structure of [1].

The device utilization of the resources in the implementation of the proposed methodology is presented in Table 2.

The collection of the basic structure like flip-flops and Configurable Logic Block (CLB) is termed as a slice. The utilization parameter shows the percentage of the resources being worked upon. The lower the utilization value, the better is the performance of the algorithm employed.

6 Conclusion

Waveform generator acting as radar source has better stability in the phase, due to the use of AWG technology. In this paper, we have proposed AWG using the CORDIC algorithm. The proposed design is also implemented on the Spartan-6 FPGA board. It is clear that from the waveform shown by a digital oscilloscope, the proposed design provides better frequency resolution than the structure of [1]. The proposed structure provides more output frequency as compared to the structure of [1] with less number of input bits. Synthesis report shows that the proposed design required less hardware.

Design of a whole waveform generator using AWG technique increased the flexibility to control and generate a different modulated waveform generator. In near future, for smart radar system function generator and controlling electronic devices.

References

1. A. Pawar, Direct digital synthesizer based on FPGA. *Int. J. Adv. Res. Electr. Electron. Instrum. Eng. (IJAREEIE)* **4**(7), 6132–6133 (2015)
2. R. Preethi, T. Muruganantham, K. Vigneshwaran, Design of fast direct digital frequency synthesizer (DDFS), in *International Conference on Engineering Innovations and Solutions (ICEIS – 2016)* (2016) p. 177

3. O. Heriana, R. Indrawijaya, Radar pulse compression signal generator based on direct digital synthesizer a D9956, in *International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications* (2016), pp. 119–124
4. W. Qiu, Y. Yu, F.K. Tsang, L. Sun, A multifunctional, reconfigurable pulse generator for high-frequency ultrasound imaging. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **59**(4), 1558–1559 (2012)
5. C. Shan, C. Zhongze, Y. Hua, Design and implementation of a FPGA-based direct digital synthesizer, (IEEE, 2011), pp. 614–617
6. A. Narayanan, M.S. Boban, P.S. Kumar, VHDL implementation of a direct digital synthesizer for various applications. *Int. J. Recent. Trends. Eng. Res. (IJRTER)* **02**(04), 175–182 (2016); P. Chandramani, A.K. Jena, FPGA implementation of direct digital frequency synthesizer for communication applications. *Int. J. Sci. Eng. Technol. Res. (IJSETR)* **4**(7), 2419–2420 (2015)
7. S. Aggarwal, P.K. Meher, K. Khare, Scale-free hyperbolic CORDIC processor and its application to waveform generation. *IEEE Trans. Circuit Syst.* **60**, 315–316 (2013)

Comparative Analysis of Digital Image Watermarking Based on Lifting Wavelet Transform and Singular Value Decomposition



Sameena Naaz, Ehtesham Sana and Iffat Rehman Ansari

Abstract The various kinds of unauthorized use and modification of data has been constantly increasing day by day and very common due to the enormous growth of multimedia information on the Internet. Thus, authenticity of the data is one of the major issues in the defence data transmission as well as in secured transmission. To overcome this sort of problem, digital watermarking is supposed to be one of the legitimate solutions as it provides possible validation, safe, and secure transmission of secret data. Here, a robust and secure digital image watermarking method for the protection of copyright has been developed which involves LWT in combination with SVD. The latest approach to wavelet transform is LWT while the significant transform technique is SVD and the obtained experimental results are cross-validated by using inverse LWT and SVD. In order to create and insert a digital signature, the signature mechanism has been used after embedding the watermark and the ownership is authenticated before extracting watermarks in this research work. The robustness of the proposed method for several common image processing attacks like Mean, Median, Gaussian, Shear, Rotation, and Crop is checked by computing the two significant performance metrics, namely, PSNR and MSE. Here, the similarity is also checked by determining the NCC values. Finally, the watermarking scheme suggested in this work is compared with various existing research works which show that the algorithm used in the proposed scheme works fairly in terms of PSNR, MSE, and NCC.

S. Naaz · E. Sana

Department of Computer Science and Engineering, School of Engineering Sciences and Technology, Jamia Hamdard, New Delhi 110062, India
e-mail: snaaz@jamiyahamdard.ac.in

E. Sana

e-mail: 29es17@gmail.com

I. R. Ansari (✉)

Electronics Engineering Section, University Women's Polytechnic, A.M.U, Aligarh 202002, UP, India
e-mail: iffat_rehman2002@yahoo.co.in

© Springer Nature Singapore Pte Ltd. 2020

L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_7

Keywords LWT · Inverse LWT · SVD · PSNR · MSE · NCC · Original/cover/host image · Watermark image · Extracted watermark

1 Introduction

The multimedia messages are being shared on a very large scale at present and are tremendously increasing day by day too. These messages may include the information available on the Internet in the form of images, audios, and videos as a result of which it is a problem of concern to secure the digital information. In this way, technocrats are trying endeavors to make innovations for the security of information. The technique that is employed to communicate interactive multimedia information secure over the network by giving proprietorship rights in Digital Watermarking and it is supposed to be one of the safe, secure, and progressed scheme as compared to other effective current strategies. The branch of applied science which basically reviews the digital images and changes in these images to enhance the quality or to extract data is referred to as Digital Image Watermarking [1, 2].

The secret information of the signal can be hidden within the cover signal itself by a technique known as Watermarking and hence the cover signal can be protected. The interactive media objects within which the watermark is inserted are typically known as the original image/cover image. When the watermark is inserted, the uniqueness of the original/cover image should not be changed and later on, it can be extracted as a proof of authenticity owner. In the process of watermarking, the watermark insertion procedure is generally known and therefore it is not necessary for the message to be secret. The essential requirements of a watermarking system are listed below:

- Imperceptibility
- Robustness
- Security
- Data Payload
- Computational Complexity
- Inevitability [3].

The watermarking techniques can be typically categorized as follows:

- Spatial domain
- Transform domain.

In spatial domain methods, the watermark is inserted by changing the intensity values of the image and these methods are less complex as well as easy to implement but they are typically fragile to image processing attacks. On the other hand, in frequency domain methods, the watermark is inserted in the frequency elements of the image and these methods are simple than the spatial domain methods. However, the transform domain methods are very popular as they are more resistant to noise and the commonly used techniques are LWT, DCT, DFT, DWT, and SVD. In order to perform multilevel decomposition, DWT is generally used that breaks down the

signal/image into approximation and detail coefficients at each level and this kind of process is similar to low-pass and high-pass filtering [4].

In this research paper, digital image watermarking scheme based on LWT-SVD technique is presented to insert the watermark image into the cover/original image, thereby maintaining overall robustness property of the watermark, excellence of carrier media and security. The proposed scheme is also compared with the other existing strategies on the basis of PSNR, MSE, and NCC.

2 Literature Review

In order to hide data in digital color images, literature review survey was done on wavelet transformation and wavelet transformation in combination with SVD techniques.

The insertion of watermark into a cover image based on 2-level DWT and SVD has been proposed in [5] and the algorithm used in this work is very secure for guarding the multimedia objects. The qualitative as well as quantitative results confirm the robustness, imperceptibility, and security of the watermark.

A number of digital watermarking techniques for the protection of copyright based on SVD, DWT, and DCT have been proposed in [6–10]. In order to check the robustness of these methods, various attacks have been applied to the watermarked images. The PSNR and correlation coefficients were used as parameters for estimating the quality of the watermarked images. The simulation results obtained show that the watermarks are imperceptible in the domain.

Other robust watermarking methods based on DWT and SVD has also been suggested in [11–15]. The results achieved from these methods are very effective in terms of imperceptibility, security, capacity, and robustness and these techniques are found to be very robust against various types of attacks.

3 Preliminaries

The main objective of watermarking is the embedding of secret information that cannot be extracted easily. To do this, information should be embedded in particular parts of the image. The main terminologies employed in the proposed system are the cover image, LWT, and SVD and these are explained as follows.

3.1 Cover Image

It is the image in which the watermark will be inserted. There are various schemes available to protect this type of signal when it is transmitted over the network. Thus, the cover image is considered as the main carrier signal.

3.2 Lifting Wavelet Transform (LWT)

It is also known as second-generation wavelet transform or forward wavelet transform (FWT) and was proposed in [16]. Let $I(m, n)$ be a 2-dimensional signal. First, the 1-dimensional wavelet transform is done in the vertical direction and after that in the horizontal direction on the given signal without any loss of generality. Each 1-dimensional wavelet transform is factored into one or more lifting stages. The basic principle and structure of lifting stage are mentioned in [17]. As far as the lifting stage is concerned, it comprises three steps and these are named as split, predict, and update.

1. **Split operation:** It is also known as lazy wavelet transform. Here, all the samples will be divided into two sets, namely, even polyphase sample set, i.e., $I_e(m, n)$ as defined in Eq. (1) and the other is odd polyphase sample set, i.e., $I_o(m, n)$ as given in Eq. (2):

$$I_e(m, n) = I(2m, 2n) \quad (1)$$

$$I_o(m, n) = I(2m, 2n + 1) \quad (2)$$

2. **Predict operation:** Predict operation is referred to as dual lifting. In this operation, the polyphase samples which are odd can be predicted from the neighboring polyphase samples which are even. The error is evaluated as the prediction of the odd samples from the even samples by using a prediction operator P and this error is actually the high-pass coefficient $h(m, n)$ which is given by Eq. (3):

$$h(m, n) = I_o(m, n) - P[I_e(m, n)] \quad (3)$$

Now, the odd sample set can be recovered as given by Eq. (4):

$$I_o(m, n) = h(m, n) + P[I_e(m, n)] \quad (4)$$

3. **Update operation:** Update operation is also known as primal lifting. In this particular operation, the even samples are updated with the updating value $U_h(m, n)$ to generate the low-pass coefficient $l(m, n)$ as defined in Eq. (5):

$$l(m, n) = I_e(m, n) + U_h(m, n) \quad (5)$$

The predictions continuously come back from the vertical neighboring polyphase samples or even polyphase samples in the traditional lifting.

LWT is also known as second era quick wavelet transform could be a substitute methodology for discrete wavelet transform and in this particular transform, split and merge replaces the up- and downsampling in each of its levels. The computational complexities are decreased to half by the split and merge process in LWT. The loss of information is also less on comparing to DWT procedure because up and down sampling are not used in LWT-based procedure. In a parallel process, both even and odd polyphase segments of the signal are filtered with the help of their wavelet filter coefficients, thus produce the best result and an image reconstructed by lifting wavelet results in increased smoothness and decreased aliasing effects as compared to general wavelets [18]. Thus, by employing LWT flawlessness of embedded watermark within the image is increased, information loss is decreased, and increases the robustness of watermark. There are several advantages of LWT [19, 20] such as fewer memory prerequisites, low aliasing effects, distortion, reconstruction is good, and computational complexities are less. One of the demerits of lifting wavelet transform is that it will become difficult to take decisions on coefficient values as small values represent large coefficients. During this disintegration, filter coefficients will be regenerated into lifting coefficients represented as predict $s(z)$, update $t(z)$, and scaling (k) utilizing Euclidian procedure. The original image is divided into an odd set of samples and even set of samples and then approximate and detailed sub-bands are obtained by applying lifting coefficients to the sampled original image.

3.3 Inverse Lifting Wavelet Transform (ILWT)

As far as the inverse LWT is concerned, it is just exactly the reverse method of LWT/FWT and is determined by simply altering the sign. Hence, it can be easily computed. It consists of the following three main steps:

1. **Inverse Primal lifting operation:** This operation is also called as undo update. In this particular operation, original even samples can be reconstructed by just subtracting the update information.
2. **Inverse Dual lifting operation:** This is also known as undo predict. When prediction information is simply added to the loss of information, original odd samples can be recovered.
3. **Merge operation:** In this operation, the original signal can be obtained by merging together the recovered odd and even samples. In order to perform merging of the signal, even samples are interpolated by inserting zeros in between the samples and after that the odd samples are placed in place of zeros.

3.4 Singular Value Decomposition

It is one of the numerical examination strategy. It has been found long back and previously used to serve numerous applications. The SVD plan was presented in the 1870s by Beltrami and Jordan, utilized for real square matrices and afterward for complex matrices in 1902 by Autonne's. The researchers Eckart and Young enhanced the SVD system to incorporate rectangular matrices in 1939. At present, the SVD plan is successfully used in image watermarking, image compression, image hiding, and reduction of noise [21]. An SVD-based image watermarking scheme in the spatial domain is found to be very robust and is discussed in [22]. As far as SVD scheme is concerned, it is actually derived from linear algebra theorem. In this, a rectangular matrix I can be decomposed into three matrices in which the two matrices are singular matrices and third matrix is the diagonal matrix. Let I be a 2-D matrix having $M \times N$ dimensions. When SVD is applied to I , it gets disintegrated as defined by Eq. (6):

$$I = U * S * V^T \quad (6)$$

$$I = \sum_{i=1}^r u_i s_i v_i^T$$

$$U = [u_1, u_2 \dots u_N]$$

where symbols U and V are singular orthogonal matrices and I denotes diagonal matrix. They actually determine the detailed geometry of the original image. Let r ($r \leq n$) be the rank of the matrix I . After that the diagonal values of the matrix S are placed in the descending order and now, the matrix I can be defined by Eq. (7):

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \sigma_r \geq \sigma_{r+1} \geq \dots \sigma_n = 0$$

$$I = \sum_{k=1}^r \sigma_k u_k v_k^T \quad (7)$$

where u_k and v_k^T are the k th eigenvector of U and V and s_k denotes the k th singular value, respectively. The quality of the cover image in which the watermark is to be inserted can be very well maintained by using SVD. So rather than inserting a watermark into 2-D cover, it is much better to insert the watermark into one of the decomposed matrices as given by Eq. (7).

4 Proposed System

The watermarking scheme proposed in this research work is using an LWT method in combination with SVD. Our objective is to provide improvement and enhancement in the robustness of the watermark that also results in its improved imperceptibility. Here, the image is decomposed first into four frequency bands, namely, LL, LH, HL, and HH bands, respectively. LL depicts the band of low-frequency providing the approximate details, LH and HL both are the bands of middle frequency providing the vertical details and horizontal details, respectively, while the HH band depicts the band of high frequency providing the diagonal details of an image. In the proposed scheme, HH band has been used for inserting the watermark as it consists of the finer details of the image. The basic model of digital image watermarking consists of two main sections, namely:

- Watermark embedding
- Watermark extraction.

The method of inserting the watermark into the original/cover image is referred to as watermark embedding. The watermarked image is the output of watermark embedding process. This procedure is done on sender's side while the watermark extraction is the method of watermark detection from the watermarked image. The scheme proposed here comprises two main algorithms, namely, watermark embedding algorithm and watermark extraction algorithm as shown in Fig. 1a, b.

5 Performance Evaluation Metrics

In this research paper, the following quality measures have been used to estimate the performance of the watermarked images:

5.1 Peak Signal-to-Noise Ratio (PSNR)

Here, the imperceptibility between the original host image and the watermarked image can be measured by PSNR. As far as the watermark is concerned, it should not be noticeable to the user as well as it should not corrupt the quality of the original image. For a gray-scale image, PSNR is defined as

$$PSNR = 10 \log_{10} \left[\frac{255^2}{MSE} \right] \quad (8)$$

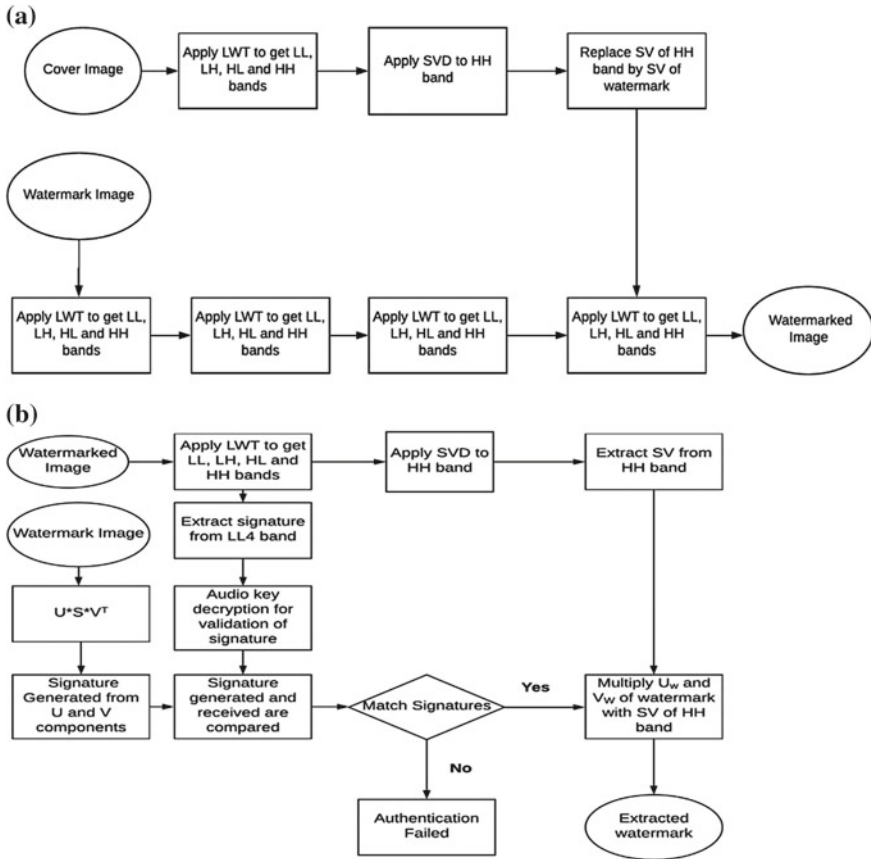


Fig. 1 a. Watermark embedding procedure. b. Watermark extraction procedure

5.2 Mean Square Error (MSE)

Another performance metric used in this work is MSE that can be defined as the mean square error between the cover image and the watermarked image. The MSE can be defined as

$$MSE = \frac{1}{M \times N} \sum_i \sum_j [I(i, j) - I_w(i, j)]^2 \quad (9)$$

where $I(i, j)$ indicates the original host image consisting of $M \times N$ pixels and $I_w(i, j)$ denotes the watermarked image.

5.3 Normalized Cross-Correlation (NCC)

To check the similarity between the two images having same size, the correlation coefficient (CC) is used. Its value varies between 0 and 1 but the ideal value for CC is supposed to be 1, i.e., unity. Here, the similarity between the inserted and the extracted watermark can be measured by the correlation coefficient which is defined as follows:

$$\rho(W, W^*) = \frac{\sum_{i=1}^N \sum_{j=1}^N (w_{ij} - \mu_w)(w_{ij}^* - \mu_w^*)}{\sqrt{\sum_{i=1}^N \sum_{j=1}^N (w_{ij} - \mu_w)^2} \sqrt{\sum_{i=1}^N \sum_{j=1}^N (w_{ij}^* - \mu_w^*)^2}} \quad (10)$$

where w denotes the watermark inserted, w^* denotes the watermark extracted, μ_w indicates the mean value of the inserted watermark, and μ_w^* indicates the mean value of the extracted watermark, respectively.

6 Experimental Results

In this research work, LWT–SVD scheme is proposed and the simulation of this scheme has been implemented in MATLAB. The digital color image “Lena” of dimension 512×512 is chosen as the original/cover image while the digital color image “Cherry” of dimension 512×512 is used as the watermark image to perform the experiment. Here, the 1-level and the 4-level LWT decomposition of the cover/original image is also done as SVD which is applied to only HH band and the digital signature is embedded into LL4 band.

6.1 Test for Imperceptibility

Imperceptibility is the measure of transparency. By using performance metrics such as PSNR and MSE, imperceptibility can be measured and it is good when the watermarked image almost resembles/looks like the original/cover image. Hence, the watermark embedding process hardly affects the cover image. The cover image and the watermarked image are shown in Fig. 2a, b, respectively. The watermark logo/image or simply the watermark is shown in Fig. 2c. The values of PSNR and MSE for the watermarked image before and after the attack with respect to the original cover image are given in Table 1. Figure 3a shows the PSNR values for original/cover and watermarked image while Fig. 3b shows the MSE values for the original/cover and watermarked image.

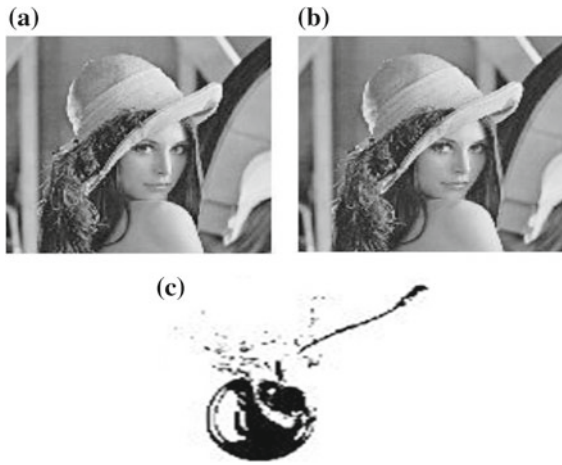


Fig. 2 a. Cover image b. Watermarked image signed with secret key c. Watermark logo

Table 1 Values of PSNR and MSE for cover and watermarked image

Attacks	Performance metrics	
	PSNR	MSE
No attack	70.31	0.0059
Mean	28.28	96.42
Median	35.21	19.59
Gaussian	39.28	7.675
Rotation	10.43	5.8851e + 03
Shear	9.07	8.0404e + 03
Crop	13.90	2.6480e + 03

6.2 Test for Robustness

The robustness of the method proposed in this work has been checked by performing various geometric attacks on the watermarked image and then extracting the watermark. The robustness of the image was checked by comparing the similarity of the extracted watermark with the original watermark. The robustness of the watermarking scheme against various types of attacks can be checked by using values of NCC. The robustness is tested under six types of attacks, namely, mean attack, median attack, Gaussian attack, rotation attack, shear attack, and crop attack, respectively. Figure 4 shows the extracted watermark before the attack. Figure 5 shows the extracted watermark after applying different types of attacks. In this research work, the values of the correlation coefficient between extracted watermark and the original watermark for original/cover image “Lena” before and after the attack is given in

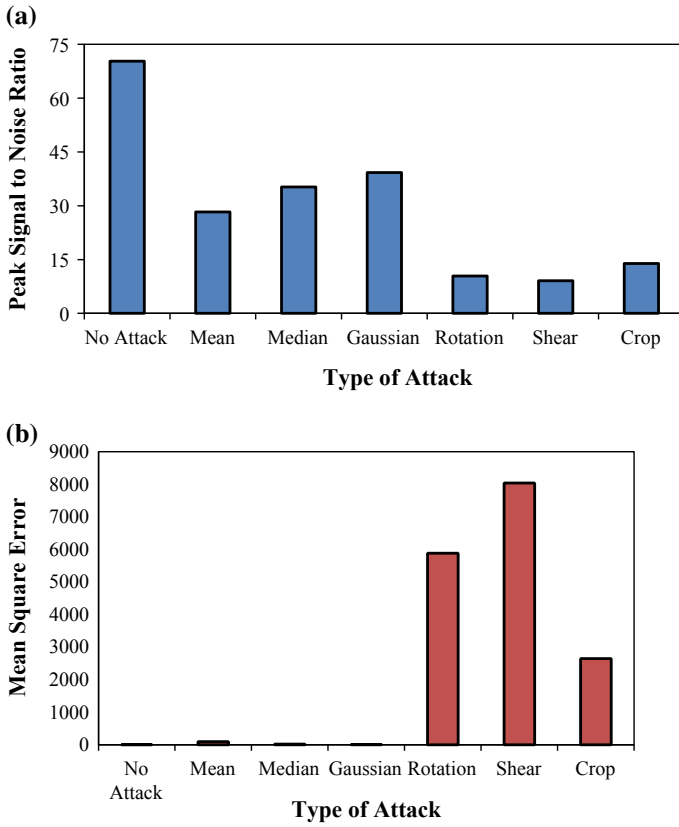


Fig. 3 a. PSNR values for cover and watermarked image. b. MSE values for cover and watermarked image



Fig. 4 Extracted watermark without attack

Fig. 5 Extracted watermark images after applying various attacks

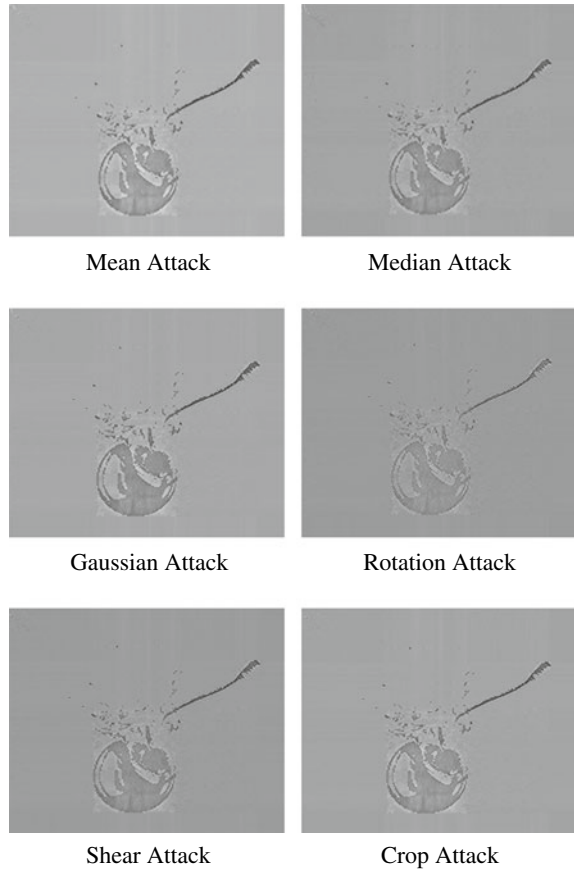


Table 2. The graph showing the values of the NCC for the extracted watermark is given in Fig. 6

Table 2 Values of NCC for extracted watermark image

Attacks	Performance metric
	Normalized cross-correlation
No attack	1.00
Mean	0.7687
Median	0.9592
Gaussian	0.9354
Shear	0.8154
Crop	0.8289

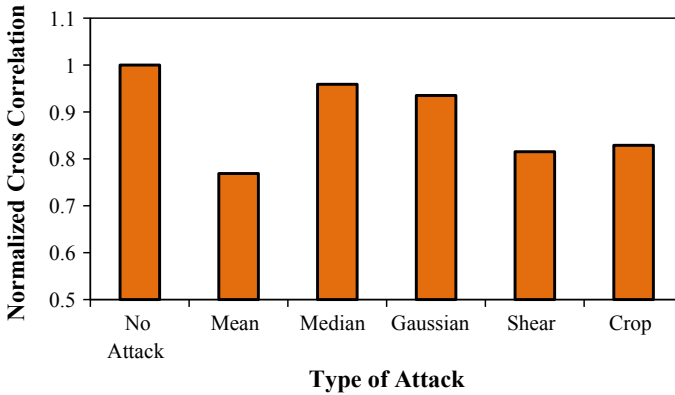


Fig. 6 Normalized cross-correlation values for extracted watermarks

6.3 Comparison of Proposed Method with Kaur et al. [5] and Radouane et al. [6] Based on Peak Signal-to-Noise Ratio

The results of the proposed scheme are compared with the results of [5, 6]. One of the schemes to embed a watermark into a cover image has been proposed in [5] which works on the combination of 2-level DWT and SVD. The other scheme has been proposed in [6] to insert the watermark in different coefficients of DWT (LH, HL, HH) and SVD technique was used. Other than this, in this particular scheme, an optimal block based on maximum entropy has been utilized for inserting the watermark in the original/cover image. Table 3 gives a comparison of the proposed work with that of [5, 6] in terms of imperceptibility which can be measured by one of the significant performance metrics, i.e., PSNR. The proposed algorithm clearly outperforms the two algorithms based upon the quality of watermarked image. The results are plotted in the form of bar chart as shown in Fig. 7.

Table 3 Comparison of proposed method with Kaur et al. and Radouane et al. based on peak signal-to-noise ratio

Attacks	Proposed method	Kaur et al.	Radouane et al.
No attack	70.39	68.79	60.052
Median	35.21	–	25.173
Gaussian	39.28	32.111	18.973
Crop	13.90	11.38	12.599

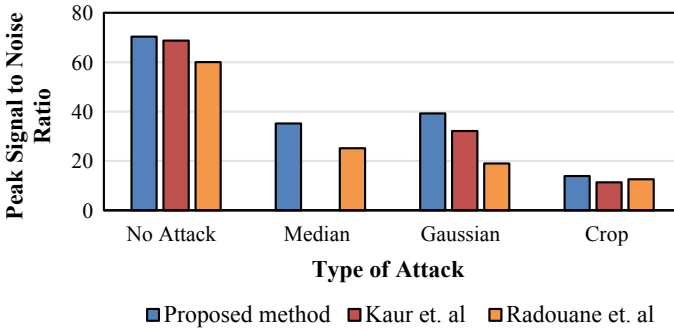


Fig. 7 Comparison of proposed method with Kaur et al. and Radouane et al. based on peak signal-to-noise ratio

6.4 Comparison of Proposed Method with Kaur et al. [5], Loukhaoukha and Chouinard [21] and Zhang et al. [22] Based on Normalized Cross-Correlation

One of the hybrid robust watermarking methods based on LWT and SVD has been proposed in [21] in which the original/cover image was decomposed into a 2-level lifting LWT and then the ILWT of a selected SB was calculated. The singular values are also modified for the watermark embedding process. Another watermarking method based on an SVD was proposed in [22] in which the watermark was embedded in the largest singular value of the image blocks. The results presented in Table 4 show that our proposed scheme outperforms the other three methods based upon the normalized cross-correlation. The same results have been also depicted in Fig. 8.

Table 4 Comparison of proposed method with Kaur et al., Loukhaoukha and Chouinard and Zhang et al. based on normalized cross-correlation

Attacks	Proposed method	Kaur et al.	Loukhaoukha and Chouinard	Zhang et al.
No attack	1.0000	0.9707	1.0000	1.0000
Median	0.9592	–	0.4700	0.9386
Gaussian	0.9354	0.9754	0.8500	0.9693
Crop	0.8289	0.3254	0.7200	0.7828

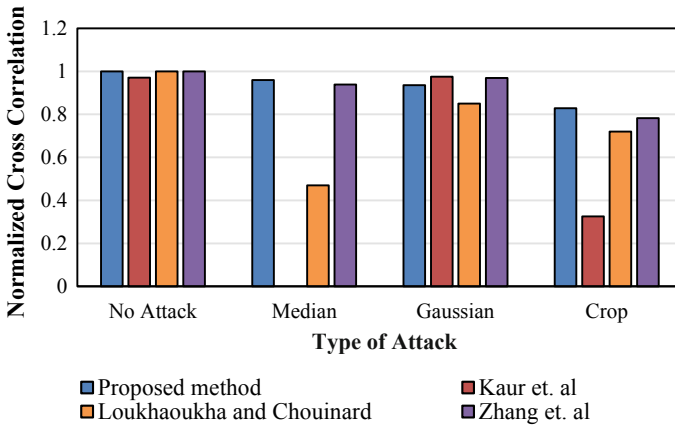


Fig. 8 Comparison of proposed method based on normalized cross-correlation

7 Conclusion

This research paper discusses an improved, robust, and secure image watermarking scheme based on LWT and SVD. The proposed scheme is implemented using MATLAB. Here, the steps used for image processing include embedding of the watermark (i.e., encoding), attacking the watermarked image, extracting the watermark (i.e., decoding), and evaluation of the results achieved. The scheme proposed here utilizes the properties of LWT as well as SVD transform to obtain the requirements of the watermarking, i.e., imperceptibility, robustness under various attacks and security. The main goal of the research is to provide improvement in the efficiency of digital image watermarking under various types of attack. To analyze the performance of the watermarked images in our work, various performance metrics have been computed, namely, NCC, PSNR, and MSE. This watermarking scheme gives NCC value 1 for no attacks with good PSNR and MSE values. The parameter values achieved by this scheme are good for most of the common image processing attacks and as far as the security is concerned, it is also improved and guaranteed by using a digital signature authentication mechanism.

The proposed method is compared with four different research works given in the literature and it has been found that our algorithm outperforms all the others in terms of PSNR, MSE, and NCC.

References

1. V. Sthul, N.N. Khalsa, Digital watermarking of an image using wavelet transform. *Int. J. Pure Appl. Res. Eng. Technol. (IJPRET)* **1**(8), 365–378 (2013)

2. K.S. Rawat, S. Goyal, R. Gupta, Discrete wavelet based image watermarking: an idea leads to security. *J. Biom.* **1**(1), 06–10 (2010)
3. H. Yang, Y. Liang, X. Wang et al., A DWT- based evaluation method of imperceptibility of watermark in watermarked color image, in *International Conference on Wavelet Analysis and Pattern Recognition*, vol. 1 (2007), pp. 198–203
4. Iffat Rehman Ansari, Shahnawaz Uddin, Signal denoising using discrete wavelet transform. *Int. J. Eng. Technol. Sci. Res. (IJETSR)* **3**(11), 23–31 (2016)
5. J. Kaur, N. Singh, C. Jain, An improved image watermarking technique implementing 2-DWT and SVD, in *IEEE International Conference on Recent Trends in Electronics Information Communication Technology* (2016), pp. 1855–1859
6. M. Radouane, R. Messoussi, R. Touahni, T. Boujiha, Robust method of digital image watermarking using SVD transform on DWT coefficients with optimal block, in *International Conference on Multimedia Computing and Systems (ICMCS)* (2014)
7. N. Shrinivas, N. Naveena, K. Manikantan, Robust digital image watermarking using dwt + svd approach, in *IEEE International Conference on Computational Intelligence and Computing Research* (2015)
8. T. Bhuyan, V. Srivastava, F. Thakkar, Shuffled SVD based robust and secure digital image watermarking, in *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)* (2016)
9. K.R. Sheth, V.V. Nath, Secured digital image watermarking with discrete cosine transform and discrete wavelet transform method, in *International Conference on Advances in Computing, Communication and Automation (ICACCA)* (Spring, Berlin, 2016)
10. T. Pardhu, B.R. Perli, Digital image watermarking in frequency domain, in *International Conference on Communication and Signal Processing* (2016)
11. A. Furqan, M. Kumar, Study and analysis of robust DWT-SVD domain based digital image watermarking technique using MATLAB, in *IEEE International Conference on Computational Intelligence and Communication Technology* (2015)
12. N.M. Makbol, B.E. Khoo, Robust blind image watermarking scheme based on redundant discrete wavelet transform and singular value decomposition. *AEU–Int. J. Electron. Commun.* **67**(2), 102–112 (2013)
13. Y. Dejun, Y. Rijjing, L. Hongyan, Z. Jiangchao, A digital watermarking scheme based on singular value decomposition and discrete wavelet transform, in *International Conference on Computer Science and Network Technology* (2011)
14. V. Santhi, A. Thangavelu, DWT-SVD combined full band robust watermarking technique for color images in YUV color space. *Int. J. Comput. Theory Eng.* **1**(4), 424–429 (2009)
15. H. Tao, L. Chongmin, J.M. Zain, A.N. Abdalla, Robust image watermarking theories and techniques; a review. *J. Appl. Res. Technol.* (2014)
16. W. Sweldens, The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal.* **29**(2), 511–546 (1998)
17. A. Manjunath, Comparison of discrete wavelet transform (DWT), lifting wavelet transform (LWT), stationary wavelet transform (SWT) and S-transform in power quality analysis. *Eur. J. Sci. Res.* **39**(4), 569–576 (2010)
18. H. Kiya, M. Iwahashi, O. Watanabe, A new class of lifting wavelet transform for guaranteeing losslessness of specific signals, in *IEEE International Conference on Acoustics, Speech and Signal Processing* (2008), pp. 3273–3276
19. S.K. Jinna, L. Ganesan, Lossless image watermarking using lifting wavelet transform. *Int. J. Recent Trends Eng.* **2**(1), 191–195 (2009)
20. K. Loukhaoukha, Optimal image watermarking algorithm based on LWT-SVD via multi-objective ant colony optimization. *J. Inform. Hiding Multimed. Signal Process.* **2**(4), 303–319 (2011)
21. K. Loukhaoukha, J. Chouinard, Hybrid watermarking algorithm based on SVD and lifting wavelet transform for ownership verification, in *11th Canadian Workshop on Information Theory* (2009), pp. 177–182

22. H. Zhang, C. Wang, X. Zhou, A robust image watermarking scheme based on SVD in the spatial domain. *Future Internet* **9**(3), 45 (2017)

Analysis of Capacitance–Voltage Characteristics for Ultrathin Si/SiGe/Si Hetero-Layered MOS Structure



Rudra Sankar Dhar, Lalthanpuii Khiangte, Parvin Sultana and Ankit Kumar

Abstract Strained Silicon Metal–Oxide–Semiconductor Field-Effect Transistor (MOSFET) has already proven to be a superb candidate for high-speed electronic devices. A novel strained heterostructure device has been developed consisting of Si/SiGe/Si layer in the channel forming hetero-layered MOS structure. To develop a mature and in-depth understanding of the heterolayer, the analysis for study of behavior and characteristics has been the crux of in this recent works. Thus, the need of understanding this advanced structure through capacitance–voltage (CV) became the essentiality and most fundamental onus leading to device modeling and thorough expectation and control of the device performance.

Keywords Capacitance–voltage · Strained silicon · Heterostructure · MOSFET

1 Introduction

With the increased advancement in our electronics world, green electronics devices have been promoted which led us to the requirement of novel material structures for high speed and ultralow power consumption in both static and dynamic implementation [1–6]. Conduction band minima of silicon occur at six points of equal energy in the momentum space E-K diagram. These six minima energy points are known as valleys as they are surrounded in by the constant energy of ellipsoidal shape [7, 8]. The mobility of electrons in a particular direction is inversely proportional to the mass component in that direction. Thus for the electrons in x-valley, the electron mobility will be lower in the x-direction and higher in the y- and z-directions. If the mobility is denoted as μ_l (parallel to axis) and μ_h (perpendicular to axis) for x-valley electrons and the electric field is applied in the x-direction, then velocity v_x (for electrons in x-valley) acquired are given as $v_x^x = \mu_l E_{xl}$ $v_x^y = \mu_h E_x$ and $v_x^z = \mu_h E_x$, where μ_l and μ_h are the low and high mobility, respectively. The total

R. S. Dhar (✉) · L. Khiangte · P. Sultana · A. Kumar
National Institute of Technology Mizoram, Aizawl 796011, Mizoram, India
e-mail: rdhar@uwaterloo.ca

© Springer Nature Singapore Pte Ltd. 2020
L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_8

current is determined by the sum of currents in the three valleys with the average mobility of $\mu_{avg} = (\mu_l + 2\mu_h)/3$ [9, 10].

This work is concerned with the investigation of the capacitance–voltage (CV) of n-channel Si/SiGe/Si heterostructure on insulator (HOI) Metal–Oxide–Semiconductor Field-Effect Transistor (MOSFET). This work provides detail insight into the frequency response and its dependence on the design parameters, given the complex heterolayers forming strained layers in MOS capacitor of the HOI device.

2 Hetero-Layered MOS Structure

2.1 Device Structure

The heterostructure device [11] designed consists of three layers of Si/SiGe/Si in the channel region of the MOSFET which designated to its heterogeneity. The novelty of this structure to the conventional strained silicon MOSFETs being designed so far is the additional epitaxial growth of the bottom silicon layer on the buried oxide (BOX) which is optimized to have the conducting active region on the Si layers for the development of n-channel improved strained silicon HOI MOSFET. This structure clubbed the benefits of strain silicon technology in channel engineering with the silicon-on-insulator device. Such device with gate-length of 50 and 30 nm has been developed with source/drain doped by n-type Arsenic concentration of 10^{20} cm^{-3} and substrate doped by Boron concentration of 10^{16} cm^{-3} . The tri-layered Si/SiGe/Si thicknesses are 2 nm/6 nm/2 nm, respectively.

2.2 Hetero-Layered M–O–(Si/SiGe/Si) Structure

The constructions of the active region of the device based on the heterogeneity provide the Metal–Oxide–Si/SiGe/Si system to be the core concept for the novel device structure as depicted in Fig. 1. The energy band diagram (flat band) of the three-layered system has also been specified. For ideal condition, these three layers (Si/SiGe/Si) in the channel region introduces three threshold voltages $V_{t_{\text{Si-Si(top)}}$, $V_{t_{\text{Si-Ge}}}$, and $V_{t_{\text{Si-Si(bottom)}}$, respectively. On application of gate voltage (V_{gs}), inversion occurs in sequence depending on the respective threshold voltages of the layers as displayed in Fig. 2. Now as the structure employed in this heterogeneous system includes strain effect in the silicon, this induced strain splits energy band valleys into reformed energy levels. This effect of strain on the energy minima does not have significant alteration on the shape of the constant energy surrounding the valleys [12]. If the interface of the silicon and substrate is normal to the z-axis, then the strain induced raises the energies of both the x- and y- valleys and lowers the energy in the z-axis valley. Thus, electrons tend to occupy the lowest conduction band energy

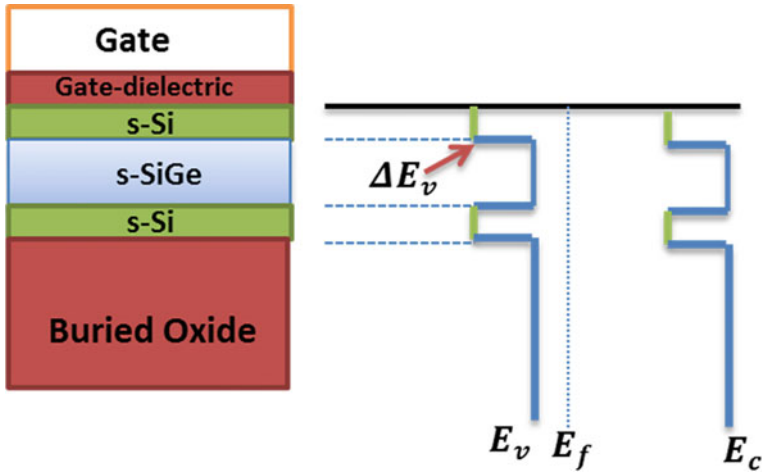


Fig. 1 Si/SiGe/Si hetero-layered HOI MOS system

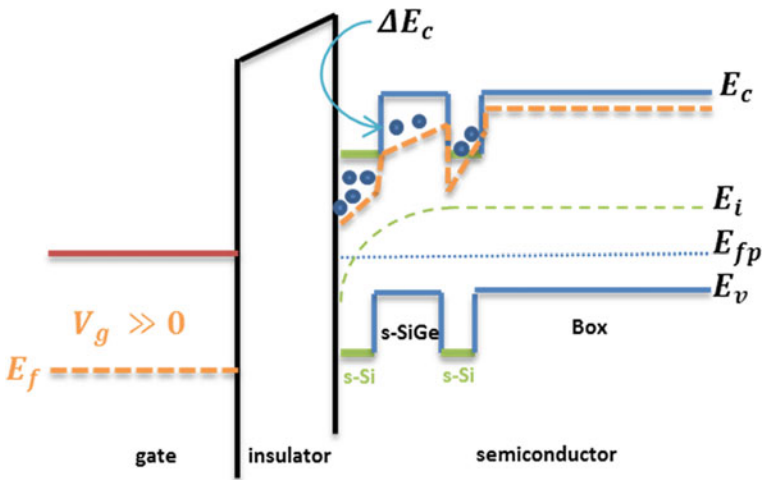


Fig. 2 Energy band structure of p-MO(Si/SiGe/Si) system in inversion

state, i.e., z-valley, in this case, resulting in probable accumulation of electrons in the layer. And in contrary, the mobility of these accumulated electrons are higher in the x- and y- direction which is in plane of the film.

Silicon epilayer is grown on the (001) surface of the $Si_{0.6}Ge_{0.4}$ alloy. The strain in the direction normal to the surface is $(-2C_{12}/C_{11})e$ assuming the lattice constant of the alloy is $(1 + e) \times$ lattice constant of silicon. Valleys in the (001) axis are lowered by $(2C_{12}/C_{11})\mathcal{E}_{ue}$ and (100) and (010) axes energies are risen by \mathcal{E}_{ue} where \mathcal{E}_u is the deformation potential. [8] For silicon value of \mathcal{E}_u and $(2C_{12}/C_{11})$ are about

8.6 eV and 0.77, respectively, resulting in (15xe) eV valleys separation by the strain. This separation energy should be larger than the thermal energy (kT) for significant electrons repopulation in the valleys [12, 13].

3 Results and Discussions

Variation of charge distribution with respect to voltage is defined as capacitance and is used for the characterization of electrical behavior of the hetero-layered M–O–(Si/SiGe/Si) system.

$$C = \frac{Q}{V} \tag{1}$$

Figure 3 illustrates the CV-curve for 50 nm HOIMOS structure, which is similar to the conventional MOS capacitance–voltage characteristics; there are three regions, accumulation, depletion, and inversion. On application of the negative voltage at the gate, accumulation of majority carriers (holes) from the p-substrate is induced at the surface, which is an exponential function of valence band bending [7, 14, 15]. The kink that is detected in the accumulation plateau (labeled as k) at initial negative voltage is induced by holes of the SiGe layer, which is an enriched hole

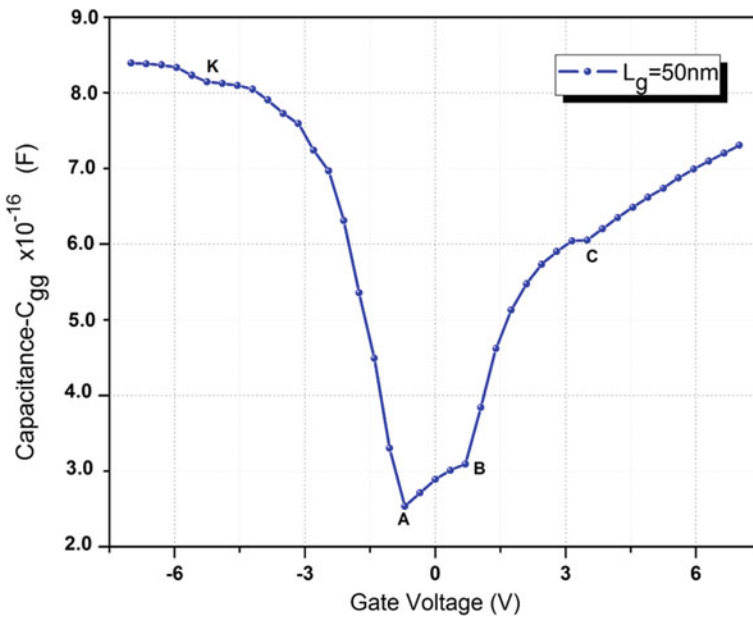


Fig. 3 CV-curve for the hetero-layered Si/SiGe/Si p-MOS system

mobility layer [14] and on increasing the negative voltage, accumulation of holes from the Si layers is perceived. As holes are the minority carriers in both the upper and lower Si layers of the structure, there is a decreased capacitance at the onset of the accumulation. Now as the gate voltage is made, more positive holes are repelled from SiO₂/Si, Si/SiGe, and SiGe/Si interfaces resulting in the formation of depleted layers of ionized acceptors.

As the depletion width continues to increase by gate voltage being more positive, the capacitance decreases simultaneously as illustrated in the curve of Fig. 3 and reaches a minimum value at the maximum depletion width (x_{dmax}). On increased positive biasing at the gate, minority carriers (electrons) are generated resulting in increasing capacitance again. As shown in the obtained inversion region of the CV-curve, the subtle increment in capacitance from point A is due to the inversion effect occurred in the bottom Si layer and thereby the curve reaches a brink point B for the onset of reversal of the layer. Furthermore enhancement in the positive gate biasing leads to inversion of SiGe that causes the capacitance increment at B to C and finally, the full inversion of the heterolayer occurs from point C onwards with the inversion of the top Si layer. The apparent charge density of mobile and immobile charges in the semiconductor can be calculated using Eq. 2.

$$\frac{1}{N_{app}} = \frac{q\epsilon_{Si}\epsilon_o}{2} \frac{\delta \frac{1}{C^2(V_g)}}{\delta V_g} \quad (2)$$

The CV analysis of scaled 30 nm gate-length device has also been carried out and is compared with the 50 nm gate-length structure as shown in Fig. 4. As indicated, the maximum depletion x_{dmax} occurs at G and F for 50 nm and 30 nm devices, respectively. This shift in the CV-curve clearly indicates the early depletion phenomenon in the shorter gate-length device.

In conventional ideal MOSFET CV analysis, the flat band voltage is zero when the work function of gate is equal to that of the silicon substrate. Whereas in the depiction observed by Fig. 4, the flat band voltage for this novel device is defined as work function difference of gate and Si/SiGe/Si layers combination resulting in the flat band voltage shift toward the left side. The work function of Si_{0.6}Ge_{0.4} is calculated to be approximately equal to 4.15 eV [16]. The analysis frequency is varied (100 kHz, 1, and 5 MHz) as shown in Fig. 5 to observe the frequency band till which this three-layered structure remains stable. Similar to the conventional MOSFET, for increasing frequency, weaker inversion is formed indicating the decreasing of the capacitance in the CV-curve, though the structural feature remains the same for the curve. This is the resultant of the fast change in applied component ac signal in the gate voltage which gives no time for generation of minority carriers (electrons) in the depletion region.

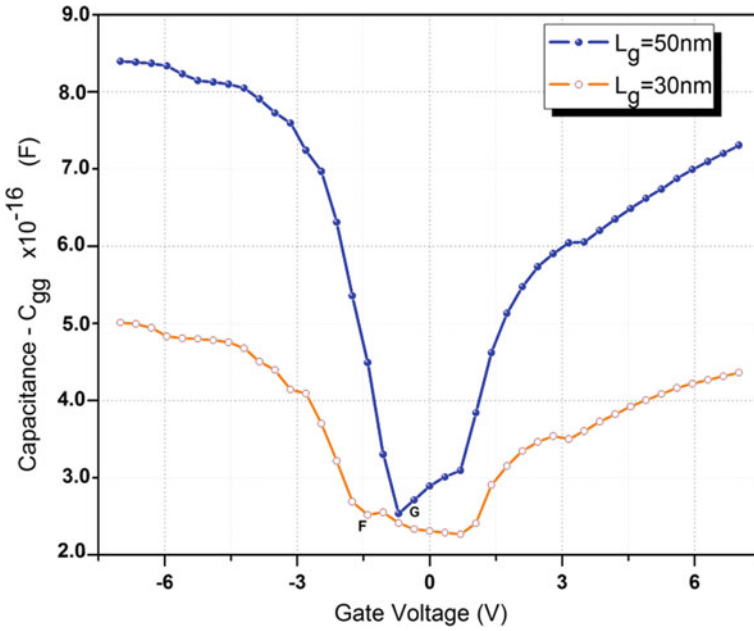


Fig. 4 Comparative analysis of CV characteristics for scaled-down 30 nm device with 50 nm gate-length device

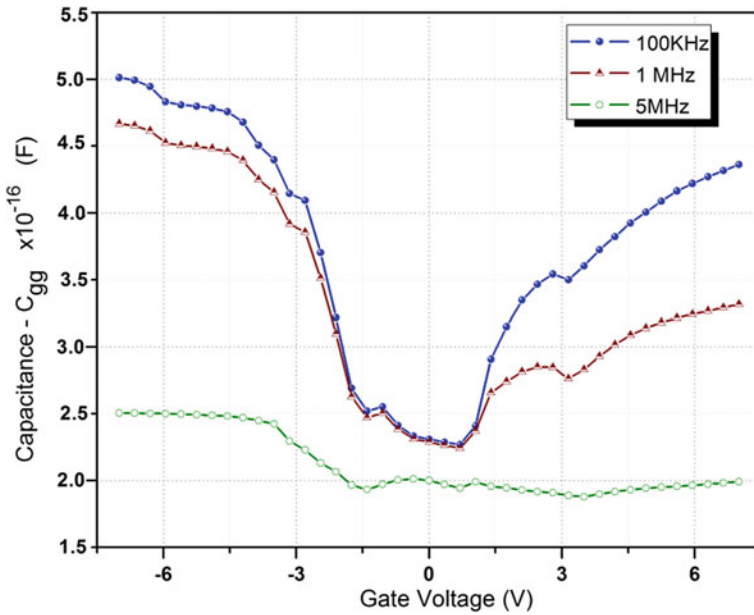


Fig. 5 CV-characterization of 30 nm gate-length device for various frequencies

4 Conclusion

Capacitance–Voltage characteristics for Si/SiGe/Si heterostructure channel for MOS-FET has been carried out using Synopsis TCAD. The analysis has been done for both the 50 and 30 nm length three-layered MOS structure. The effect of the three Si/SiGe/Si layers in the inversion region is evidently visible in the accumulation and onset of the inversion in the CV characteristics. Flat band voltage shift due to the trapped charges and induced strain is additionally prominent in the shorter gate-length device. The in-depth study of the MOS device performance in varied frequency range and capacitance dependency on the layer's thickness gives an insight into the fundamental physics.

References

1. V.K. Joshi, Spintronics: a contemporary review of emerging electronics devices. *Eng. Sci. Technol. Int. J.* **19**(3), 1503–1513 (2016)
2. K.O. Petrosyants, D.A. Popov, High-k gate stacks influence on characteristics of nano-scale MOSFET structures, in *2nd International Conference on Modelling, Identification and Control* (2015)
3. C. Dekker, How we made the carbon nanotube transistor. *Nat. Electron.* **1**(9), 518 (2018)
4. S. Datta, How we proposed the spin transistor. *Nat. Electron.* **1**(11), 604 (2018)
5. S. Manipatruni et al., Scalable energy-efficient magnetoelectric spin–orbit logic. *Nature* **565**(7737), 35 (2019)
6. J.-A. Carballo, W.-T.J. Chan, P.A. Gargini, A.B. Kahng, S. Nath, ITRS 2.0: Toward a re-framing of the semiconductor technology roadmap, in *2014 32nd IEEE International Conference on Computer Design (ICCD)* (2014), pp. 139–146
7. M.V. Fischetti, Z. Ren, P.M. Solomon, M. Yang, K. Rim, Six-band k p calculation of the hole mobility in silicon inversion layers: Dependence on surface orientation, strain, and silicon thickness. *J. Appl. Phys.* **94**(2), 1079–1095 (2003)
8. L. Kleinman, Deformation potentials in silicon. I. uniaxial strain. *Phys. Rev.* **128**(6), 2614 (1962)
9. I. Aberg, C.N. Chl eirigh, O.O. Olubuyide, X. Duan, J.L. Hoyt, High electron and hole mobility enhancements in thin-body strained Si/strained SiGe/strained Si heterostructures on insulator, in *IEEE International on Electron Devices Meeting, 2004. IEDM, Technical Digest* (2004), pp. 173–176
10. T. Pe i -Brđanin, B.L. Doki , Strained silicon layer in CMOS technology. *Electronics* **18**(2), 63–69 (2014)
11. L. Khiangte, R.S. Dhar, Development of tri-layered s-Si/s-SiGe/s-Si channel heterostructure-on-insulator MOSFET for enhanced drive current. *Phys. Status Solidi* 1800034 (2018)
12. C.S. Smith, Piezoresistance effect in germanium and silicon. *Phys. Rev.* **94**(1), 42 (1954)
13. R.W. Keyes, High-mobility FET in strained silicon. *IEEE Trans. Electron Devices* **33**(6), 863 (1986)
14. R.W. Keyes, Explaining strain [in silicon]. *IEEE Circuits Devices Mag.* **18**(5), 36–39 (2002)
15. K. Datta, A. Shadman, E. Rahman, Q.D.M. Khosru, Trilayer TMDC heterostructures for MOS-FETs and nanobiosensors. *J. Electron. Mater.* **46**(2), 1248–1260 (2017)
16. M. Amato, M. Bertocchi, S. Ossicini, Work function bowing in Si_{1-x}Ge_x heterostructures: Ab initio results. *J. Appl. Phys.* **119**(8), 85705 (2016)

Diabetic Retinopathy Detection Using Twin Support Vector Machines



Manisha Singla, Samyak Soni, Prateek Saini, Abhishek Chaudhary
and K. K. Shukla

Abstract It is essential to get the regular check-up of our eye so that the earlier detection of diabetic retinopathy (DR) can be made possible. DR is the disease because of retinal damage due to prolonged diabetic mellitus. The detection of DR using eye fundus images has been a current research topic in the area of medical image processing. Several methods have been developed to automate the process of DR detection. Researchers have made use of different classifiers to efficiently detect the presence of diabetes. SVMs contribute to the latest modelling of DR detection. Although, it proved to be an efficient technique but time consuming, especially when the dataset is large. Also, there is noticeable decrease in the performance of SVM when the dataset is corrupted by noise and outliers. This paper presents the idea of making use of twin support vector machines (TWSVMs) and its robust variants for DR detection. We give the detail of feature extraction from the digital fundus images which are fed to the TWSVMs and its robust variants. The comparison with the previous works of SVM illustrates the superiority of our model. It should be noted that the choice of TWSVM classifier not only solved the problem of time consumption but also made the DR detection robust.

Keywords Diabetic retinopathy detection · Support Vector Machines · Twin Support Vector Machines · Feature extraction · Robust variants

1 Introduction

1.1 Diabetic Retinopathy

As reported by WHO, the 7th leading reason for death worldwide is diabetes. Diabetic patients tend to show the abnormality in their retina. As the disease progresses slowly, it is suggested by medical experts that diabetic patients should refer to the doctors

M. Singla (✉) · S. Soni · P. Saini · A. Chaudhary · K. K. Shukla
Indian Institute of Technology (BHU), Varanasi, India
e-mail: manishasigla.rs.cse17@itbhu.ac.in

© Springer Nature Singapore Pte Ltd. 2020
L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_9

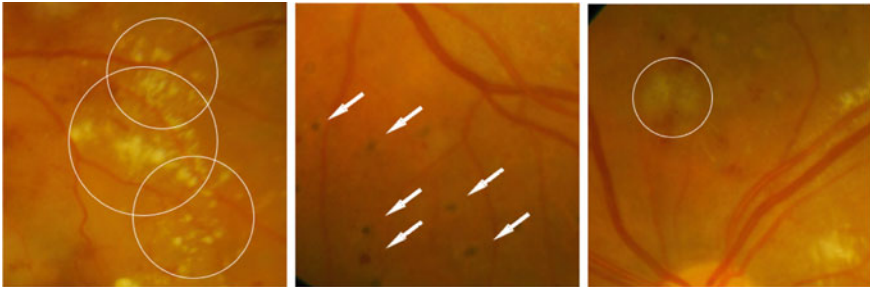


Fig. 1 Hard exudates, microaneurysms, and soft exudates

at least twice a year so that the signs of an illness can be timely diagnosed. This process not only is time-consuming but it also takes a lot of money. Furthermore, increasing diabetic patients lead to a lack of medical resources in some parts of the world. This may lead to visual loss, which can never be reversed. Also, if the patient is in the early stage of DR, detection and immediate treatment can control the whole process, and hence it can be delayed. Therefore, it is very essential to get the required diagnosis on time. Sometimes, we cannot believe the manual diagnosis completely as the lack of experience may lead to misdiagnosis too. In such cases, the computer-aided diagnosis works better. Therefore, to get a fast and reliable DR diagnosis, one can trust the computer-aided diagnosis.

DR emerges as small differences in the retinal capillaries. The very first abnormality is the presence of microaneurysms (Ma) (Fig. 1) which leads to the rupture of intraretinal hemorrhage. The severity of this disease can be classified as mild non-proliferative diabetic retinopathy (NPDR) when the microaneurysms are seen for the first time in the retina. The next abnormality that arises is the presence of hard exudates and retinal edema. This leads to increased permeability through the capillary walls. The next stage is the moderately proliferative DR (MPDR) in which hard exudates start leaking because of weak blood vessels. The last stage is the proliferative DR (PDR) in which the blood vessels disrupt and lead to microinfarcts in the retina. These microinfarcts are known as soft exudates (SE).

Various researchers have worked in this direction to automate the process of DR detection so that misdiagnosis can be prevented. As the problem is related to classifying the cases into diabetic and nondiabetic, various classifiers had already been used by the researchers to enhance the quality of DR detection. These classifiers may work well when the dataset is small but when the dataset is large, model training takes too long and makes the overall processing slow. This calls for a faster classifier which can train the model faster compared to other standard classifiers.

Our contributions to this work are listed below:

1. We have used TWSVM as the classifier to solve the problem of DR detection.
2. We have also used a robust variant of TWSVM, pin-TWSVM, to add robustness to our model.
3. We have used eight features from the images to detect its presence.

1.2 Related Work

In this part, we have discussed those works which contributed to the problem of DR detection. DR detection using machine learning has currently been an interesting research area, and many pieces of research are trying to detect this problem using machine learning models accurately. First, we are discussing here the computer-aided screening system, which was designed in 2013 that analyzes the fundus images and generates the report of its severity for DR [14]. In this work, classifiers like Gaussian Mixture Model (GMM), SVM, Adaboost, and k-nearest neighbor (KNN) are analyzed over classifying retinopathy lesions. It was observed that GMM and KNN are the best classifiers for red and bright lesion classification [14].

Similarly, the use of GMM was further extended in this area using the three-stage segmentation algorithm [6]. In this work, eight features were extracted and were passed through the preprocessing and post-processing stages defined in [6]. The algorithm managed to achieve an accuracy of 95.2, 95.3, and 95.15% over DRIVE [7], CHASE-DB1 [2], and STARE [18] datasets respectively.

In 2015, STARE dataset was used further to test an another iterative algorithm, which was proposed by [15]. In this work, authors have proposed a new algorithm which is robust to the rate of new vessel pixel addition and hence it was computationally efficient and consistent throughout its performance [15].

Another work which used morphological operations together with machine learning also grabbed attention. In this work, after preprocessing, authors applied some morphological operations to find the microaneurysm. After this, they extracted the features and trained the classifiers. They used SVM and KNN [11] and observed that SVM performed better than KNN.

In 2016, the detailed analyses of DR detection using machine learning was provided by [1].

Furthermore, another work proposed by [17] emphasized more on the data preprocessing by applying histogram equalization, green channel extraction and image enhancement, and resizing techniques. They used the Kaggle DR dataset.

Another important contribution in this area was provided by [10]. In this paper, authors found that only two features, area and number of microaneurysms are sufficient to predict DR in the fundus images. This work was proved to be computationally efficient. Similarly, segmentation techniques were also applied to DR detection problems by [12].

2 Dataset Used

We have used the DIARETDB1 dataset [9] which is obtained from the Standard Diabetic Retinopathy Database. This dataset comprises 89 colored eye images. Among these, 84 contain mild symptoms of DR, and the remaining five are normal as these do not have any of the above-discussed abnormalities. Similarly, the dataset contains

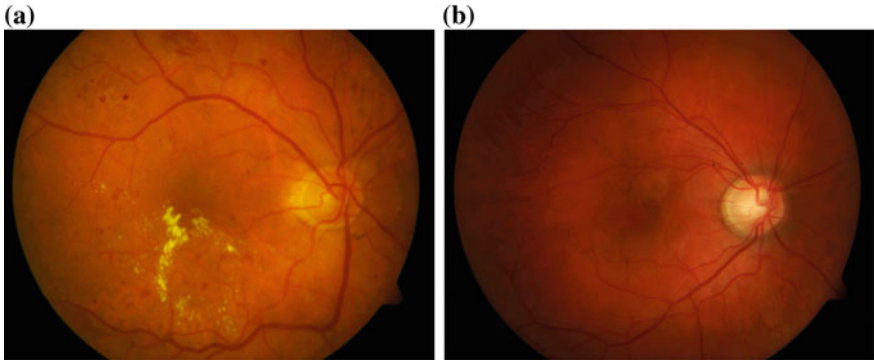


Fig. 2 a Diseased image. b Normal image

89 already extracted microaneurysms images and 89 already extracted hard exudates images (Fig. 2).

3 Feature Extraction

We have used eight features in this work and the flowchart is given below in Fig. 3 illustrates the steps followed in extracting the features. The features which are used in our work are given below:

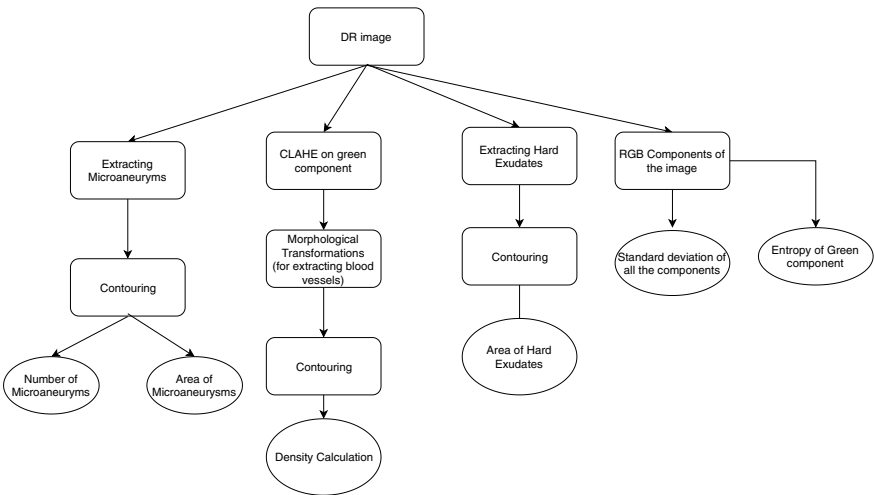


Fig. 3 Flowchart for feature extraction

1. Number of microaneurysms,
2. Area of microaneurysms,
3. The red components' standard deviation,
4. The green components' standard deviation,
5. The blue components' standard deviation,
6. The entropy of green component of the image,
7. Hard exudates' density,
8. The density of blood vessels.

3.1 Number and Area of the Microaneurysms with a Density of Hard Exudates

Inspired by the work of [10], we have also included these two features in our work. We first found edges of the microaneurysms using contours, and these contours are filtered by the area. Contours with large areas are removed. Remaining contours are used to find the number and area of the microaneurysm.

This method is also used for calculating the area of hard exudates and dividing it by total area leads to the density of hard exudates.

3.2 Blood Vessels Extraction

These are the steps which are performed for blood vessels extraction.

1. Apply ADHE [13] on the green component of the image.
2. Apply three times the morphological opening–closing operations on different size kernels to remove the noise.
3. Apply binary thresholding to further remove noise.
4. Using contours, filter the image using area and perimeter to remove circular structures that are not blood vessels.
5. Dilate the final image (Fig.4).

3.3 Entropy

Entropy is the measure of randomness [16]. In image processing, entropy might be used to classify textures, a particular texture might have one specific entropy as specific patterns repeat themselves in approximately certain ways. So, this helps in categorizing the images. The formula for entropy is given by

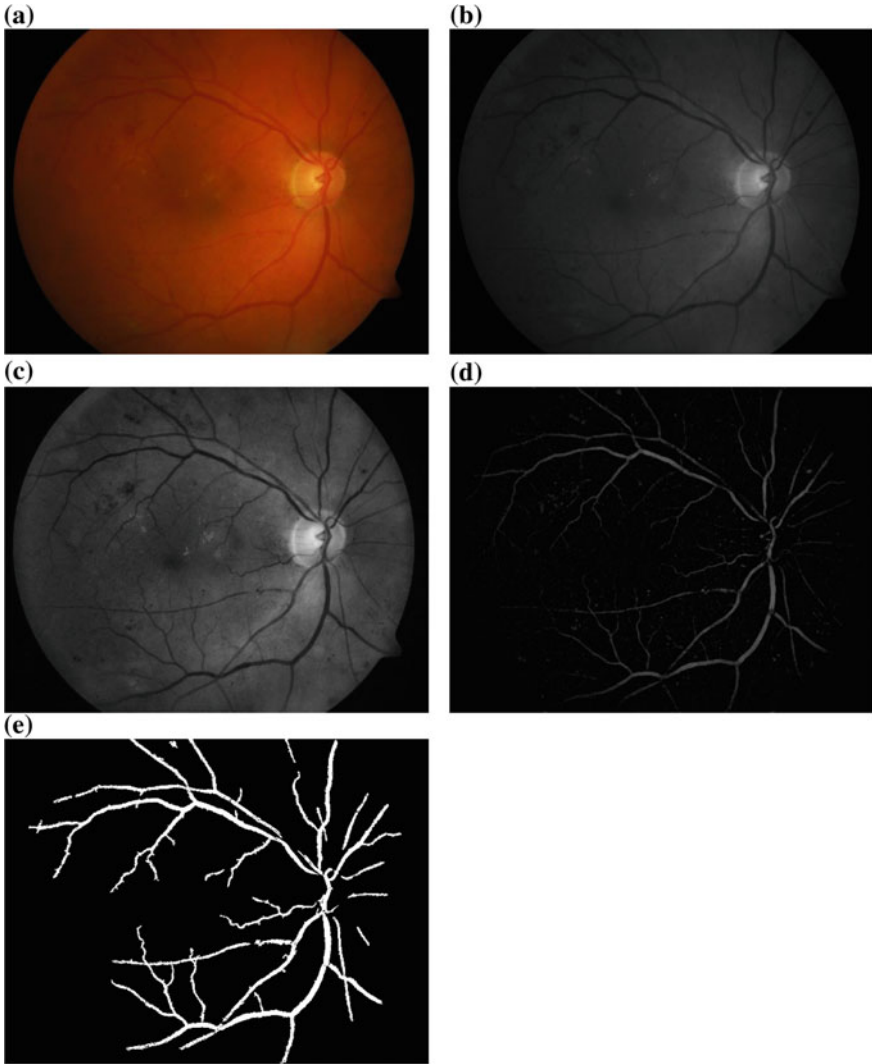


Fig. 4 a Original image. b Green component. c CLAHE. d Noise reduced. e Dilated final image

$$H(s_m) = - \sum_{n=1}^{256} p_n(s_m) \cdot \log_2(p_n(s_m)), m = 1, \dots, M \quad (1)$$

3.4 Standard Deviation

It gives the measure of variability from the mean value. The high value of standard deviation confirms that the values are dispersed more from the mean and a less standard deviation shows that the values are close to the average value. The representation of standard deviation is given as follows [4]: we have used the predefined function to find standard deviations of red, blue, and green components of the image.

3.5 Morphological Transformations

Morphological operations include basic operations like dilation and erosion. To perform these operations, two kinds of images are required and these are, one, the original image and the other one is the kernel to decide the nature of this operation.

4 Classifiers Used

1. Support Vector Machines (SVM)

Support Vector Machines (SVMs) [3, 5, 8] are the classifiers categorized in supervised machine learning models. SVM finds an optimal hyperplane by the help of two boundary lines which are maximally separable from each other. The distance between them is the maximal possible distance and the middle of these two boundary lines is considered as the final hyperplane. In SVM, support vectors are those points which touch these boundary lines. It is very clear that a good model will have less number of support vectors so the dependency can be minimized in case of any variation among the data points.

In this classifier, hinge loss is the loss function which is conventionally used with it [19]. For a training set $\{X_i, Y_i\}_{i=1}^m$ where m denotes the number of examples, the hinge loss is given by

$$L_{\text{hinge}} = \max(0, 1 - Y_i(w^T X_i + b)) \quad (2)$$

where w is the weight vector and b is the bias term. SVMs are also used for performing nonlinear classification by the help of kernel trick which implicitly maps the inputs into high-dimensional spaces.

2. Twin Support Vector Machine

Twin SVM (TWSVM) splits the complex quadratic equation of SVM into two simpler quadratic equations. It can be seen like dividing the complex problem into two parts. Because of two equations, two nonparallel hyperplanes are generated by TWSVM which are at least a distance of one from each other. These two hyperplanes are used to separate the two classes and the data points which are

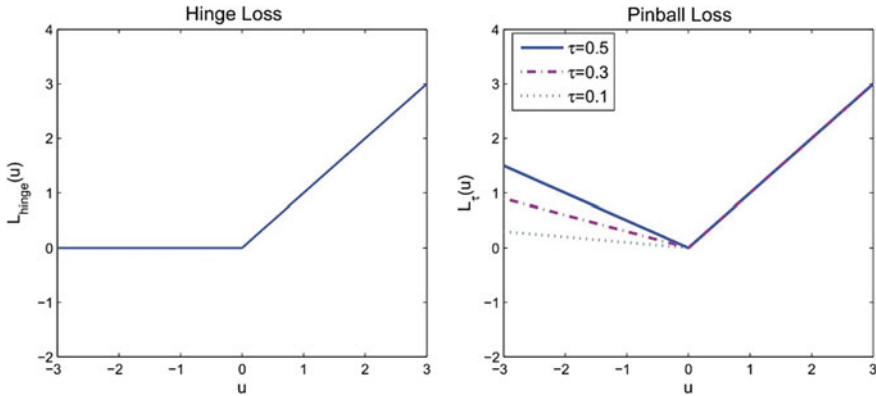


Fig. 5 Comparison of hinge loss and pinball loss

close to one hyperplane will belong to that class and rest to the others.

Like SVM, the conventional TWSVM also uses hinge loss function.

3. Twin Support Vector Machine using pinball loss. In this classifier, hinge loss function from the standard TWSVM is replaced by the pinball loss function to introduce robustness toward the noise. As TWSVM with hinge loss is sensitive to outliers, the new TWSVM was proposed which replaced the hinge loss function by the pinball loss function [20] as the pinball loss function is robust to noise and outliers. The pinball loss function is given as follows, which can be regarded as a generalized L1 loss (Fig. 5):

$$L_{\tau}(u) = \begin{cases} u & u \geq 0 \\ -\tau u & u < 0 \end{cases} \tag{3}$$

Now, the question arises: Why are we using TWSVM for DR detection problems and how pin-TWSVM affects the performance of a classifier in this problem? TWSVM is four times faster than SVM and hence the resulting classifier of this problem would be computationally efficient than SVM. Also, the presence of noise in the dataset severely affects the performance of the classifier, therefore, the introduction of the pinball loss function with TWSVM to the problem of DR detection can make the model robust to such noises.

4.1 Gaussian Noise

Gaussian noise is the statistical noise which incorporates a probability density function (PDF) equal to the normal distribution function, which is also known as the Gaussian distribution. The probability distribution function N of a Gaussian random variable x is given by

$$N_G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4)$$

where x represents the gray level, μ is the mean value, and σ is the standard deviation.

5 Numerical Experiments and Results

After performing the feature extraction as described earlier, we have implemented the classifiers, namely SVM, TSVM, and Pinball TSVM on the training set obtained from the division of feature set in the ratio of 80:20.

5.1 Hyperparameter Tuning Without Adding Noise

In this part of our work, we have shown the comparison of the above-described classifiers on the basis of accuracy over different regularization parameters. It should be noted here that till this point, we have not added any Gaussian noise in the feature set to test the robustness.

For the extracted features, we have compared the three classifiers described above on the basis of accuracy over different regularization parameters. It should be noted here that this is the problem of two-class classification which reports whether the fundus image is diseased or not.

From Fig. 6, we have observed that the TWSVM with pinball loss is giving comparable results.

The next part of this work discusses the effects of adding noise to the dataset.

Hyperparameter tuning for noisy data In this part of our work, we have shown the comparison of above-described classifiers based on accuracy over different regularization parameters. To test the robustness of the model, we have manually added Gaussian noise in the dataset (as we have described in Sect. 4.1) (Fig. 7).

It should be noted here that all the experiments are performed over ten times corresponding to all the three classifiers. To describe the variation in the accuracies, box plots are used. Furthermore, these box plots also show the variation in the accuracies after addition of Gaussian noise in our feature set.

From Figs. 8, 9, and 10, we can observe that the pinball loss function with TWSVM works better for DR detection problem (for both noise-free and noisy data).

The next part is the comparison of all the classifiers for DR detection problem over computation time taken.

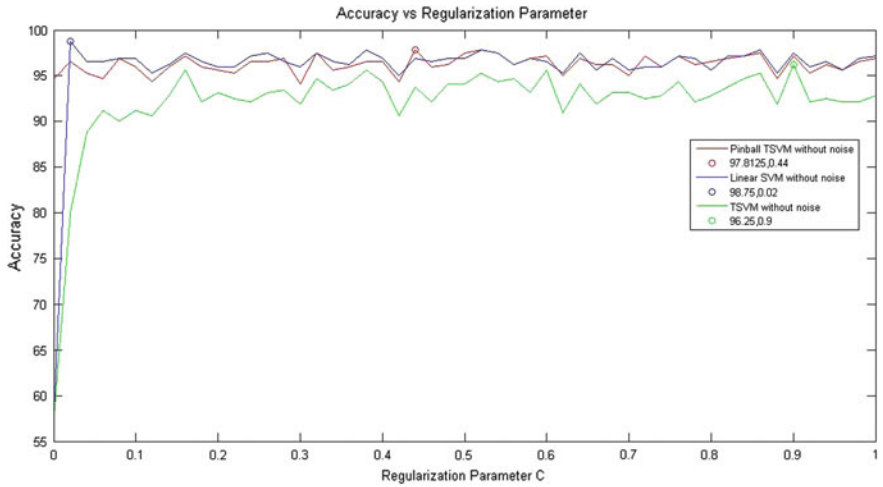


Fig. 6 Plot of accuracy versus regularization parameter (without noise) where $c = [0, 1]$

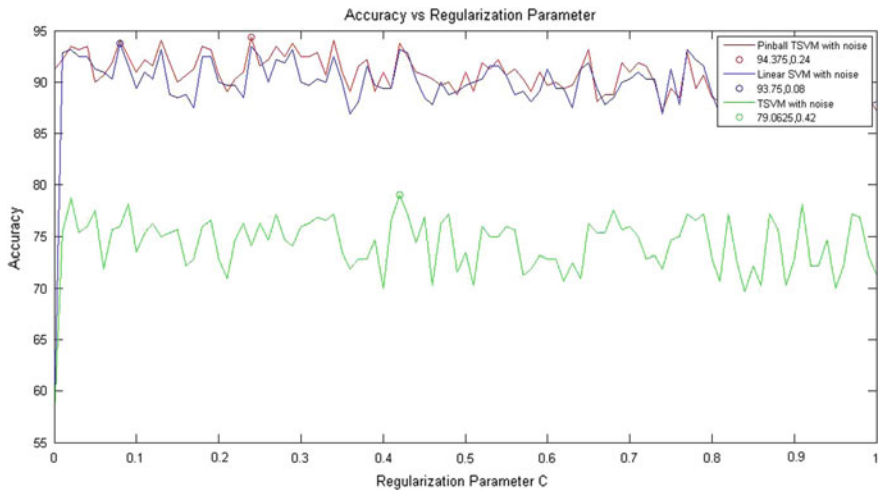


Fig. 7 Plot of accuracy versus regularization parameter for with noise data where $c = [0, 1]$

Now we have shown the comparison of accuracies of SVM and its variants in the tabular form (see Table 4.3). It should be noted here that we have also made this comparison on the basis of time consumption.

In Table 4.3, the best accuracies and the least time consumption are shown in bold (Table 1).

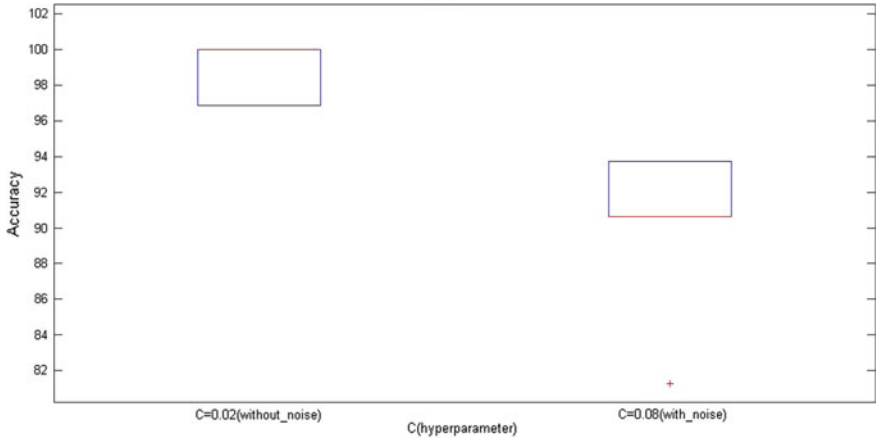


Fig. 8 Box plot of accuracies on linear SVM

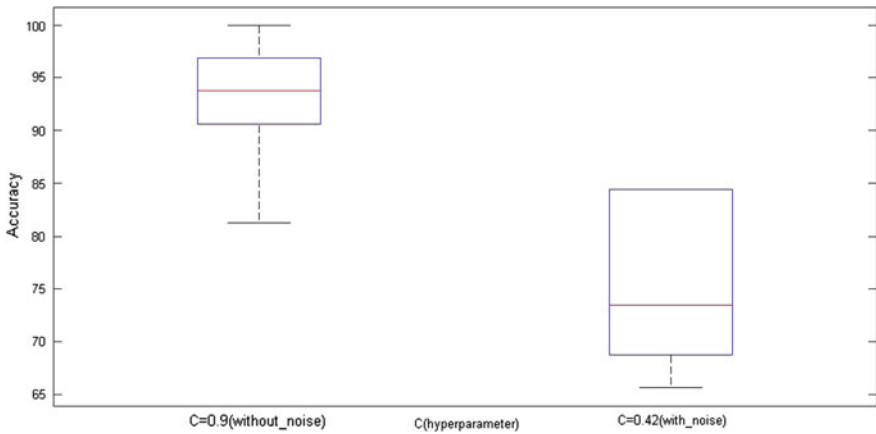


Fig. 9 Box plot of accuracies on TSVM with hinge loss

This table compares the performance of classifiers for DR detection problem. The comparison has been made on the basis of accuracy and time taken. From this, it can be observed that the TWSVM and TWSVM with pinball loss function can also be effectively applied to the problem of diabetic retinopathy detection.

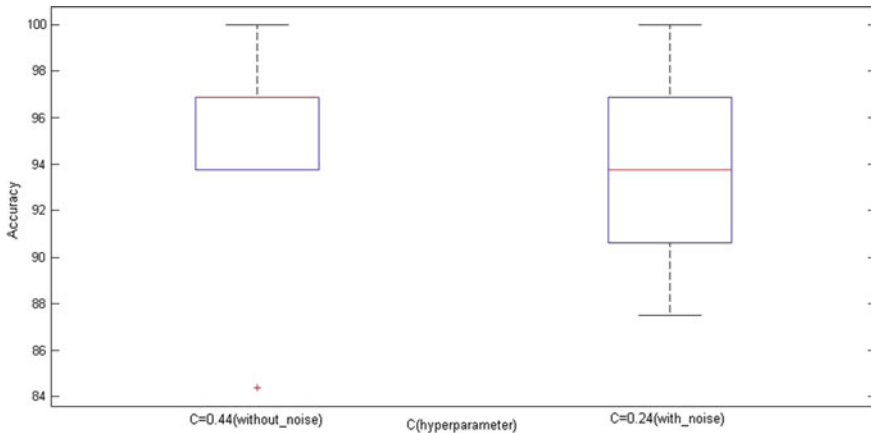


Fig. 10 Box plot of accuracies on TSVM with pinball loss

Table 1 Accuracy and time for different datasets using different SVMs

		Linear SVM	TSVM with hinge loss	TSVM with pinball loss
Diaretdb1 with noise	Accuracy (in %)	93.75	79.06	94.37
	Time (in s)	6.28	1.76	0.73
Diaretdb1 without noise	Accuracy (in %)	98.75	96.25	97.21
	Time (in s)	4.46	1.73	1.09

6 Conclusion

In this paper, we have used TWSVM with pinball loss function for the problem of diabetic retinopathy detection. To the best of our knowledge, it is for the first time that twin SVM has been applied to this problem. Also, this work reports a comparison with the earlier work of DR detection using SVM. We have used pin-TWSVM for DR detection so as to make the model robust to noise.

Therefore, the following conclusions can be drawn from this work:

1. TWSVM with pinball loss function for DR detection is the fastest among the rest of the above described classifiers.
2. TWSVM with pinball loss function for DR detection gives the best accuracy as compared to all the above described models for noisy data as well.
3. TWSVM with hinge loss is highly affected by noise whereas TWSVM with pinball loss function tolerates noise well.

As the use of TWSVM and its variant makes the training process computationally fast, this work can be extended to large datasets as well. Also, the use of data normalization techniques can help in improving the results.

References

1. J. Amin, M. Sharif, M. Yasmin, A review on recent developments for detection of diabetic retinopathy. *Scientifica* **2016** (2016)
2. G. Azzopardi, N. Strisciuglio, M. Vento, N. Petkov, Vessels delineation in retinal images using cosfire filters. *Reconstruction* **25**(9), 1200–1213 (2006)
3. C.J. Burges, A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* **2**(2), 121–167 (1998)
4. P.J. Burt, Fast filter transform for image processing. *Comput. Graph. Image Process.* **16**(1), 20–51 (1981)
5. C.C. Chang, C.J. Lin, Libsvm: a library for support vector machines. *ACM Trans. Intell. Syst. Technol. (TIST)* **2**(3), 27 (2011)
6. R. Gargeya, T. Leng, Automated identification of diabetic retinopathy using deep learning. *Ophthalmology* **124**(7), 962–969 (2017)
7. A. Hoover, V. Kouznetsova, M. Goldbaum, Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response, in *Proceedings of the AMIA Symposium*. American Medical Informatics Association (1998), p. 931
8. T. Joachims, Text categorization with support vector machines: Learning with many relevant features, in *European conference on machine learning* (Springer, Berlin, 1998), pp. 137–142
9. T. Kauppi, V. Kalesnykiene, J.K. Kamarainen, L. Lensu, I. Sorri, J. Pietila, H. Kalviainen, H. Uusitalo, Diaretdb1-standard diabetic retino-pathy database. *Last Modif.* **6**, 19 (2007)
10. S. Kumar, B. Kumar, Diabetic retinopathy detection by extracting area and number of microaneurysm from colour fundus image, in *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)* (IEEE, 2018), pp. 359–364
11. J. Lachure, A. Deorankar, S. Lachure, S. Gupta, R. Jadhav, Diabetic retinopathy using morphological operations and machine learning, in *2015 IEEE International Advance Computing Conference (IACC)* (IEEE, 2015), pp. 617–622
12. P. Powar, C. Jadhav, Retinal disease identification by segmentation techniques in diabetic retinopathy, in *Information and Communication Technology for Sustainable Development* (Springer, Berlin, 2018), pp. 255–265
13. R. Raju, Optimisation of image processing networks for neuronal membrane detection. Ph.D. thesis, University of Nottingham (2016)
14. S. Roychowdhury, D.D. Koozekanani, K.K. Parhi, Dream: diabetic retinopathy analysis using machine learning. *IEEE J. Biomed. Health Inform.* **18**(5), 1717–1728 (2013)
15. S. Roychowdhury, D.D. Koozekanani, K.K. Parhi, Iterative vessel segmentation of fundus images. *IEEE Trans. Biomed. Eng.* **62**(7), 1738–1749 (2015)
16. J.B. Saaddine, A.A. Honeycutt, K.V. Narayan, X. Zhang, R. Klein, J.P. Boyle, Projection of diabetic retinopathy and other major eye diseases among people with diabetes mellitus: United states, 2005–2050. *Arch. Ophthalmol.* **126**(12), 1740–1747 (2008)
17. D.S. Sisodia, S. Nair, P. Khobragade, Diabetic retinal fundus images: preprocessing and feature extraction for early detection of diabetic retinopathy. *Biomed. Pharmacol. J.* **10**(2), 615–626 (2017)
18. J. Staal, M.D. Abràmoff, M. Niemeijer, M.A. Viergever, B. Van Ginneken, Ridge-based vessel segmentation in color images of the retina. *IEEE Trans. Med. Imaging* **23**(4), 501–509 (2004)
19. Y. Wu, Y. Liu, Robust truncated hinge loss support vector machines. *J. Am. Stat. Assoc.* **102**(479), 974–983 (2007)

20. Y. Xu, Z. Yang, X. Pan, A novel twin support-vector machine with pinball loss. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(2), 359–370 (2016)

Semantic-Based System for Exercise Programming and Dietary Advice



Givi-Giorgi Mamatsashvili, Konrad Ponichtera, Mikołaj Małkiński,
Maria Ganzha and Marcin Paprzycki

Abstract Growing health awareness results in interest in healthy eating and “fitness”. While information about exercising and dieting is readily accessible, it is difficult, for an inexperienced person, to find what is right for her/him in terms of *both* diet and exercise. The aim of this work is to introduce a system, based on semantic technologies, which addresses user goals with joint exercise programs and meal suggestions.

Keywords Ontology · Semantic technologies · Inference · Nutrition · Fitness · Cloud · Scalability · High availability

1 Introduction

Recently, people become, more than ever, conscious about diet and exercise. This results in the creation of multiple “fitness tools”. Since numerous exercise programs and diet plans can be found online (see, Sect. 2), why do we need yet another application? However, while certain exercise programs work for some, they might not work for other users. Therefore, existing programs might not be “optimal” for the given person. Note also that such programs ignore user health, experience, and fitness goals. Here, personal trainer can help, but such service can be expensive.

Overall, while human input seems needed, the question arises: how to minimize it (before it becomes counterproductive). Here, applications should generate exercise programs, while allowing customization (where variants should work as well as the original one). Similarly, diets should offer variants, as people often do not want

G.-G. Mamatsashvili · K. Ponichtera · M. Małkiński · M. Ganzha
Warsaw University of Technology, Warsaw, Poland

M. Ganzha · M. Paprzycki
Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

M. Paprzycki (✉)
Warsaw Management University, Warsaw, Poland
e-mail: marcin.paprzycki@ibspan.waw.pl

© Springer Nature Singapore Pte Ltd. 2020
L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_10

to follow strictly defined diets. Hence, the application should collect personalizing information during registration, and use it to generate advice.

Take, for example, an intermediate lifter who wishes to get stronger. Assume that (s)he formulates goal(s), history, and characteristics. Here, a personalized program should be developed, aimed at the development of muscles that (s)he specified while suggesting low-calorie meals, to help loose weight. Moreover, if (s)he does not like certain exercise(s), the application should propose a modified program, which still takes into account her/his preferences.

To develop the needed application, domain knowledge must be captured. We have selected ontologies, to represent exercises, meals, ingredients, muscles, etc., and their relations (in the context of fitness). The developed ontology allows the application of semantic technologies to “ask proper questions” and infer answers. Thus, in what follows, we discuss (i) related works, (ii) design and implementation of the proposed application, and (iii) present an example of how it works.

2 Related Work

Let us start with fitness applications. Here, Stronglifts¹ provides exercise programs and the ability to track them. It suggests weights adjustments, e.g., when to decrease the weights, during certain exercises. It is possible to log details of workouts and export logs to other devices. However, user preferences are not considered. Furthermore, support for experienced lifters is limited.

Freeletics² specializes in nutrition and fitness. Here, few features are free, but more advanced ones require subscription. In the application, one has access to various workouts and tools, such as nutrition guidance and a digital AI coach.

Both these apps can be a great resource for newcomers. However, they lack adaptivity, flexibility, and features for experienced users. Although Freeletics features the digital coach, it does not *combine* exercises with meal suggestions.

Semantic technologies have been used by authors of [1], to help understand diet. Proposed agent system relies on an ontology containing nutritional information about foods, collected from convenience stores in Taiwan. Here, users “inform the system” what food they consume. This information is analyzed and confronted with what is determined to be a “healthy diet”. Overall, this is a good example of how ontologies can support a better understanding of health.

Similarly, work presented in [2] discusses the use of biomedical ontology, to support clinical decisions. Here, the ontology, containing relevant clinical information, aids detecting irregularities, such as wrong diagnoses, unobserved diseases, etc. Due to the intricacy of the problem, ontologies are used to represent complex knowledge. This illustrates how semantic technologies can be used to develop an “assistant”, tasked with aiding the user.

¹<https://stronglifts.com/>.

²<https://www.freeletics.com/>.

3 Proposed Approach

In this context, we have decided to use ontologies to represent knowledge about physical exercises and nutrition. The proposed system consists of two main components, (1) related to exercise and (2) dietary advice.

3.1 Knowledge Representation

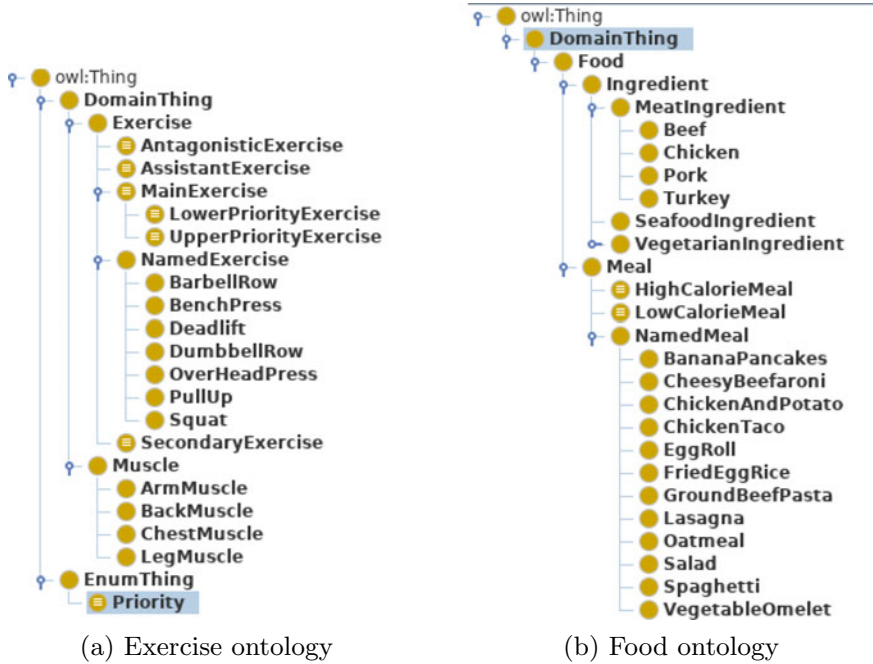
In [3], an ontology is defined as an explicit specification of a conceptualization. Ontologies formally represent knowledge, by capturing entities within a domain and expressing relations between them. Ontologies are expressed in standardized formats (RDF, RDFS, and OWL), which can be interpreted by a machine.

Our solution introduces two ontologies, representing “fitness” and “food”. Ontology of fitness captures relations between physical exercises and muscles (body function), while food ontology models nutritional attributes of meals. Ontologies, forming knowledge base, are independent of the remaining parts of the application and can be modified, without changing other components of the system. Furthermore, stored knowledge can be shared between applications, or exposed to the outside world (e.g., within Linked Open Data³) to be (re)used.

Exercise ontology To the best of our knowledge, there are no ontologies of physical exercises. Hence, we have developed our own (if a better one exists, our ontology can be replaced by it). It is based on a specific “exercise philosophy”, well recognized in the community. Here, exercises are classified as *antagonistic*, *assistant*, *main* or *secondary*, assigned into groups, and arranged in a specific order. Moreover, different arrangements are recommended, based on users experience: *beginner*, *intermediate* or *advanced*. Figure 1a (left panel) illustrates how these concepts are represented in the exercise ontology. The top level concept is *DomainThing*. It has two subclasses: *Exercise* and *Muscle*. The former contains one *primitive* class, the *NamedExercise*, which has subclasses for each exercise type. Exercise classes can have, one or many, instances, capturing their variations. For example, *Deadlift* has two instances: *RomanianDeadlift* and *StiffLeggedDeadlift*, which are similar, but technically different. Note that this is not a comprehensive ontology of fitness. We have developed it to the point where its usefulness can be shown. Hence, if the proposed approach is to become a real-world application, this ontology has to be further developed/extended.

All instances in exercise ontology can have object properties, listed in Fig. 2. Based on these properties, an exercise is inferred as belonging to one of the following *defined* classes: *AntagonisticExercise*, *AssistantExercise*, *MainExercise*, or *SecondaryExercise*. Namely, an exercise is a *MainExercise* if it has *lower* or *upper* priority, represented by an attribute *hasPriority*, or a *SecondaryExercise* if it develops a specific muscle and is secondary to a main exercise, represented by attributes

³<https://lod-cloud.net>.



(a) Exercise ontology

(b) Food ontology

Fig. 1 Exercise and food ontologies—top level concepts

Fig. 2 Object properties from the exercise ontology

- hasAntagonist
- hasAssistant
- hasDevelopee
- hasPriority
- hasSecondary
- hasSpecialization
- isAntagonisticTo
- isAssistantTo
- isDevelopedBy
- isPrioritizedBy
- isSecondaryTo
- isSpecializedBy

hasDevelopee and *isSecondaryTo*. Additionally, each *MainExercise* can be classified as either *Lower* or *UpperPriorityExercise*. This division is possible thanks to a sibling class of *DomainThing*, called *EnumThing*, which has a single subclass *Priority*. These rules are described as equivalence statements for the defined classes (see, Fig. 3). Here, *defined* classes, and *object properties*, are key to the exercise program generation algorithm.

Fig. 3 Definition of a *SecondaryExercise*



Users can specify muscles they want to develop. For this purpose, we have introduced the *Muscle* class, with four subclasses: *ArmMuscle*, *BackMuscle*, *ChestMuscle*, and *LegMuscle*. Each individual muscle, e.g., *Triceps*, is an instance of a subclass. This allows capturing relationships between exercises and muscles.

Food ontology Several ontologies deal with food/nutrition (see, also, Sect. 2). However, they are either too detailed [4], or focused on a “problem not relevant in our context”. Moreover, choices like [5], do not capture all factors needed for meal recommendations. Hence, we have created an ontology, representing meal ingredients, their calories, and how they are composed into meals. The structure of the food ontology, is presented in Fig. 1b. Here, ingredients have been separated into: *MeatIngredient*, *SeafoodIngredient* and *VegetarianIngredient* classes. Each of them has subclasses. For instance, class *MeatIngredient* has the following subclasses: *Beef*, *Chicken*, *Pork* and *Turkey*.

Composition of ingredients into meals is expressed through instances of subclasses of the *NamedMeal* class, e.g., class *Salad* has instances, representing different salads (with slightly different ingredients). Ingredients are represented through *hasIngredient* property. Additionally, meal instances have an approximate number of calories, described using *hasCalories* property.

Again, this is a “minimal ontology” developed for the prototype, and should be viewed as such. However, it allows to capture user preferences; e.g., availability of a number of calories allows recommending meals appropriate to lose, maintain or gain weight. Moreover, the structure of the ontology allows dealing with special requirements, such as food allergies or suggesting vegan meals (see, [6]). Finally, it can allow suggesting meals composed of ingredients present in the users kitchen (in the presence of an IoT-enabled refrigerator, see Sect. 4).

4 Technical Aspects

Let us now briefly describe the key technical aspects of the developed system/application and its components.

4.1 System Description

The application should be accessible from a web browser on a desktop, or a mobile device (phone/tablet). Hence, the interface has been created with Angular framework.

Fig. 4 Diagram showcasing individual parts the solution is composed of. In the IoT module, the vendor API connection is replaced with mocked database

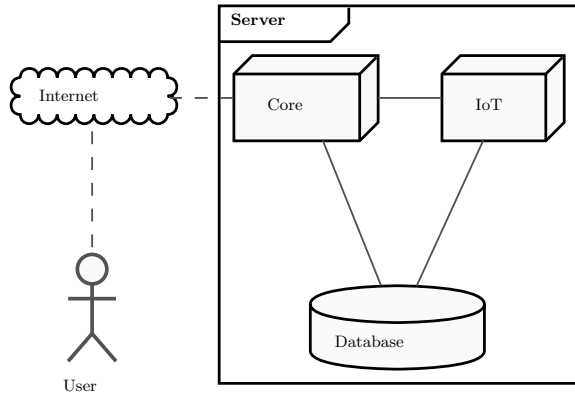


Fig. 5 Sample SPARQL query, listing exercises assistant to classic bench press

```
PREFIX sf: <http://www.stayfit.oof/ontologies/exercise#>
SELECT ?exercise
WHERE {
    ?exercise sf:isAssistantTo sf:ClassicBenchPress
}
```

The server-side part was developed in Java, using Spring Boot suite. It consists of two modules: *Core* and Internet of Things (*IoT*), depicted in Fig. 4. *Core* takes care of all aspects of the application, e.g., connection with the database, serving Angular interface, creating and authenticating users, managing session and handling user requests. The *IoT* is to deal with home appliances, such as smart refrigerators. It serves as a proxy between the *Core* and appliance(s). Currently, it allows only delivering individual dietary advice, where meals are composed of ingredients found in “smart fridges”.

The proposed approach allows separation of business logic (*Core* module), from handling external interfaces. Since modules are “independent”, if one of them requires an update (e.g., due to change in vendor API), it can be updated without altering the other one. Currently, for testing purposes, the *IoT* module is emulated by a database with information about ingredients “found in smart fridge”.

4.2 Application of Semantic Technologies

Presented solution uses ontologies (from two domains) and reasoning (see, Sect. 3). Ontologies are represented in OWL, while operations on them use Apache Jena,⁴ which allows formulating SPARQL⁵ queries. A sample SPARQL query, returning all exercises “assistant to classic bench press”, is depicted in Fig. 5.

⁴<https://jena.apache.org/>.

⁵<https://www.w3.org/TR/rdf-sparql-query/>.

```
sf:MainExercise(?e) ^ sf:Deadlift(?e)
-> sf:isAntagonisticTo(sf:HipHinge, ?e)
```

Fig. 6 Example SWRL rule. It states that if exercise e is a *main exercise* and a *deadlift*, a *hip hinge exercise* will be *antagonistic* to e

A query can be executed against the ontology graph, or obtained through an inference process (see, Sect. 4.3). The latter is used for extraction of knowledge, which was not explicitly stated during ontology creation. Moreover, Semantic Web Rule Language (SWRL⁶) was used to express logical implications, inferring additional triples. An example SWRL rule is presented in Fig. 6.

The left-hand side of the implication contains prerequisites for the right-hand side to hold. In this case, if exercise e is the main exercise and is of deadlift type, then every hip hinge exercise will be antagonistic to it.

It is possible to build not only primitive rules, but also defined ones, like `MainExercise`, subclasses of which are populated during reasoning, as opposed to primitive classes where their subclasses are found explicitly in the ontology. This significantly increases reasoning capabilities and allows to “keep the ontology clean” by splitting reasoning into taxonomy-based and rule-based. However, misuse of SWRL rules might lead to reasoning problems. Hence, every rule should be very well documented, so that its selective testing is possible.

4.3 Inference

The proposed approach depends on semantic reasoning and the use of SWRL. Here, we have found that the native reasoner, built into Apache Jena, does not support SWRL. Thus we have used an external reasoner, the Openllet⁷ (fork of Pellet⁸), capable of performing both OWL DL reasoning and resolving SWRL rules.

4.4 Scalability and High Availability

Today, monolithic solutions are, very often, being replaced by microservice-based ones. This results in the creation of deployment methods independent from physical machines [7]. The key factor is fast launching of new instances, without excessive and time-consuming configuration. One of the possible approaches is containerization. Containers are similar to virtual machines, but are created from templates and,

⁶<https://www.w3.org/Submission/SWRL/>.

⁷<https://github.com/Galigator/openllet>.

⁸<https://github.com/stardogunion/pellet>.

usually, don't have preallocated resources. In the developed solution all components were containerized using Docker⁹ container engine.

Although containers do not offer the same level of isolation as the virtual machines, they became an important building block of cloud environments. However, they are still just a runtime method. They do not provide load balancing, high availability, or failure recovery. This created demand for another layer of abstraction, for dynamic creation and management of the pool of containers (which became known as container orchestration). For this purpose we have decided to use Kubernetes¹⁰ orchestrator. It takes care of managing the pool of physical container hosts and deploying the application in a highly available manner (resilient to failure). The example of Kubernetes deployment, of the presented solution is showcased in Fig. 7.

Having the cluster configured, it is possible to deploy the whole solution at once. Moreover, the orchestrator will try to evenly schedule the pods among cluster nodes in order to achieve the best availability in case of node failure. All the pods communicate with each other thanks to the virtual pod network, which spans among all nodes of the cluster. User, trying to load the application, sends a request to the global load balancer. It can be configured in the cluster, or rented from the cloud service provider. The global load balancer ensures uniform load distribution among pods of the same type so that no pod is flooded with an excess amount of requests.

Due to the modular structure, it is easy to add nodes to the cluster, to increase overall computing power, and to secure availability in case of node failure. Kubernetes tries to automatically ensure that all deployed applications are available and reschedules pods from the dead nodes on the remaining ones.

Choosing Docker and Kubernetes, as fundamental for the deployment, resulted in creating extremely scalable and reliable environment, ready for the cloud-oriented market. What's more, due to adoption of standardized open source solutions, it was possible to achieve that without falling into vendor lock-in, typical for Platform as a Service (PaaS) environments.

Obviously, such complex deployment was not needed for the demonstrator. However, we have decided to ensure that large-scale deployment is possible. We have tested the infrastructure and found that, as expected, it was as scalable and as resilient as the Docker+Kubernetes pair guarantees (see, Sect. 5).

5 Experimental Verification

The proposed application has been thoroughly tested for various usage scenarios. Let us discuss a simple example of a typical usage, showcasing the main functionality of the application, along with its current shortcomings.

⁹<https://www.docker.com>.

¹⁰<https://kubernetes.io>.

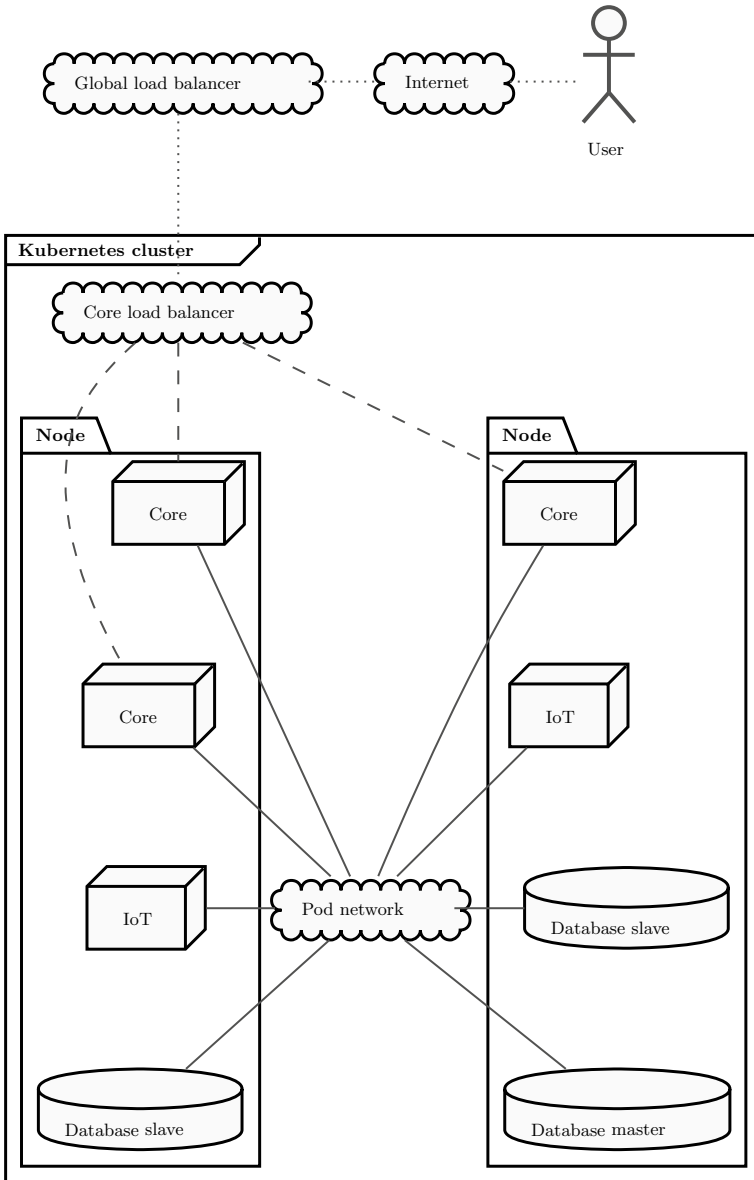


Fig. 7 Diagram example double node Kubernetes deployment of three instances of the core module and two instances of IoT module as well as the master-slave database. The presented IoT module usage is the testing one, where vendor API connection is replaced with mocked database

5.1 Typical Usage Scenario

Let us assume that user is a 23 year old male, 179 cm tall, who weighs 87 kg. He creates an account and provides the above data during registration. He also states that he wishes to lose weight, and wants only basic training—he has time to train only a few times per week, for a limited number of hours. Furthermore, he identifies himself as an intermediate lifter, and chooses to train hamstrings during deadlifts, triceps during bench presses, upper back during overhead presses, and quadriceps during squats. Once the registration is complete, the application generates an exercise program, which may look as in Fig. 8.

The proposed program includes four workouts, each of them based on specific barbell movements: Bench Press, Deadlift, Overhead Press, and Squat. Each of these workouts includes the main exercise, primarily used the strength and body development as it requires the use of a variety of muscles. These exercises are preceded by antagonistic exercises, i.e., exercises focusing on the muscles, which oppose those used during the main exercise. As a third exercise, an assistant exercise, which helps strengthening of the core for a smoother training experience is suggested. The latter half of the workout is very similar to the former. Once again, we have antagonistic and assistant exercises, but the main exercise is replaced by a secondary exercise. The

Current programme

Workout 1:

- Part 1:
 - Explosive jump variation (4x8)
 - Low-bar squat (4x8)
 - Sit-up (4x8)
- Part 2:
 - Explosive jump variation (4x8)
 - Front squat (4x8)
 - Plank (4x8)

Workout 3:

- Part 1:
 - Kettlebell swing (4x8)
 - Conventional deadlift (4x8)
 - Plank (4x8)
- Part 2:
 - Keg throw (4x8)
 - Romanian deadlift (4x8)
 - Sit-up (4x8)

Workout 2:

- Part 1:
 - Bodyweight pull-up (4x8)
 - Push press (4x8)
 - Core stabilization (4x8)
- Part 2:
 - Weighted pull-up (4x8)
 - Z press (4x8)
 - Bicycle crunch (4x8)

Workout 4:

- Part 1:
 - Two-arm dumbbell row (4x8)
 - Classic bench press (4x8)
 - Anti-rotational press (4x8)
- Part 2:
 - Bent-over row (4x8)
 - Close-grip bench press (4x8)
 - Core stabilization (4x8)

Generate new

Fig. 8 Example exercise program, generated for intermediate user

secondary exercise is a variation of the main exercise and specializes in developing the user specified muscle group.

Main exercises have their priority specified, i.e., if we declare some exercise to prioritize lower or upper part of the body, it can be treated as the main exercise. Here, the inference is done by checking the property of the existence. Antagonistic exercises are inferred through SWRL rules. For each exercise, where antagonism can be declared, there is a dedicated SWRL rule, which specifies classes of exercises. For example, hip hinge is antagonistic to each exercise e , which is also a deadlift. Note that e is *not* explicitly asserted as a main exercise. This fact is resolved by the reasoner first so that the SWRL rule engine will be provided with an already inferred taxonomy. Similarly in the case of assistant exercises; each of them has a dedicated SWRL rule, which defines the assistance to some main exercise. For instance, if some exercise prioritizes lower part of the body (and, by definition, is the main exercise), sit-ups are known to be assistant to it. Inference of secondary exercises is also done by defining an SWRL rule for each such exercise (having a decline bench press, it is secondary to some exercise if that exercise is the main one and a bench press).

Muscles are individuals, divided into four classes: arm, back, chest, and leg muscles. They can be specialized by some exercises. That fact is resolved in two ways; either by explicitly asserting that some individual exercise has specialization of an individual muscle (e.g., front squat has specialization of quadriceps) or by using SWRL rules. Due to limitation of OWL DL, it is not possible to assert a triple of the form individual property class, e.g., incline bench press (individual) has specialization (property) of the chest muscle (class). As a workaround, it is possible to apply SWRL rules, to state that if some muscle is a chest one, it is specialized by the incline bench press. Because “is specialized by” and “has specialization” properties are inverses of each other, having such rule defined, the SWRL engine resolves that each incline bench press has specialization of every asserted and inferred chest muscle, which is the desired result.

In summary, we can state that the inference on the exercise ontology worked just as expected. An intermediate lifter has been presented with a program consisting of four workouts, each with relevant exercises.

Having the program generated, the user is able to log the workout in the application, as (s)he has completed it, by stating the number of successful repetitions of each prescribed set (see, Fig. 9).

Assuming that eight successful repetitions took place during each set, once the user finishes her/his workout, a new entry will appear in the workout history dashboard (see, Fig. 10).

Specifically, the newly registered session will be marked as successful, due to the fact that 8 repetitions were completed. In the case when the session was not successful, it will still be registered in the history, along with the number of completed repetitions.

Separately, in the meal advice view (see, Fig. 11) the user is presented with three meal choices—chicken taco, peanut butter oatmeal, and an egg roll. The meals, shown on the screen, were all described as having less than 650 calories per serving. This property has been used to classify them as low-calorie meals. The suggestion is made

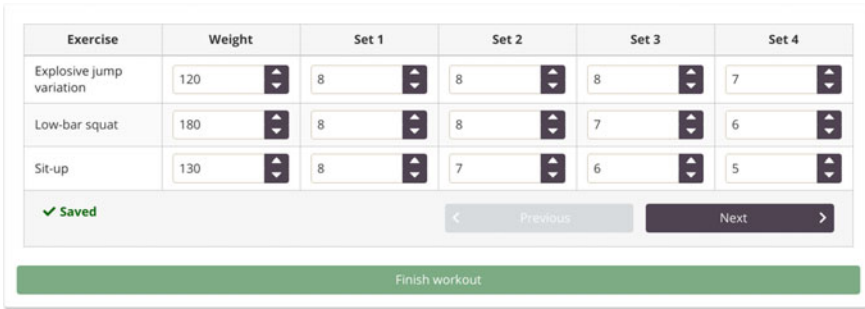


Fig. 9 The view of ongoing workout

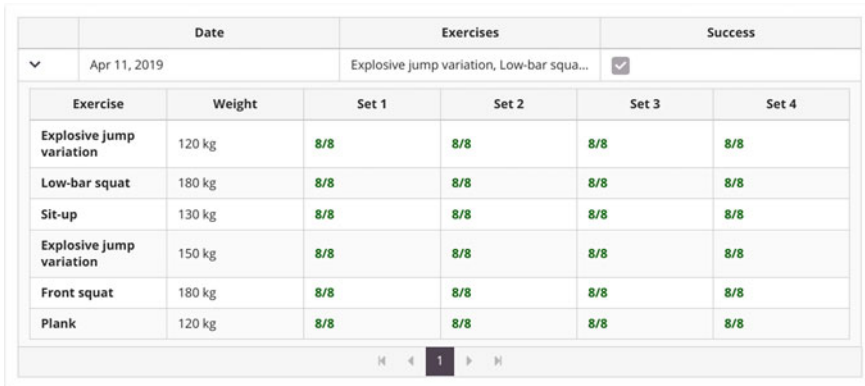


Fig. 10 Successfully finished workout

based on the fact, that the user is attempting to lose weight (see, above). In this case, the user should be presented only with low-calorie meals. Similarly, the user who wants to gain weight would be presented with meals inferred to be high-calorie ones (having more than 750 calories per serving) while in case of “weight maintaining” all types of meals would be presented, including high-calorie, low-calorie, and those in between. It is up to the users to control the number of calories they consumed on the given day and reach their daily calorie goal. This is why they are presented with a progress bar they can fill by marking the meals they consumed.

Each meal has a list of ingredients, along with an approximate number of calories. When the user marks the egg roll, as the consumed meal, the distance from reaching daily calorie goal changes from 1937 to 1374 (as egg roll has approximately 561 calories). This means the user has already consumed the 29% of calories (s)he should consume during the given day, in order to lose weight.

Meals are OWL individuals, of specific primitive classes, having constraints in the form of superclasses. For example, a peanut butter oatmeal is an individual of type *Oatmeal*, which is a subclass of a meal that has oat flakes as an ingredient.

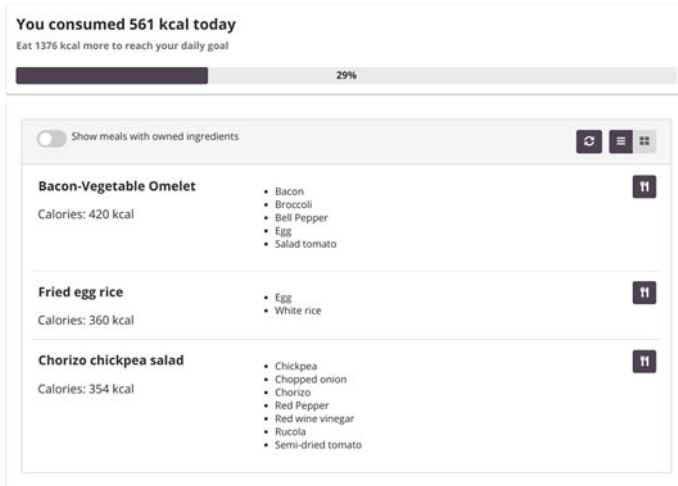


Fig. 11 Meal advice screen, showing amount of consumed calories on a given day, as well as other meal propositions

Since ingredients are also individuals of some classes formed into a taxonomy (for example, bacon, which is known to be meat) there is a broad possibility for extending the ontology and program functionality to include potential user allergies and intolerances, as well as individual preferences (see, also [6]).

Users have also access to meals that can be made using ingredients stored in their smart fridges (but these would be suggested in a separate view of the application; see, also, Sect. 4.1).

5.2 *Kubernetes’ High Availability Scenario*

Separately, we have tested the use of Docker+Kubernetes. Due to Kubernetes’ health probing mechanisms, the whole system should be immune to failures of its components, no matter if the failure occurred on the application or machine level.

The fundamental high availability principle requires that each component of the developed solution is available in at least two instances. It can be easily seen, in the Fig. 7 that, no matter which logical or physical component of the solution is taken down, the whole system is still going to function normally, without users knowing that some incident has occurred on the server-side.

Software failure scenario Software failure can occur in the case when the application in one of the Kubernetes’ pods crashes (e.g., due to lack of available memory) or becomes unable to serve the requests properly (e.g., loses connection with the database or the IoT vendor servers). It can be easily simulated by enforcing pod

shutdown. Although Kubernetes will immediately proceed with the creation of a new pod instance, it will take some time before it “makes sure” that the pod is working properly and won’t redirect the requests to it, in the meantime. One can try to use the functionality of the killed pod before the new one becomes ready. We have tested that in the *Core* module pod, which died, it was still possible to perform operations like logging in, generating new workout, etc. Similarly, when the IoT module pod was taken down, it was still possible to request dietary advice, composed of ingredients from the emulated refrigerator.

Hardware failure scenario Hardware failure is more serious because, unlike in the case of software one, Kubernetes is not able to revive the unavailable node; it has to be done manually by the administrator. The failure can be simulated either by detaching the network cable from the node, or simply by unplugging its power cord. All the pods of the node will become detached and unable to process forwarded requests. The control plane will probe the node and then mark it as unavailable, which will result in all internal load balancers to omit that node during traffic handling. In the meantime, the cluster itself will remain operational, having at least one pod of each type deployed on the remaining nodes. Moreover, Kubernetes might schedule pods from a failed node on the remaining ones, in order to enhance load balancing. When all of the mentioned things are happening, the user is not able to notice that failure and recovery occur on the server-side. In the worst case, if the user’s request was meant to be just handled by the pod from the failed node, there might be a (slight) delay in request serving, while the cluster’s internal mechanisms redirect the requests to the working node. The delay is linearly decreasing as the number of worker nodes of the cluster increases because the probability that the user will be served by that particular node decreases proportionally to the cluster’s size.

It is worth noting that administrator’s intervention is required only in the case of the bare-metal cluster, deployed manually on physical machines. In case when the cluster is provisioned by the cloud provider, recovery from the node failure is going to be automated, just as in case of software pod failure.

Maintenance scenario Cluster nodes might also need maintenance, which requires disabling their pod scheduling capabilities and gracefully taking down all of currently working pods. This process is called node cordoning and draining. Nodes that are shut down are treated just like failed ones, with the exception that Kubernetes is not probing them to check if they are available again. During node draining, each evicted pod has its copy created (on one of the remaining nodes) so that user is not able to notice that some server instances are being shut down. Once the maintenance is complete, nodes can be uncordoned and reattached to the cluster, which will proceed with pod scheduling.

In summary, all performed tests illustrated that the Docker+Kubernetes infrastructure is capable of efficiently “protecting” the running application.

5.3 *Shortcomings and Limitations*

One of the main assumptions, behind the developed application, is to minimize the human involvement in providing fitness and dietary advice. This is to be achieved by utilizing semantic technologies and ontologically capturing domains of interest into a semantic knowledge base. However, decision-making is not perfect. While the generated exercise programs might be efficient in terms of achieving one's weight goal, user satisfaction cannot be guaranteed, as there are far too many factors that the application cannot foresee (being in its current form). It is also worth noting that there is no way to guarantee the accuracy of daily caloric intake provided by the system, as every person is unique. This can cause the numbers to be significantly off for some people. This will also require users to readjust the number of calories, by experimenting with the suggestions. This, basically, means reintroduction of human involvement that the application tries to get rid of in the first place.

However, most importantly, the application is not universal. The exercise ontology, used in it, adopts a certain exercise philosophy and, while being capable of generating numerous programs, it lacks the variety that users might be looking for. Moreover, users may believe that a different exercise philosophy would be better for them. This problem can be solved, by replacing fitness ontology by a different one (with the remaining parts of the application unchanged).

As noted, the current version of food ontology does not take account food allergies, or strict diets, as its based solely around the idea of calories and classifying food based the calorie count. Acknowledging these limitations, it is crucial to take into account that this application is only a prototype and can be improved in numerous ways. Furthermore, it was designed in such a way that incorporating such changes should be relatively easy, e.g., due to modular design and extensible nature of ontologies.

6 **Concluding Remarks**

In this paper, we have presented a semantic-based system for exercise programming and dietary advice. A demonstrator has been developed, in order to test the presented solution and to better understand the needs of the users. The system has successfully generated exercise programs, based on provided preferences, while also suggesting some meals that the user could be interested in.

Shortcomings and limitations discussed above will be prioritized in the upcoming versions. Moreover, the system can be further improved by adding more knowledge to the ontologies, more specifically, we aim to improve the food ontology so that it can satisfy the needs of various users. This can be planned by considering the diets the users are following, such as plant-based or ketogenic diets, or the foods they are unable to consume due to certain intolerances and allergies. Finally, for convenience, the system can also be integrated with personal assistants like Amazon

Alexa or Google Assistant. Our goal is to expand and further improve the solution in the future. This is going to be relatively easy, thanks to the implementation and the technologies used.

References

1. M.H. Wang, C.S. Lee, K.L. Hsieh, C.Y. Hsu, G. Acampora, C.C. Chang, Ontology-based multi-agents for intelligent healthcare applications. *J. Ambient. Intell. Humaniz. Comput.* **1**, 111–131 (2010), <https://doi.org/10.1007/s12652-010-0011-5>
2. D. Riaño, F. Real, J.A. López-Vallverdú, F. Campana, S. Ercolani, P. Mecocci, R. Annicchiarico, C. Caltagirone, An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients. *J. Biomed. Inform.* **45**(3), 429–446 (2012)
3. T.R. Gruber, A translation approach to portable ontology specifications. *Knowl. Acquis.* **5**(2), 199–220 (1993), <https://doi.org/10.1006/knac.1993.1008>
4. D. Celik, Foodwiki: Ontology-driven mobile safe food consumption system. *Sci. World J.* **2015**, 475410 (2015), <https://doi.org/10.1155/2015/475410>
5. Pizza ontology, <https://protege.stanford.edu/ontologies/pizza/pizza.owl>
6. K. Ponichtera, M. Małkiński, J. Sawicki, M. Ganzha, M. Paprzycki, Fusing individual choices into a group decision with help of software agents and semantic technologies (2019)
7. G. Chen, The rise of the enterprise container platform. IDC White Paper (2018)

Offline Handwritten Gurumukhi Character Recognition System Using Deep Learning



Udit Jindal, Sheifali Gupta, Vishal Jain and Marcin Paprzycki

Abstract Currently, Gurumukhi—a religion-specific language originating from India, ranks as the 14th most spoken language and the 18th most popular writing script language of the entire world. However, while there exists a large body of literature related to recognition of handwritten texts in various languages, number of publications related to recognition of Indian handwritten scripts is considerably smaller. It concerns also the case of Gurumukhi. Hence, in the current contribution, we consider Gurumukhi handwritten character recognition, to fill the existing practical gap. The proposed approach is based on deep convolutional neural networks and has been applied to the 35 core Gurumukhi characters. Obtained results are promising, as accuracy of 98.32% has been achieved for the training dataset, and 74.66%, on the test data. These results are comparable to results reported in earlier research but have been obtained without any feature extraction or post-processing.

Keywords Character recognition · Deep learning · Convolutional neural network · Handwritten script · Gurumukhi

U. Jindal · S. Gupta

Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura,
Punjab, India

e-mail: uditjndl@gmail.com

S. Gupta

e-mail: sheifali.gupta@chitkara.edu.in

V. Jain

Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi, India

M. Paprzycki (✉)

Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

e-mail: marcin.paprzycki@ibspan.waw.pl

© Springer Nature Singapore Pte Ltd. 2020

L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_11

1 Introduction

Handwritten text recognition has a prominent role, among others, in business, health-care, or cultural heritage preservation. Observe that, while computer systems slowly replace handwritten documents in day to day practices of today, there exists a very large body of documents that have been handwritten and should be digitized. This has to be done, among others, to assure preservation of content as pen-and-paper documents that have limited lifespan (due to normal aging and degradation). Here, digitization has to involve not only creation of images (which is required primarily for historical manuscripts, which are work of art in their own right, and preserving them as images are required), but also extraction of their actual content/text (needed for further processing/searching/storage). For the latter, text images are usually converted into a machine-encoded form, using Optical Character Recognition (OCR). Recently, a lot of works have been devoted to recognition of printed, as well as handwritten, characters. Obviously, characters handwritten by different persons differ in both size and shape; since every individual has different handwriting, and thus different form of writing of individual characters. This, in turn, is the main source of difficulty for application of computers to handwritten character recognition [1, 2]. To address this problem, recently, it was proposed to use deep learning and convolution neural networks, which tend to be substantially more accurate than the approaches used in the past.

Abundant literature exists, concerning handwriting character recognition for non-Indian languages and scripts that represent them. However, only very few articles are available related to recognition of Indian scripts, such as the Gurumukhi. The Gurumukhi is utilized primarily within the Punjabi dialect. It has been established to be the world's 14th spoken language (as far as number of people who use it is concerned). It is also the 18th most popular writing script language of the world [3]. Note that speakers communicating in the Punjabi language (Gurumukhi), are not limited only to the states of Northern India, like Haryana and Punjab, and to the Delhi area. This language has also spread around the world. Because of its popularity, there exist also a very large number of written texts in this language (content of which is represented using the Gurumukhi script). Here, the body of written language includes, among others, sacred texts, books, verses, etc. Therefore, it is important to create an OCR system for such a rich, and generally utilized, dialect, which may become useful in different regions [4]. Moreover, note that there is still large number of persons who prefer to take records manually (in a written form). Here, note, obviously a lot of drawbacks exist in traditional approaches to handwritten documents management. These are, among others, storage space, preservation against decay, impossibility of making backups, etc. Moreover, the searching operation is extremely difficult (and very time-consuming) in the case of handwritten documents, existing on paper. Therefore, to eliminate these issues, recognition of handwritten characters is clearly needed for the Gurumukhi script.

Our goal is to propose an approach to recognize Gurumukhi handwritten text. Based on analysis of available state of the art approaches, we have decided to apply

Convolution Neural Networks (CNN), which seem to hold the highest promise (at the time of writing this contribution). Here, let us note that the earlier work, reported in [3–8], has already used machine learning-based approaches. However, in this work, feature extraction has been done manually and explicitly. Hence, the accuracy of the proposed models depended on the type of features extracted (and on the quality of feature extraction itself). To avoid such issues, we have decided to apply a “global approach”, where no feature extraction is applied at all, to see what level of accuracy can be achieved. Obviously, if successful, the proposed approach can be combined with feature extraction to further improve quality of results.

The remaining part of this contribution is organized as follows. Section 2 presents the main features of the Gurumukhi script. Next (in Sect. 3), we summarize that state of the art of handwritten Gurumukhi character recognition. Section 4 outlines the research methodology of the proposed CNN-based deep learning model. Initial results of our experiments, with the proposed architecture, are discussed in Sect. 5. A brief conclusion and future scope of possible research can be found in Sect. 6.

2 Gurumukhi Script

Gurumukhi script is used for writing in the Punjabi language. The word “Gurumukhi” means “from the mouth of the Guru”. Gurumukhi script is eighteenth most popular writing script language of the world [3]. The Gurumukhi script has 3 vowel bearers, 32 consonants, 6 additional consonants, 9 vowel modifiers, 3 auxiliary signs and 3 half characters [6], as shown in Fig. 1. The Gurumukhi script is written from top to bottom and left to right, and it is not case sensitive. In the Gurumukhi script, the characters have an upper horizontal line called headline. In this script, some characters are quite similar to each other, which makes their recognition a relatively difficult task. For example, in the consonants set, two characters shown by red arrow are differentiated only with header lines. In the same way, characters shown by blue arrow are also very similar to each other (the difference being the “closing of the loop”). In this work, we have focused our attention on recognition of 35 selected characters, i.e., 32 consonants and 3 vowel bearers, because these are the core characters in Gurumukhi script.

3 State of the Art

Let us now briefly summarize main pertinent research results. Let us start from observation that various machine learning-based techniques have already been applied to recognize characters written in different languages. Here, let us first note contributions, in which researchers have extracted different types of features for the classification purpose. For instance, Sharma and Jain [3] have proposed the Gurumukhi character recognition system based on application of machine learning. In their work,

S-cell and C-cell have been used, in the neocognitron technique, to extract local features. Kumar et al. [6] proposed a model based on the diagonal and transitional feature extraction and used the K-nearest neighbors algorithm (KNN) classifier for character recognition. However, in this technique, it is hard to determine the nearest parameter value of K . Kumar et al. [7] have extracted various features, like diagonal, directional, and zoning features, and further used the KNN and the Bayesian classifiers, for distinguishing handwriting of different writers. Kumar and Gupta [8] discussed the feature extraction techniques, like diagonal features, linear binary pattern (LBP), and transition features and used KNN and Support Vector Machines (SVM)-based classifiers. Finally, Lehal and Singh [4] proposed a Gurumukhi script recognition system based on multi-classifiers. In all these articles, different combinations of types of features that were to be extracted were considered (with various levels of success). However, it is clear that success of the selected feature selection approach may depend on the specific features of the dataset itself. Hence, it is worthy to consider an approach that is independent of the specific feature extraction method and see how successful can it become.

In the last few years, deep leaning demonstrated remarkable performance in the domain of machine learning, and pattern recognition, in particular. Among existing deep learning approaches, convolution neural networks (CNN) are one of the most popular models [9]. The CNN approach has an edge over other image processing techniques, as the images are processed using a highly optimized structure, while

Fig. 1 Full Gurumukhi character set



the basic idea behind its development was to imitate the human brain processing capabilities. The capacity of CNN to effectively display the informative information can be improved by varying characteristics of hidden layers, as well as parameters used for training in each layer [10].

In the literature survey, only very small body of work has been found on attempts at applying CNNs to the Gurumukhi character recognition [5]. Here, Kaur and Rani [11] proposed a CNN-based approach. However, it was combined with manual(!) feature extraction, preceding feeding the CNN, which was then used for the actual classification of handwritten characters. This being the case, we have decided to apply the CNN directly to the preprocessed handwritten images of Gurumukhi characters. However, it has to be stressed immediately that, as will be seen, the applied preprocessing does not involve any form of feature extraction.

4 Proposed Approach

The proposed approach consists of three main steps. First, dataset to be used in the experiments are prepared. Second, it is preprocessed to be ready to be fed into the CNN. Finally, the specific CNN architecture is designed and applied to the prepared dataset. In Fig. 2, we summarize the main aspects of each one of these three stages and follow with a more detailed description of each one of them.

4.1 Dataset Preparation

A dataset has been created for all 35 fundamental Gurumukhi isolated characters by creating 3,500 images (one image containing a single “version” of one character). For preparing such database of 3500 images, 50 people from various age groups and education levels, and both genders wrote the 35 selected Gurumukhi characters on the A4 size paper. Obviously, in this way, the constructed dataset includes a wide variety of individual renditions of characters, in view of individual writing styles of persons who participated in character writing. Each “individual dataset” consists of 35 Gurumukhi characters, created by a single participant. However, each individual participant has written all 35 characters two times on separate paper sheets (resulting

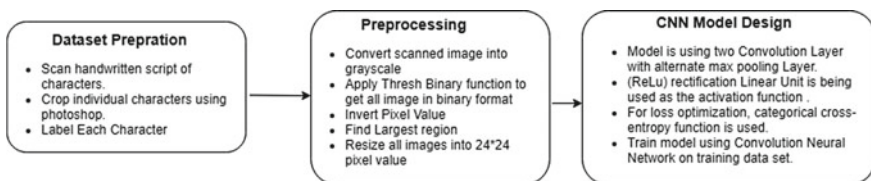


Fig. 2 Summary of preparatory stages of the proposed approach

in 70 renditions of characters; two of each). A sample dataset is shown in Fig. 3. Next, obtained sample datasets have been scanned. All individual characters have been cropped manually, from the scanned images, using Adobe Photoshop. In this way, 70 cropped images of characters, written by each individual participating in our study have been prepared. Example of cropped character images can be found in Fig. 4. Each cropped image, of an individual character, was saved in a separate file, with its corresponding (distinct) label name. Overall 100 sample datasets were collected from 50 participants. Thus, the final dataset consisted of $70 \times 50 = 3,500$ images of 35 characters. Next, the dataset has been partitioned into two sets, the

Fig. 3 Sample dataset from one participant

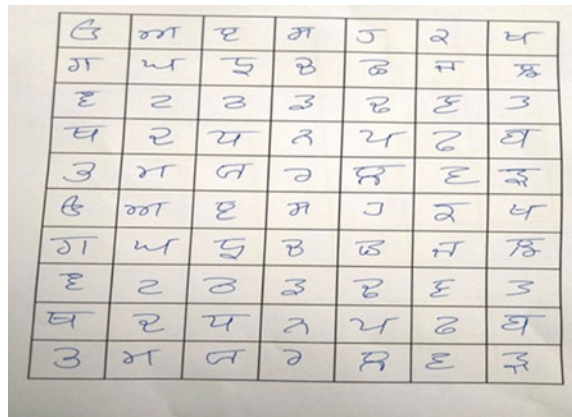
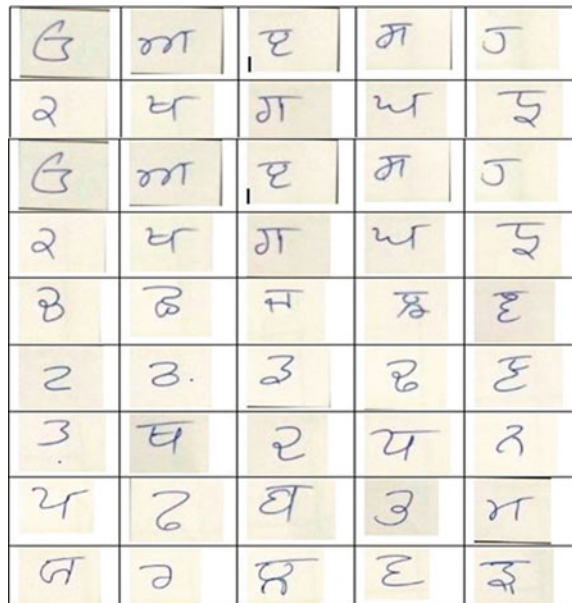


Fig. 4 Cropped character images



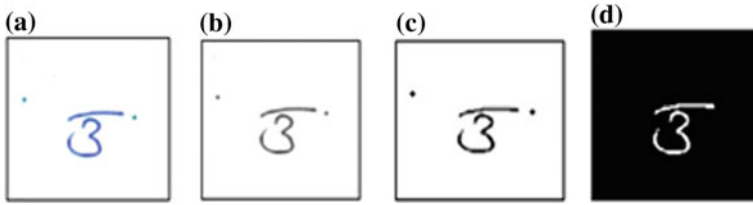


Fig. 5 Phases of the image during preprocessing **a** original image, **b** gray-scale image, **c** binary image, **d** inverted binary image with largest region (without noise)

first set containing 80% of images, to be used as a training set, and the remaining 20% to be used as the test set. This is a standard approach used in machine learning experiments.

4.2 Preprocessing

Preprocessing converts all obtained images into a common size, to feed them into the CNN. Here, the process is illustrated in Fig. 5. The scanned and cropped image, of single character, is shown in Fig. 5a. Such character image is, first, converted into the gray-scale image, as shown in Fig. 5b. In the gray-scale image, every pixel value represents the brightness level of that pixel and it from the 0–255 range. Next, a binary function, with a threshold value of 0.7, is applied to the gray-scale image, to convert it to a binary image, as shown in Fig. 5c. From that, the largest region is segmented out, to remove the noise, as shown in Fig. 5d. Before training the CNN, on the prepared images, they are resized to 24×24 pixels.

4.3 CNN Model Design

A CNN can be conceptualized using three basic layers: (a) convolution, (b) max-pooling, and (c) the final (or output) layer. In the convolution layer, features are derived from an image using a predefined weight filter. Depending upon the size of the weighting filter, feature maps are produced [12, 13]. The convolution layer, at the beginning of the model, merely identifies the generic features [12]. As the model's depth increases, the complexity of the extracted features also increases. The feature maps generated by the last convolution layer of the model are much closer to the problem at hand. Sometimes these feature maps are too large so, to decrease the number of trainable parameters, pooling layers are introduced. The most commonly used pooling layer form is the max-pooling. Max-pooling layer is used to minimize the number of parameters, in the case when images are too large. After convolution

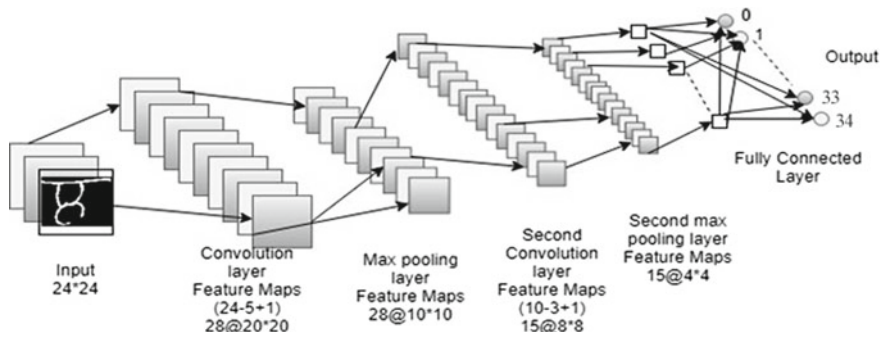


Fig. 6 Layer diagram of the proposed CNN model

and pooling layers, the output is formulated. Note that, the convolution and max-pooling layers are only able to derive features and decrease the number of parameters. Thus, the role of the output layer (also known as final layer), which is a fully connected layer, is to deliver output in the form of the output classes. The architecture of the proposed CNN model, used in our work, comprises of two convolution layers, with alternate max-pooling layers, and an output layer, as shown in Fig. 6.

In our work, the first convolution layer uses 28 weight filters of size (5×5) on (24×24) images, from which 28 feature maps, of size 20×20 , are obtained. From the first layer, the corresponding 28 feature maps are passed to the max-pooling layer with the filter size of (2×2) . The max-pooling layer produces 28 feature maps of size (10×10) , which are again passed through a second convolution layer, with 15 weight filters of size (3×3) . This creates 15 feature maps of the size of (8×8) . Resulting features are then passed through a max-pooling layer with the same filter as the last max-pooling layer. The final output from these layers is passed to a final layer, i.e., a fully connected layer, which uses the Softmax as the activation function. This fully connected layer transforms the feature maps to the 35 classes. As a result, the designed CNN model is classifying the images of the corresponding 35 Gurumukhi characters [14–16]. All layers in this model, except the pooling layers, use the Rectified Linear Unit (ReLU) as an activation function. In summary, the steps used to apply the proposed CNN architecture to solve the problem of Gurumukhi script recognition are as follows:

i. **Input:**

Scan the handwritten manuscript.

Label each character. [3,500 images written by 100 participants].

ii. **Preprocessing:**

- (a) Convert all character images into the gray-scale format and then invert all pixel value changing them into binary form.

*/**Rgb2gray converts: Change over RGB esteems to gray-scale esteems by framing a weighted sum of the R, G, and B value of pixel: **/*

$$Y = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (1)$$

(b) Resize all images into 24 * 24-pixel value matrices.

iii. **Splitting Dataset:**

Split all image dataset into 80% of training and 20% of testing data at random.

iv. **Use CNN model for character recognition using two convolutional layers with alternate max-pool layer.**

(a) Apply Rectified Linear Unit (ReLu) activation function on the convolutional layer.

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (2)$$

(b) Apply max-pooling layer with (2 * 2) pool-size (filter)

- Input a volume of an image $W1 \times H1 \times D1$
- Output:

$$W2 = (W1 - F)/S + 1 \quad (3)$$

$$H2 = (H1 - F)/S + 1 \quad (4)$$

$$D2 = D1 \quad (5)$$

Here, $W1$, $H1$, and $D1$ are the width, height, and channel of an input image. $W2$, $H2$, and $D2$ are the width, height, and channel of an output image. F is the size of Filter, S is the stride, i.e., the step size of the convolution operation.

(c) Apply Softmax Regression (often used in the final layer of the neural network) to handle multiple classes classifications.

$Y^{(i)} = \{1, \dots, K\}/K$ -> number of classes

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (6)$$

where $\sigma(z)$ = Previous layer weighted sum of output.

(d) In the end, use ‘‘Categorical Cross-entropy’’ function to determine loss optimization. Specifically, use the following function:

$$H(T, q) = - \sum_{i=1}^N \frac{1}{N} \log_2 q(x_i) \quad (7)$$

/**Where the size of the test set is N and the probability of event x evaluated from the training set is $q(x)$. This is also known as Monte Carlo estimation**/

v. **Output:**

This experiment, for the Gurumukhi character recognition, results in accuracy, for the training data, of 98.32%. Furthermore, the testing data accuracy is 74.66% (see, the next Section).

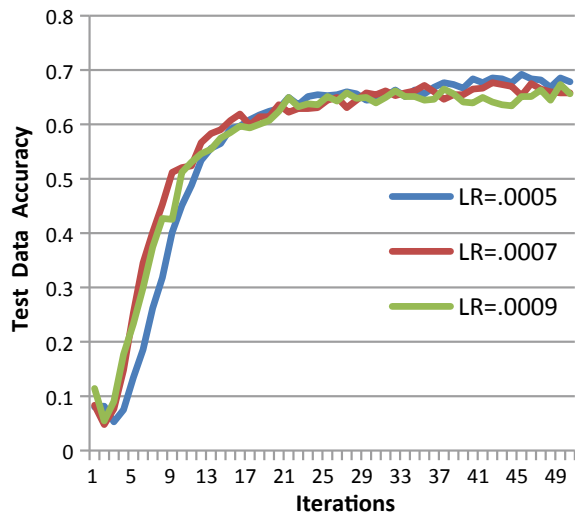
5 Experimental Results and Discussions

The proposed model has been experimented with using the dataset, which was prepared as described in Sect. 4.1. As noted, for training, 80% of the dataset was used whereas the remaining 20% was used for testing purposes. This was done in such way that each of the 35 core Gurumukhi characters has been split into 80–20% groups, to assure balance between “character classes” in the training and the testing datasets.

The CNN model was implemented in Python, using machine learning libraries Tensorflow and Keras. The two parameters that impact the efficiency of the CNN model are the learning rate and batch size. The learning rate (LR) of a model implies how rapidly the weights of the neurons will be adapted. Here, three different LR shave been tried: 0.0005, 0.0007, and 0.0009. The test set accuracy graph for different LR’s, at different iterations, is shown in Fig. 7.

From Fig. 7, it can be observed that, for more than 30 iterations, use of LR = 0.0005 results in slightly higher accuracy of character recognition (for the test dataset) than the remaining two LR’s. However, the results are relatively close to each other.

Fig. 7 Test data accuracy for each iteration



More precise data can be found in Table 1. Here, the training set accuracy and the test dataset accuracy for three different LR's are presented. It can be seen that the highest accuracy for LR = 0.0005 (after 50 iterations) has been achieved for both the training and test dataset accuracy. Hence, LR = 0.0005 has been used in subsequent experiments.

Note that the 3,500 resized images (24 * 24 pixels), used in the experiments, cause a memory overflow when considered all at once. Hence, a batch size becomes also a parameter for evaluating the performance of the proposed model. Test dataset error has been analyzed for three different batch sizes (BS) of 50, 60, and 70. Figure 8 demonstrates the impact of BS on testing dataset error for different iterations. From Fig. 8, it is clear that the test dataset accuracy is highest at BS = 60 for iterations 16–50. Hence, batch size of 60 is preferable when larger number of iterations can be applied.

Table 2 shows the training and test dataset accuracy for different batch sizes at LR = 0.0005.

Table 1 Training and test data accuracy at different LR's after 50 iterations

Learning rate	Training dataset accuracy	Test dataset accuracy
0.0005	0.9788	0.678571
0.0007	0.9778	0.658163
0.0009	0.9744	0.656463

Fig. 8 Effect of batch size on test dataset accuracy at LR = 0.0005

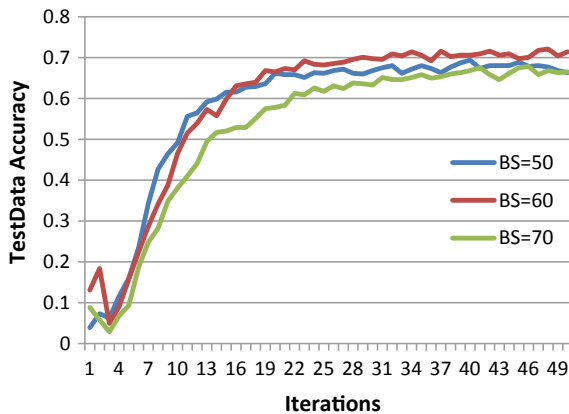


Table 2 Performance at different batch size at LR = 0.0005

Batch size	Test dataset accuracy	Training dataset accuracy
50	66.32	94.04
60	71.42	97.83
70	66.49	98.00

Table 3 Test data accuracy of CNN model at different LR and BS

Learning rate	Batch Size	Test set accuracy (%)
0.0005	50	74.66
	60	71.42
	70	66.49
0.0007	50	74.32
	60	69.39
	70	69.90
0.0009	50	74.51
	60	72.96
	70	69.39

Finally, in Table 3 we compare test data accuracy for all learning rates and batch sizes. This table is collectively showing the effect of learning rate and batch size on test dataset accuracy.

The simulations have been done for different combinations of learning rate and batch size with Gaussian filter and it is observed that the highest accuracy of 74.66% was achieved for learning rate of 0.0005 and batch size 50. It can also be observed, from this Table, that the system is always giving better accuracy for the batch size of 50 for all the three learning rates. Similarly said, system performs comparatively better for learning rate of 0.0009. So if batch size is kept small and learning rate is high, the best system performance is observed.

6 Conclusion and Future Scope

In comparison to feature extraction, Convolution Neural Networks (CNN) can directly draw visual patterns from the pixelated images. Hence, in this study, a CNN model was applied to Gurumukhi character classification. A handwritten dataset of 35 fundamental Gurumukhi characters comprising of 100 images for each character written by 50 different persons was prepared to represent different writing styles of people. It was established that, for the proposed CNN model, the best learning rate is 0.0005 and the batch size of 50. Accuracy obtained for the training data was 98.32%, while the testing data accuracy was 74.66%. In the future, we plan to (a) apply the proposed model to the complete set of all Gurumukhi characters, (b) develop automatic feature extraction approach(es) and combine them with the CNN, and (c) apply the proposed approach to the complete scripts/words in the Gurumukhi language.

References

1. R.D. Zarro, M.A. Anwer, Recognition-based online Kurdish character recognition using hidden Markov model and harmony search. *Eng. Sci. Technol. Int. J.* **20**(2), 783–794 (2017)
2. Z. Alom, P. Sidike, M. Hasan, T.M. Taha, V.K. Asari, Handwritten Bangla character recognition using the state-of-art deep convolutional neural networks. *Comput. Vis. Pattern Recognit.*, 1–12 (2017)
3. D. Sharma, U. Jain, Recognition of isolated handwritten characters of Gurumukhi script using neocognitron. *Int. J. Comput. Appl.* **10**(8), 975–8887 (2010)
4. G. Lehal, C. Singh, A Gurumukhi script recognition system. *Int. Conf. Pattern Recognit.* **2**(2), 557–560 (2000)
5. N. Kumar, S. Gupta, A novel handwritten Gurumukhi character recognition system based on deep neural networks. *Int. J. Pure Appl. Math.* **117**(21), 663–678 (2017)
6. M. Kumar, M.K. Jindal, R.K. Sharma, k-nearest neighbor based offline handwritten Gurumukhi character recognition, in *Proceedings of the 2011 International Conference on Image Information Processing (ICIIP)* (2011)
7. M. Kumar, M.K. Jindal, R.K. Sharma, Classification of characters and grading writers in offline handwritten Gurumukhi script, in *2011 International Conference on Image Information Processing*, Shimla (2011), pp. 1–4
8. N. Kumar, S. Gupta, Offline handwritten Gurumukhi character recognition: a review. *Int. J. Softw. Eng. Its Appl.* **10**(5), 77–86 (2016)
9. S. Acharya, A.K. Pant, P. K. Gyawali, Deep learning based large scale handwritten Devanagari character recognition, in *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, Kathmandu (2015), pp. 1–6. <https://doi.org/10.1109/skima.2015.7400041>
10. S. Roy, N. Das, M. Kundu, M. Nasipuri, Handwritten isolated Bangla compound character recognition: a new benchmark using a novel deep learning approach. *Pattern Recognit. Lett.* **90**, 15–21 (2017)
11. H. Kaur, S. Rani, Handwritten Gurumukhi character recognition using convolution neural network. *Int. J. Comput. Intell. Res.* **13.5** **90**, 933–943 (2017)
12. S. Naz, A.I. Umar, R. Ahmad, I. Siddiqi, S.B. Ahmed, M.I. Razzak, F. Shafait, Urdu Nastaliq recognition using convolutional–recursive deep learning. *Neurocomputing* **243**, 80–87 (2017)
13. B. Chakraborty, B. Shaw, J. Aich, U. Bhattacharya, S.K. Parui, Does deeper network lead to better accuracy: a case study on handwritten Devanagari characters, in *2018 13th IAPR International Workshop on Document Analysis Systems* (2018), pp. 411–416
14. M.M. Rahman, M.A.H. Akhand, S. Islam, P. Chandra Shill, M.M. Hafizur Rahman, Bangla handwritten character recognition using convolutional neural network. *Int. J. Image, Graph. Signal Process.* **7**(8), 42–49 (2015)
15. M. Mhiri, C. Desrosiers, M. Cheriet, Convolutional pyramid of bidirectional character sequences for the recognition of handwritten words. *Pattern Recognit. Lett.* **111**, 87–93 (2018)
16. M.Z. Alom, P. Sidike, T.M. Taha, V.K. Asari, *Handwritten Bangla Digit Recognition Using Deep Learning* (2017). [arXiv:1705.02680](https://arxiv.org/abs/1705.02680)

A Hybrid Self-constrained Genetic Algorithm (HSGA) for Digital Image Denoising Based on PSNR Improvement



Divya Verma , Virendra Prasad Vishwakarma and Sahil Dalal

Abstract The problem of denoising of images can be traced back to ancient times. Feature preservation remains an integral part of camera manufacturers around the world and drastic improvements have been achieved of late for the same. This work proposes a novel mathematical solution to the problem of image denoising. Images have been denoised using genetic algorithm evolutionary programming based on a self-constrained equational concept. A sample image is added with five different types of noise (Table 1) and they are denoised using existing filters (Table 2) and proposed algorithm (Table 3). The performance for different parametric functions has been compared using Peak Signal-to-Noise Ratio (PSNR) values in decibels. Consistent improvement is noted for five different noise models and compared with existing filters and the results are tabulated and graphically depicted (Figs. 4, 5, 6 and 7).

Keywords Image processing · Denoising · Genetic algorithm · Optimization · PSNR · Computer vision · Image restoration

1 Introduction

Optimization remains a key concern in almost any field of business and science. Achieving a trade-off amongst cost, quality and effectiveness has been nature's law, as exemplified by Charles Darwin's theory of natural selection of species in his work 'On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life' (1859). Nature favours the fittest species, individuals and races in that order. A digital form of this classical theory is the Genetic Algorithm (GA). The GA is a form of evolutionary algorithm that works on initial candidate solutions to generate the next pool and so on until the specified termination criteria are reached. It finds applications in various decision engineering problems such as scheduling, operations research and other problems of the optimization class.

D. Verma (✉) · V. P. Vishwakarma · S. Dalal
USICT, Guru Gobind Singh Indraprastha University, New Delhi, India
e-mail: dverma0732@gmail.com

© Springer Nature Singapore Pte Ltd. 2020
L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_12

One such problem is the denoising of corrupt images. Transmission of data over serial lines and packets adds noise to the useful data, which leads to corruption of pixels. Considering an image as a pure mathematical matrix, it is possible to manipulate images with addition of various types of noise and attempt to denoise them using our own algorithm. A correct and true evaluation of the robustness of the algorithm would be to conduct a comparison of the same versus existing algorithms.

2 Literature Review

The theory and literature reviewed are Evolutionary Algorithms (EAs) for the optimization task, which is reviewed in Section A and the image restoration problem, which is reviewed in Section B.

2.1 *Evolutionary Algorithms for Optimization Tasks*

The literature reviewed led us to the recently developed population search techniques, out of which differential evolution algorithm, as ideated by Storn and Price [1–3], is relatively more competitive and also reliable. A Pareto front can be defined as the edge of the criterion space, on which any point is called the Pareto optimal. By moving along this curve, either of the objective functions can be minimized but not both at once.

EAs have gained importance due to many reasons, some of which are listed below:

- They are inherently simple to apply in practice as no complex mathematics is involved such as gradient and vector information.
- The population can be made to converge on optimal solutions.
- They're applicable to several problems which can be depicted as optimization problems, such as the Travelling Salesman Problem (TSP) and scheduling on the factory floor.
- The robustness to dynamic changes is a welcome change in the event of changing conditions of run.

The parameters for control and strategies involved in DE are highly dependent on the constraints of the problem under consideration, as observed by Liu et al. [4]. The population size NP, however, remains user-defined. Performance of the DE algorithm may be improved through parallelization which can be based on the objective level, population level or the elemental level. The elements level parallelization is generally used for higher level parallelization [5].

Consumption time and accuracy play an important role in the control parameter settings. DE has three parameters: the vector amplification factor—F, control parameter for crossover—CR and population size—NP.

The description of the control parameters—F and CR—are as given below:

$$\begin{aligned}
 F_i, G+1 &= (F_i + r_1 \times F_u) \text{ if } r_2 < T_1 \\
 &F_i, G \quad \text{otherwise} \\
 CR_{i, G+1} &= r_3 \quad \text{if } r_4 < T_2 \\
 &CR_{i, G} \quad \text{otherwise}
 \end{aligned}$$

where T_1 and T_2 represent probabilities to represent factors F and CR , and r_j, j belongs to $[1, 2, 3, 4]$.

The dynamic allocation method of the penalty function is accountable to the range of infeasibility in the current population, and also the distribution of the infeasible population [6]. In other words, the measure of infeasibility is used to form a two-stage dynamic penalty function which is applied to the infeasible solutions, such that less infeasible solutions having a low-valued objective function are allowed to remain fit. It has been observed that this approach gives improved or comparable results to existing methods [7]. Both continuous and discontinuous search spaces can be traversed with this.

The main advantages of the approach are

1. Optimum or near optimum solutions can be found from a totally infeasible candidate population.
2. The robustness is proven for finding optimum solutions over a number of run of trials.
3. It may be applied to find the solution of non-linear equality and inequality constraints-based equations (Fig. 1).

de Paiva et al. [8] have used a novel Hybrid Genetic Algorithm (HGA) for improving a previous method introduced by Toledo et al. [9]. The process of initialization occurs at the start of the method as well as each time the population re-initializes [4]. Mutation is carried out to introduce diversity over the population over the already denoised images (first generation results).

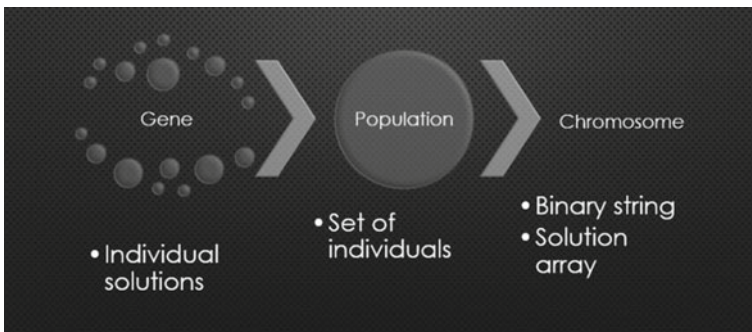


Fig. 1 Construction of a GA population

The stopping criteria may be described as any one of the following:

- (a) Maximum number of generations
- (b) Maximum time required to compute
- (c) Goal(s) achieved
- (d) Stalling generations (i.e. stuck on local minima for a finite number of seconds, or the results may get stuck for a number of generations, etc.) (Fig. 2).

Once one or more of these stopping criteria are met, the next generation is taken up for the iteration and so on.

The mathematical equation [8] of the process can be defined as

$$Fitness, I = \sum \sqrt{1 + \beta^2 |\nabla I|^2} + \frac{\lambda}{2} (I - I_0)^2 \quad (1)$$

where v = variance of observations or candidate populations, $\lambda = \frac{1}{\sqrt{v}}$, β = performance parameter (set as default after some iterations), ∇I = gradient of image I , I = original image and I_0 = noisy image.

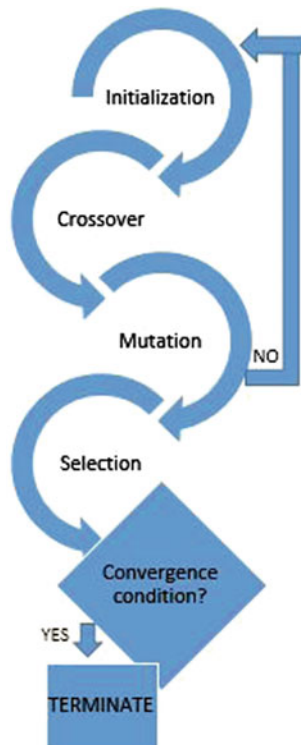


Fig. 2 GA working flowchart

2.2 Image Restoration Problem

Image processing can be divided into directions of solutions to be taken up into stochastic and probability methods, Markov random field theory and partial differential equation approaches [10]. The PDE approach looks at the problem of image denoising as a physical problem with well-established theory. A Banach space is defined as a linear space in which any Cauchy sequence is convergent. A Cauchy sequence is one in which the elements progressively become closer to each other as they progress but never quite equal, i.e. an asymptotic relationship [8]. For analyzing the inverse problem of image reconstruction, the geometrical Beltrami framework represents a balance between feature preservation and is invariant to parametrization [6].

One of the three broad classes of boundary conditions for partial differential equations involves the normal derivative of the independent variable to be specified on the boundary of the equation. For the Euler–Lagrange equation

$$(R \times Ru) - (R \times u_0) - (\lambda \cdot \Delta u) = 0 \tag{2}$$

In the above equation, the Neumann boundary condition is defined as $\partial u / \partial N = 0$ on $\partial \Omega$ where N is the projected normal to the surface of $\partial \Omega$ in the outward direction. The boundary is smooth such that the normal derivative can exist [8].

This finds application in heat flux equations of thermodynamics [11].

Using differential geometry, two processes essentially make up the Beltrami flow [6]:

- (a) Using mathematical Riemannian manifold for embedding the image
- (b) Metric tensor to provide regular measurements of the image.

Image restoration is essentially a task to reconstruct a degraded image, making it an inverse problem. The problem may be approached in two ways: integral calculus of definite sized pixels or voxels (as explained ahead); and that of matrices. The matrix approach is a build-up of the calculus approach in itself.

The fundamental components of image restoration are image signal and noise. Noise is defined as any unwanted interruption in useful information. In case of degraded images, it is random pixels that distort the appearance of the real image. If a is taken to be a particular pixel value, the Probability Density Function (PDF) $p(a)$ is the probability of having that particular value and is defined by

$$E\{a\} = \int_{-\infty}^{\infty} a p(a) da \tag{3}$$

If Eq. (3) is squared,

$$E\{a^2\} = \int_{-\infty}^{\infty} a^2 p(a) da \quad (4)$$

The variance of the noise may be defined as

$$\sigma^2 = E\{a^2\} - (E\{a\})^2 \quad (5)$$

Noise may be independent, i.e. additive in nature or dependent, i.e. multiplicative. Additive noise is the most common for electronic noise. Several types of noise are as follows:

- (a) *Gaussian noise*: This is the most commonly used noise model. Its usual sources are thermal noise and black-body radiation.
- (b) *Impulse (salt and pepper) noise*: Corrupted pixels are set to either zero or one, the classical binary format.
- (c) *Poisson noise*: In electromagnetic wave terms, this may be physically represented as ‘photon’ noise with a PDF of Poisson distribution. This noise is not independent of the signal nor additive.
- (d) *Speckle noise*: It is a multiplicative noise, with random values multiplied to pixel values [5].

Image filters are used to remove noise from images which are in-built in the application program. A few examples are:

- (a) *Wiener filtering*: It minimizes the Mean Square Error to the best estimate. It is a low-pass filter, i.e. it allows lower than threshold frequencies to pass while stopping the rest. It works for greyscale images

Wiener filter estimates the local mean and variance around each pixel and creates a piecewise filter using the estimates [4]. It is a 2D filter function inbuilt in MATLAB.

- (b) *Median filtering*: It is a non-linear mask that smooths out noise without blurring the edges [6]. It is more effective when it is required to simultaneously reduce noise and preserve edges [12]. In MATLAB, it is implemented as a 2D filter [4].
- (c) *Unsharp masking filter*: It is a technique to sharpen the image quality by means of subtracting an unsharp version of the image from itself [12]. In MATLAB, it is implemented as a 2D filter as a correlation kernel. A correlation kernel defines the redundancy or similarity between pixel frames [4].

3 Proposed Methodology and Mathematical Statement of Problem

The fitness function under consideration for our case would be a degraded image function $g(x, y)$ such that (Fig. 3)

$$g(x, y) = [h(x, y) \times f(x, y)] + \eta(x, y) \tag{6}$$

The problem may finally be depicted as a self-constrained mathematical optimization problem of the form [14–22]

$$Y = \sum_{i=1}^6 a_i d_i \tag{7}$$

such that

$$a_1 + a_2 + a_3 + a_4 + a_5 + a_6 = 1 \tag{8}$$

where $a_1, a_2 \dots a_6$ are the fractional constraining parameters, $d_1, d_2 \dots d_6$ are the noisy images using various image filters on the reference image (*ref*) and Y is the denoised image using the algorithm [23–25].

This paper proposes a hybrid genetic algorithm which would work on the basis of self-optimization using the genetic algorithm toolbox for the purpose of denoising the

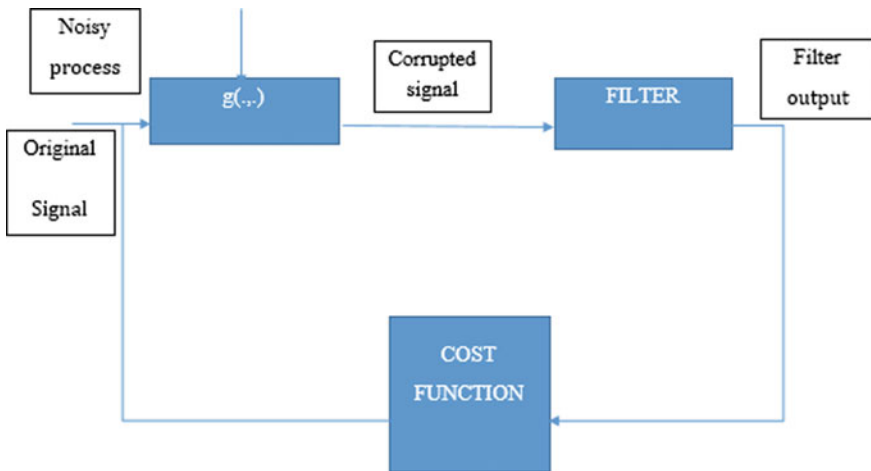


Fig. 3 Model optimal filter design for minimizing optimization function—a flowchart

image. The Peak Signal-to-Noise (PSNR) values for Y w.r.t. the reference images are calculated and compared in a tabular format with previously discussed noise removal methods. The genetic algorithm will act on the above objective function and find the minimum (- PSNR) value, hence the maximum PSNR value. Various noise models are selected and the results are tabulated and charted in the following section.

4 Simulation Results and Analysis


The images are read and fed into the program in MATLAB v R2013A on a 64-bit processor of Intel Core i5-3230 M CPU @ 2.60 GHz, 2601 MHz, with a physical RAM of 4.00 GB with Microsoft Windows 10 Home OS. The plots are analyzed on MS Office Excel 2013.

An original image is read into the system, and five noise models are added to it (Sect. 4.1). These noisy images are denoised using existing filters and then using our algorithm (Sect. 4.2). The MATLAB GA toolbox results are plotted in Section 4.3. The analysis presents a clear improvement in an a priori mannerism in PSNR values as tabulated. Section 4.4 present analysis of the obtained PSNR values (unit = dB) for various permutations and combinations of functions. The conclusion is obtained on the basis of charts prepared on Excel 2013, presented in Sect. 5.

4.1 Noisy Images





See Table 1.

Table 1 Noise added models to reference image

Serial No.	Noise and characteristics	Image
1.	Gaussian noise (mean = 0.001, variance = 0.002), N1	

(continued)

Table 1 (continued)

Serial No.	Noise and characteristics	Image
2.	Salt and pepper noise (noise density = 0.001), N2	
3.	Poisson noise, N3	
4.	Speckle noise (variance = 0.002), N4	
5.	Zero-mean, Gaussian noise, N5	

Noisy images for noise models N1–N5 in rows 1–5 of this table

4.2 *Denoised Images Using Existing Filters*

The noisy images are denoised using existing filters and the results are presented below. For each case, kernel size of the filter mask is a determiner of the smoothness of denoising obtained. It is observed that Wiener filter works optimally on all five noise models, followed by median filter and unsharp filter (Table 2).

4.3 *Denoised Image Using Proposed HSGA*

The image is denoised using the proposed algorithm. The results obtained are tabulated in Fig. 4 (Section D) which achieves the best PSNR values in comparison to filters in Section B.

The population type is taken to be a double vector of size = 20 with default values. The fitness scaling is taken as a Rank function. The selection function is chosen as stochastic uniform and the crossover function to be a heuristic function. Migration is chosen to be forward.

4.4 *Results*




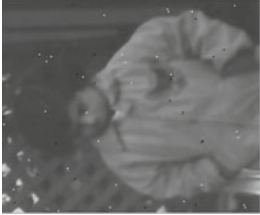


The PSNR values were calculated through Mean Square Error (MSE) values for each run and plotted. Further observations and charts are mentioned as below. The values are in decibels (dB).

It is observed that the PSNR values are highest for salt and pepper noise, N2. Plotting the above tabular results into a graph, it is observed the proposed algorithm achieves the highest PSNR values, i.e. the highest denoising for salt and pepper noise N2 and ZMG noise N5 (Fig. 4).

The genetic algorithm toolbox results are plotted in Fig. 5 where the peak PSNR value for Gaussian noise N1 is obtained to be 37.585 dB for the HSGA. The values for a_1 , a_2 , a_3 , a_4 , a_5 and a_6 for this is obtained to be 0.6655, 0.06132, 0.06267, 0.06210, 0.10558 and 0.04376 rounded off to the fifth decimal place. The result was obtained on the 51st iteration for a double precision population array of size $[20 \times 6]$.







Further inter-comparison among the GA parameter functions led us to some interesting observations. For constraint dependent mutation function taken to be constant in two instances, analyses were made for selection versus crossover functions (Table 4) and selection versus fitness scaling functions (Table 4). A column-wise analysis of Table 5 reveals that the different selection functions output similar valued results. The average PSNR values for different selection functions are given in Table 5. Increase in PSNR values denotes a greater denoising power (in decibels).

Table 2 Denoised images using existing filters

Serial No.	Noise model	Denoised image using Wiener filter of 4×4 matrix size	Denoised image using Median filter of 4×4 matrix size	Denoised image using Unsharp masking filter
1.	Gaussian, N1			
2.	Salt and Pepper, N2			

(continued)

Table 2 (continued)

Serial No.	Noise model	Denoised image using Wiener filter of 4×4 matrix size	Denoised image using Median filter of 4×4 matrix size	Denoised image using Unsharp masking filter
3.	Poisson, N3			
4.	Speckle, N4			

Denoised images in rows 1–5 of this table using the proposed HSGA algorithm for noise-added images

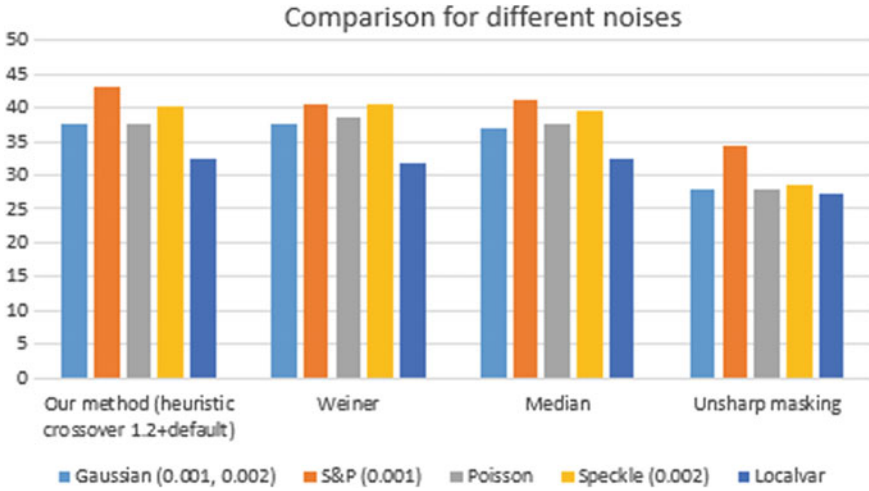






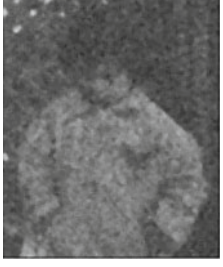
Fig. 4 Bar graph chart for PSNR value comparison (plot for Table 3)

Table 3 Denoised images using the proposed algorithm

Serial No.	Noise model	Denoised images using proposed algorithm
1.	Gaussian, N1	
2.	Salt and Pepper, N2	

(continued)

Table 3 (continued)

Serial No.	Noise model	Denoised images using proposed algorithm
3.	Poisson, N3	
4.	Speckle, N4	
5.	Zero-mean Gaussian, N5	

Denoised images in rows 1–5 of this table using the proposed algorithm for noise-added images

Plotting the results for Table 5, it is observed that the heuristic crossover function achieves the highest PSNR value in a line plot (Fig. 6) and a bar chart representation (Fig. 7). Heuristic crossover randomizes the genetic pool by making a cut across the parent chromosome vector strings and choosing the shorter distance node from the cut point. It substantially improves the performance of the algorithm which is itself metaheuristic in nature [26–28].

Plotting the results for Table 6, it is observed that stochastic uniform selection function and rank scaling fitness scaling function outperform the rest (Fig. 7).

After obtaining the above results, further analyses were made keeping the best performing functions namely, stochastic uniform selection, heuristic crossover and rank scaling fitness functions constant, and calculating results for population sizes of 20 and 50 versus population creation functions (Table 7). Constraint dependent

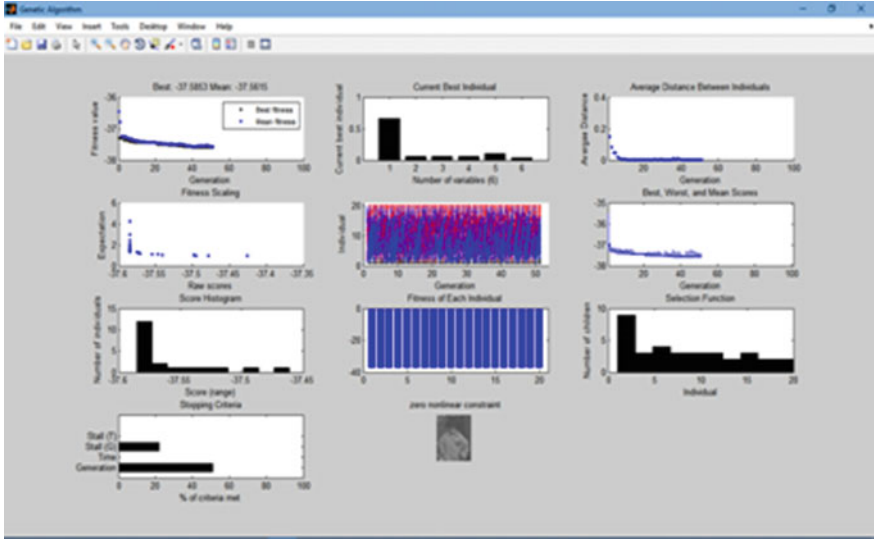


Fig. 5 Genetic algorithm toolbox results for Gaussian noise N1

Table 4 PSNR value comparison (dB) for noise models N1–N5 against proposed HSGA algorithm and filtering methods

Noise models	Using HSGA (heuristic crossover 1.2 + default parameters) PSNR (dB)	Wiener PSNR (dB)	Median filter PSNR (dB)	Unsharp masking filter PSNR (dB)
Gaussian, N1	37.58525857	37.701447	36.8878357	27.7544014
Salt and pepper, N2	42.9849714	40.6605295	41.2956311	34.3480405
Poisson, N3	37.56324662	38.416361	37.6214171	27.8604596
Speckle, N4	40.2370814	40.52071	39.5399246	28.5077744
ZMG, N5	32.30948812	31.7661634	32.3063935	27.3307968

Table 5 PSNR values for selection versus crossover functions

Selection versus crossover functions	Single point	Heuristic	Two point	Arithmetic
Stochastic uniform	37.25349	37.46107	37.24395	37.29040
Roulette	37.19697	37.24954	37.16203	37.23092
Tournament (size = 3)	37.16276	37.15298	37.13976	37.17965
Remainder	37.2569	37.38257	37.13886	37.18877

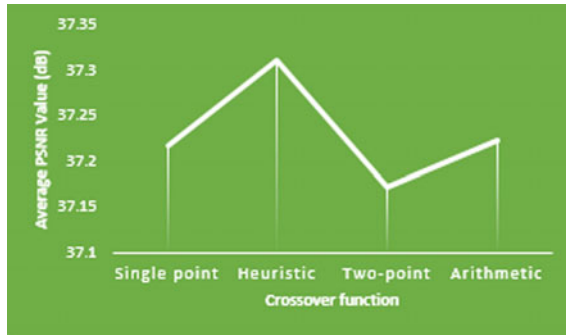


Fig. 6 Line plot for average values of PSNR (dB) for stochastic uniform, roulette, tournament and remainder selection functions against crossover functions (plot for Table 5)

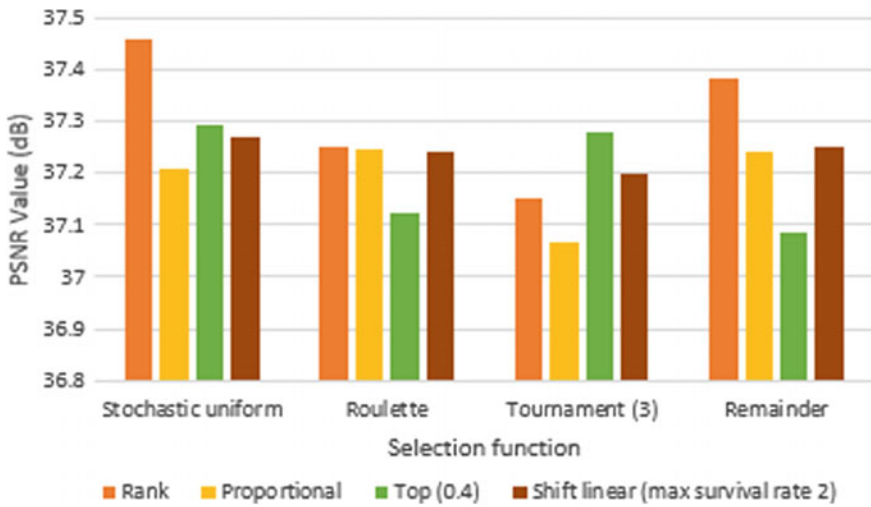


Fig. 7 Bar chart representation for selection versus fitness scaling function PSNR values (plot for Table 4)

Table 6 Average PSNR values for crossover function

Crossover	Average
Single point	37.21753
Heuristic	37.31154
Two point	37.17115
Arithmetic	37.22243

Table 7 PSNR values for selection versus fitness scaling functions

Selection versus fitness scaling	Rank	Proportional	Top	Shift linear
Stochastic uniform	37.46107	37.20706	37.29607	37.27232
Roulette	37.24954	37.24757	37.12228	37.23997
Tournament	37.15298	37.06420	37.27951	37.19751
Remainder	37.38257	37.24085	37.08337	37.25166

Table 8 PSNR Values for population sizes of 20 and 50 versus population creation function

Population size/creation function	Constraint dependent	Uniform	Feasible population
Population size 20	42.98497	42.15553	45.01786
Population size 50	44.05217	43.33957	54.90042
Population size 60	46.36317	44.57920	46.13222

mutation was analyzed to be the best performing and selected for this iteration as well. The noise selected was salt and pepper noise N2 to observe the effects on the classical binary noise. The time taken to calculate increases but an interesting trend can be observed from the table.

The average number of iterations at which the optimization converges is approximately 51, which tends to increase with the population size. The user fitness and constraint function evaluation are done in serial.

Table 8 reveals an almost constant trend for population sizes for constraint dependent and uniform population creation function; a break from the trend is clearly visible for adaptive feasible function and the highest value of denoising is obtained thus far at 54.9004 dB.

5 Conclusion and Future Scope

The importance of optimization problems in a world of diminishing resources and rising population is discussed. In the coming days for science and robotics, it shall be of prime importance for researchers to consider a vast number of constraints while coming up with a solution. Regardless of the diversity of factors involved, the criterion source being chosen should be such as to benefit the maximum and harm none of the users. Genetic algorithms with their multiple parameters can be applied successfully to almost any optimization problem for industrial, computer and production engineering [13]. To summarize

1. Our method achieves the highest PSNR for two runs for Salt and Pepper (S&P) and zero-mean Gaussian noise.
2. The overall PSNR values are highest for S&P noise.
3. The best performing selection function is stochastic uniform.

4. The best performing crossover function is heuristic (37.3 dB).
5. Adaptive feasible achieves a more constant result across different selection functions.
6. Average PSNR values (dB) are greater for constraint dependent by 0.1 dB.

The above analyses led us to the conclusion that stochastic uniform selection function, heuristic crossover function, constraint dependent mutation function and a population size of 20 for a constraint dependent population generating function works the best for our algorithm. These functions are also the default operating choices for the optimization toolbox for genetic algorithm in MATLAB 2013. The proposed HSG algorithm achieves the highest PSNR values for salt and pepper noise N2 and zero-mean Gaussian noise N5 and comes a close second to Wiener filter for Gaussian noise N1, Poisson noise N3 and Speckle noise N4. Hence, our work presents a robust self-constrained hybrid genetic algorithm-based image denoising.

The study undertaken was a cumulative work in the domains of image processing, genetic algorithms and optimization tasks. Optimization tasks were studied as were classic Pareto trade-off problem common across fields. The problem of image restoration was taken up for the application of a metaheuristic algorithm toolbox. A new self-constrained equation-based code was developed and the results analyzed on different toolboxes. The work developed would be carried forward to improve the robustness of the algorithm and to further improve the parameter through the usage of custom functions.

References

1. M. Ali, M. Pant, A. Abraham, A simplex differential evolution algorithm: development and applications. *Trans. Inst. Meas. Control.* **34**(6), 691–704 (2011). SAGE
2. K.V. Price, R.M. Storm, *Differential Evolution With 292 Figures, 48 Tables and CD-ROM A Practical Approach to Global Optimization* (Springer, 2005)
3. Gonzalez, Woods, *Digital Image Processing*, 3rd edn. (Prentice Hall, 2008)
4. Gonzalez, Woods, *Digital Image Processing using MATLAB*, 2nd edn. (Gatesmark Publishing, 2009)
5. G. Dougherty, *Digital Image Processing for Medical Applications* (Cambridge University Press, 2009)
6. D. Zosso, A. Bustin, A primal-dual projected gradient algorithm for efficient Beltrami regularization. UCLA CAM Report, 14-52 (2014)
7. A. Naït-Ali, Genetic algorithms for blind digital image stabilization under very low SNR. *IEEE Trans. Consum. Electron.* **53**(3) (2007)
8. J.L. de Paiva, C.F.M. Toledo, H. Pedrini, A hybrid genetic algorithm for image denoising, in *IEEE Congress on Evolutionary Computation (CEC)*, IEEE (2015)
9. Toledo, L. de Oliveira, R.D. da Silva, H. Pedrini, Image denoising based on genetic algorithm, in *IEEE Congress on Evolutionary Computation (CEC)*, pp. 1294–1301 (2015)
10. G. Ercan, P. Whyte, Digital image processing. US Grant US6240217B1 (1997)
11. H.A. Bruck et al., Digital image correlation using Newton-Raphson method of partial differential correction. *Exp. Mech.* **29**(3), 261–267 (1989)
12. MATLAB and Optimization Toolbox 6.3 Release R2013a, The MathWorks, Inc., Natick, Massachusetts, United States

13. S.N. Sivanandam, S.N. Deepa, *Introduction to Genetic Algorithms* (Springer, Berlin, Heidelberg, New York, 2008)
14. A. Konaka, D.W. Coitb, A.E. Smith, Multi-objective optimization using genetic algorithms: a tutorial. *Reliab. Eng. Syst. Saf.* **91**(2006), 992–1007 (2006)
15. J.-S. Lee, Digital image enhancement and noise filtering by use of local statistics. *IEEE Trans. Pattern Anal. Mach. Intell., PAMI* **2**(2) (1980)
16. Z. Liao et al., An automatic filtering method based on an improved genetic algorithm—with application to rolling bearing fault signal extraction. *IEEE Sens. J.* **17**(19) (2017)
17. A.K. Qin, P.N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC* (2005)
18. U. Aiman, V.P. Vishwakarma, Face recognition using modified deep learning neural network, in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC* (2005)
19. R. Farmani, J.A. Wright, Self-adaptive fitness formulation for constrained optimization. *IEEE Trans. Evol. Comput.* **7**(5) (2003)
20. J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evol. Computation* **10**(6), 646–657 (2006)
21. Q. Fan, X. Yan, Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies. *IEEE Trans. Cybern.* **46**(1) (2016)
22. F. Grimaccia, M. Mussetta, R.E. Zich, Genetical swarm optimization: self-adaptive hybrid evolutionary algorithm for electromagnetics, 0018-926X, IEEE (2007)
23. K. Sakthidasan Sankaran, N. Velmurugan Nagappan, Noise free image restoration using hybrid filter with adaptive genetic algorithm. *Comput. Electr. Eng.* **54**, 382–392 (2016). Elsevier
24. G. Aubert, P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations* (Springer Science + Business Media LLC, 2006)
25. M. Gen, R. Cheng, L. Lin, *Network Models and Optimization: Multiobjective Genetic Algorithm Approach* (Springer London Limited, 2008)
26. O. Kramer, *Genetic Algorithm Essentials*, Studies in Computational Intelligence, vol. 679 (Springer Nature, 2017)
27. A. Tamchenko, Visual-PSNR measure of image quality, 1047-3203/Ó (Elsevier, 2014)
28. Sheikh, Sabir, Bovik, A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Trans. Image Process.* **15**(11) (2006)

Deep Convolutional Neural Network with Transfer Learning for Detecting Pneumonia on Chest X-Rays



Prateek Chhikara , Prabhjot Singh , Prakhar Gupta 
and Tarunpreet Bhatia

Abstract Increasing medical abnormalities has led to life insecurity. Detection and treatment of such abnormalities at an early stage can help save lives. Developing tools and algorithms to make human life better is the main purpose of technology. Detection of such abnormalities through X-ray images is a problem which requires a better solution. There is a need to develop reliable and interpretable deep learning models that can detect such abnormalities with high accuracy. Many models have been developed over time but still there is scope for improvement. In this study, an automatic diagnostic tool based on neural network framework is proposed for the diagnosis of pediatric pneumonia from chest X-ray images. For this task, a deep convolutional neural network model with transfer learning has been proposed. Before passing images to model, preprocessing is done using filtering, gamma correction, equalization, and compression methods. The proposed model is compared with ResNet, ImageNet, Xception, and Inception in terms of precision, recall, accuracy, and ROC accuracy score. Experimentation is done on standard X-ray dataset collected from the Women and Children's Medical Center. The experimental results clearly reveal that our proposed algorithm gives better results than other conventional models.

Keywords Pneumonia · Machine learning · Transfer learning · Image processing

P. Chhikara · P. Singh (✉) · P. Gupta · T. Bhatia
Thapar Institute of Engineering and Technology, Patiala, India
e-mail: psingh4_be16@thapar.edu

P. Chhikara
e-mail: pchhikara_be16@thapar.edu

P. Gupta
e-mail: pgupta7_be16@thapar.edu

T. Bhatia
e-mail: tarunpreet@thapar.edu

1 Introduction

Lungs are the primary organs of the respiratory system. The exchange of gases (intake of O_2 and breathing out of CO_2) takes place in the lungs and respiratory system. Gases entering through our nostrils pass through the trachea (windpipe) which filters the impurities from the air. The trachea splits into two branches dividing air into both left and right lung [1]. Lungs contain small pouches called Alveoli which are responsible for the exchange of gases. In alveoli exchange of gases takes place, these alveoli act as the functional units for the exchange of gases. In the alveoli, the inhaled oxygen is mixed with the hemoglobin present in the blood whereas CO_2 goes out of the blood.

Respiration is one of the most important life processes. Any abnormality can lead to loss of life. Detection of this abnormality is a prime concern to avoid any life threat. Various lung diseases can be caused by the abnormalities and improper functioning of the lungs. Lung disease can be divided into airway, lung tissue, and lung circulation types [2]. Airway disease is caused by the infection in the O_2 carrying tubes, making the tubes narrow. Inflammation of the tissues in the lungs causes lung tissue disease, which makes the expansion of the lungs difficult. Lung circulation disease causes inflammation and clotting of the blood vessels, damaging the lungs' ability to inhale O_2 and exhale CO_2 . The most common lung disease includes Bronchitis, Lung cancer, Pneumothorax, Asthma, Pneumonia, Pulmonary Embolus, and Pulmonary Edema. Pneumonia is the most contagious lung disease which causes inflammation to our alveoli, leading to children's death worldwide. Due to the filling of the alveoli with fluid or pus, difficulty in breathing takes place and causes a severe respiratory problem. Bacteria, viruses, and fungi are the main causes of Pneumonia. Pneumonia infects mainly the children less than 2 years old and people over 60 years old. People having a weak immune system and who often smoke, take drugs, and drink excessive amounts of alcohol are more likely to be affected by pneumonia. UK based NGO conducted a study in India [3] which stated that by 2030 India will be in the top five countries in the world having the highest number of deaths due to pneumonia. Pneumonia is one of the main causes for the high mortality rate amongst children under-five years of age. India is one-third of the total WHO southeast Asian countries having under-five mortality rate very high. Keeping in mind that pneumonia is a treatable disease, still, it has killed 1.7 million people in India and 10.1 million people across the globe. Countries like India, Pakistan, Nigeria to name a few are the countries having the highest number of deaths due to pneumonia. The study also stated that around 4 million people could have been saved if appropriate actions were taken.

There is a significant improvement in image acquisition devices over the last decade. These devices provide high-resolution images from the radiological machines (MRI scans, X-ray, CT-scans, etc.). Due to this reason, deep learning-based models have become the prime choice for classification of medical images. Medical image analysis is a widely popular research field for deep learning across the globe, due to the availability of data which is labeled and structured. It is significantly popular because it checks whether Artificial Intelligence (AI) based systems

will improve patient outcomes and survival, and it provides ground for human-AI interaction [4]. In this study, Digital Image Processing (DIP) is used on the X-ray dataset to extract important features from the image, so that unrecognizable features could be made eminent. With DIP minor errors in the image can be rectified and the whole gamut of the contrast in the image can be increased. This paper analyzes different deep learning approaches used in medical image processing. Deep learning in medical big data has a huge dominance because the significant hierarchical correlation inside the data can be found by using deep learning models rather than tedious and laborious handcrafting of features. with the large dissimilarity in patient's data, regular learning approaches are untrustworthy. Over the past few years, there is big progress in Deep learning approaches due to its capability to adapt with complex data [5].

1.1 Organization

This article is organized as follows. The previous research works are mentioned in Sect. 2. Section 3 focuses on the dataset and the base technique. The proposed approach of this paper is described in Sect. 4. In Sect. 5, the obtained results are discussed. Finally, Sect. 6 concludes the paper.

2 Previous Research Works

The parallelism between healthcare and technology has made the diagnosis of disease simple and effective. Technology has transformed healthcare. It's being used to diagnose lung cancer, pneumonia, and other diseases. Deep learning has unlocked new keys for image analysis, making the diagnosis of disease faster, and accurate. With the availability of large datasets, algorithms cover a wide range of applications which involves chest pathology [6], hemorrhage identification [7], arrhythmia detection [8], diagnosis of cardiovascular disease [9], and cancer detection [10] to name a few. Among them, in deep learning, the most important and widely used application in image analysis is X-ray. The bone age of the patient, the gender of the patient, etc., can be found out just by scanning the patient body with the X-ray.

Pneumonia endemic countries face many problems, like the unavailability of a rapid field-deployable diagnostic tool, which is cheap for the patient and offers accurate diagnosis results. Several methods have been proposed till now for identification of pneumonia. Kosasih et al. [11] used wavelet augmented analysis for the cough sound in Rapid Childhood Pneumonia Diagnosis. They aggregated the mathematical features that were calculated to develop a classifier, through which pneumonia can be separated from many other respiratory diseases. Sharma et al. [12] proposed an approach that uses Image Processing in detecting the pneumonia clouds in chest X-ray. Otsu thresholding was used in the technique which differentiated the healthy

part of lung from the part which is affected by pneumonia. Deepika et al. [13] used Gray Level occurrence matrix texture feature to classify pneumonia as mild, moderate, and severe. Abdullah et al. [14] used Cellular Neural Network to detect the symptoms of pneumonia in a short amount of time. It was based on the principle that the connection occurs between adjacent units only. Jiansheng et al. [15] used the fuzzy neural network based on dynamic Kohonen network for the diagnosis of pneumonia syndromes. Fisher-iris data was used to check the reliability of the model. They achieved 86% accuracy when compared with the original standard diagnostic data. Huang et al. [16] predicted pneumonia readmission by designing a Decision Support Model. Different value combinations of SVM parameters, C and γ were used, and they achieved the accuracy of 83.85% and 82.24%, respectively, for the two different models. The model was found out to be powerful in spotting pneumonia patients at high risk.

3 Materials and Methods

3.1 Dataset

Kermary et al. [17] produced the dataset by X-ray scanning of the patient under 5 years of age at Guangzhou Women and Children's Medical Center [18]. There are 5866 chest X-ray images of children present in the dataset. The resolutions of the images vary from 400p to 2000p. There are 4283 pneumonia cases and 1583 normal cases present in the dataset. The training is done on the dataset and after that, the performance is measured on different transfer learning models. The dataset categorization is shown in Table 1.

3.2 Transfer Learning

In real world machine learning applications, it's not necessary for upcoming future test data and present training data to have the same feature space with a similar distribution. Many times the classification task and training data are in different domain set (having different distribution). For example, a person has already trained a model on Cats and Dogs dataset and he wants to classify the emotions of a person.

Table 1 Dataset categorization

Type	Train	Validation	Test
Normal	1349	234	234
Pneumonia	3883	400	400
Total	5232	634	634

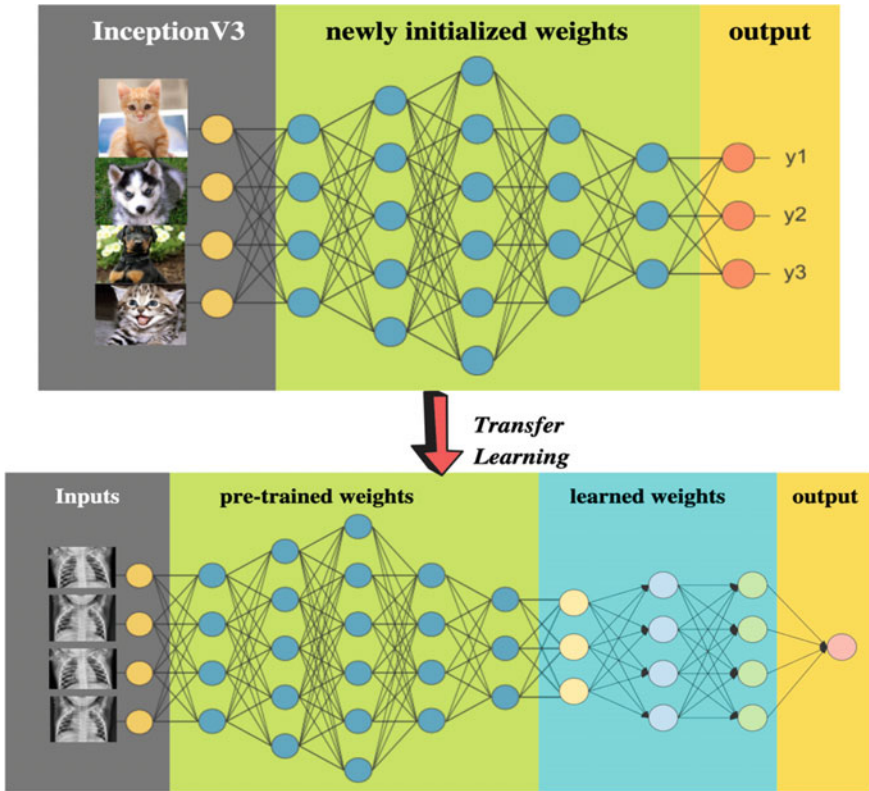


Fig. 1 Transfer learning intuition

In such situations, knowledge transfer is a good alternative to improve the learning performance by discarding data labeling efforts which are time-consuming. To solve this problem, transfer learning is used as a new learning framework [19]. It extracts the knowledge from multiple source tasks and the knowledge is applied to a different target task. It is different from multitasking learning, despite learning from source and target at the same time, transfer learning focuses on the target task only. Here, the symmetricity of the source and target tasks is not mandatory. The basic workflow of transfer learning is shown in Fig. 1.

4 Proposed Model

An automated diagnostic tool is proposed in this work. Initially, the dataset is pre-processed using DIP algorithms. Then the dataset is divided into train and validation set.

Finally, the dataset is ready to be passed through the model training stage. Further, the performance of the model is analyzed.

4.1 Phase 1-Preprocessing of Dataset

Median Filter. It is a pre-processing step used to remove noise from the image, by preserving edges. In this, a window (size smaller than the actual image) is to run through the image and replacing each entry with the median of the window. **Histogram Equalization.** It is performed to improve the contrast of the image by adjusting the image's histogram [20]. The transformation function is of the form:

$$Out_k = f(Inp_k) = (L - 1) \sum_{i=0}^k P_r(Inp_i) \quad (1)$$

where, ' L ' is the pixel range, ' Inp ' is original image, ' Out ' is output image, ' $P(Inp_k)$ ' is the estimate of the occurrence probability of intensity level ' Inp_k ' in a specified range, ' $f(Inp_k)$ ' is strictly monotonically increasing function in the domain $0 \leq Inp \leq L-1$.

Gamma Correction. Many devices captures, prints, display images, and respond according to a power law. It is important for displaying an image accurately, required for further analysis and decision making.

$$Out = Inp^{1/\gamma} \quad (2)$$

where, γ is a constant, ' Inp ' is the original image and ' Out ' is the output image.

Contrast Limited Adaptive Histogram Equalization (CLAHE). CLAHE is used to enhance the visibility level of pneumonia clouds in the X-ray image [21].

JPEG Compression. This algorithm is used to minimize the image file size of both color and gray image without compromising the details of the image. Discrete Cosine Transform is used for lossless compression [22].

The X-ray dataset images before preprocessing and after preprocessing on Normal cases are shown in Fig. 2, and for the Pneumonia cases are shown in Fig. 3.

4.2 Phase 2-Preparing Dataset

After preprocessing of images, the dataset is labeled. The images belonging to pneumonia class get label 1 while images belonging to normal class get label 0. Preparation of the training set was done with 5232 images, in which there were 1349 normal images and 3883 pneumonia images. The validation set is prepared with 634 images, out of which 234 normal images and 400 pneumonia images. Each image was resized to 299 pixel * 299 pixel keeping RGB channels intact.

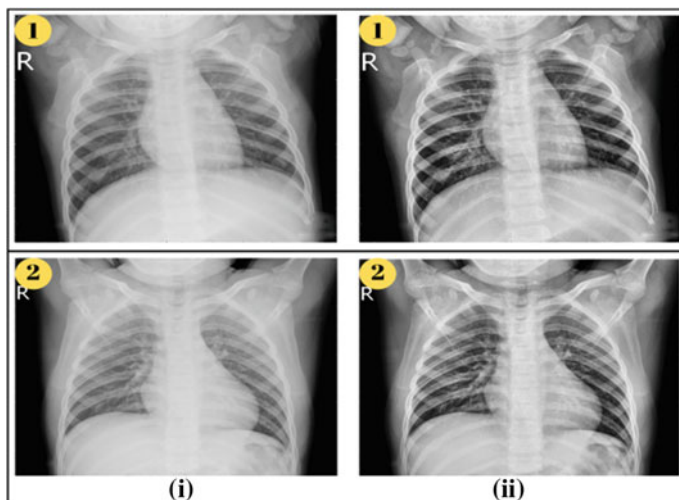


Fig. 2 Normal cases-original image and image after preprocessing

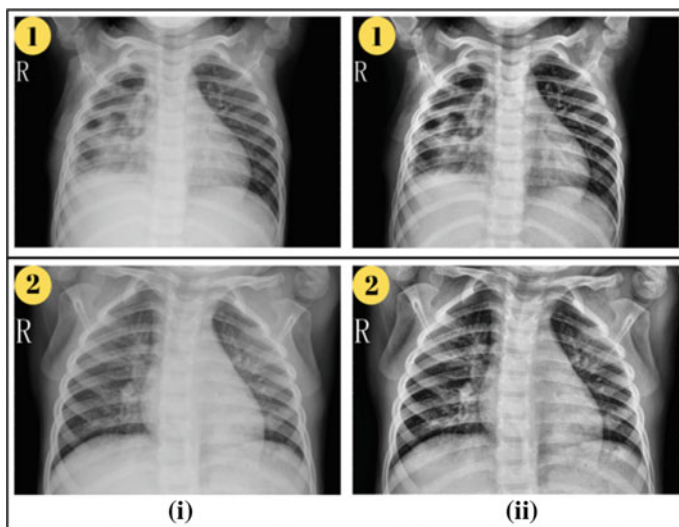


Fig. 3 Pneumonia cases-original image and image after preprocessing

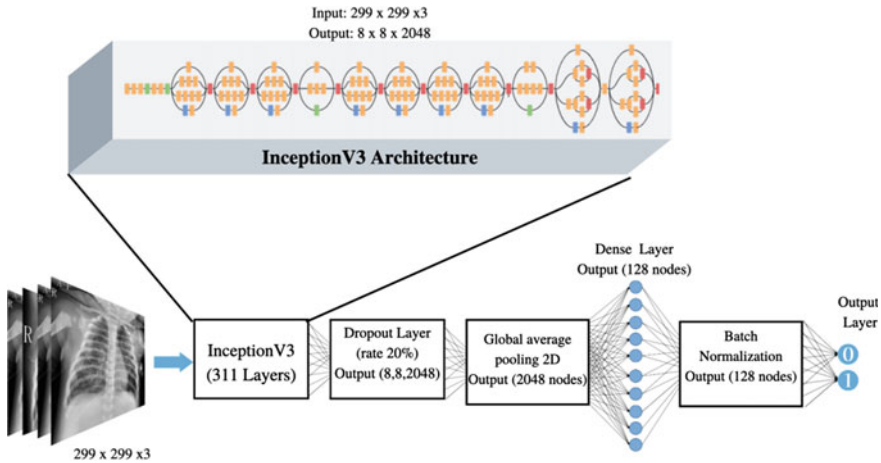


Fig. 4 Architecture of the proposed method with 316 layers

4.3 Phase 3-Model Training

In our study, transfer learning is used to enhance our model performance. The architecture of the proposed model is shown in Fig. 4. For this paper, standard InceptionV3 model is used with additional 5 layers.

Inceptionv3. Google developed this architecture which is an extension of InceptionV1. InceptionV3 contains repeating blocks known as Inception Block in deep networks in which parameter hyper-tuning is possible. Multiple convolutional filters which range from 1×1 to 7×7 are present that gives results after each iteration. It preserves the overall features of the image, thus there is no loss of localized information. In this, the number of parameters to be trained are reduced compared to other algorithms in existence [23]. There is flexibility to choose from either convolutional layers or appropriate pooling layers. By tuning the hyperparameters, a faster and more accurate architecture is achieved which works for smaller datasets.

The top layer of this model is discarded and at the end, few layers are added. The output layer of InceptionV3 model is the concatenation of four layers, in which two layers are activation layers directly and two mixed layers consisting of the concatenation of other layers. Then to this layer, a dropout layer with dropout rate of 20% is added to avoid overfitting and stopping hidden units from being dependent on particular units from the previous layer. After that, the Global average pooling 2D layer is added to minimize the overfitting in the model by minimizing the number of parameters. The volume of Dropout layer is (8, 8, 2048) in channel last representation, which was further reduced to (2048) using Global average pooling 2D layer. To this layer, dense layer with 128 nodes is added and used ReLU as the activation function. It reduces computational costs significantly and prevents gradient disappearance. Further, a layer of Batch normalization is added keeping default parameters for beta_initializer and gamma_initializer. Using batch normalization improves the

performance of the neural network and standardize input to layers. Further to this layer, a dense layer is added with two nodes. This layer will act as the output layer with two nodes, one for the pneumonia class and one for the normal class. The activation function used for this layer is Sigmoid Function.

To build our model, validation accuracy is considered as the measure to find the best model. The learning rate is reduced timely while the model is being trained by using callbacks. The model is built using Keras library in python. Categorical_crossentropy is used for loss function. The “Nadam” optimizer is used to minimize the cross-entropy loss function and used accuracy as metrics. The parameters of Nadam optimizer were kept default. Finally, the model with training dataset is built and the validation set is used for cross-validation. The final model contains 316 layers, 311 layers of which comes from InceptionV3 model. The layers distribution is shown in Table 2.

The final model contains 94 convolution 2d layers. All layers used linear activation function and dilation rate equals to 1. In each layer, there is a variation of the size of the kernel and number of filters. All activation layers use ReLU as the activation function except the output layer which used sigmoid as the activation function. If there is no significant increase in the accuracy (say no increment than 0.001) for the model on the validation set, or till 5 epochs until the model performance improvement is not significant, the learning rate is reduced by 20%. Total of 20 epochs is performed and saved the model which gives maximum accuracy on the validation set.

5 Experimental Results

The training dataset is used to train our model along with hyperparameters tuning to enhance the performance on the validation set, and then the overall performance is evaluated by using the test set. For evaluating the response of the model and calculating the validation accuracy of the overall classification precision, recall, and f1-score metrics is used as a benchmark which is based on the confusion matrix. In medical image classification, the four most frequently used CNN models are - ResNet, ImageNet, Xception, and InceptionV3 and are used to compare with the proposed model [24]. The proposed model is designed on the top of InceptionV3 model, for better learning of the effective characteristics of the X-ray image. In order to accelerate the training of the neural network and to overcome the problems of insufficient data Transfer Learning is used. The network consists of 311 layers of InceptionV3 model which contain 94 convolutional layers, 94 batch normalization layers, 94 ReLU activation layers, and a few pooling layers. The output layer of Inception model is followed by one of each dropout layer, global average pooling layer, batch normalization layer, and two dense layers. Training is accomplished by keeping the dilation rate of 1 and the cross-entropy loss function is minimized by using ‘Nadam’ optimizer.

After completion of the training phase, the results of the proposed model is compared with the four pre-existing network test results as shown in Table 3. Also, the

Table 2 Distribution of layers in the proposed method

	1	2	3	4	5	6	7	8	9	10	11
Layer name	Input	Conv2d	Batch normalization	Activation	Max pooling2D	Average pooling2D	Mixed	Concatenate	Dropout	Global average pooling 2D	Dense
No. of layers	1	94	95	94	4	9	13	2	1	1	2

Table 3 Comparison of classification metrics between proposed method with existing models

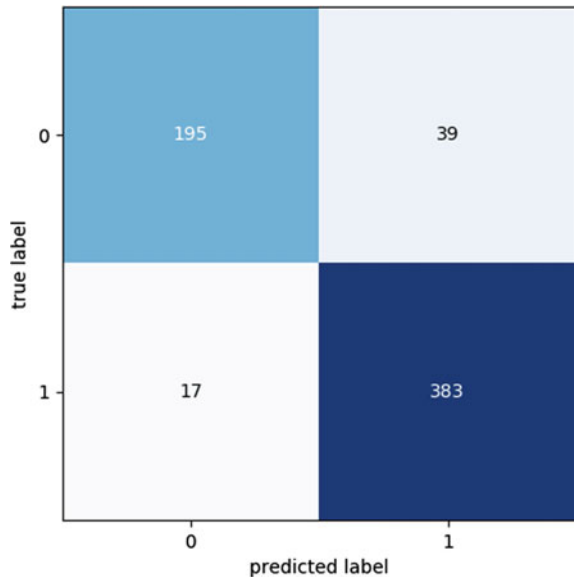
Model	Precision	Recall	F1-score	Validation accuracy
ResNet	0.742	0.961	0.837	78.3%
ImageNet	0.785	0.952	0.861	83.7%
Xception	0.857	0.967	0.908	87.8%
InceptionV3	0.916	0.841	0.877	85.3%
Proposed model	0.907	0.957	0.931	90.1%

confusion matrix of the proposed method is shown in Fig. 5. The test results shows that our proposed model has better precision (90.7%) than ResNet (74.2%), ImageNet (78.5%), and Xception (85.7%). In terms of recall, the proposed model has a recall rate of 95.75%, which is better than recall rate of ImageNet (95.2%), and InceptionV3 (91.6%). The proposed model outperforms other models with F1-score of 93.15%. The proposed model has validation accuracy of 90.16% whereas ResNet, ImageNet, Xception, and InceptionV3 has validation accuracy of 78.3%, 83.7%, 87.8%, and 85.3%, respectively.

In the end, the Receiver Operating Characteristic (ROC) curve is plotted to get complete access to the proposed model performance because the number of pneumonia and normal samples are imbalanced.

The curve gives a synopsis of the haggle between the true positive and false positive rate for a model by varying the values of different thresholds. The ROC curve is shown in Fig. 6 and the area under the curve is calculated and found out to be 0.911. The area under the curve signifies the adequacy of the test, area with the

Fig. 5 Confusion matrix of the proposed method



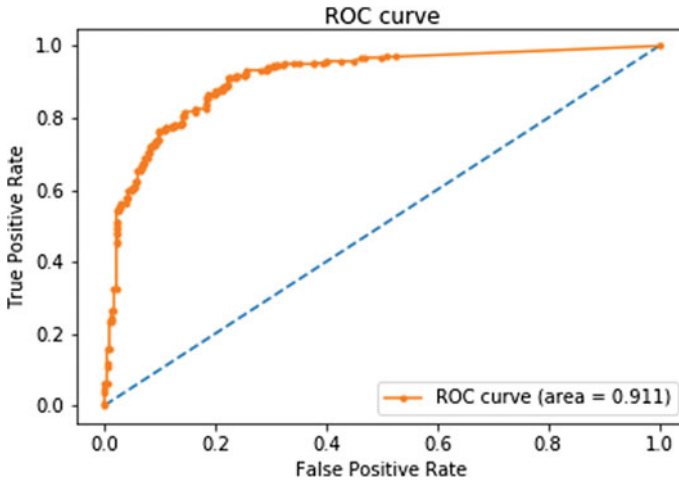


Fig. 6 ROC curve of the proposed method

largest magnitude means the test is better. The proposed model reduces the problem of overfitting by including dropout layer and global average pooling layer. On the mentioned X-ray database, the performance of our proposed method in detecting the pneumonia is better than the pre-existing algorithms.

6 Conclusion

An automated diagnostic tool is presented in this paper, which classifies chest X-ray images of the patient into normal and pneumonia. Various image preprocessing steps are applied like, median filter, histogram equalization, gamma correction, CLAHE, and JPEG compression on images to enhance some important image features so that our machine learning models can benefit from this improved data to work on. Transfer learning is used along with fine-tuning of hyperparameters to build our model. The model is built on the top of InceptionV3 model and added one dropout layer, one batch normalization layer, one global max-pooling layer, and two dense layers to it. Comparison of performance of four CNN models is performed for the diagnosis of pneumonia disease. Thus, the test results of the proposed model are compared with ResNet, ImageNet, Xception, and InceptionV3 model. Cross-validation accuracy of our model is 90.16% which outperforms ResNet, ImageNet, Xception, and InceptionV3 by 11.86%, 6.46%, 2.36%, and 4.86%, respectively.

References

1. B.M. George, S.B. Nayak, S. Marpalli, Morphological variations of the lungs: a study conducted on indian cadavers. *Anat. Cell Biol.* **47**(4), 253–258 (2014)
2. C. Beigelman-Aubry, C. Godet, E. Caumes, Lung infections: The radiologist's perspective. *Diagn. Interv. Imaging* **93**(6), 431–440 (2012)
3. F. Online, 17 lakh indian children to die due to pneumonia by 2030; here's how it can be averted. [Online] (2018). Available: <https://www.financialexpress.com/economy/17-lakh-indian-children-to-die-due-to-pneumonia-by-2030-heres-how-it-can-be-averted/1380390/>
4. J. Ker, L. Wang, J. Rao, T. Lim, Deep learning applications in medical image analysis. *IEEE Access* **6**, 9375–9389 (2017)
5. M.I. Razzak, S. Naz, A. Zaib, Deep learning for medical image processing: overview, challenges and the future, in *Classification in BioApps.* (Springer, Berlin, 2018), pp. 323–350
6. P. Lakhani, B. Sundaram, Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology* **284**(2), 574–582 (2017)
7. M.J. Van Grinsven, B. van Ginneken, C.B. Hoyng, ea Theelen, Fast convolutional neural network training using selective data sampling: application to hemorrhage detection in color fundus images. *IEEE Trans. Med. Imaging* **35**(5), 1273–1284 (2016)
8. P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, A. Y. Ng, Cardiologist-level arrhythmia detection with convolutional neural networks (2017). [arXiv:1707.01836](https://arxiv.org/abs/1707.01836)
9. B. Mann, J.S. Whiting, N.L. Eigler, Method for detecting, diagnosing, and treating cardiovascular disease. *Nov. 29 2005*, US Patent 6,970,742
10. X. Zhen, J. Chen, Z. Zhong, B. Hrycushko et al., Deep convolutional neural network with transfer learning for rectum toxicity prediction in cervical cancer radiotherapy: a feasibility study. *Phys. Med. Biol.* **62**(21), 8246 (2017)
11. K. Kosasih, U.R. Abeyratne, V. Swarnkar, R. Triasih, Wavelet augmented cough analysis for rapid childhood pneumonia diagnosis. *IEEE Trans. Biomed. Eng.* **62**(4), 1185–1194 (2014)
12. A. Sharma, D. Raju, S. Ranjan, Detection of pneumonia clouds in chest x-ray using image processing approach, in *2017 Nirma University International Conference on Engineering (NUiCONE).* (IEEE, 2017), pp. 1–4
13. N. Deepika, P. Vinupriya, D. Kathirvelu, Classification of lobar pneumonia by two different classifiers in lung ct images, in *2018 International Conference on Communication and Signal Processing (ICCSP).* (IEEE, 2018), pp. 0552–0556
14. A.A. Abdullah, N.M. Posdzi, Y. Nishio, Preliminary study of pneumonia symptoms detection method using cellular neural network, in *International Conference on Electrical, Control and Computer Engineering 2011 (InECCE).* (IEEE, 2011), pp. 497–500
15. L. Jiansheng, H. Jinliang, S. Jianjing et al., Research on diagnostic models of pneumonia syndromes in tcm based on fuzzy-neural net, in *2009 Chinese Control and Decision Conference.* (IEEE, 2009), pp. 933–937
16. J.S. Huang, Y.F. Chen, J.C. Hsu, Design of a clinical decision support model for predicting pneumonia readmission, in *2014 International Symposium on Computer, Consumer and Control.* (IEEE, 2014), pp. 1179–1182
17. D. Kermany, M. Goldbaum, Labeled optical coherence tomography (oct) and chest x-ray images for classification. *Mendeley Data* **2** (2018)
18. D.S. Kermany, M. Goldbaum, W. Cai, C.C. Valentim, H. Liang, S.L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan et al., Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell* **172**(5), 1122–1131 (2018)
19. S.J. Pan, Q. Yang, A survey on transfer learning. *IEEE Trans. Actions knowl. Data Eng.* **22**(10), 1345–1359 (2009)
20. S.M. Pizer, E.P. Amburn, J.D. Austin, R. Cromartie et al., Adaptive histogram equalization and its variations. *Comput. Vis. Graph. Image Process.* **39**(3), 355–368 (1987)

21. G. Yadav, S. Maheshwari, A. Agarwal, Contrast limited adaptive histogram equalization based enhancement for real time video system. in *2014 International Conference on Advances in Computing, Commu-Nications and Informatics (ICACCI)* (IEEE, 2014), pp. 2392–2397
22. G.K. Wallace, The jpeg still picture compression standard. *IEEE Trans. Consum. Electron.* **38**(1), xviii–xxxiv (1992)
23. C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in *Thirty-First AAAI Conference on Artificial Intelligence* (2017)
24. Saket, 7 best models for image classification using keras. [Online] (2018). Available: <http://www.it4nextgen.com/keras-image-classification-models/>

A Review on Video Analytics Its Challenges and Applications



Shaharyar Alam Ansari and Aasim Zafar

Abstract Nowadays the vast amount of video data is generated, which needs to be processed fast for getting correct insights into it. Manpower is not enough to process all of it in a short span of time due to their limited capacity. Video analytics facilitates useful tools to extract more valuable video data. This paper lays stress on different videos analysis activities. Here we explore various techniques which are used in video analysis along with presenting a comparison with image analysis techniques. Further, the application of video analytics along with the challenges being faced while analyzing the real time videos is discussed.

Keywords Video processing · Image processing · Object detection · Object tracking · Pattern recognition · Action recognition

1 Introduction

The demand for video analytics has increased over the years owing to growing security concerns. Video Analytics can mean several things. Mostly it is used to mean Video content Analysis (VCA) or just analytics whereby you study patterns of video consumption, performance or other parameters. Video analytics is used to detect and find temporal events and spatial events.

In video analytics, video processing is done which is a special case of signal processing. In signal processing, video filters are used in which video files or streams of video are input as well as output signals.

VCA is used in several applications including entertainment, health care, safety, and security, etc. One of the most well-known application of this would be Motion Detection, wherein you track the changes in the video using a fixed background scene. Video tracking is another important application of VCA. Both of which are typically used for surveillance. The other application areas may include robotics, medical applications, and multimedia. Video analytics application is just like any

S. A. Ansari (✉) · A. Zafar
Aligarh Muslim University, Aligarh, UP, India
e-mail: shaharyard@gmail.com

© Springer Nature Singapore Pte Ltd. 2020
L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_14

other analytics applications but based on Videos. Which is used to study the consumption patterns in Videos to identify patterns in the audience, to optimize the overall viewing experience and other purposes. It is mostly used in relation to online video analytics on sites like YouTube, Vimeo, and Facebook, etc. It answers questions like; How popular is your video(s)? What age group are the majority of your viewers in? How many times are they watching it? How long are they watching it? Are they clicking on the ads that appear on that video?, etc. Video Analytics will play the dominant role in intelligence extraction, which can be easily understood by Fig. 1

Video Analytics is one of the most important and essential element when we want to get insights about daily traffic because this helps the traffic monitoring system in controlling congestion. Video analytics applications can be deployed at the centralized location or at the edge devices or both.

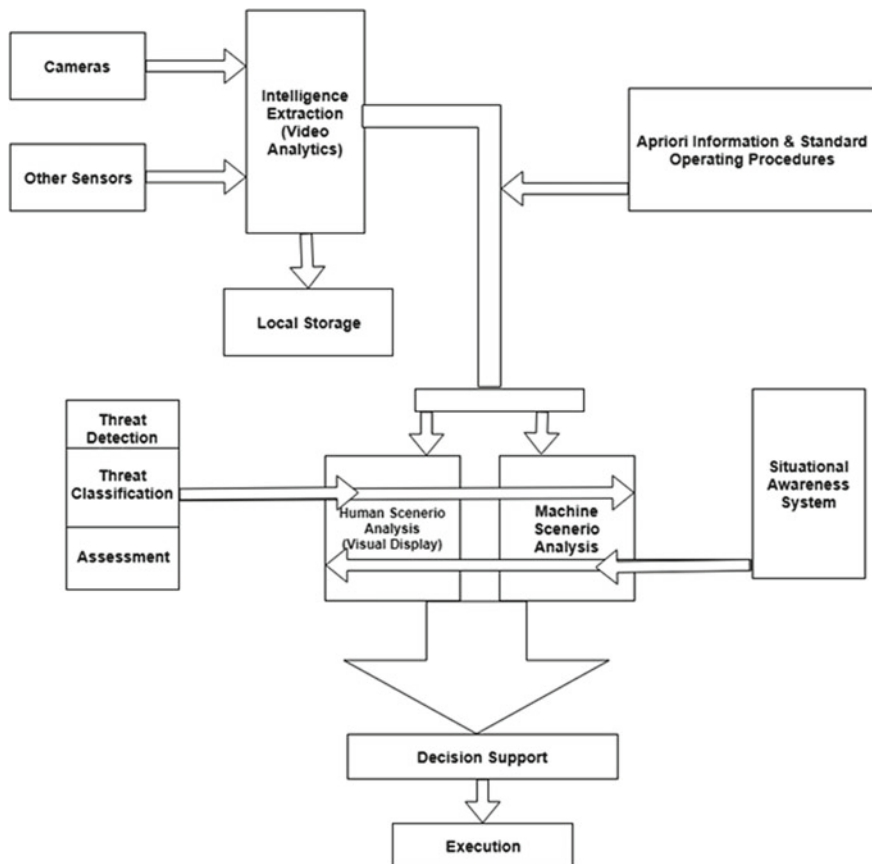


Fig. 1 An overview of the decision-making process

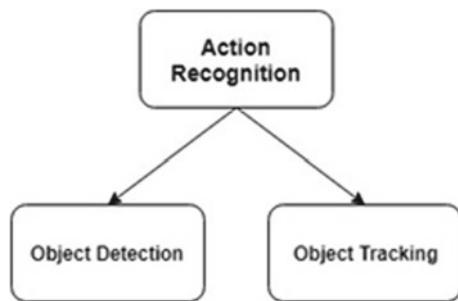
1.1 How Image Analysis Is Different from Video Analysis

Image processing is a part of signal processing where we are interested in finding useful objects. This goal is achieved by the process of segmentation. This segmentation can be described as a technique in which areas containing desired objects can be identified while ignoring the rest. In video processing, though the methods of image processing are extensively used on the other hand, it exploits the temporal nature of video which is being processed. In this case both the input and output are the video files.

At the simplest level, it could be running an object detector such as HoG (or a sliding window convnet) on each frame, and then some means of assigning nearby detections in adjacent frames to be the same object, and discarding detections which do not seem to have continuity in time. Many of the algorithms in this field seem to look at a single frame as a building block to looking at the whole sequence, where perhaps the data from adjacent frames is combined, aggregated, and/or used to disambiguate the current frame. Another approach is to first estimate object motion between frames (using optical flow, phase correlation, pyramid block matching or another method) and then treat areas of multiple frames that are collocated after accounting for motion as being the same object. This is very powerful but limited by the accuracy of the motion estimation.

In newer research, there is a back-and-forth between finding where things are (detection) and how things move (tracking), where each task can help the other one, for example [1] or [2], to the point that the two parts of the algorithm are no longer separable. Kalal’s TLD algorithm is a recently famous version of this. There are also algorithms which work directly in the spatio-temporal (or sometimes just temporal) domain. An example of purely temporal would be detecting vehicles by the periodicity of variation of their wheel spokes (Figs. 2 and 3).

Fig. 2 An overview of action recognition activities



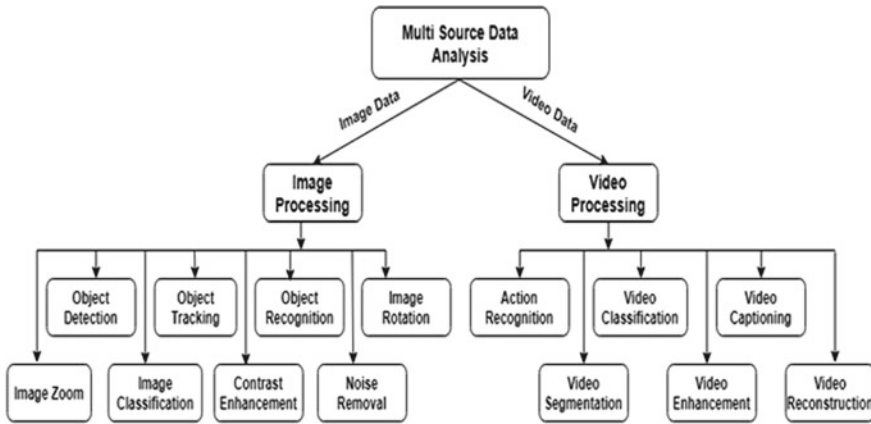


Fig. 3 An overview of activities in video analysis and image analysis

2 Applications of Video Analytics

In the current era where video has more demands over the image, video surveillance are adding new and better functionalities by exploring areas such as data mining, deep learning, and pattern recognition. Human tracking and detection aid society by helping old age and physically challenged people. Detecting, Identifying, and monitoring process can lead to behavior understanding and analysis. All these activities in Video analysis are providing a platform so that new applications of surveillance in indoor and outdoor which can provide security and crime prevention in highway, shopping malls, and parking. Further applications may be in the area of Automatically Detecting Vehicle, Recognition, and Tracking of License Plate in Real Time Videos, etc.

Key applications of video analytics may contain various features and functionalities such as: People management which further comprises crowd detection, people counting, people scattering, queue management, and people tracking. Video analytics proves to play a vital role in behavior monitoring which includes detecting motion of humans, their face detection together with masking their privacy and suspicious activity.

Vehicle management is essentially the important applications of video analytics which include traffic monitoring, gathering road data, classification of vehicles, and recognition of license plate. Device protection such as camera tampering, theft, threat detection, and intrusion detection are also one of the key application of video analytics.

3 Literature Survey

As the amount of video data generated today becomes huge, it becomes difficult to handle such data. So there is a need to properly execute that data for getting correct insight into it. The proper execution includes proper action recognition, video classification, video captioning, video segmentation, video enhancement, and video reconstruction. Since there exists a lot of research work concerning action recognition in video analysis, therefore, we have started from traditional value representation [3–8] together with shallow and deep encodings [9, 10]. Here our main focus is on end to end alternatives which have become powerful and gaining popularity these days [11].

In the following sections, we will discuss recent works in different areas of video analytics.

3.1 Action Recognition

Action recognition is the need of the hour in video analytics, the author has proposed a model [12] 3D histograms of texture (3DHoTs) in which they extract distinct focal points (features) from a series of depth maps and then combines these features with texture description for an efficient action representation of sequence of videos. Also to change the Adaboost optimization function he had added the inequality constraints which is derived from multiclass SVM. Further authors have used inequality constraint obtained from SVM so as to make the changes in AdaBoost optimizing function at decision level fusion. The basis of this function is Kernel-based extreme learning machine (KELM) which is the best classifier as described by the author.

Other method of action recognition is rank pooling [13] in which the author has used the latent structure of video sequence, which is a function-based on temporal polling. To know how features at frame level evolve over time in different videos latent structures are employed. Further, they have proposed a technique to extract the temporal dynamics of the video. This task is performed by ranking the features in chronological order which are obtained from videos. This overall aids in recognizing suitable actions.

3.2 Video Classification

In video classification author has proposed a model [14] in which he introduced two-stream architecture that aggregates audio and visual features, it also explores aggregation layers based on clustering technique. Further, he introduced context gating which is a nonlinear unit which aims to model interdependencies between network activations. In another method [15] of video classification, the author has proposed

an approach which is based on generalized maximum clique problem (GMCP). This function uses the co-occurrence of concepts that yields a richer representation of video compared to most of the current approaches. The matching is performed using GMCP that finds the strongest clique in a video. Additionally, he uses Mixed Binary Integer Programming (MBIP) to optimize the solution through GMCP.

3.3 Video Captioning

The authors have proposed an LSTM-TSA for sentence generation. This LSTM-TSA tries to traverse video representation together with traversing semantic attributes which are being employed in video captioning. Here authors basically focused on mining the attributes and fusing them all together so that enhanced sentence generation could be achieved. Just to give a clear verification of the proposed work authors have presented an MIL framework. This framework explores the semantic information being present in videos and learns about the impacts of attributes which are being used to contextually control the impacts of attributes in videos and images. To claim the validity of proposed work authors have carried out experiments on widely adopted datasets. Authors have claimed to have obtained improved performance when compared with other techniques. Their proposed LSTM-TSA achieves up to date and MSVD 52.8 and 74% in terms of BLEU@4 and CIDEr-D. Authors have achieved results on M-VAD and MPII-MD which are superior to the previous method.

3.4 Video Segmentation

In video segmentation the author has proposed a model OSVOS [16]; in this, they have focused on video segmentation using semi-supervised techniques. In this technique they have separated object from the background in a video. One-Shot Video Object Segmentation (OSVOS) is totally based on CNN architecture. This is used to transfer the semantic cues which are being learned on ImageNet so that the process of foreground segmentation could be achieved and finally the appearance of single annotated objects in video sequence could be marked.

3.5 Video Enhancement

Authors in the paper [17] have introduced a novel method for video enhancement. They proposed compressive sensing (CS) technique, which uses a non-sampled contourlet transform. In the proposed CS framework, authors assumed that signals

in the transform domain are sparse, the transform domain is assumed to be contourlet domain. This is used by the proposed algorithm, which then extracts the high-resolution frame. The quality of the frame is further increased by a post-processing step which applies to deblur of the frame using a bilateral filter and hence enhancing the consistencies. This method is computationally inexpensive while providing the high-quality output of captured video streams. The algorithm can even increase the quality of video streams in consumer applications where capturing device captures the video in low-resolution format.

In another paper the author has proposed real time video enhancing algorithm which uses the Graphical Processing Units (GPUs) for faster parallel computations. Also, the performance gains with GPU respect to Central Processing Units (CPUs) is considered by them. The author has used optimizations to have real time performance that is needed to enhance and play videos up to 4 k videos.

3.6 Video Reconstruction

An adaptive method for reconstructing the video is being proposed in [18]. This adaptive reconstruction method for coded aperture is used for parallel compressive imaging. The techniques mentioned in this paper makes use of speckle like feature together with pixel-wise equal exposure mask so that the moving areas could easily be separated from the clear stationery background (Table 1).

4 Issues and Future Work in Video Processing

Some of the frequently studied model problems are:

- People, vehicles or other objects detection and tracking throughout a video sequence. For example, using the CalTech Pedestrians dataset or other standard benchmarks.
- Hand gesture recognition, either for games, user interface, or sometimes sign language recognition. Often using depth data, for example, Kinect video.
- Activity recognition, for example, walking versus standing. Due to the nature of this problem, it is much more common to see pure spatio-temporal algorithms here, for example [19].
- Simultaneous location and mapping (SLAM). Often found in robotics, this problem is basically to build a 3D model of the environment from the video of a single camera (or stereo or depth video) which is moving around; the environment is often assumed to be static. This is often done by running some kind of feature detectors such as SIFT or SURF, matching up features in successive frames, and then building point clouds from the relative 3D motion implied by features.

Table 1 Comparison table for various schemes in different activities

Activities	Classification parameters					
	Authors	Approach	Model	Features extracted	Suitable for different size large scale dataset	Capability to deal with time and space factor
Action recognition	Fernando et al. (2017) [13]	Function-based temporal pooling method	Rank pooling	Ranking functions, parametric	Yes (better in performance than other)	Temporal with best result
	Wang et al. (2017) [20]	Feeding frame level CNN sequence features to LSTM model	(3-D) CNN with LSTM	Representative features extracted	Yes	Both spatial and temporal with best result
	Ji et al. (2014) [30]	Histogram of oriented gradient feature (HOG)	AdaBoost algorithm, nearest neighbor classifier algorithm	Raw featured data	Yes	Both spatial and temporal with best result
Video classification	Miech et al. (2017) [14]	Propose a two-stream architecture aggregating audio and visual features	LSTM, GRU, CNN	Audio and visual features extracted	Yes	Both spatial and temporal with best result
	Shayan Assari et al. (2014) [15]	Contextual approach for event classification	GMCP model and SVM model	Motion based feature	No	No one
	Andrej Karpathy et al. (2014) [31]	Context stream and fovea stream	CNN model, multilayer neural network	Color based and object based feature	Yes (better in performance than other)	Both spatial and temporal
Video captioning	Pan et al. (2017) [25]	CNN plus RNN framework	LSTM with transferred semantic attributes (LSTM-TSA)	Semantic attributes	Yes	Both
	Yang et al. (2017) [32]	Adversarial learning and long short-term memory (LSTM)	LSTM-GAN (generative adversarial network) system architecture	Semantic attributes and video features	Yes	Both
	Gao et al. (2017) [24]	attention-based LSTM model with semantic consistency	LSTM with attention mechanism	salient structures of video	Yes	Both spatial and temporal with best result

(continued)

Table 1 (continued)

Activities	Classification parameters					
	Authors	Approach	Model	Features extracted	Suitable for different size large scale dataset	Capability to deal with time and space factor
Video segmentation	Perazzi et al. (2017) [26]	Pixel labeling convnet for frame by frame segmentation	Convolutional neural network	Video object segmentation	Yes	None
	Caelles et al. (2017) [16]	One-shot video object segmentation (OSVOS)	Convolutional neural networks (CNNs)	Video object segmentation	Yes	None
	Tokmakov et al. (2017) [27]	Video object segmentation with visual memory	Neural network with an explicit memory module	Video object segmentation in unconstrained videos	Yes	Both
Video Enhancement	Ko et al. (2017) [17]	Artifact free low-light video restoration algorithm	Lowlight video degradation model with flickers	Similar patches among a given set of temporally adjacent frames	Yes	Both
	Deshmukh et al. (2018) [33]	Compressive sensing (CS) theory along with non-subsampled contourlet transform	Bilateral filter non-subsampled contourlet transform (NSCT) interpolation adaptive filter	Sparse expansion	Yes	Both
	Silva et al. (2018) [28]	Real time video enhancing algorithm	RGB (Red Green Blue) to HSI (Hue Saturation Intensity) conversion	Features extracted	Yes	Both
Video reconstruction	Zhao et al. (2017) [29]	Compressive sensing (CS) Approach	Multi hypothesis (MH) prediction Bregman iteration algorithm	Video data	Yes	Both
	Chen et al. (2017) [18]	Adaptive reconstruction method	Pixel-wise equal exposure coding	Features	Yes	Both

(continued)

Table 1 (continued)

Activities	Classification parameters					
	Authors	Approach	Model	Features extracted	Suitable for different size large scale dataset	Capability to deal with time and space factor
	Li et al. (2018) [34]	Intrinsic tensor sparsity model	Intrinsic tensor sparsity (ITS) Gaussian joint sparsity (GJS)	Patches	Yes	Both

As video processing includes action recognition, video classification, video captioning, video segmentation, video enhancement, and video reconstruction. So we will see the challenges and issues in each of these activities separately.

4.1 Issues in Action Recognition

Action recognition in videos are common nowadays and it has received increasing attention in the understanding of multimedia with CNN features. The author has proposed a model [20] in which an activity is conducted by a subject or multiple subjects to consider only salient features instead of mapping an entire frame into a static representation. The detection and recognition are not very clear in the current state of the art so there is a need for the optimization of algorithms by analyzing them in the frequency domain. Also, the speed and scalability are not satisfactory which can be improved by using memory processing cluster coupled with the computation power of GPU's [21].

4.2 Issues in Video Classification

Most of the current literature on Video classification focused on individual feature extraction which does not provide the accurate classification so there is a need to classify the videos on the basis of a collection of features which also saves the computation time.

Also to improve the video classification model we need to add attention layer to help our model to find focused regions [22]. The current state of the art in video classification has achieved 87% which is not satisfactory so it needs further improvement by integrating different algorithm or modifying the existing algorithm [23].

4.3 Issues in Video Captioning

The current literature [24] on video captioning does not focus on a particular domain so, in the future, we can modify these model to work on specific domain datasets, e.g., movies.

The performance of video captioning is still not up to the mark as expected so an attention mechanism needs to be further incorporated into the present mechanism, i.e., LSTM-TSA architecture for further boosting video captioning. Also there is a need to maximize semantic attributes for multiple sentence or paragraph generation for videos which need to be addressed further [25].

4.4 Issues in Video Segmentation

The literature [26] uses a pixel labeling ConvNet for frame segmentation. Use of labeling ConvNets for video object segmentation is a promising strategy. The future work should be exploring more sophisticated network architectures, incorporating temporal dimension, and global optimization strategies for efficient segmentation.

The ConvGRU [27] module encodes object which is evolving in spatial and temporal dimensions in a video and uses this encoding to improve motion segmentation this model lacks instance-based segmentation. So, Instance-level video object segmentation is a promising direction for future work.

4.5 Issues in Video Enhancement

The author has proposed an algorithm [17] for restoring videos which are captured in low light and do not possess artifacts. In his work for restoring videos, he first finds adjacent frames which have same patch and then adaptively accumulate those frames to increase their brightness. Further he reduces the color distortion by improving color assignment and by using guide map he performed fusion of image so as to reduce saturation. In this the author has presented restoration of lowlight algorithm that selectively accumulates similar patches for artifact-low-light video enhancement which neglect the noise removal process so the noise reduction step will be included in the lowlight restoration algorithm for detail preserving noise removal.

The literature [28] discussed on only single aspect of video enhancing so there is a need to fine-tune the implementation to have real time performance.

4.6 Issues in Video Reconstruction

The current literature on video reconstruction lacks in speed and performance due to the high complexity of algorithms [29]. In the current state of art it is found that the methods only separate the moving area and the area having high frequency, when the object motion is very small (maximum motion length during the exposure time is less than 3 pixels). Here the observer is interested in the areas with large motion but the proposed method treats the moving area as a high frequency one [18]. So such issues need to be addressed further.

5 Conclusion

In this paper we have done the thorough survey of most of the papers related to video analytics and found that by using video analytics with soft computing techniques, we can help in providing smarter and secured cities, implementation of laws to be more foolproof, intelligent transportation solutions, retail companies to act smartly, and be customer friendly. Though a lot of research has been done in this regard, more improvements need to be done for the proper functioning of the system. Motion smoothening, noise removal, proper restoration of image in low light, domain-specific captioning, etc. need to be done in the near future. Also the efficiency of the estimation and stabilization needs to be improved to reduce the computational cost.

References

1. Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(7), 1409 (2012)
2. M. Andriluka, S. Roth, B. Schiele, People-tracking-by-detection and people-detection-by-tracking, in *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008* (IEEE, 2008), pp. 1–8
3. M. Jain, H. Jegou, P. Bouthemy, Better exploiting motion for better action recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), pp. 2555–2562
4. H. Wang, C. Schmid, Action recognition with improved trajectories, in *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 3551–3558
5. B. Fernando, E. Gavves, J.M. Oramas, A. Ghodrati, T. Tuytelaars, Modeling video evolution for action recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 5378–5387
6. X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition: comprehensive study and good practice. *Comput. Vis. Image Underst.* **150**, 109–125 (2016)
7. X. Peng, C. Zou, Y. Qiao, Q. Peng. Action recognition with stacked fisher vectors, in *European Conference on Computer Vision* (Springer, Cham, 2014), pp. 581–595

8. Z. Lan, M. Lin, X. Li, A.G. Hauptmann, B. Raj, Beyond gaussian pyramid: multi-skip feature stacking for action recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 204–212
9. L. Wang, Y. Qiao, X. Tang, Action recognition with trajectory-pooled deep-convolutional descriptors, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 4305–4314
10. M. Jain, J.C. van Gemert, C.G. Snoek, What do 15,000 object categories tell us about classifying and localizing actions? in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 46–55
11. Z. Li, K. Gavriluk, E. Gavves, M. Jain, C.G. Snoek, VideoLSTM convolves, attends and flows for action recognition. *Comput. Vis. Image Underst.* **166**, 41–50 (2018)
12. B. Zhang, Y. Yang, C. Chen, L. Yang, J. Han, L. Shao, Action recognition using 3D histograms of texture and a multi-class boosting classifier. *IEEE Trans. Image Process.* **26**(10), 4648–4660 (2017)
13. B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, T. Tuytelaars, Rank pooling for action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(4), 773–787 (2017)
14. A. Miech, I. Laptev, J. Sivic, Learnable pooling with context gating for video classification (2017). arXiv preprint [arXiv:1706.06905](https://arxiv.org/abs/1706.06905)
15. S. Modiri Assari, A. Roshan Zamir, M. Shah, Video classification using semantic concept co-occurrences, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 2529–2536
16. S. Caelles, K.K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, L. Van Gool, One-shot video object segmentation, in *CVPR 2017 (IEEE, 2017)*
17. S. Ko, S. Yu, W. Kang, C. Park, S. Lee, J. Paik, Artifact-free low-light video enhancement using temporal similarity and guide map. *IEEE Trans. Industr. Electron.* **64**(8), 6392–6401 (2017)
18. Y. Chen, C. Tang, Z. Xu, Q. Li, M. Cen, H. Feng, Adaptive reconstruction for coded aperture temporal compressive imaging. *Appl. Opt.* **56**(17), 4940–4947 (2017)
19. S. Sadanand, J.J. Corso, Action bank: a high-level representation of activity in video, in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2012), pp. 1234–1241
20. X. Wang, L. Gao, J. Song, H. Shen, Beyond frame-level CNN: saliency-aware 3-D CNN with LSTM for video action recognition. *IEEE Signal Process. Lett.* **24**(4), 510–514 (2017)
21. M.U. Yaseen, A. Anjum, O. Rana, R. Hill, Cloud-based scalable object detection and classification in video streams. *Future Gener. Comput. Syst.* **80**, 286–298 (2018)
22. A. Burney, T.Q. Syed, Crowd video classification using convolutional neural networks, in *2016 International Conference on Frontiers of Information Technology (FIT)* (IEEE, 2016), pp. 247–251
23. N. Najva, K.E. Bijoy, SIFT and tensor based object detection and classification in videos using deep neural networks. *Proc. Comput. Sci.* **93**, 351–358 (2016)
24. L. Gao, Z. Guo, H. Zhang, X. Xu, H.T. Shen, Video captioning with attention-based LSTM and semantic consistency. *IEEE Trans. Multimedia* **19**(9), 2045–2055 (2017)
25. Y. Pan, T. Yao, H. Li, T. Mei, Video captioning with transferred semantic attributes, in *CVPR*, vol. 2 (2017), p. 3
26. F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, A. Sorkine-Hornung, Learning video object segmentation from static images, in *Computer Vision and Pattern Recognition*, vol. 2, no. 7 (2017)
27. P. Tokmakov, K. Alahari, C. Schmid, Learning video object segmentation with visual memory (2017). arXiv preprint [arXiv:1704.05737](https://arxiv.org/abs/1704.05737), p. 3
28. R. Silva, A. Dasanayaka, R. Ragel, Real-time video enhancement using graphical processing units, in *2018 Moratuwa Engineering Research Conference (MERCon)* (IEEE, 2018), pp. 549–554
29. C. Zhao, S. Ma, J. Zhang, R. Xiong, W. Gao, Video compressive sensing reconstruction via reweighted residual sparsity. *IEEE Trans. Circuits Syst. Video Technol.* **27**(6), 1182–1195 (2017)

30. X. Ji, L. Zhou, Y. Li, Human action recognition based on AdaBoost algorithm for feature extraction, in *2014 IEEE International Conference on Computer and Information Technology (CIT)* (IEEE, 2014), pp. 801–805
31. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1725–1732
32. Y. Yang, J. Zhou, J. Ai, Y. Bin, A. Hanjalic, H.T. Shen, Y. Ji, Video captioning by adversarial LSTM. *IEEE Trans. Image Process.* **27**(11), 5600–5611 (2018)
33. H.B. Deshmukh, V.D. Shinde, Video enhancement using non subsampled contourlet transform and bilateral filtering techniques. *Algorithms* **6**(10) (2018)
34. W. Li, F. Liu, L. Jiao, F. Hu, S. Yang, Video reconstruction based on intrinsic tensor sparsity model. *Signal Proc. Image Commun.* (2018)

Digital-Controlled Dual-Mode Switching Mode Power Supply for Low-Power Applications



K. Shubha Rao  and Veena S. Chakravarthi 

Abstract This paper discusses the modeling of a multi-mode digitally controlled switching mode buck converter to power a system-on-chip (SOC). It operates in digital pulse width modulation mode under heavy and medium load conditions, whereas in digital pulse frequency modulation mode it operates under light load conditions to obtain high-power conversion efficiency over a wide range of load current. In PWM mode of operation, $\Sigma-\Delta$ modulator-based digital pulse width modulator (DPWM) is implemented which provides high-resolution, low-power architecture to supply accurate, precise and low ripple content voltage to SOCs. A type III compensator is designed to improve the dynamic performance of the buck converter. Maximum efficiency of 88.48% is obtained at 0.2 A under PWM mode of operation. An output voltage accuracy of 0.25% is achieved against a target value of 1%. The load current and line voltage transient responses are verified by considering the different range of load current and line voltage variations. As an example, for a load variation from 150 to 400 mA, the undershoot is 30 mV, and when the load varies from 400 to 150 mA, the overshoot is 32 mV. The % peak overshoot/undershoot is around 3%. The response time of the power supply when load variations occur (150–400 mA and vice versa) is 3 μ s. The steady-state output voltage is 1.002 V, when the input battery voltage varied from 2.5 to 3.2 V and vice versa; in addition, the undershoot and overshoot in output voltage are 20 and 25 mV, respectively. The time taken by the power supply to settle to a steady value of 1.0025 is 10 μ s. Light load or power save mode efficiency is improved by using a DPFM mode of operation. The minimum efficiency obtained is 66.62% at 0.001 A. The % efficiency lies between 66.62 and 85% for a load current range of 0.001–0.1 A

Keywords Buck converter · Digital pulse width modulation · Digital pulse frequency modulation

K. S. Rao (✉) · V. S. Chakravarthi
B.N.M. Institute of Technology, Bengaluru, India
e-mail: shubharaok@bnmit.in

V. S. Chakravarthi
e-mail: scveena@gmail.com

© Springer Nature Singapore Pte Ltd. 2020
L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_15

1 Introduction

As technology evolves, more functionalities are being added to the SOCs. At the same time, the speed of operation of the system needs to be high with operating and stand-by power loss maintained low. Latest systems-on-chip and networks-on-chip are multi-core systems and they need efficient power supply regulators that provide a well-regulated constant DC voltage to increase their performance, reliability and lifespan. In addition, the regulated output voltage should contain minimum ripple. It is very challenging to design such highly efficient power supply regulators that deliver a low voltage with fewer ripples to a load varying over a wide range.

Changes in load current occur when SOC switches between various modes of operation. When SOC is operating in active mode, the load current demand is high whereas in sleep mode it is low. Since there are billions of transistors involved in a SOC operation, voltage regulators must have a fast transient response, to respond quickly to the changes in load current.

Efficiency is a good measure of the superiority of power supply circuitry. If the power efficiency is low, it would result in large power dissipation and hence enormous amount of heat gets generated. This large heat generation needs expensive cooling systems which increase the area occupied by the system. Hence a high-efficiency converter is essential so that elements can be packaged with high density, resulting in small size and weight, and low temperature rise.

Accordingly, this paper explains the modeling of a buck switching mode power supply that achieves regulated voltage with less ripple content, high frequency of operation and high efficiency. The switching mode power supply is controlled digitally and it operates in two modes of operation according to the load current. It operates in PWM mode during medium and heavy load condition and in PFM mode during light load/sleep mode of operation to achieve high power efficiency over the entire range of load condition.

The paper is organized as follows: Section 2 discusses the structural view of the multi-mode buck converter. Section 3 provides modeling of DPWM mode buck converter. Section 4 explains modeling of DPFM mode of operation. The results are analyzed in Sect. 5, and conclusions are drawn in Sect. 6.

2 Digital Multi-Mode Buck Switching Mode Power Supply

Figure 1 shows the digital power supply regulator using a buck converter. It operates in two modes of operation, namely PWM and PFM. PWM mode consists of digital type III compensator and second-order sigma–delta modulator, whereas PFM comprises hysteretic comparator. As closed-loop control is a digital controller, the analog voltage V_{out} is converted into a digital/discrete signal. This is done with the help of analog to digital converter (ADC). The compensator block generates a discrete signal which is proportional to the duty ratio. This discrete signal is given to DPWM

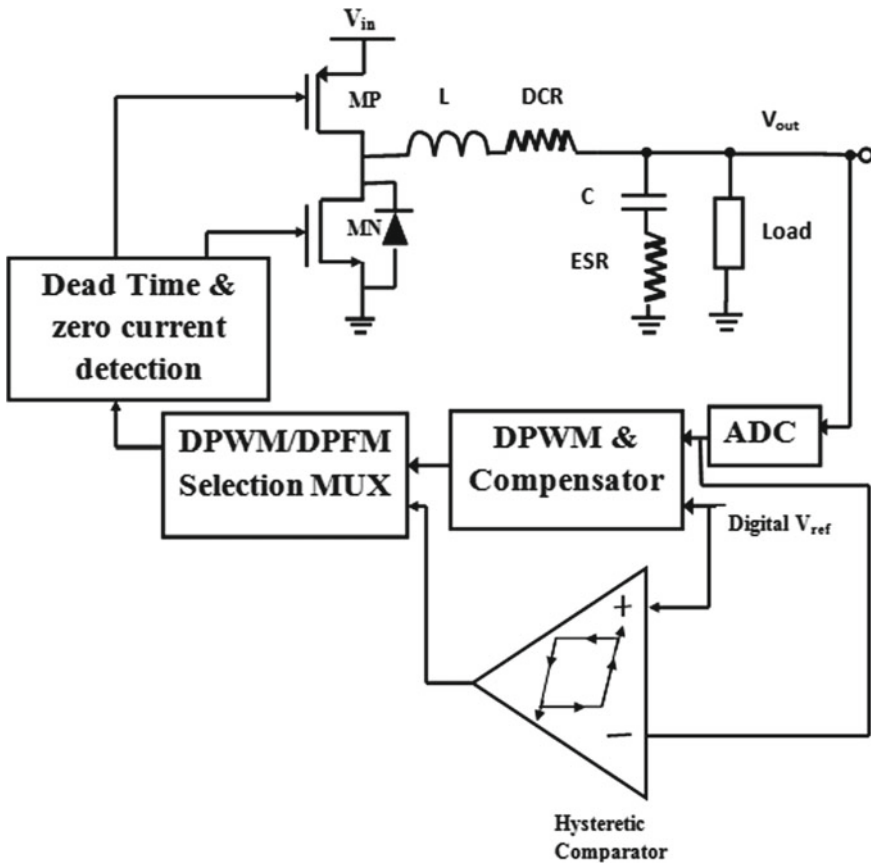


Fig. 1 Block diagram of the digital multi-mode buck power supply module

block wherein the discrete signal is converted back into an analog signal to drive the switches of the buck converter to maintain the output voltage to the desired value [1].

The PWM mode of operation is used under heavy and medium load conditions. The PFM mode of operation comprises a hysteretic comparator which compares the output voltage with an upper and lower threshold voltage to generate PFM pulses [2]. A multiplexer is used to select the PWM or PFM mode of operation depending on load condition [3, 4]. Finally, we have a dead time generator to generate two non-overlapping pulses to drive power MOSFETs. Under light load condition, a zero current detection is done to operate the buck converter in discontinuous current mode (DCM) of operation. The buck converter is designed to provide 1 V, 800 mA output from an energy source, that is, battery with an input voltage of 3.2 V.

Considering 20% current ripple, 1% voltage ripple and 5% maximum output voltage overshoot, the synchronous buck converter is designed. A MATLAB program is written to find the values of the LC filter. Table 1 shows the main specifications of the synchronous buck converter.

3 PWM Mode of Operation of the Buck Converter

In PWM architecture, the desired output voltage is obtained by varying the duty cycle D of the switching elements. The frequency of operation is kept constant which simplifies the design of the feedback loop and the output LC low-pass filter. PWM method of control also results in high efficiency and low output ripple content during moderate and heavy load conditions. However, in PWM control technique the efficiency of the converter reduces drastically under light load conditions as the switching losses which are load independent becomes predominant over the conduction losses. The two main components of PWM mode of operation of buck converter are DPWM and type III compensator, which will be explained in the following sections.

3.1 Design of 13-Bit $\Sigma - \Delta$ Modulator DPWM

A high-resolution DPWM is necessary to achieve precise output voltage and avoid limit-cycle oscillations [5–7]. In this paper, a 13-bit DPWM structure which consists of a second-order $\Sigma - \Delta$ modulator having 6-bit resolution and a counter-comparator block with the 7-bit resolution is designed. The hybrid architecture consists of one

Table 1 Buck converter specifications

Parameter	Minimum	Nominal	Maximum
Input voltage (V_{in})	1.8 V	3.2 V	3.6 V
Output voltage (V_{out})	0.9 V	1 V	1.2 V
Output current (I_{out})	2 mA		800 mA
Switching frequency (f_s)	–	10 MHz	–
Output inductor (L)		1 μ H with DCR = 1 $m\Omega$	
Output capacitor (C_0)		2 μ f with ESR = 9 $m\Omega$	

hardware method (counter comparator DPWM) and one soft method ($\Sigma-\Delta$ modulator) to ease or lessen the requirement of high clock frequency and hence reduce power consumption [8, 9]. Figure 2 shows the structural representation of the hybrid DPWM.

Figure 3 shows the structural configuration of the $\Sigma-\Delta$ modulator-based DPWM. The effective resolution of the 7-bit core DPWM is increased to 13-bit by 6-bit $\Sigma-\Delta$ modulator by using noise shaping concept. The hybrid $\Sigma-\Delta$ modulator-based DPWM is modeled in the digital domain using Xilinx system generator tool and timing simulation is performed to verify the design.

3.2 Design of Type III Compensator

A type III compensator provides three poles (one at the origin) and two zeroes which results in phase boost to offset the phase lag introduced by output LC filter complex pole [10]. The poles and zeroes of the type III compensator network should be carefully placed and selected to achieve the desired crossover frequency, sufficient phase margin, good steady-state output and fast transient response. The transfer function in the digital domain is obtained as:

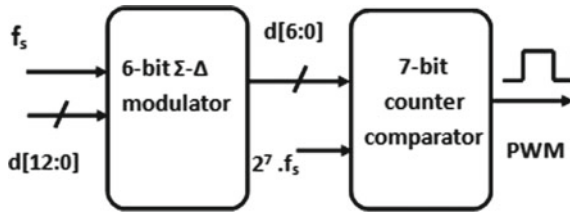


Fig. 2 Structure of $\Sigma-\Delta$ DPWM

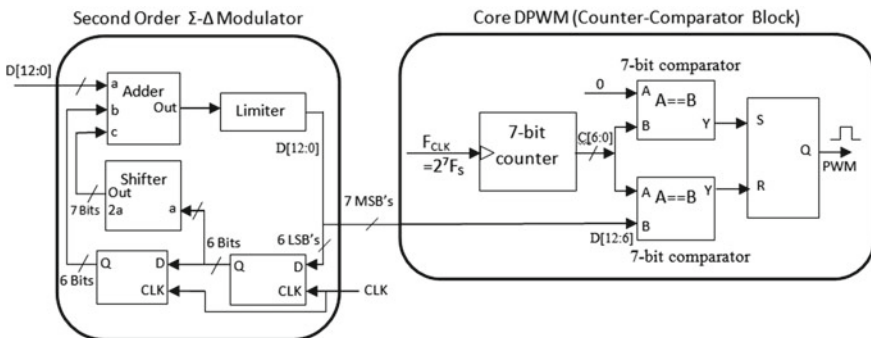


Fig. 3 The architecture of high-resolution $\Sigma-\Delta$ modulator-based DPWM

$$G_{TYPE3}(z) = K + \frac{R_1}{(z - P_1)} + \frac{R_2}{(z - P_2)} + \frac{R_3}{(z - P_3)} \quad (1)$$

Based on the specifications of the buck converter as given in Table 1, pole placement technique is used to designing type III compensator for a phase margin of 70° and a crossover frequency of $1/16$ of switching frequency. A MATLAB program is written to obtain a digital representation of type III compensator and Fig. 4 shows its bode plot. The phase margin and crossover frequency of buck converter without compensator is 12.3° and 1.44×10^6 rad/s, respectively, and is increased to 62.2° and 4.16 rad/s by using type III compensator.

Once the compensator is designed, it is implemented digitally in a parallel configuration for a buck converter. In order to perform digital control of buck converter, Xilinx system generator tool is used, which is as shown in Fig. 5.

4 PFM Mode of Operation of the Buck Converter

PFM is a nonlinear control method in which a train of pulses is applied to the converter switches to maintain the output voltage within the predetermined voltage range. The frequency of operation is reduced in PFM method, which in turn lowers the switching

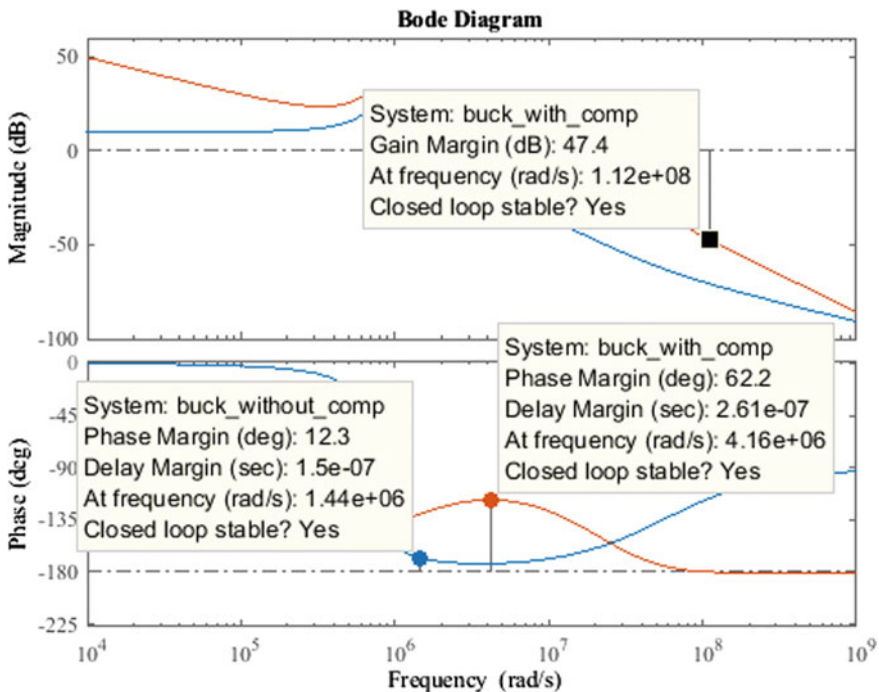


Fig. 4 Bode plot for buck converter with type III compensator

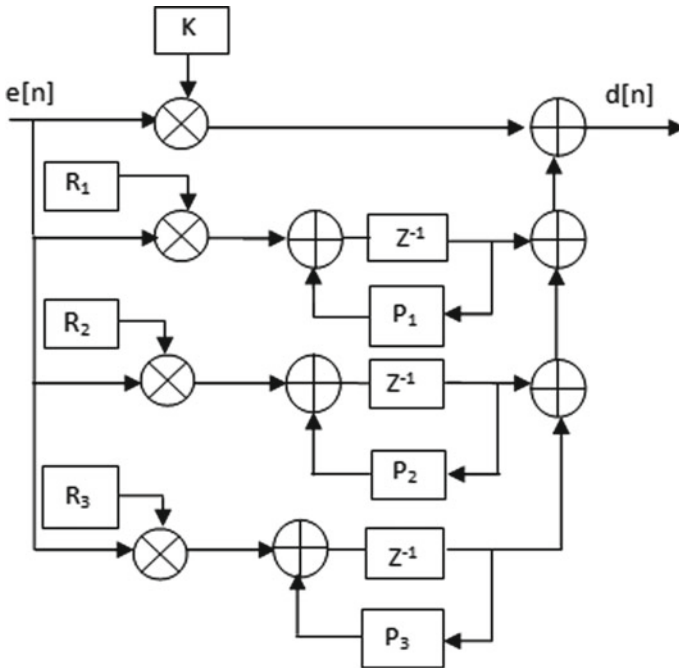


Fig. 5 Type III compensator architecture

losses and increases the efficiency under light load condition. The PFM converter circuits do not require loop compensators and hence they are simple and easy to design. Digital PFM architecture is implemented using a hysteretic comparator which compares the output voltage with upper and lower threshold voltages to control the ON-OFF of the power MOSFET switches. A zero inductor current detector algorithm is implemented to prevent negative inductor current. Figure 6 shows the DPFM architecture.

5 Results and Its Analysis

The hardware–software co-simulation of digital power supply regulator is performed in the MATLAB/Simulink environment using the Vivado system generator tool to validate the design. The co-simulation is carried out with type III compensator, 13-bit $\Sigma-\Delta$ DPWM and digital reference voltage with soft start in PWM mode of operation to analyze its performance under various load and line transient conditions. The power supply module operates under PWM mode of operation during medium and heavy load conditions and under PFM mode of operation during light load conditions (<0.1 A) to obtain good efficiency. Figure 7 shows the digital power supply regulator

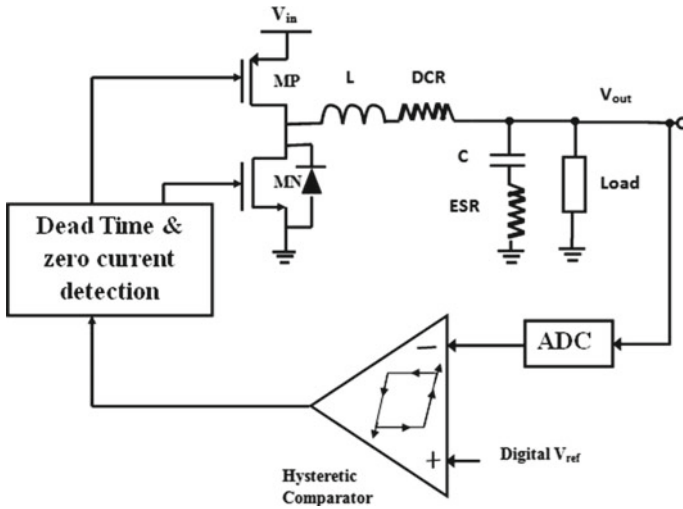


Fig. 6 The architecture of hysteresis DPFM control

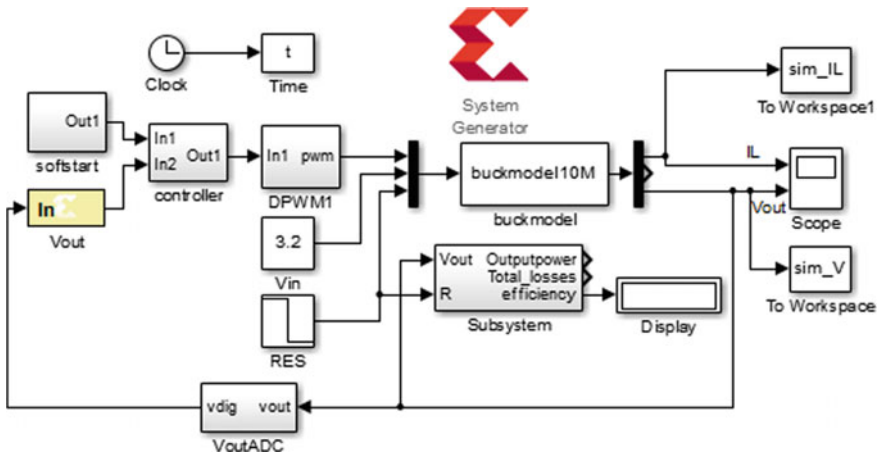


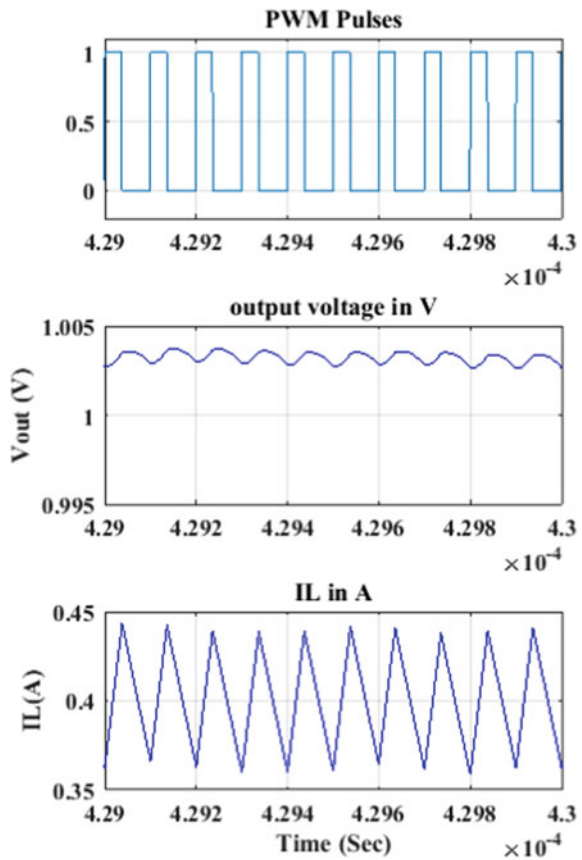
Fig. 7 The hardware–software modeling of a digital power supply regulator

hardware/software model in the Simulink environment.

5.1 Static Performance Analysis

The performance of the buck switching mode power supply under various load conditions for three different input voltage conditions are analyzed and results are tabulated. Figure 8 shows the PWM pulses, output voltage and inductor current at a load current of 0.4 A. It is observed that the output voltage obtained is 1.0024 V, the ripple content in inductor current is between 70 and 85 mA and ripple content in output voltage is 0.8 mV. Figure 9 shows the plot of output voltage against various load conditions for three input voltages ($V_{in} = 3.2, 2.5$ and 1.8 V). The average accuracy of 0.24% is obtained which is within the limit of the desired accuracy of 1%.

Fig. 8 PWM pulses, output voltage and inductor current response when the load current is 0.4 A



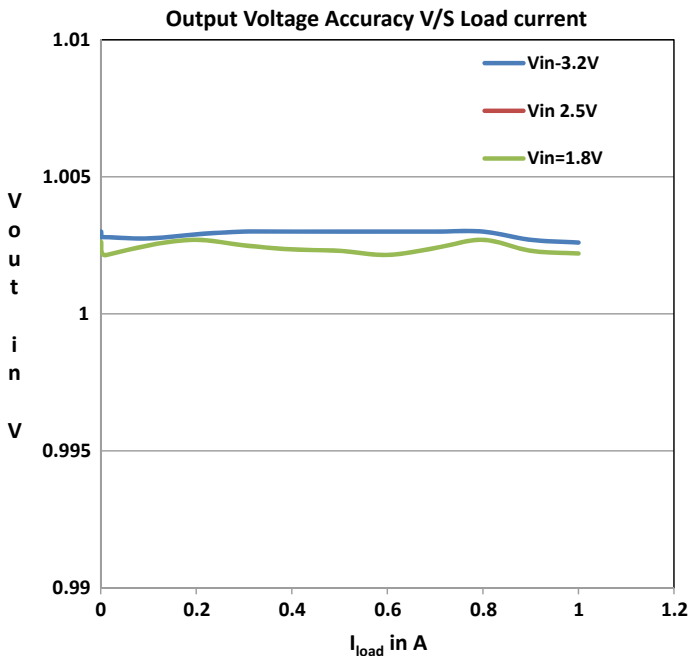


Fig. 9 Output voltage accuracy versus load current

5.2 Dynamic Performance Analysis

The dynamic performance of digitally controlled switching mode buck converter for different load and input battery voltage variations is analyzed, compared and the results are tabulated.

Figure 10a shows the load transient, output voltage and inductor current when load conditions change from 150 to 400 mA and vice versa. It is observed that undershoot in output voltage when the load varies from 150 to 400 mA is 30 mV and overshoot when the load varies from 400 to 150 mA is 32 mV. The % peak overshoot/undershoot is around 3%. The response time of the power supply when load variations occur (150–400 mA and vice versa) is 3 μ s. The steady-state output voltage is 1.002 V. Figure 10b shows the output voltage transient response when the input battery voltage is varied from 2.5 to 3.2 V and vice versa. The undershoot and overshoot in output voltage are 20 and 25 mV, respectively. The time taken by the power supply to settle to a steady value of 1.0025 is 10 μ s.

The DPFM mode of operation is used to improve the efficiency under light load condition. In this paper, the DPFM mode of operation is chosen to load current values of <0.1 A. Figure 11 shows the %Efficiency versus load current in DPFM mode of operation. The minimum efficiency obtained is 66.62% at 0.001 A. Figure 12 shows the power conversion efficiency for forced DPWM and DPFM mode of operation.

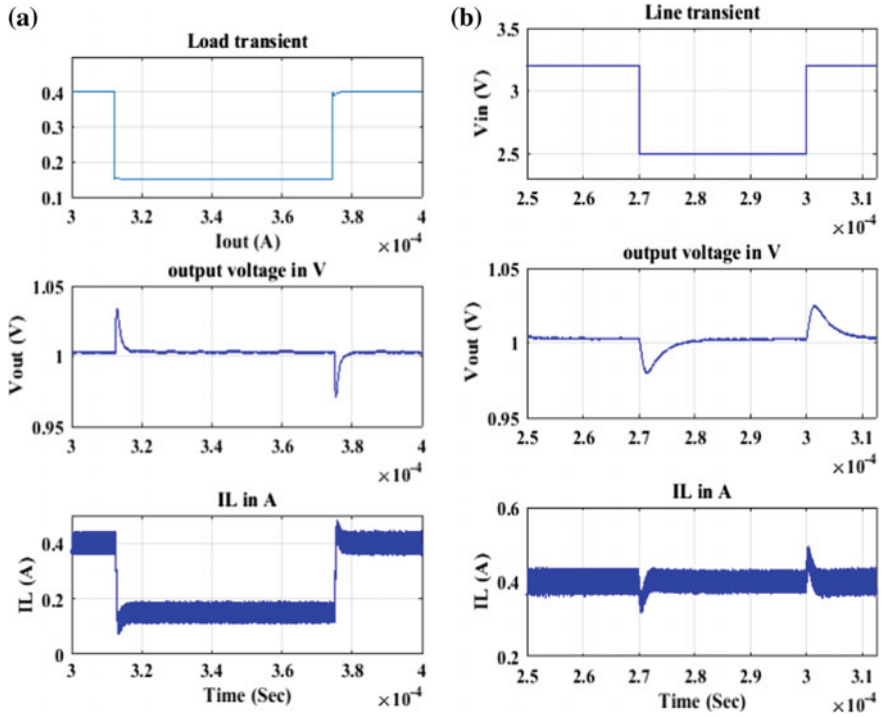


Fig. 10 a Load transient response for load variations of 150–400 mA and vice versa. b Transient response for an input battery voltage change of 2.5–3.2 V

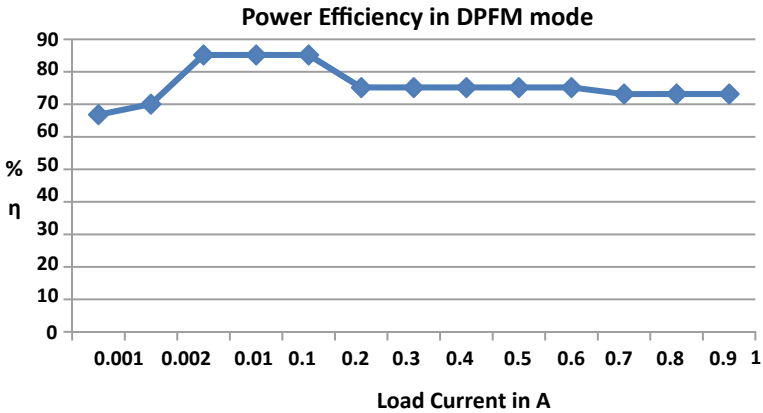


Fig. 11 Power efficiency versus load current for DPFM mode of operation

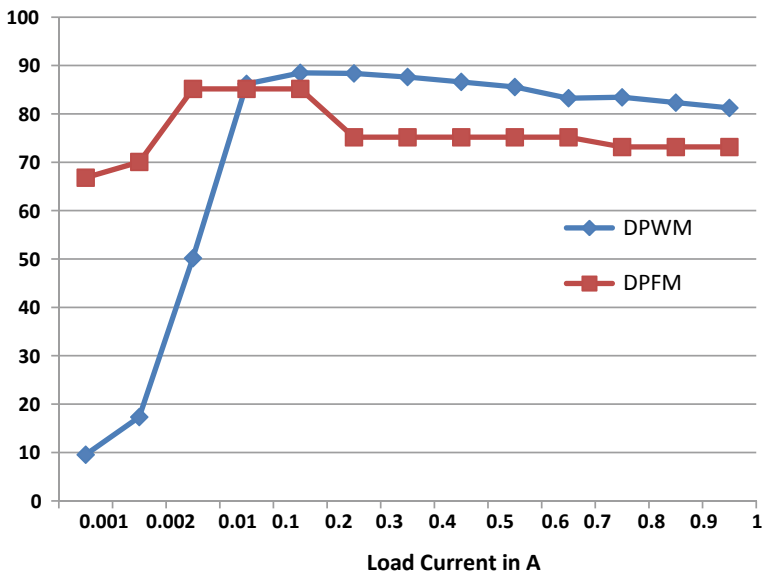


Fig. 12 Power efficiency versus load current for DPFM and DPWM mode of operation

The DPFM has a much higher efficiency of 66.82–85% for load currents <0.1 A. The forced DPWM has a power conversion efficiency of merely 10% at 0.001–86% at 0.1 A. Hence power supply circuitry is operated in DPFM mode of operation for load currents <0.1 A.

6 Conclusion

A multi-mode digitally controlled switching mode buck converter is implemented to power system-on-chip (SoC). It operates in DPWM mode under heavy and medium load conditions, whereas in DPFM mode under light load conditions to obtain high power conversion efficiency over a wide range of load current. Maximum efficiency of 88.48% is obtained at 0.2 A under PWM mode of operation. An output voltage accuracy of 0.25% is achieved against a target value of 1%. The load current and line voltage transient responses are verified by considering the different range of load current and line voltage variations. Light load or power save mode efficiency is improved by using a DPFM mode of operation. The minimum efficiency obtained is 66.62% at 0.001 A. The %Efficiency lies between 66.62 and 85% for a load current range of 0.001–0.1 A. The voltage ripples during DPFM mode of operation are observed to be 3 mVpp, which is within the desired value of 10 mVpp.

References

1. C.-S. Wu, K.-C. Lin, Y.-P. Kuo, P.-H. Chen, Y.-H. Chu, W. Hwang, An all-digital power management unit with 90% power efficiency and ns-order voltage transition time for DVS operation in low power sensing SoC applications, in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1370–1373 (2015)
2. K. Wang, N. Rahman, Z. Lukić, A. Prodić, All-digital DPWM/DPFM controller for low-power DC-DC converters, in *Twenty-First Annual IEEE Applied Power Electronics Conference and Exposition*, pp. 719–723 (2006)
3. X. Zhang, D. Maksimovic, Digital PWM/PFM controller with input voltage feed-forward for synchronous buck converters, in *Twenty-Third Annual IEEE Applied Power Electronics Conference and Exposition*, pp. 523–528 (2008)
4. C.-H. Tsai, C.-H. Yang, J.-H. Shiau, B.-T. Yeh, Digitally controlled switching converter with automatic multimode switching. *IEEE Trans. Power Electron.* **29**(4), 1830–1839 (2014)
5. A.V. Peterchev, S.R. Sanders, Quantization resolution and limit cycling digitally controlled PWM converters. *IEEE Trans. Power Electron.* **18**(1), 301–308 (2003)
6. A. Kelly, K. Rinne, High-resolution DPWM in a DC-DC converter application using digital sigma-delta techniques, in *IEEE 36th Power Electronics Specialists Conference*, pp. 1458–1463 (2005)
7. P.-K. Leong, C.-H. Yang, C.-W. Leng, C.-H. Tsai, Design and implementation of sigma-delta DPWM controller for switching converter, in *IEEE International Symposium on Circuits and Systems*, pp. 3074–3077 (2009)
8. S. Guo, Y. Gao, Y. Xui, X. Lin-Shi, A. Bruno, Digital PWM controller for high-frequency low-power DC-DC switching mode power supply, in *IEEE 6th International Power Electronics and Motion Control Conference (IPEMC 2009)*, pp. 1340–1346 (2009)
9. Z. Lukić, N. Rahman, A. Prodić, Multibit $\Sigma-\Delta$ PWM digital controller IC for DC-DC converters operating at switching frequencies beyond 10 MHz. *IEEE Trans. Power Electron.* **22**(5), 1693–1707 (2007)
10. M.K. Alghamdi, A.A. Hamoui, A spurious-free switching buck converter achieving enhanced light-load efficiency by using a $\Delta\Sigma$ -modulator controller with a scalable sampling frequency. *IEEE J. Solid-State Circuits* **47**(4), 841–851 (2012)

Depth Analysis of Different Medical Image Segmentation Techniques for Brain Tumor Detection



Kapil Kumar Gupta, Namrata Dhanda and Upendra Kumar

Abstract Segmentation is the most significant step in studying and assimilating medical CT and MR images. To identify the feature areas in the medical images and to clip them, segmentation is used. Owing to the continued growth in technology and research areas, it becomes more challenging to take out the best method of image reconstruction. Therefore, many technologies have been made for the identification, detection and classification of brain tumor parts in MR images. Different radiology tools are used for identification and detection of the inner components of the human anatomy by physicians and radiologists without any surgery. In this paper, the authors paid attention on the segmentation analysis of brain tumor MR images. For the comparison purpose, the authors consider the parameters like run-time complexity, sensitivity and segmentation accuracy.

Keywords Pattern recognition · Medical imaging · MRI · Brain tumor diagnosis · Image segmentation

1 Introduction

Highly efficient and reliable images are being produced by modern three-dimensional medical analysis tools and techniques. The analysis and research of medical MR and CT radiological images are rapidly growing nowadays. Any image that is taken up by

K. K. Gupta (✉)

Department of CSE, Amity University, Lucknow, India

e-mail: kapilkumargupta2007@gmail.com

Department of CSE, Institute of Technology,

Shri Ramswaroop Memorial University, Barabanki, India

N. Dhanda

Amity University, Lucknow, India

e-mail: ndhanda510@gmail.com

U. Kumar

IET, Lucknow, India

e-mail: upendra.ietlko@gmail.com

© Springer Nature Singapore Pte Ltd. 2020

L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits*

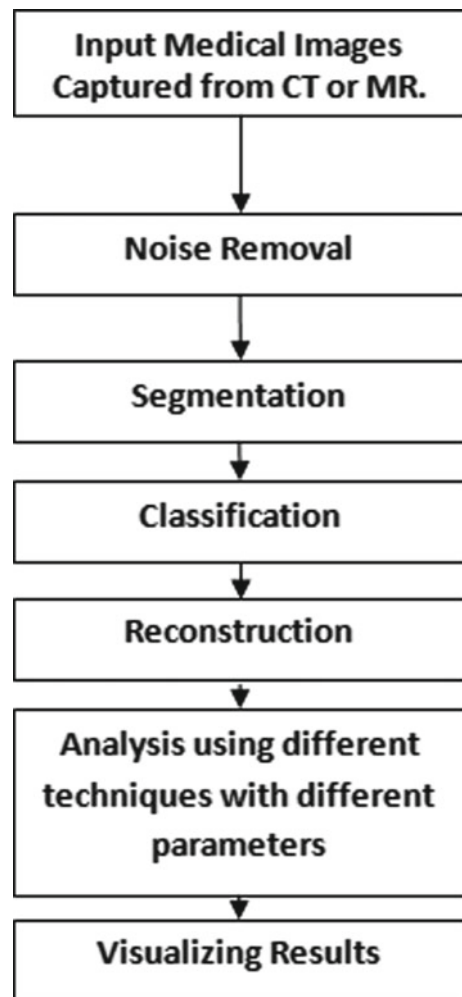
and Signals, Advances in Intelligent Systems and Computing 1064,

https://doi.org/10.1007/978-981-15-0339-9_16

a two-dimensional camera, graphical images and text images produce 2-D radiology images and satellite pictures taken from the recent satellites are also the 2-D medical images [1]. Figure 1 shows the typical image processing process. Initially, we take input from the MRI radiological tool. The second step is about detecting the type of noise and removing it by using appropriate filters. The next stage is to segment the tumor region of the MR pictures of the brain tumor. Then the segmented images go through the next step of the classification and reconstruction. So, as a result, we will get the required medical image.

In our normal brain, cells develop in a regulated way, replacing damaged cells with fresh cells. But tumor cells produce uncontrollably into the brain and this reason is still not being understood. A brain tumor is an uneven anomalous growth that

Fig. 1 Visual representation of the general medical process



begins in the brain and usually does not spread to other parts of the body. Benign or malignant may be the primary brain tumors. A slowly produced benign brain tumor has definite limits, and scarcely diffuses. Although their cells are not malignant, if they are situated in a critical area, benign tumors can be life-threatening. A malignant brain tumor develops rapidly, has sporadic limits and spreads to neighboring regions of the brain. Although often referred to as brain cancer, malignant brain tumors do not fit the definition of cancer because they do not spread to organs outside the brain and spine.

- Tumors can influence the brain if ordinary tissue is destroyed, ordinary tissue is compressed or intracranial pressure is increased. Symptoms differ with the type and place of the tumor in the human brain. General symptoms are as follows:
- Tendentially worsening headaches
- Difficulty in the natural walking
- Natural language speaking issues
- Vision and eye rotation problems
- Increased intracranial pressure causing somnolence, headaches, nausea and vomiting, and sluggish reactions.

Our focus in this paper is to distinguish between distinct methods of segmentation. The segmentation is the processing of a digital image into various sections with comparable characteristics such as color and intensity. It is used to divide the items into an extensive picture used for the profound study. The primary objective of segmentation of the medical image is:

- A thorough study of the inner human body composition
- Locate tumor and other abnormalities to define the lesion
- Calculate tissue quantity to determine tumor development
- Calculation of the radiation part.

Radio images are multiplex in nature. These images provide soft-tissue delineation in different areas of interest [2]. Automatic segmentation of medical images is therefore very hard [3]. Owing to the following reasons, the whole process can be changed:

- Substantial volume modification
- Intensity variation in the medical images
- Presence of the artifacts
- Similarities of different tissues present in the black and white images.

2 Segmentation Techniques

Medical image processing is always the center of attraction for the researchers. In the medical imaging detection of brain tumor is a challenging task [4]. Over the past

30 years, many study papers on medical imaging containing brain tumor have been released. In the last decade research is growing exponentially. So, authors compared techniques from 2009 to 2018. The details of brain tumor detection technique are listed below:

2.1 Wavelet Transform-Based Brain Tumor Segmentation in MRI Images

Ahmed Kharrat et al. presented research paper on brain tumor identification using the segmentation method “wavelet transform” to decompose MR images. After that k-means is applied to remove the suspicious locales or tumors.

The wavelet transform is deterioration of an image into a group of functions called a wavelet family $\Psi_{x,y}(s)$, “Eq. (1)”, in which the majority of the premise wavelets are gotten from scaling and interpretation of a solitary function called the mother wavelet $\Psi(s)$, “Eq. (2)”.

$$B_{x,y} = \int_{-\infty}^{\infty} a(s)\Psi_{x,y}(s)ds \quad (1)$$

$$\Psi_{x,y}(s) = \frac{1}{\sqrt{x}}\Psi\left(\frac{s-y}{x}\right) \text{ with } x \neq 0 \quad (2)$$

This transformation represents image at different resolution levels. Wavelet investigation is a successful procedure fit for uncovering parts of information which other flag examination systems. By dissecting the pictures at various levels, the technique can extricate better subtle elements from them and thus moves forward the nature of the picture. Furthermore, wavelet testing is fitted to compact or de-noise a signal without significant degradation [5].

2.2 Multi-parameter Watershed Segmentation-Based Brain Tumor Detection

Rajeev Ratan et al. did their research by using multi-parameter MR images analyses. The perception and assessments of the division results show the adequacy of this methodology.

For segmentation of MR images watershed segmentation has been applied. By deciphering the inclination guide of a force picture as tallness esteems, we get lines which seem, by all accounts, to be edges. On the off chance that the forms were a landscape, falling precipitation would discover the path from the partitioning lines toward the associated catchments bowl. These isolating lines are called watersheds.

It has been used for detection of 2D and 3D images of tumor. Watershed division uses the force as a parameter to section the entire picture informational collection [6] (Fig. 2).

2.3 Segmentation of Medical Images Using Texture Analysis

Qurat-Ul-Ain et al. proposed a well-organized and fast method of brain tumor diagnosis. First, texture feature extraction takes place. In the second step, the texture-based classification of timorous images by using the ensemble base classifier. Using two-stage segmentation process, the extracted tumor region is classified as malignant.

The process of segmentation in this method includes skull removal and tumor extraction phases.

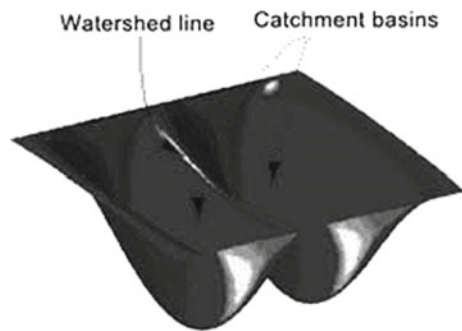
Authors propose a unique approach to extract texture feature, which is the combination of two methods. First, the calculation of texture based on the first order of the histogram and secondly, the calculation of texture based on the matrix [7] (Fig. 3).

2.4 Computer-Aided Thresholding and Morphological Tool for Brain Tumor Detection

M. Akram et al. did study on automatic diagnostic system for medical images. They defined their process under three stages:

1. MR images of brain is procured and preprocessing is done to evacuate the comotion and to hone the image.
2. Global threshold segmentation is done.
3. Fragmented image is treated by the tolls of morphological operations and tumor concealment is performed to evacuate the false positive (Fig. 4).

Fig. 2 Watershed segmentation simplified to 2D



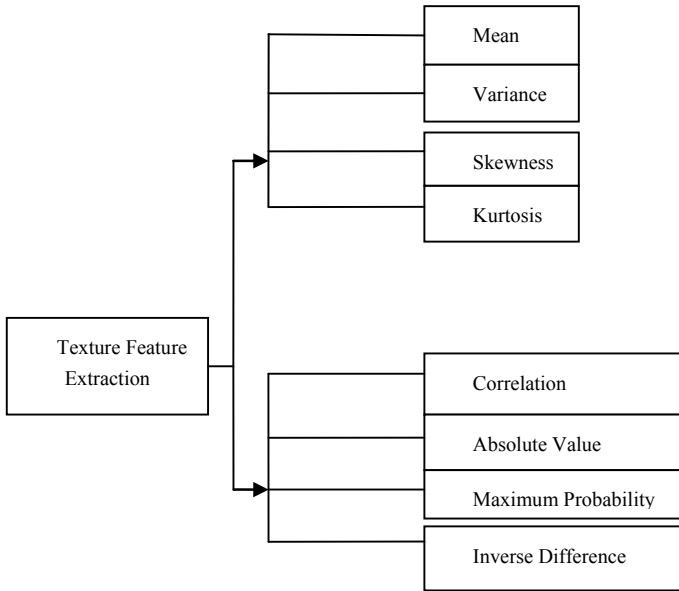


Fig. 3 Texture-based extraction technique for brain tumor MRI classification

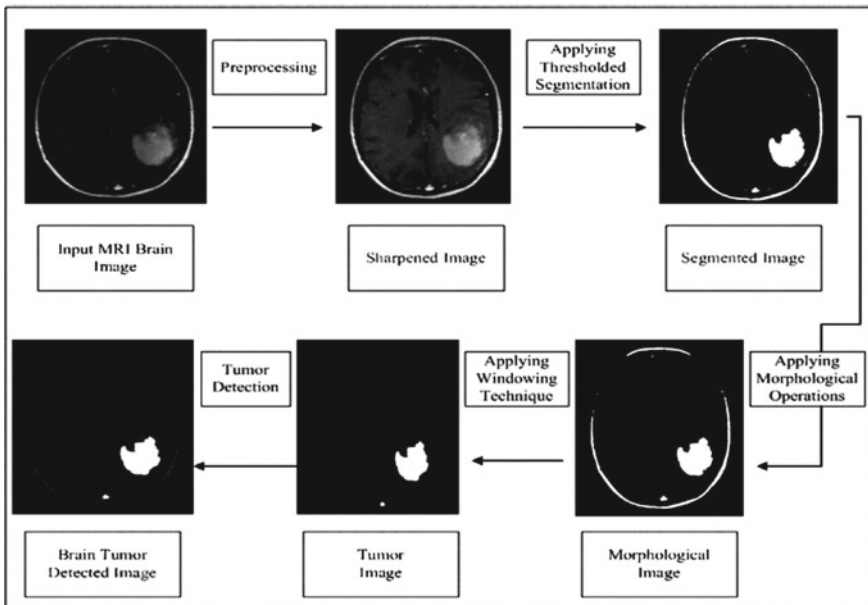


Fig. 4 Flowchart of the computer-aided brain tumor detection

MR image segmentation of the brain tumor is based on worldwide threshold segmentation. The steps of this segmentation process are:

1. Select threshold value.
2. To transform grayscale image to binary image, apply limit value.
3. If the value of pixels exceeds the limit value, then select a foreground image; otherwise background [8].

2.5 Image Analyses of Brain Tumor Through Multi-scale Modeling

In this frame of reference, Stefan Baurer et al. put forward a unique way to select a healthy brain images from the timorous images. Tumor growth modeling in conjunction with licensed algorithms is chosen to build a comparison between a healthy and a patient brain picture. Finally, a close comparison is made using non-rigid registration between the mutated atlas and the patient picture.

This method provides liberty in the segmentation of atlas-based brain tumor picture as well as enhanced patient-specific simulation and tumor diagnosis. Segmentation based on atlas has the ability to segment the picture without leaving any relationship between areas and intensities of pixels.

The different steps engaged in the method of segmentation are:

1. Using affine registration procedure, the first step is to register images.
2. Next, positioning seed point alongside mean of the medical images, ignoring areas containing CSF.
3. Compare with the manually segmented eroded images. When the volume matches with one of eroded patient tumor then the tumor growth get stopped.
4. Using non-rigid registration step, exact deformation of tumor and nearby tissues is obtained [9].

2.6 Probabilistic Neural Network-Based Brain Tumor Detection

Dina Abo Edahab et al. proposed an enhanced technique that applied on MR images using probabilistic neural network. Gray-level image segmentation is used to organize data such as acknowledging the area of concern, that is, lesion, locating tumor and other anatomical structure and defects. The data of the healthy parts' anatomical structure are contrasted with the components that are infected. The numbers below indicate the distinction between an ordinary and anomalous brain [10] (Fig. 5).

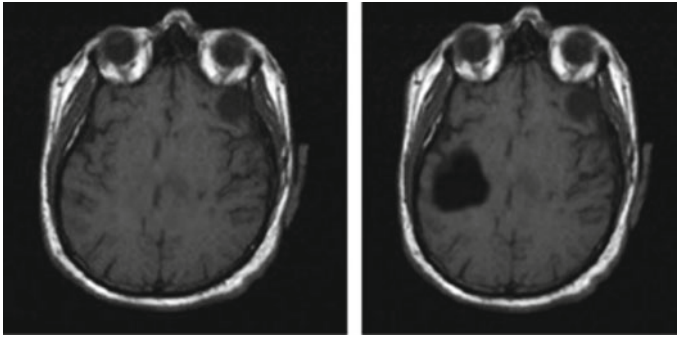


Fig. 5 Normal brain and abnormal brain MRI

Authors presented enhancement and smoothing step presumed that main image noise is random and additive, that is, spurious and random signal, $m(a, b)$, added to true pixel value $l(a, b)$:

$$I(a, b) = l(a, b) + m(a, b) \quad (3)$$

Spatial domain is based on the use of pixels directly in a small pixel is shown as:

$$h(i, j) = S[f(i, j)] \quad (4)$$

where $f(i, j)$ is the image input, $h(i, j)$ is the processed image and S is an operator on f , defined over some neighborhood of (i, j) . The proposed noise enhancement algorithm depends on using spatial filters and includes smoothing and sharpening filters [11].

Smoothing by linear filters: In this algorithm we presumed L as an $M \times N$ image, m is an odd no. smaller than both M and N , and B is the convolution kernel. The filtered version of L is as follows:

$$L_B(a, b) = L * B = \sum_{h=-n/2}^{n/2} \sum_{l=-m/2}^{m/2} B(f, l) L(a - f, b - l) \quad (5)$$

where $a = 1$ to M and $b = 1$ to N . The value $L(a, b)$ is get replaced with weighted sum of L values in a neighborhood of (a, b) [12].

In the next step smoothing through Gaussian filter takes place. Let G_f is a non-negative, real-valued column matrix given by:

$$G_f(a, b) = \frac{1}{d} \exp \left[\frac{-[a^2 + b^2]}{2\sigma^2} \right] \quad (6)$$

where d is expressed as $d = \sqrt{2\pi\sigma^2}$.

Noise reduction level gets enhanced by this type of filters, compared with linear filters (Fig. 6).

2.7 Detection of Brain Tumors Using Segmentation Based on Watershed and Thresholding

Anam Mustaqeem et al. proposed a unique procedure for identification of tumor by using watershed method, thresholding segmentation and morphological operations [13]. The proposed methods follow the following steps:

Step 1. Threshold segmentation: It is a simple segmentation method. This technique depends on the limiting value which converts the gray picture to a binary image format. Usually, this technique uses maximum entropy and k-means clustering techniques [14].

Step 2. Watershed segmentation: In this method, on the bases of intensity of pixels, group of pixels is being formed. Pixels with same intensity are grouped together. To distinguish a timorous image, watershed mathematical morphological tool is used. This method usually suffers from over and under segmentation [12].

Different methods are used for using watershed segmentation:

- (a) The computed local images, gradient maximum is selected [14].
- (b) Using markers, the specific marker positions are used [15].

Step 3. Morphological operations: When the picture is transformed into binary format, it is implemented with few morphological operations. The primary aim of this is to separate portion of the picture from the tumor.

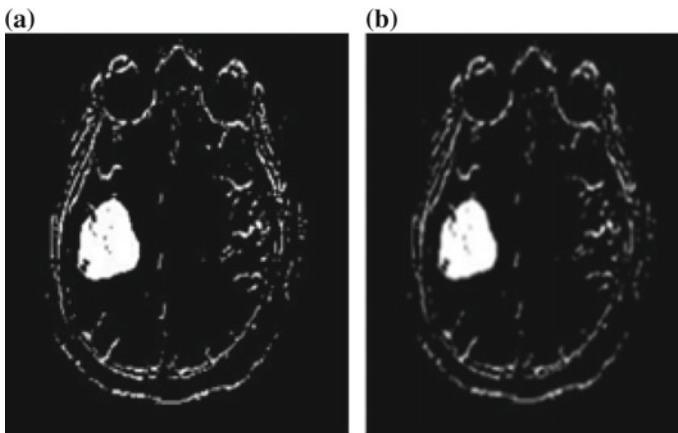


Fig. 6 a. Applying Gaussian filter. b After applying averaging filter

2.8 Brain Tumor Detection Using Watershed Segmentation

Padmakant Dhage et al. put forward a research paper on brain tumor detection using watershed segmentation in 2015. This document represents the capacity of the watershed division to isolate the irregular tissue from the typical tissue in order to obtain a true idea of included and non-included property, which definitely helps the specialist to separate the region. The right location and shape of the tumor are fixed and various parameters such as edge, whimsy, entropy and centroid were calculated. Watershed segmentation is edge-based segmentation algorithm. Any gray-scale picture can be seen as a topographic surface where high force signifies pinnacles and slopes while low power indicates valleys. We begin filling each secluded valleys (nearby minima) with various colored water (labels). As the water rises, contingent upon the pinnacles (angles) adjacent, water from various valleys, clearly with various hues will begin to blend. To stay away from that, we fabricate obstructions in the areas where water blends. We proceed with crafted by filling water and building obstructions until the point when every one of the pinnacles is submerged. At that point, the obstructions we made give you the division result. This is the “logic” behind the watershed [16, 17].

2.9 Analyses of Segmentation and Classification Through Levenberg–Marquardt Approach for Brain Tumor MRI Images

A. Shenbagarajan et al. presented an analysis over brain tumor MRI images by using active contour model-based segmentation and Levenberg–Marquardt-based classification techniques.

Analysis over brain tumor MRI images is done in the following steps:

1. Processing of data available.
2. Segmentation through ACM segmentation technique.
3. Implementation of feature extraction techniques.
4. Classification using ANN-LM technique.

In ACM (active contour model) segmentation technique, we assume that the image is statistically homogenous.

This process examines all effects of images like contour lengths and gradients. The energy can be minimized by using the following Eq. (4):

$$\ln Q(J_s|q) = \int \int_B J_s(a, b) dB \quad (8)$$

where $J_s(a, b)$ is the pixel intensity (a, b) of the MRI, and the integral part of the equation gives area enclosed by the curve q .

The quality of segmentation gets improved using region-based information instead of using gradient information [3].

2.10 Segmentation and Classification by Using Wavelet and Zernike Approach

M. Sornam et al. focus to begin an automated T1 and T2-based tumor diagnostic approach for weighted MR pictures. The proposed working process is as follows:

1. The initial step is to segment benign and malignant tissue from brain MR by filtering using the algorithm named k-means.
2. Feature extraction using wavelet and Zernike methods.
3. Evaluation through ELM (extreme learning machine) algorithm.

For segmentation process authors used traditional k-means algorithm [18].

The algorithm is as follows:

1. Select no. of cluster value k .
2. Randomly select centers of k cluster.
3. Performed mean computation of the cluster.
4. Now the distance to each cluster core between each pixel is calculated. If the distance calculated is close to the middle, migrate to the cluster.
5. Else move to the next cluster.
6. Recalculate the cluster center (Fig. 7).

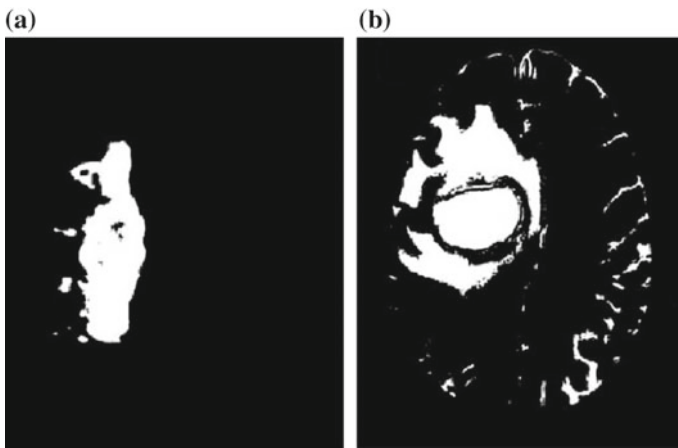


Fig. 7 Segmentation using k-means **a**—Benign, **b**—Malignant

2.11 Using SVM and ANN Classification for Detection of Brain Tumor

Rasel Ahmmed et al. proposed a brain tumor MR image classification technique using SVM and ANN. Temper-based k-means integration and altered fuzzy C-means (TKFCM) clustering algorithm is used for MRI image segmentation. Use support vector machine and artificial neural network to extract features and then classify MRI pictures after segmentation.

TKFCM algorithm assimilates k-means and fuzzy C-means with few modifications. Temper is added with typical k-means, which are identified in the brain picture by gray intensity. For coarse picture $A(a_i, b_j)$, gray-level-based temperature and picture complexity is determined by Eq. (9)

$$A(a_i, b_j) = \sum_{i=m+1}^{P+m} \sum_{j=m+1}^{Q+m} R(a_i, b_j) S_{PQ(resize)} \quad (9)$$

Temper window is chosen by S_{PQ} , given by Eq. (10)

$$S_{PQ} = \sum_{i=m+1}^{P-m} \sum_{j=m+1}^{Q-m} R(a_i, b_j) \quad (10)$$

In Eq. (9), The temper-based image matrix has a gray intensity number G and a number of bins B used to detect image temper $R(a_i, b_j)$, where m is defined as $m = (\text{winsize} - 1)/2$. The desired temper is obtained by putting exact values of temper, row and column.

For segmentation, clustering algorithm based on temper k-means and modified fuzzy C-means is shown separately:

$$L_g = \sum_{i=1}^D \sum_{j=1}^G A(a_i, b_j) \|a_i - d_j\|^2 \quad (11)$$

$$L_p = \sum_{j=1}^Q \sum_{i=1}^G (V_{ij}) * (h_{ij})^2 \quad (12)$$

From Eqs. (11) and (12) the contour with the help of which tumor region can be found is:

$$L_{gp} = \oint_d (L_g, L_p) \quad (13)$$

where P and Q are row and column of $A(a_i, b_j)$ d is contour value. Number of cluster, centroid of cluster and number of data points are defined by G , Q and D , respectively [19].

2.12 Fuzzy C-Means Clustering-Based Brain Tumor Detection

Alexander Zotin et al. proposed a unique method to detect edge in the brain tumor. Author uses fuzzy C-mean clustering algorithm to solve the brain tumor detection problem.

Fuzzy C-means clustering performs as follows:

- Identify clusters by iteratively searching for a collection of blurry clusters. Find out the cluster centre of each cluster.
- Cluster centres represent the structure of the data.
- It enables the separating of the current set of points of a certain amount of fluffy sets.
- Associated blurry membership matrix to determine the degree of membership of the initial set of vectors k th element in the i th cluster is prepared.
- Given n clusters, fuzzy C-means dataset is $X = \{x_1, x_2, \dots, x_m\}$ into n fuzzy clusters centres $C = (c_1, c_2, \dots, c_n)$.
- The objective function is

$$f(D, C) = \sum_{i=1}^n \sum_{k=1}^m d_i (x_m - c_n)^2 \quad (14)$$

where d is the degree of membership value, x_m is the m th cluster and c_n is the centre of the n th cluster.

The canny edge detector is implemented after dividing the picture into a sequence of homogeneous clusters using the FCM algorithm [20, 21].

3 Comparison

3.1 Comparison Based on the Different Parameters

For the comparison purpose of different above said techniques authors selected the following parameters:

- Imaging modality
- Image task
- Accuracy of the segmentation process.

These parameters are selected for comparison because of detection rate is promptly dependent on the given parameters. Comparison of above said parameters is listed in Table 1.

3.2 Comparison Based on Time Complexity of Different Algorithms

It is difficult to compare one technique with other. There are a lot of parameters for comparison. Result is dependent upon the quality of the image as well as technique applying. If all the techniques are tested on the same data set, then the time complexity of the technique plays big role in the fast detection of brain tumor. Comparison of the time complexity is listed in Table 2.

3.3 Histogram Analysis of Segmentation Accuracy

The given below plot presents comparison of the different segmentation techniques accuracy rate. Authors prefer histogram analysis for the visualization of percentage accuracy rate listed in Table 1 (Fig. 8).

4 Conclusion

Based on the comparative analysis of the different techniques, the accuracy level of the segmentation is better in texture-based analysis, local independent-based transformation and fuzzy C-mean clustering algorithm. However, the comparison made on the basis of the time complexity predicts different result. On the basis of time complexity computer-aided tool containing morphological erosion and dilation technique is better than other techniques. The robustness of the algorithm against the different parameters like the variance of illumination, focus and scales has been still unchecked for a number of sample images.

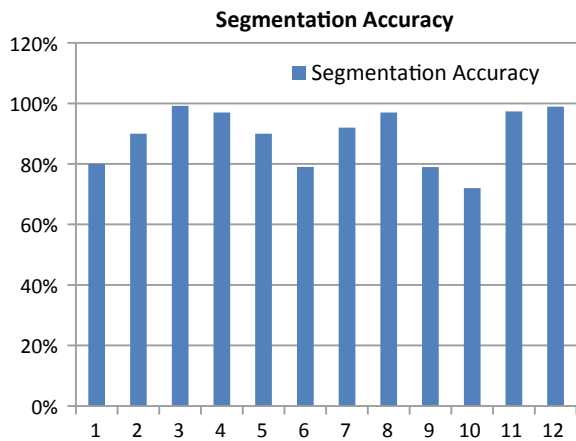
Table 1 Comparison of different techniques based on segmentation accuracy

S. no.	Research	Segmentation technique	Imaging modalities	Image task(s)	Segmentation accuracy (%)
1	Kharrat et al. [5]	Wavelet transform with morphology	MRI	Segmentation	80
2	Ratan et al. [6]	Watershed	MRI	Segmentation	90
3	Qurat-UI-Ain et al. [7]	Texture and ensemble classifier	MRI	Segmentation and classification	99.20
4	Usman Akram et al. [8]	Computer-aided tool containing morphological erosion and dilation	MRI	Segmentation	97
5	Baurer et al. [9]	Multistage modeling using Eulerian approach	MRI	Registration and segmentation	90
6	Edhab et al. [10]	Probabilistic neural network techniques	MRI	Segmentation and classification	79
7	Mustaqeem et al. [13]	Watershed and thresholding	CT images and MRI	Segmentation	92
8	Dhage et al. [16]	Watershed segmentation	MRI	Segmentation	97
9	Shenbagarajan et al. [3]	Artificial neural network based Levenberg-Marquardt	MRI	Segmentation and classification	78.96
10	Sornam et al. [18]	Wavelet and Zernike transform	MRI	Segmentation	72
11	Ahmmmed et al. [19]	SVM and ANN	MRI	Segmentation and classification	97.37
12	Zotin et al. [20]	Fuzzy C-means clustering	MRI	Edge detection	98.9

Table 2 Comparison of time complexity

S. no.	Research	Segmentation technique	Imaging modalities	Time complexity
1	Kharrat et al. [5]	Wavelet transform with morphology	MRI	$O(n^2 \log n)$
2	Ratan et al. [6]	Watershed	MRI	–
3	Qurat-Ul-Ain et al. [7]	Texture and ensemble classifier	MRI	$O(n^2 \log n)$
4	Usman Akram et al. [8]	Computer-aided tool containing morphological erosion and dilation	MRI	$(n \log m)$
5	Baurer et al. [9]	Multiscale modeling using Eulerian approach	MRI	–
6	Edhab et al. [10]	Probabilistic neural network techniques	MRI	$O(mn)$
7	Mustaqeem et al. [13]	Watershed and thresholding	CT images and MRI	–
8	Dhage et al. [16]	Watershed segmentation	MRI	–
9	Shenbagarajan et al. [3]	Artificial neural network-based Levenberg–Marquardt	MRI	$O(mn)$
10	Sornam et al. [18]	Wavelet and Zernike transform	MRI	$O(2^m)$
11	Ahmmmed et al. [19]	SVM and ANN	MRI	$O(mn)$
12	Zotin et al. [20]	Fuzzy C-means clustering	MRI	$O(n^m)$

Fig. 8 Histogram analysis of different techniques based on segmentation accuracy



Acknowledgements Prof. Dr. Anil Kumar, Director, ASET, Amity University, Lucknow, supports this work. They would also thank the anonymous reviewers for their important and constructive criticisms and suggestions, which significantly enhanced this paper's quality.

References

1. N. Goel, A. Yadav, B.M. Singh, Medical image processing: a review, in *IEEE International Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity (CIPECH)* (2016), pp. 57–62
2. Neeraj Sharma, Lalit M. Aggarwal, Automated medical image segmentation techniques. *J. Med. Phys.* **35**(1), 3–14 (2010)
3. A. Shenbagarajan, V. Ramalingam, C. Balasubramanian, S. Palanivel, Tumor diagnosis in MRI brain image using ACM segmentation and ANN-LM classification techniques. *Indian J. Sci. Technol.* **9**(1), 1–12 (2016)
4. K.K. Gupta, N. Dhanda, U. Kumar, A comparative study of medical image segmentation techniques for brain tumor detection, in *2014th International Conference on Computing Communication and Automation (ICCCA)* (2018)
5. A. Kharrat, M.B. Messaoud, N. Benamrane, M. Abid, Detection of brain tumor in medical images, in *International Conference on Signals, Circuits and Systems* (2009), pp. 1–6
6. R. Ratan, S. Sharma, S.K. Sharma, Brain tumor detection based on multi-parameter MRI image analysis. *ICGST-GVIP J.* **9**(III), pp. 9–17 (2009)
7. G.L. Qurat-Ul-Ain, S.B. Kazmi, M.A. Jaffar, A.M. Mirza, Classification and segmentation of brain tumor using texture analysis, in *Recent Advances in Artificial Intelligence, Knowledge Engineering and Data Bases* (2010), pp. 147–155
8. M. Usman Akram, A. Usman, Computer aided system for brain tumor detection and segmentation, in *International Conference on Computer Networks and Information Technology* (2011), pp. 299–302
9. S. Bauer, C. May, D. Dionysiou, G. Stamatakos, P. Büchler, M. Reyes, Multiscale modeling for image analysis of brain tumor studies. *IEEE Trans. Biomed. Eng.* **59**(1), 25–29 (2012)
10. D.A. Edhab, S.S.A. Ghoniemy, G.M. Selim, Automated brain tumor detection and identification using image processing and probabilistic neural network techniques. *Int. J. Image Process. Visual Commun.* **1**, 1–8 (2012)
11. D. Shevad, *Multi Object Segmentation and Tracking in Image Sequence* (Lamar University, Beaumont, April 2005)
12. J. Shi, Adaptive local threshold with shape information and its application to object segmentation, in *2009 IEEE International Conference Robotics and Biomimetics (ROBIO)*, 19–23 December 2009, pp. 1123–1128
13. A. Mustaqeem, A. Javed, T. Fatima, An efficient brain tumor detection algorithm using watershed and thresholding based segmentation. *IJ Image Graphics Signal Process.* **10**, 34–39 (2012)
14. G. Li, Improved watershed segmentation with optimal scale based on ordered dither halftone and mutual informational, in *2010 3rd IEEE International Conference Computer Science and Information Technology (ICCSIT)*, 9–11 July 2010, pp. 296–300
15. H. Hedberg, F. Kristensen, P. Nilsson, V. Owall, A low complexity architecture for binary image erosion and dilation using structuring element decomposition, in *Vol. 4, Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium, 23–26 May 2005*, pp. 3431–3434
16. P. Dhage, M.R. Phegade, S.K. Shah, Watershed segmentation brain tumor detection, in *International conference on pervasive computing (ICPC)* (2015)
17. J. Jaya, K. Thanushkodi, M. Karnan, Tracking algorithm for de-noising of MR brain images. *IJCSNS Int. J. Comput. Sci. Netw. Secur.* **9** (2009)

18. M. Sornam, M.S. Kavitha, R. Shalini, Segmentation and classification of brain tumor using wavelet and Zernike based features on MRI, in *IEEE International Conference on Advances in Computer Applications (ICACA), Coimbatore* (2016), pp. 166–169
19. R. Ahmed, A.S. Swakshar, M.F. Hossain, M.A. Rafiq, Classification of tumors and it stages in brain MRI using support vector machine and artificial neural network, in *Proceedings of IEEE International Conference on Electrical, Computer and Communication Engineering (ECCE), Bangladesh* (2017), pp. 229–237
20. A. Zotin, K. Simonov, M. Kurako, Y. Hamad, S. Kirillova, Edge detection in MRI brain tumor images based on fuzzy C-means clustering, in *22nd International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (2018), pp. 1262–1270
21. Z. Stosic, P. Rutesic, An improved canny edge detection algorithm for detecting brain tumors in MRI images. *Int. J. Signal Process.* **3**, 11–15 (2018)
22. A. Tom, P. Jidesh, Geometric transform invariant brain-MR image analysis for tumor detection, in *IEEE conference on Circuits, Controls and Communications (CCUBE), 1–6, December 2013*
23. P. Jidesh, S. George, Curvature driven diffusion coupled with shock for image enhancement/reconstruction. *Int. J. Signal Imaging Syst. Eng.* **4**(4):238–247 (2011)
24. G. Vijaya, A. Suhasini, An adaptive preprocessing of lung CT images with various filters for better enhancement, *Acad. J. Cancer Res.* **7**(3), pp. 179–184 (2014)
25. Meiyang Huang, Wei Yang, Wu Yao, Jun Jiang, Wufan Chen, Qianjin Feng, Brain tumor segmentation based on local independent projection-based classification. *IEEE Trans. Biomed. Eng.* **61**(10), 2633–2645 (2014)
26. A. Zotin, K. Simonov, F. Kapsargin, T. Cherepanova, A. Kruglyakov, L. Cadena, Techniques for medical images processing using shearlet transform and color coding, in *Computer Vision in Control Systems-4. Intelligent Systems 136*, ed. by M. Favorskaya, L. Jain (Springer, Cham, 2018)
27. J. Huo, K. Okada, E.M. van Rikxoort, H.J. Kim, J.R. Alger, W.B. Pope, J.G. Goldin, M.S. Brown, Ensemble segmentation for GBM brain tumors on MR images using confidence-based averaging. *Med. Phys.* **40**(9), 093502 (2013)
28. M.K. Date, M.S.P. Akarte, Brain image segmentation algorithm using K-means clustering. *Int. J. Comp. Sci. Appl.* **6**, 285–289 (2013)
29. C. Jason, E. Sharon, S. Dube, S. El-Saden, U. Sinha, A. Yuille, Efficient multilevel brain tumor segmentation with integrated bayesian model classification. *IEEE Trans. Med. Imaging* **27**(5), 629–640 (2008)
30. M. Lugina, N.D. Retno, R. Rita, Brain tumor detection and classification in magnetic resonance imaging (MRI) using region growing, fuzzy symmetric measure, and artificial neural network back propagation. *Int. J. ICT* **1**, 20–28 (2015)
31. W. Shuihua, D. Sidan, A. Abdon, L. Aijun, L. Zeyuan, Application of stationary wavelet entropy in pathological brain detection. *Multimed. Tools Appl.* 1–14 (2016)
32. J.L. Prince, J.M. Links, *Medical Imaging Signals and System* (Pearson Education, 2006)
33. D.J. Withey, Z.J. Koles, Three generations of medical image segmentation: methods and available software. *Int. J. Bioelectromag.* **9**, 67–80 (2007)
34. Neeraj Sharma, Lalit M. Aggarwal, Automated medical image segmentation techniques. *J. Med. Phys.* **35**(1), 3–14 (2010)

Comparative Analysis of Classification Techniques for Diagnosis of Diabetes



Paramjot Kaur and Ramanpreet Kaur

Abstract Diabetes is a disease with which many people are affected, and diagnosing diabetes is becoming an important task. Machine learning algorithms are widely used for detection and classification process. In this work, we have used five classifiers to diagnose disease. The dataset, Pima Indian diabetes database, used to validate our work is taken from an online repository. We evaluated different machine learning algorithms for their accuracy. The classification accuracy was comparable to the state-of-the-art ranging from 70.12 to 79.22%. In this work, we suggested that the Naïve Bayes algorithm is an optimal algorithm, which is good in terms of accuracy as well as running time complexity.

Keywords Machine learning · Classifier · Confusion matrix · ANN · SVM · Diabetes

1 Introduction

Diabetes is a chronic condition when the body does not produce the accurate amount of insulin. Diabetes increases the risk of many diseases, namely heart disease, kidney malfunctioning, nerve system damaging, and so on. For diagnosing a diabetic patient, an expert relies on his past experiences. Several studies are carried out to diagnose diabetes based on various parameters. There are four kinds of diabetes: Type1, Type2, gestational diabetes, and pregestational diabetes.

Several machine learning, neural network, and deep learning techniques have been applied in various fields, including medical data. These algorithms include linear discriminant analysis, support vector machine, K-nearest neighbor, multilayer neural network, and so on. Dogantekin et al. proposed a diagnosis system based on linear discriminant analysis (LDA) and adaptive network-based fuzzy system

P. Kaur (✉)

Shaheed Darshan Singh Pheruman Memorial College for Women, Rayya, Amritsar 143112, India
e-mail: Jotchandi1@gmail.com

R. Kaur

Chandigarh University, Mohali 140413, India

© Springer Nature Singapore Pte Ltd. 2020

L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_17

for diabetic data classification [1]. Another approach focused on diabetes diagnosis with the use of generalized discriminant analysis (GDA) and least square support method (LS-SVM) [2]. Sean N. Ghazavi and Thunshun W. Liao obtained good classification accuracy using fuzzy K-nearest neighbor algorithm and the adaptive network-based fuzzy inference system [3]. The GDA-LS-SVM technique had an accuracy of 82.05%. Mostafa Fathi and Mohammed Sainee Abadeh [4] made use of ant colony optimization for extraction of fuzzy rules for building a fuzzy classification system for diagnosis of diabetes disease. Kayaer and Yildirim [5] used the LM algorithm for the neural networks on a Pima Indian dataset and achieved an accuracy of 77.08%. Okan et al. investigated two different feed-forward artificial neural networks (FFANN) for diagnosing diabetes and concluded that Newman–Watts small world feed-forward artificial neural network (SW-FFANN) gives better results as compared to Watts–Strogatz SW-FFANN [6]. Izhan addressed class imbalance in various medical databases on clinical predictions using the method of bagging neural network [7]. One method used genetic algorithm as a feature selection algorithm and Naïve Bayes for classification purpose [8]. Another work explored popular machine learning algorithms for diabetes identification and validated the model against 10-fold validation [9]. One approach investigated multilayer perceptron learning on Pima Indian diabetes database and concluded that the algorithm is giving optimal results [10]. A recent work used multilayer perceptron classifier with R Studio platform for classification of diabetes data, and this work was also verified with MATLAB [11]. Particle swarm optimization for reducing number of attributes of data followed by support vector machine and fuzzy decision tree on Pima Indian diabetes dataset improved the accuracy for diabetes classification [12].

This work focuses on classifying UCI diabetes data with given attributes using five different learning algorithms and getting comparable accuracy as of other methods. We find that Naïve Bayes classifier outperforms all other classifiers with 79.22% accuracy. Further, this paper contains the proposed system, dataset description, classification method, results, and conclusion.

2 Proposed System

In this work, we trained various machine learning algorithms and validated their results on the test set. We took a benchmark dataset with eight predictors and one binary outcome variable. Then this dataset is partitioned randomly into 90% training set and 10% testing set. Then various classification models were trained and computed performance metrics in terms of accuracy.

A typical machine learning process is depicted in Fig. 1. In machine learning classification process, at first, classifier gets trained on certain training data samples and then it is validated against test data samples.

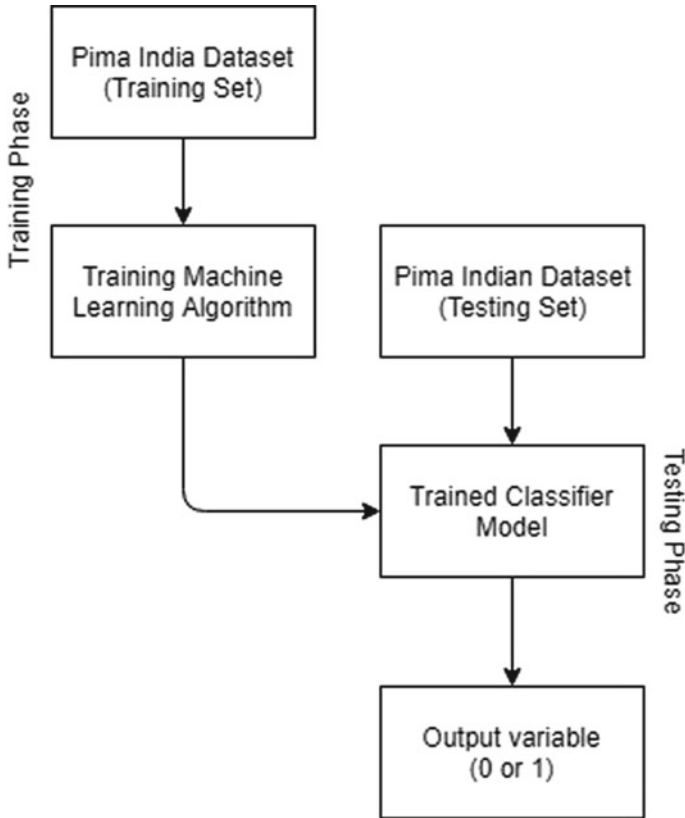


Fig. 1 Machine learning training and testing phase for Pima Indian dataset

2.1 Dataset Description

We chose a benchmark dataset, Pima Indian diabetes database, for diagnosing diabetes from Kaggle [13], which is an online machine learning repository. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. This dataset consists of 769 entries of females at least 21 years old.

There are eight predictors in the database and one binary outcome class to be predicted. These are described as follows:

- Number of time pregnant: Number of times pregnant
- Glucose concentration: Plasma glucose concentration 2 h in an oral glucose tolerance test
- Diastolic blood pressure measured in mm Hg
- Skin thickness: Triceps skin fold thickness measured in mm
- Insulin: 2 h serum insulin measured in μ U/ml

- Body mass index: Body mass index (weight in kg/(height in m)²) measured in kg m⁻²
- Diabetes pedigree function: Diabetes pedigree function
- Age: Age in years
- Outcome: Binary class variable (0 or 1).

2.2 Classification Techniques

- A. *Artificial Neural Network (ANN)*: Artificial neural network consists of multiple processing layers to represent data with multiple layers of abstraction [14].
- B. *Decision Tree*: Decision tree algorithms are commonly based on a divide and conquer technique. The main motive of these algorithms is to divide the training set T into multiple subsets so that each subset belongs to the single target set. Decision trees are constructed by partitioning the numerical attributes. Researchers from various fields create a decision tree based on observational data [15].
- C. *Random Forest*: Random forest algorithm is used for classification and regression. Random forests are a combination of decision trees and values of a random vector of each tree. The random forest is sampled independently [16]. It produces two types of information predictor variable and internal structure of data.
- D. *Support Vector Machine (SVM)*: Support vector machines are based on maximum margin principle. They construct a hyperplane to separate different categories to be classified [17].
- E. *Naïve Bayes Classifier*: Naïve Bayes is a simple probabilistic model which can distinguish different objects or classes based on Bayes' theorem [18]. Naïve Bayes is a practical classifier with advanced applications like a cancer diagnosis, face recognition, and diabetes diagnosis.

3 Classifier Results

In this work, we used classifiers explained in Sect. 2.2 to conclude the results. These classifiers allow the user to validate various models for data consisting of feature space. The benchmark dataset taken from online repository was fed to respective machine learning algorithm. Then classifiers were trained on training set in various iterations until convergence is reached and give validated results in terms of accuracy on test set. The minimum accuracy achieved is 70.12% using artificial neural network and maximum accuracy achieved is 79.22% using Naïve Bayes classifier. From these classifier results, we can observe that the Naïve Bayes classifier gives the most satisfactory results in terms of accuracy.

In Table 1, average recall for each model is given. In medical diagnosis, recall is an important criterion which tells a fraction of relevant samples that have been

Table 1 Average recall corresponding to each model used for classification

Models	Average recall (%)
Artificial neural network (ANN)	70.00
Decision tree	77.00
Random forest	75.00
Support vector machine (SVM)	78.00
Naïve Bayes	79.00

correctly identified from total samples. Deep learning provides us least, while Naïve Bayes achieves most recall value.

In Table 2, f1-score is provided against each model used. f1-score is another parameter which measures classifiers accuracy. Again, the Naïve Bayes classifier is leading among other classifiers used in this work.

Figure 2 comprises an overall comparison corresponding to each classifier used in this work, which is showing that the Naïve Bayes algorithm is the best algorithm for future predictions.

The most important metric for classifier’s performance is the confusion matrix which tells us the number of samples correctly classified and misclassified. The diagonal elements of the confusion matrix are correctly classified samples, while elements other than diagonal elements are misclassified samples. Tables 3, 4, 5, 6 and 7 consist of a confusion matrix corresponding to each classifier used in this work.

Table 2 Average f1-score corresponding to each model used for classification

Models	Average f1-score (%)
Artificial neural network (ANN)	67.00
Decision tree	77.00
Random forest	76.00
Support vector machine (SVM)	78.00
Naïve Bayes	79.00

Fig. 2 Bar chart comparison of accuracy, recall, and f1-score of each classifier

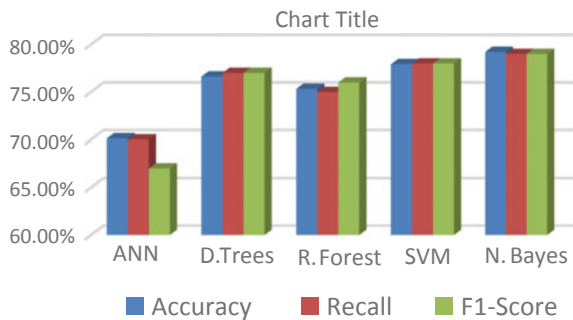


Table 3 Confusion matrix for ANN classifier with accuracy 70.12%

	True Class 0	True Class 1
Pred. Class 0	44	3
Pred. Class 1	20	10

Table 4 Confusion matrix for decision tree classifier with accuracy 76.62%

	True Class 0	True Class 1
Pred. Class 0	44	5
Pred. Class 1	18	15

Table 5 Confusion matrix for random forest classifier with accuracy 75.32%

	True Class 0	True Class 1
Pred. Class 0	48	11
Pred. Class 1	8	10

Table 6 Confusion matrix for SVM classifier with accuracy 77.92%

	True Class 0	True Class 1
Pred. Class 0	40	8
Pred. Class 1	9	20

Table 7 Confusion matrix for Naïve Bayes classifier with accuracy 79.22%

	True Class 0	True Class 1
Pred. Class 0	41	5
Pred. Class 1	11	20

4 Conclusions

This work comprises various machine learning algorithm trained and tested on Pima Indian dataset taken from the online repository and observing various performance metrics. In classification accuracy, Naïve Bayes classifier performs best for classifying diabetes data. Naïve Bayes achieves 79.22% accuracy whereas artificial neural network achieves least accuracy of 70.12%. If all parameters are considered, then the Naïve Bayes algorithm is performing the best. So, based on these parameters we can easily explore the best training algorithm required for diagnosing diabetes.

References

1. E. Dogantekin, A. Dogantekin, D. Avci, L. Avci, An intelligent diagnosis system for diabetes on linear discriminant analysis and adaptive network based fuzzy inference system: LDA-ANFIS. Digit. Signal Process. (2010)

2. K. Polat, S. Güneş, A. Arslan, A cascade learning system for classification of diabetes disease: generalized discriminant analysis and least square support vector machine. *Expert Syst. Appl.* (2008)
3. S.N. Ghazavi, T.W. Liao, Medical data mining by fuzzy modeling with selected features. *Artif. Intell. Med.* (2008)
4. M.F. Ganji, M.S. Abadeh, A fuzzy classification system based on ant colony optimization for diabetes disease diagnosis. *Expert Syst. Appl.* (2011)
5. K. Kayaer, T. Yildirim, Medical diagnosis on pima indian diabetes using general regression neural networks, in *International Conference on Artificial Neural Networks and Neural Information Processing* (2003)
6. O. ErKaymaz, M. Ozer, M. Perc, Performance of small-world feedforward neural networks for the diagnosis of diabetes *Appl. Math. Comput.* (2017)
7. I. Fakhruzi, An artificial neural network with bagging to address imbalance datasets on clinical prediction, in *2018 International Conference on Information and Communications Technology, ICOIACT 2018* (2018)
8. D. Choubey, S. Paul, S. Kumar, S. Kumar, Classification of Pima indian diabetes dataset using naive bayes with genetic algorithm as an attribute selection. *Commun. Comput. Syst.* (2016)
9. S. Wei, X. Zhao, C. Miao, A comprehensive exploration to the machine learning techniques for diabetes identification, in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, (2018), pp. 291–295
10. S.A. Saji, K. Balachandran, Performance analysis of training algorithms of multilayer perceptrons in diabetes prediction, in *Conference Proceeding—2015 International Conference on Advances in Computer Engineering and Applications, ICACEA 2015* (2015), pp. 201–206
11. S.K. Mohapatra, J.K. Swain, M.N. Mohanty, Detection of diabetes using multilayer perceptron, in *International Conference on Intelligent Computing and Applications* (2019), pp. 109–116
12. D.K. Choubey, S. Paul, K. Bala, M. Kumar, U.P. Singh, Implementation of a hybrid classification method for diabetes, in *Intelligent Innovations in Multimedia Data Engineering and Management*. IGI Global (2019), pp. 201–240
13. <https://www.kaggle.com/uciml/pima-indians-diabetes>.
14. J.S. Tiruan, Artificial neural network. *Neuron* (2011)
15. L. Rokach, O. Maimon, Decision tree, in *Data Mining Knowledge Discovery Handbook* (2005), pp. 165–192
16. University of California, L. Breiman, Random forest. *Mach. Learn.* **45**(5), 1–35 (1999)
17. C. Cortes, V. Vapnik, Support vector machine. *Mach. Learn.* (1995)
18. Naive Bayes classifier The naive Bayes probabilistic model

Multimedia Tampering Detection: A Comprehensive Review of Available Techniques and Solutions



Gautami Tripathi, Mohd Abdul Ahad and Zeeshan Ali Haq

Abstract The popularity of internet usage has resulted in the exponential growth of data in one form or the other. The exhaustive use of massive amount of artistic and communicative multimedia for several purposes has resulted in a need to develop tools and techniques to handle such multimedia in a secure and privacy-preserved manner. The legitimacy, ownership and authentication of such datasets are important as it can have a significant effect on the lives of human beings and organizations as a whole. With the advancement of technology, there exists an ocean of freely available software that can be used to create fake and tampered data without being detected by the normal visual perceptions resulting in dangerous consequences. Today, issues like copyright, ownership and legitimacy are imperative when we deal with multimedia data. This has made multimedia tampering detection an important area for research. Many techniques are being developed to identify different types of tampering in multimedia. This paper aims to present the state-of-the-art about multimedia tampering detection tools and techniques. Furthermore, it provides a comprehensive survey of the current researches in the area of multimedia tampering detection and related technologies.

Keywords Multimedia · Tampering · Authentication · Legitimacy · Data privacy · Image processing

G. Tripathi (✉) · M. Abdul Ahad · Z. Ali Haq
Department of Computer Science and Engineering, School of Engineering Sciences and
Technology, Jamia Hamdard, New Delhi 110062, India
e-mail: gautami1489@gmail.com

M. Abdul Ahad
e-mail: itsmeahad@gmail.com

Z. Ali Haq
e-mail: aliz.inam@gmail.com

© Springer Nature Singapore Pte Ltd. 2020
L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits
and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_18

1 Introduction

The ability of multimedia to capture vast amounts of information in a fast and efficient manner has made it one of the most important components of the professional information coverage. These assets are widely used in areas like journalism, law, scientific information capturing and so on. However, recent years have seen a rapid rise in cases related to fake news, falsified documents and untrue stories. The dominance of internet and other social media platforms has resulted in the near extinction of the traditional methods of communication and journalism. Today, multimedia sources are playing a major role in the information sharing domain. This far-spread reach of multimedia has raised concerns related to the authenticity and integrity of the media content.

With the growing popularity of digital cameras and easy access to freely available graphics editing tools, media manipulation has become very common [1–3]. Multimedia tampering is as old as the inception of photography. History reveals several notable instances of photo manipulation some of which raises interesting ethical questions. Figure 1 shows the historic picture of the US President “Abraham Lincoln”, which is a classic example of image tampering in history. Although it is one of the most famous portraits of Lincoln, he never posed for it. Rather it is a combination of “Lincoln’s” head and “John Calhoun’s” body [1].

Today, multimedia messages have become an effective and preferred means of communication. With the widespread use of user-friendly multimedia editing tools

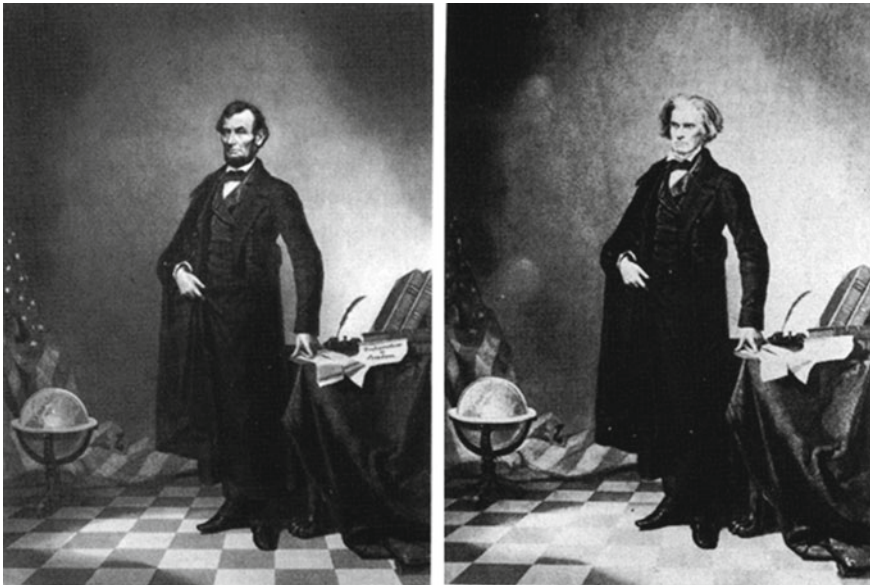


Fig. 1 Tampered photo of US President Abraham Lincoln [1]

and digital devices, multimedia tampering is not limited to the sphere of a few domain experts; rather it has become a game that anyone can play. In the last few years, we have witnessed a huge growth in the social networking sector and online journalism resulting in rapid sharing of photos, audios and videos.

Facebook alone has around 2.23 billion monthly active users worldwide [27], to which there are an additional billions connected to WhatsApp, Instagram and other social platforms. These huge and far-reaching digital platforms have deeply penetrated into the lives of individuals and are creating a parallel virtual world. With the ease of availability of such tools, the spreading of false information is easier than ever. The real challenge lies in generating evidence to prove them wrong. Today, algorithms are technologically advanced and capable of forging the interactive identity of someone else. Technology has made it possible to create 3D facial models with just one picture [29]. Recent times have witnessed cases where the multimedia content is tampered and modified to create false perceptions and influence specific communities, organizations and individuals, thus creating a threat to communal harmony and political stability. In addition, there are reported cases where image morphing was used to defame individuals as well as organizations. With such large number of reported incidents, the legitimacy of the available multimedia content is highly doubtful.

This paper is an attempt to provide awareness about the know-how of the common types of multimedia contents tampering and to present a comprehensive overview of the prominent techniques and solutions which are used to detect and prevent the tampering of the multimedia data. In this digital era wherein every data and information is getting digitized, techniques must be developed to protect the legitimacy and privacy of the data. The current paper is organized into five sections. Section 2 presents the overview of multimedia content tampering and some of the common techniques to prevent such tampering are also discussed here. Section 3 highlights the existing state-of-the-art techniques for preventing multimedia content tampering in both spatial and frequency domain. Section 4 presents the challenges associated with tampering detection. The final section concludes and discusses about the best practices of multimedia contents prevention have been discussed. Furthermore, it also highlights the future trends in multimedia content protection.

2 Classification of Multimedia Tampering

The rapid proliferation of media houses has resulted in the exponential growth of media content in different forms like audio, video, images and so on. With a number of sources contributing toward this universe of multimedia content, its tampering has become a major issue. In some cases, multimedia content tampering is done to enhance the contents, while in others it is done to fabricate the contents with the intent to cause harm to a person, community or organization.

Multimedia tampering is defined as any change in the original media characteristics [4, 5]. Multimedia data include audio, video, images, animations, text, that is,

any mode through which an information can be recorded, played and displayed. For all these different types of media, the tampering can be categorized as: authorized and unauthorized. When we talk about tampering of videos and images we have to consider that videos are not much different from images. They are moving images only but the movement of one image after another is in a very quick succession. These images forming the video are called frames and have to be homogeneous in nature to give an impression of continuity. So while considering tampering in images, it can also be considered for tampering of a video. An image can be tampered in either of the two ways, technical touching or the creative touching. Technical touching involves manipulation of contrast, white balance, hue and changing of the pixel values, whereas the creative touching are used for enhancing the already available image and the best example are the photo shopped images, that is, red eye correction feature, sharpening of any feature in the image, reflection correction and so on. [6–8].

Although video tampering includes many features of image tampering, but it is much more complex as if not done properly the tampering becomes obvious and gets prone to easy (visual) detection. In video tampering, we have to do image tampering on all the successive images making sure that continuity is maintained. Apart from this, there are several other types of tampering that can be performed on a video file such as compression of video, changing the sampling frequency while converting the analog video to a digital video, changing the number of frames per second and changing the number of pixels in order to change the quality of the video. There also exists a type of tampering where a part of image is superimposed onto the frame, thus changing the frame structure [6–8].

Another type of multimedia in which tampering can be performed is audio. Audio is a combination of amount of disturbances created in the form of change in air pressure at source. If somehow this change in pressure can be altered, it leads to tampering of the audio file. This includes changing the frequency at which the signal is being transmitted, introduction of noise and all the parameters associated with the sound signal. Once the file is stored in a digital medium, there are other types of tampering that can be performed such as changing the bits for conversion, altering the loudness and pitch of the audio signal, harmonics involved in the audio file while transferring it from one source to another [6–8].

Today, social networking sites are flooded with such tampered multimedia messages that it becomes difficult to distinguish between the genuine and counterfeit messages. It is imperative to identify the legitimacy of the multimedia messages as otherwise it may lead to serious consequences. Figures 2 and 3 show some of the significant media characteristics that can be tampered and the different types of tampering, respectively [6–8].

There are several common techniques which are used to protect the multimedia data from getting tampered. Cryptography, steganography and watermarking are the very basic techniques to prevent simple multimedia tampering [9]. However, it is imperative to understand that only protecting the multimedia is not sufficient in today's time. Techniques must be devised to identify the kind of tampering done on the multimedia contents. Furthermore, we should also be able to identify the regions within the contents where the tampering has been done. Only then we can

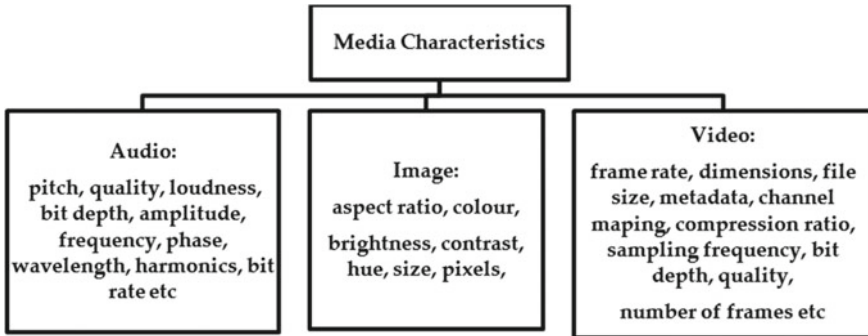


Fig. 2 Media characteristics

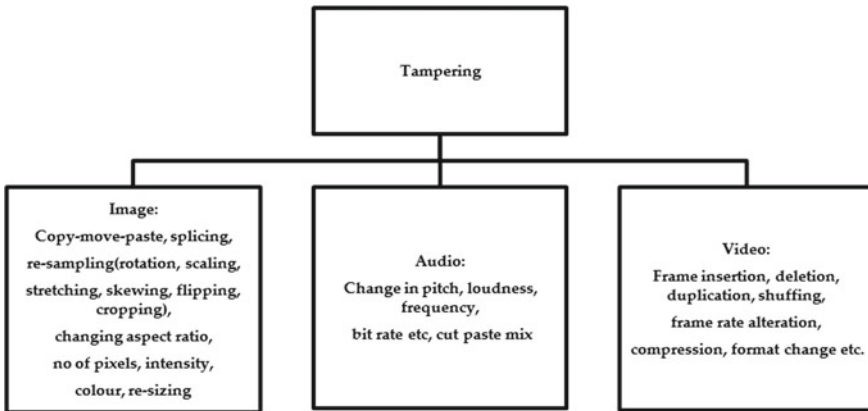


Fig. 3 Types of tampering

have a tampered proof, secured and privacy-preserved communication and transfer of multimedia data.

3 Existing State-of-the-Art Techniques in Multimedia Tampering

There are several state-of-the-art techniques proposed by the researchers in recent years to counter tampering of the multimedia contents. Some of the recent proposals are reviewed here. The existing techniques are very expensive in terms of computations required while detecting the tampering with the multimedia. The main techniques for multimedia tampering detection make use of watermarking, steganography and cryptography. In [10] watermarking technique is used for the reconstruction of

altered video. Video is considered as an “erasure communication channel” which is used to transmit the watermark consisting of compressed key frames and localization data. “Fountain coding” is used to embed the “watermark payload” in the video. Apart from tampering localization, the original video content that has been maliciously attacked can also be recovered. The decoder detects the spatial and temporal modifications using an intra and inter-frame tampering detection module, respectively. This method is capable of reconstructing the original key frames with decent quality and localizes the tampering for a tampering rate variation from 10 to 67%. In [11] a technique for tampering detection is proposed using “stationary wavelet transform (SWT)” and “discrete cosine transform (DCT)” based on copy-move forgery and localization of such tampering. The purpose of incorporating SWT in this technique is that SWT provides translational invariance and localization in both the spatial as well as spectral domain. The purpose of using DCT over the SWT is that it provides robustness to the post-processing steps, like scaling, compression and blurring after tampering has been detected. The technique disintegrates the multimedia into four regions using SWT. Approximation region is then subdivided into overlapping blocks for extracting the features. Now on each overlapping block, DCT is applied to obtain the reduced dimension of the feature. The usefulness of this technique is monitored through two metrics: the “true detection rate” and the “false detection rate”. In [12] the author presents a novel watermarking technique for detecting tampering and localization. It uses the least significant bit method to embed the watermark in the spatial domain. Watermarking involves an embedding and extraction process where the watermark extraction process is the reverse of embedding. The proposed scheme considers the size of cover image as $N \times N$. For the embedding process, Arnold cat map is applied on the cover image to obtain the scrambled image and an XOR operation is performed between six MSBs of the scrambled image to obtain a binary scrambled image. After that, logistic map is used to generate a chaotic sequence of same length $N \times N$. Binary chaotic watermark is calculated which is inserted in one of the two LSBs of the scrambled image. The watermark is obtained by applying the “Arnold cat map” (T-K) times on modified image. The “mean square error (MSE)”, “peak to signal noise ratio (PSNR)”, “structural similarity index (SSIM)”, and so on were used as the performance metrics for evaluating the proposed scheme.

The authors of [13] present the current overview of the digital audio forensics. For the sake of discussions, the audio file analysis is categorized into container and content-based analysis. The paper also discusses the history and examples of audio forensics investigations. The survey also focuses on the future directions and open research challenges, including the forensics audio enhancement, acoustic environment identification, microphone forensics, transcoding identification and codec identification.

In [14] a robust technique based on discrete SWT and multidimensional scaling is employed in detecting the copy-move forgery and image tampering. The experimental results establish that the computational complexity has been reduced many folds, and the accuracy in determining between the tampered and not-tampered image is as high as 97%. In the presented algorithm, the image is pre-processed using the luminance component of the image.

The work presented in [15] detects tampering and localization using a “multi-granularity geometrically robust” video hashing method. First, the video is divided into frame sequences and a sub-band LLL is obtained for PCET moment calculation by performing a 3D-DWT on each frame. Each frame is further divided for performing local PCET moment calculation and block-level hash formation. Furthermore, the positions of metadata of prominent objects are used as “pixel level hash” for fine “tampering localization”. Final video hash is formed using the hashes of all three granularities. The results show high accuracy and efficiency in cases even when the video is rotated.

The authors of [16] presented a convolution neural network (CNN)-based forensic approach based on deep learning. A new constrained convolution layer is developed to suppress the contents of an image and learn the features for manipulation detection from data directly. The experimental result shows up to 99.97% accuracy, outperforming the existing general purpose manipulation detectors. The proposed CNN also works in scenarios where a source mismatch exists between the training and testing data. The existing techniques are very expensive in terms of computations required while detecting the tampering with the multimedia. In [17] a simple and efficient algorithm is presented which detects the spliced regions of the image or video. A simple statistical-based approach is exploited in which each fragment of the multimedia is inspected along with the neighboring fragments. Using the information received, the algorithm classifies the fragments into tampered region or not-tampered regions. With the help of other analyzing tools such as histogram analysis, true positive score, computational performance, better results were obtained while detecting the tampered fragments for splicing. In [18] audio tampering is being detected with the help of electrical network frequency signal present in the tampered audio file. When tampering has been attempted, abrupt phase discontinuities take place that lead to abnormalities in the audio. The technique used herewith is broadly categorized into two phases. In the first phase, “ESPRIT-Hilbert ENF” estimator is used. In the next phase, “support vector machine (SVM)” is used to detect the tampering. The entire process is phased out in three bands: “pre-processing”, “feature extraction” and “classification”. In pre-processing stage, we first down sample the signal and then estimate the ENF which is assumed to be in vicinity of the SNR value. As the SNR value deteriorates, the ENF value also downgrades. This ENF estimation forms a forward feedback loop which helps in feature extraction which forms the second phase of the proposed scheme. The feature extraction is performed using the proposed ESPRIT-Hilbert feature extraction scheme. Three estimators for ENF estimation, namely HEE and 3E which provide the algorithm with two different estimates are used. From these estimates for extracting the features, “power spectral density (PSD)” is used, which provides the frequency spectrum of the estimates. For classifying the audio to be tampered, SVM has been trained over a training set on equal number of edited and unedited tests.

A novel technique based on passive image tampering technique is used in [19]. This technique is based on “local binary pattern (LBP)” and DCT. It is used for tampering based on copy-move and splicing. This technique is subdivided into two bands. In the first band, 2D DCT is applied on the LBP space. This is done from the

chrominance component of the input image. DCT helps in discriminating between localized spaces. The tampering traces are modeled. First, the image is divided into “overlapping blocks” based on its chrominance value maintaining 50% overlapping. On these overlapping images, LBP operator is applied which highlights the tampering present based on the edges of the pasted regions and its sharp edges. These LBP codes are now transferred into frequency domain using 2D DCT. In the second sub-band, SVM is used to classify the multimedia as tampered or not based on the training and testing sets. In this paper the proposed technique was evaluated over three different image datasets, that is, “CASIA TIDE v 1.0”, “CASIA TIDE v 2.0” and “Columbia dataset” and accuracies of 97, 97.5 and 97.77% were reported.

The authors of [20] perform watermarking methods in determining the tampering on the image and recovering of the tampered image. Here two images are generated from the original image based on lifting wavelet technique and half-toning technique. This provides a 2×2 metrics of overlapping blocks providing two chances of recovering the tampered image. The proposed methodology is basically divided into three phases; generation and embedding of water mark; tamper detection and authentication; and recovery and reconstruction. In generation and embedding of watermark, image is divided into two images based on lifting wavelet technique and half-toning technique. From these image bits, authentication bits are obtained. Now based on watermarked images received, the second phase detects if the image is tampered with or not. In this phase, the image is separated into 2×2 non-overlapping blocks and all the blocks are marked as valid. In [21] the author presented a tamper detection technique using block truncation coding (BTC) to ensure the integrity of images or other multimedia while in compressed form. In the proposed scheme, the size of the authentication data is user specific. The authentication data is embedded in the encryption of the media and embedded into it. Thus, the author detects the tampered portions of the BTC compressed images.

To reinforce the image details, before tampering, high-pass filtering and histogram equalization are used during the pre-processing phase in [22]. This reinforcement can achieve high accuracy while determining if the image has been tampered with or not. A metric is derived consisting of peak values of histogram derivatives after the pre-processing phase. Also in this phase, histogram illumination is estimated. Suppose a tampering is done with the image, the tampered portions of the image will be illuminated differently as compared to the rest of the image histogram illumination. To classify the images between tampered or not-tampered, probability density function (PDF) is used. The authors in [23] proposed two AVI video watermarking schemes, namely VW16E and VW8F for getting effective imperceptibility and tamper detection. The empirical evaluation proves the effectiveness of the proposals as claimed by the authors. In [24], the authors proposed a “multi-scale image hashing” method by adopting the location-context information by extracting features of the images. Afterward, the hash was appended with the image. In order to distinguish between tampered and non-tampered image, global and color hash component-based authentication was performed. Once the tampered image is identified, the exact regions within the image that are tampered are identified using multi-scale hash component. The researchers in [25] presented a survey on passive video tampering detection methods

addressing detection of “double or multiple compressed” videos, “region tampering detection” and “video inter-frame forgery” detection. Furthermore, they identified the open research challenges in passive video tampering. In [26], the authors proposed a methodology to detect tampering in the foreground or background image. The approach made use of watermarking and alpha matte. In their approach, image watermarking based on the DWT, DCT and singular value decomposition was used. Table 1 presents a tabular view of the state-of-the-art about multimedia tampering detection approaches.

Table 1 Multimedia tampering detection approaches

S. no.	Authors	Technique(s)	Remarks/Observations
1	“Amanipour, V., and Ghaemmaghami, S. (2018)” [10]	Watermarking technique for reconstruction of tampered videos	Works for detecting video tampering. Decent results obtained along with the ability to recover original video frame
2	“Mahmood, T., Mehmood, Z., Shah, M., and Saba, T. (2018)” [11]	SWT and DCT-based image tampering detection is proposed. Works by dividing images into four regions	Able to detect and resolve copy-move forgery
3	“Fatema, M., Maheshkar, V., Maheshkar, S., and Agarwal, G. (2018)” [12]	Watermarking technique is used to detect tampering. LSB is used as a region to embed the watermark on the source image	MSE, PSNR, SSIM are used to evaluate the effectiveness of the proposal and the results achieved seem promising
4	“Thirunavukkarasu, V., Kumar, J. S., Chae, G. S., and Kishorkumar, J. (2018)” [14]	DSWT and multidimensional scaling approach is used to detect image tampering	Special focus on copy-move forgery detection in images
5	“Chen, H., Wo, Y., and Han, G. (2018)” [15]	Multi-granularity geometrically robust hashing method is used to detect video tampering	Works only for detecting video tampering. Claimed to provide fair amount of accuracy
6	“Bayar, B., and Stamm, M. C. (2018)” [16]	CNN and deep learning-based approach is used for detecting image tampering	Provide very high accuracy
7	“Reis, P. M. G. I., da Costa, J. P. C. L., Miranda, R. K., and Del Galdo, G. (2017)” [18]	Audio tampering detecting technique is proposed using SVM and ESPRIT-Hilbert ENF	Requires a lot of pre-processing and thus is comparatively slower than other counterparts
8	“Arab, F., Abdullah, S. M., Hashim, S. Z. M., Manaf, A. A., and Zamani, M. (2016)” [23]	VW16E and VW8F watermarking schemes are proposed for detecting tampering of AVI videos	The approach is limited to AVI video formats only. However, the empirical evaluations show decent results

4 Challenges in Multimedia Tampering Detection

The diffusion of different forms of media with the fast growing forms of communication has reshaped the social, personal and professional life of people. In the last decade the digital communication technologies along with the volumes of multimedia content generated has controlled the way people think, behave, communicate and socialize. Despite of the numerous advantages of the multimedia communications, these advancements carry with them several risks that harm the society at large. The increasing reports of multimedia tampering and media misuse had resulted in the development of tools and techniques to detect and curb the negative effects of multimedia tampering. Many such techniques and related state-of-the-art researches are already discussed in Sect. 3. Regardless of the ongoing researches, there are still many challenges that need to be catered. Some of the challenges that multimedia tampering detection face today are given below [27–31, 34–36].

4.1 Outdated Forensics Tools and Techniques

The brisk pace at which the modern-day technology is evolving has posed real challenges for the multimedia tampering detection tools and techniques.

4.2 Heterogeneous Media Content

Today, media is being generated at a very fast pace. The available media is in different formats like text, images, audio, video, graphics, animations and so on. With such diverse media in the market it becomes a challenge to develop a single tool that can detect forgery in all media formats.

4.3 User Friendliness

Most of the multimedia tampering detection techniques and available tools in the market require some form of technical knowledge related to the media formats. It is still a challenge to develop techniques and tools that can provide easy solutions to non-expert users and can rescue the common man from the glitches of tampered multimedia.

4.4 General Awareness

The user base of social media sites is growing at an exponential rate. Today, billions of people across the globe are using social networking for their day-to-day communications that makes it possible to spread a message in a matter of seconds. This virtue of social networks can be misused to spread hatred and fake news. Recently, a popular messaging site WhatsApp has taken measures to check the spreading of fake news by disallowing the broadcast of messages to more than five people [36]. Yet, there have been reports of the usage of proxy sites to bypass this check. With such large population at the risk of being misguided, it becomes a need of the hour and a challenge to create general awareness among the general public about the consequences of multimedia tampering and how to safeguard from it.

4.5 Lack of Adequate Laws: Rules and Regulations

There are several cases of multimedia tampering reported around the globe, however, most of the culprits walk freely because of the absence of stringent rules and regulations in this regard.

5 Conclusion and Discussions

With the proliferation of the freely available and easy to use tools for media editing, multimedia tampering has become a child's play. Today, a huge quantity of vulnerable multimedia content is available online which needs to be protected. The authenticity and integrity of this content is always under question. Today, there are many online tools available to ensure the authenticity of the media content. JPEGsnoop [31] and Ghro [32] are some of the open source tools available for checking the authenticity of the multimedia content. Other tools available online include forensically [30], fotoforensics [33] and so on. These tools can help the naive users to validate the authenticity of the multimedia content before believing it completely. This paper presents an in-depth study of the researches in the area of multimedia tampering detection in the past few years and gives an overview of the various media characteristics and how these characteristics can be altered. First, we have pointed out the historical evidences of media forgery and the emerging threats due to the freely available editing tools. The paper further discusses the different categories of media tampering and the available techniques to detect them. It is imperative to understand the intricate characteristics of multimedia data and devise tools and techniques for preventing the tampering and illegal and unethical alteration of these multimedia contents. The techniques and proposals discussed in the paper apply one or the other

mechanisms to detect unlawful tampering of the multimedia data. Finally, the paper discusses the various challenges in multimedia tampering detection.

Most of the data content we have at our disposal today consists of multimedia in one or the other form. Therefore, it is indeed very crucial to come up with techniques to prevent and detect the tampering of such datasets which would otherwise have catastrophic aftereffects. Also, it is evident from the reviews that there is no universal technique available to detect all forms of multimedia tampering, and with such rapid increase in the generation of multimedia content, it becomes vital to come up with novel mechanisms to cater all forms of multimedia tampering.

References

1. <http://www.dailymail.co.uk/news/article-2107109/Iconic-Abraham-Lincoln-portrait-revealed-TWO-pictures-stitched-together.html>
2. T. Liu, Z.D. Qiu, The survey of digital watermarking-based image authentication techniques, in *2002 6th International Conference on Signal Processing*, vol. 2 (IEEE, 2002), pp. 1556–1559
3. P. Yin, H.H. Yu, Classification of video tampering methods and countermeasures using digital watermarking, in *Multimedia Systems and Applications IV*, vol. 4518 (International Society for Optics and Photonics, 2001), pp. 239–247
4. H.H. Yu, D. Kundur, C.-Y. Lin, Spies, Thieves, and Lies: The Battle for Multimedia in the Digital Era, Visions and Views Editor: Nevenka Dimitrova Phillips Research, 1070-986X/01/\$10.00 © 2001 IEEE
5. MMGD0101, Introduction To Multimedia Chap. 2 Multimedia Systems, <http://www.fms.edu.my/images/Document/MMGD0101%20-%20Introduction%20to%20Multimedia/MMGD0101%20chapter%202.pdf>
6. Digital Image Characteristics, <http://www.sprawls.org/resources/DICHAR/index.htm>
7. Significant Characteristics of Audio Files https://wiki.archivematica.org/Significant_characteristics_of_audio_files
8. Video, <https://en.wikipedia.org/wiki/Video>
9. G. Voyatzis, I. Pitas, The use of watermarks in the protection of digital multimedia products. *Proc. IEEE* **87**(7), 1197–1207 (1999)
10. V. Amanipour, S. Ghaemmaghami, Video-Tampering Detection and Content Reconstruction via Self-Embedding. *IEEE Trans. Instrum. Meas.* **67**(3), 505–515 (2018)
11. T. Mahmood, Z. Mehmood, M. Shah, T. Saba, A robust technique for copy-move forgery detection and localization in digital images via stationary wavelet and discrete cosine transform. *J. Vis. Commun. Image Represent.* **53**, 202–214 (2018)
12. M. Fatema, V. Maheshkar, S. Maheshkar, G. Agarwal, Tamper detection using fragile image watermarking based on chaotic system. in *International Conference on Wireless Intelligent and Distributed Environment for Communication* (Springer, Cham, 2018), pp. 1–11
13. M. Zakariah, M.K. Khan, H. Malik, Digital multimedia audio forensics: past, present and future. *Multimed. Tools Appl.* **77**(1), 1009–1040 (2018)
14. V. Thirunavukkarasu, J.S. Kumar, G.S. Chae, J. Kishorkumar, Non-intrusive forensic detection method using DSWT with reduced feature set for copy-move image tampering. *Wirel. Pers. Commun.* **98**(4), 3039–3057 (2018)
15. H. Chen, Y. Wo, G. Han, Multi-granularity geometrically robust video hashing for tampering detection. *Multimed. Tools Appl.* **77**(5), 5303–5321 (2018)
16. B. Bayar, M.C. Stamm, Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection. *IEEE Trans. Inform. Forensics Secur.* (2018)

17. A.M. Sundaram, C. Nandini, ASRD: algorithm for spliced region detection in digital image forensics, in *Computer Science On-line Conference* (Springer, Cham, 2017), pp. 87–95
18. P.M.G.I. Reis, J.P.C.L. da Costa, R.K. Miranda, G. Del Galdo, ESPRIT-hilbert-based audio tampering detection With SVM classifier for forensic analysis via electrical network frequency. *IEEE Trans. Inform. Forensics Secur.* **12**(4), 853–864 (2017)
19. A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, G. Bebis, H. Mathkour, Passive detection of image forgery using DCT and local binary pattern. *Signal, Image Video Process.* **11**(1), 81–88 (2017)
20. B.B. Haghghi, A.H. Taherinia, A. Harati, TRLH: fragile and blind dual watermarking for image tamper detection and self-recovery based on lifting wavelet transform and halftoning technique. *J. Vis. Commun. Image Represent.* (2017)
21. Y.C. Hu, K.K.R. Choo, W.L. Chen, Tamper detection and image recovery for BTC-compressed images. *Multimed. Tools Appl.* **76**(14), 15435–15463 (2017)
22. A. Baomy, M. Abdalla, N.F. Soiliman, F.E.A. El-Samie, Efficient implementation of pre-processing techniques for image forgery detection, in *Japan-Africa Conference on Electronics, Communications and Computers (JAC-ECC)* (IEEE, 2017), pp 53–56
23. F. Arab, S.M. Abdullah, S.Z.M. Hashim, A.A. Manaf, M. Zamani, A robust video watermarking technique for the tamper detection of surveillance systems. *Multimed. Tools Appl.* **75**(18), 10855–10885 (2016)
24. C.P. Yan, C.M. Pun, X.C. Yuan, Multi-scale image hashing using adaptive local feature extraction for robust tampering detection. *Signal Process.* **121**, 1–16 (2016)
25. K. Sitara, B.M. Mehtre, Digital video tampering detection: an overview of passive techniques. *Digit. Investig.* **18**, 8–22 (2016)
26. W.C. Hu, W.H. Chen, D.Y. Huang, C.Y. Yang, Effective image forgery detection of tampered foreground or background image based on image watermarking and alpha mattes. *Multimed. Tools Appl.* **75**(6), 3495–3516 (2016)
27. <https://eastwest.eu/en/inside-news/social-media-tampering>
28. <http://papers.nips.cc/paper/6672-unsupervised-image-to-image-translation-networks.pdf>
29. A.S. Jackson, A. Bulat, V. Argyriou, G. Tzimiropoulos, Large pose 3D face reconstruction from a single image via direct volumetric CNN regression, in *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 1031–1039
30. <https://29a.ch/2015/08/16/forensically-photo-forensics-for-the-web>
31. <https://sourceforge.net/projects/jpegsnoop/>
32. <https://www.getghiro.org/>
33. <http://fotoforensics.com/>
34. M. Barni, M.C. Stamm, B. Tondi, Adversarial multimedia forensics: overview and challenges ahead, in *2018 26th European Signal Processing Conference (EUSIPCO)* (Rome, 2018), pp. 962–966
35. K.N. Sowmya, H.R. Chennamma, A survey on video forgery detection. *Int. J. Comput. Eng. Appl.* **9**(2), 17–27 (2015)
36. <https://www.businesstoday.in/technology/news/whatsapp-promises-to-check-fake-news-before-2019-without-snooping-on-users/story/317896.html>

Optimising Fractal Encoding Using Cuttlefish Optimisation Algorithm



Smriti Sehgal, Laxmi Ahuja and M. Hima Bindu

Abstract Increase in demand for better appearance and less storage requirement of an image has led to explore various image compression techniques. Due to technological advancement of photo capturing devices such as single-lens reflex camera (SLR), digital SLR, smart phone cameras, and satellite sensors, more detailed information can be recorded in a single image. A coloured image captured by high wavelength sensors produces large-sized image as it contains highly correlated data. Many image compression and analysis techniques have been developed to aid the interpretation of images and to compress as much information as possible in it. The goal of image compression is to recreate original image with less number of bits and minimal data loss. For generating computer graphic images and compression of objects, it has been suggested that by storing images in the form of transformation instead of pixels lead to compression and can be achieved through fractal coding. In fractal image compression, encoding image blocks into fractal codes using iterated function system (IFS) takes large amount of time taken to compress it. A study of various meta-heuristics techniques, which are designed to solve complex problems approximately, has been conducted to improve upon computational time of fractal coding as well as compression ratio, while maintaining image visually. In this paper, using the property of pattern adaption of surroundings, cuttlefish optimisation algorithm is applied to minimise the time taken for fractal coding. Compression results have been compared with other meta-heuristic techniques, such as particle swarm optimisation and genetic algorithm, and has shown high compression ratio of approximately 31%.

S. Sehgal (✉)

Amity School of Engineering and Technology, Amity University, Noida, UP, India
e-mail: smriti1486@gmail.com

L. Ahuja

Amity Institute of Information Technology, Amity University, Noida, UP, India
e-mail: lahuja@amity.edu

M. Hima Bindu

North Orissa University, Baripada, Odisha, India
e-mail: profhbnou2012@gmail.com

© Springer Nature Singapore Pte Ltd. 2020

L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_19

237

Keywords Affine transformation · Cuttlefish algorithm · Fractal image compression · Optimisation · Meta-heuristics

1 Introduction

Image compression technology has gained quite a lot of importance in information storage, network transmission, military, satellite navigation, meteorology, education, medical treatment and other industries. It converts an image to be represented using less number of bits with minimal loss of information. The basic idea used in compression is to reduce the data redundancies [1], such as coding, inter-pixel and psycho-visual redundancies. In coding redundancies, repeated long words are used in the file and can be reduced by replacing long words with short similar phrases. In inter-pixel redundancies, neighbouring pixels are not independent and are highly correlated. It can be overcome by sending the difference between successive pixels. Psycho-visual redundancy is caused due to ignorance from human visual system and its elimination can lead to loss of qualitative information. Image compression is explained using the model as in Fig. 1.

Image to be compressed passes through two distinct functional components [1], one is encoder and the other is decoder. Encoding process is the main step in image compression which does the actual reduction of data and in turn uncorrelate data through mapper, quantiser separates the irrelevant data from compression and symbol coder assigns special codes/characters to original data. Output of encoding is compressed data which is helpful in transmission or storage. Decoding step is inverse of encoding and it decodes the data and undo the mapping step to recreate original image.

Compression of an image [2, 3, 4] is broadly classified into two types of techniques: lossy and lossless compression techniques. If both the functions, that is, encoder and decoder, are equal, the system is lossy, otherwise lossless. Lossy techniques construct the new image with some loss of information and high compression ratio. Output image using lossless compression techniques is almost the same as input image with negligible loss of information. But compression ratio is compromised in later case. Various lossless techniques [1, 2, 3] exist such as run length coding,

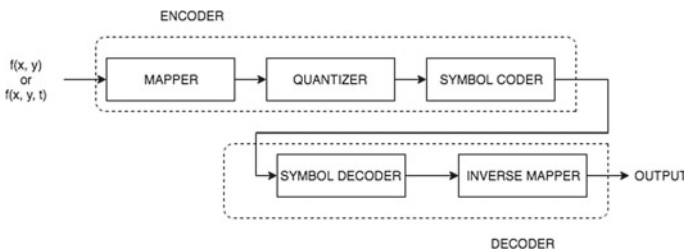


Fig. 1 Basic image compression model [1]

Huffman coding, Lempel–Ziv–Welch (LZW) compression, and lossy compression techniques [1, 2, 3] include transformation coding, vector quantisation, fractal coding and many more. One of the most commonly used lossy techniques is JPEG [3] that gives good compression ratio for small-sized images. As the size of image increases, compression ratio and quality of image decreases. To get good compression ratio for large-sized images, it was proposed in [5] that image is divided into blocks and stored in the form of transform coefficients. Storing image in the form of coefficients compresses it and is known as fractal image compression (FIC). As the original image is converted into encoded image, on application of repeated affine transformations, computation time for encoding increases. Increased execution time needs to be minimised for effective transmission.

Optimisation techniques are generally used to minimise or maximise the function to obtain approximate solution. These techniques are divided into two categories as deterministic and heuristic. The major disadvantage of deterministic techniques is that, it cannot be applied to nonlinear problems; which is overcome by heuristic approach. Under the category of heuristics approach, evolutionary algorithms [6, 7] such as genetic algorithm, particle swarm optimisation and differential evolution [8] start with random solution and search within it for best solution. New population is made after each generation depending upon current best solution. Another category is heuristic approach [9]; sometimes known as nature-inspired algorithms, these are based on simulating search strategy from nature. These have the ability to give local or global near-optimal solution to vast domain problems without in-depth knowledge. Over the last few decades, many meta-heuristic techniques have been explored and applied to different areas.

In this paper, fractal encoding in image compression is optimised through cuttlefish optimisation algorithm. Fractal image compression [10, 11] is a lossy compression technique in which the original image is converted into fractal codes using transformations. It uses the concept of self-similarity such that smaller regions of the image are similar to larger portions of the same image. In [5], Barnsley proposed that encoded image in the form of transformed codes can be stored using lesser number of bits. In fractal encoding, each range block is mapped to best domain block and requires large number of comparisons leading to high computational cost and time. Decoding in fractal image compression is much simpler and fast, as fractal codes are converted back to bitmaps. As mentioned before, optimisation of fractal encoding is done using various meta-heuristic techniques such as cuttlefish algorithm, genetic algorithm [12, 13, 14, 15] and particle swarm optimisation [14, 15]. Cuttlefish algorithm [16, 17] is based on the property of pattern changing and adaption to background either to hide in it or to show itself by emitting bright colours. It is a search strategy used to get current solution much closer to near to optimal solution. It includes two main processes: reflection and visibility. Reflection is the amount of light and colour reflected back from cuttlefish skin. Visibility represents the clarity of matching pattern and tells how far the current solution is from the best solution. Search strategy to find new solution is formulated as:

$$newp = reflection + visibility \quad (1)$$

Performance of cuttlefish algorithm on fractal encoding is compared with other meta-heuristic algorithms [9, 18]. Results show that advantages of high compression ratio are provided by fractal encoding and its execution time is optimised by cuttlefish algorithm, genetic algorithm [6] and particle swarm optimisation [6] with minimal loss. Out of these, cuttlefish algorithm provides much better results than all, keeping image quality as good as original image.

Image compression [3, 4, 19] is usually measured through compression ratio, peak signal-to-noise (PSNR) and execution time metrics. Compression ratio is ratio of original and encoded image. Higher the compression ratio, better is size reduction. PSNR is peak to signal noise ratio, depicting the ratio between signal power and noise power. It is the approximation of human perception of compressed image, that is, higher the PSNR, better is the image view. Execution time is the time in seconds/milliseconds taken to encode or decode an image.

This paper is organised in seven sections. Section 1 provided an introduction to the fractal compression and cuttlefish algorithm. Section 2 provides detailed information about fractal encoding and decoding. Section 3 describes proposed methodology; how cuttlefish algorithm is used to optimise fractal compression. Section 4 shows results obtained and comparisons made. Section 5 concludes the paper with its future aspects. References used are listed at the end of the chapter.

2 Fractal Image Compression

In fractal image compression, Barnsley [5] proposed that images can be compressed, if converted into fractal codes. Fractal compression algorithms using iterated function systems IFS or PIFS [10] give very high computational complexity. Fractal image compression consists of two main steps: fractal encoding, in which image is being converted into fractal codes and second step is fractal decoding, in which codes are converted back to bitmaps. Fractals are formed by the property of self-similarity. Encoding process takes lot of computation as it needs to form fractals which require large number of iterations. All the transformations applied must be contractive, that is, any two points in an original image must be closer in output image after affine transformations. John Hutchinson explored the basic mathematical idea [20] behind fractal image compression and was refined by Barnsley and Sloan [21], which states that application of transformations on an object reaches to a fixed point after certain number of iterations, where further operation cannot be applied.

IFS consists of sequences of affine transformations [22, 23] to depict relationship between different parts of an image. Generally, affine transformations consist of combination of rotation, scaling and translation of spatial coordinates in n-dim space. General affine transformation is denoted by w as follows:

$$w_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix} \quad (2)$$

where x, y are spatial coordinates and a, b, c, d, e, f are transformation coefficients.

Affine transformation, w , always moves any two points closer on repeated application, which makes it contractive, and distance between the two points is denoted as follows:

$$d(w(P1), w(P2)) < sd(P1, P2) \tag{3}$$

where d is a distance metric and $P1$ and $P2$ are some arbitrary points; for $s < 1$.

Collection of contractive affine transformations, w , converts the original image, S to compressed image $w(S)$ and is written as:

$$w(S) = \bigcup_i^n w_i(S_i) \tag{4}$$

In real world, we assume that each image has a unique fixed point to which each sub-image must converge into, after application of various affine transformations. In practice, it becomes difficult to find that unique point. So, we divide our image into sub-images and unique point is found out for each sub-image. This type of IFS is called partitioned IFS.

Affine transformation of PIFS applied to an image is denoted as:

$$w_i(f) = w_i(x, y, f(x, y)) \tag{5}$$

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & S_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ 0 \end{bmatrix} \tag{6}$$

where x, y = pixel coordinates; z = intensity pixel of $[x, y]$; a, b, c, d, e, f = affine transformation coefficients; S_i = contrast of transformation; O_i = brightens of transformation.

Contracted image is then represented by union of individual transforms from w_1 to w_n

$$W = \bigcup_i^N w_i \tag{7}$$

When affine transformation, w_i is applied to sub-image D_i , R_i is generated. D_i is sub-image of domain pool D and R_i is sub-image of range pool, R

$$w_i : D_i \rightarrow \hat{R} \tag{8}$$

RMS error metric is used to find error between two images f and g , of size $N*N$, as follows and must be as low as possible:

$$d_{RMS}(f, g) = \sqrt{\sum_x \sum_y^N yN} \quad (9)$$

To implement fractal encoding on image [11, 24], the image is divided into overlapping regions called domain and non-overlapping regions called range blocks. Generally, the size of domain blocks is chosen twice the size of range block. For example, if range block size is 8*8, then domain block must be 16*16. Domain blocks are best mapped onto range blocks by averaging of four neighbouring pixels into one and finding standard deviation of each block. For every range block, on application of affine transformation, fractal image format is generated, which is almost similar to domain block.

Fractal decoding [11] converts the fractal codes back to bitmap pixels by repeated applications of transformations until it reaches to an image which closely approximates original image.

Steps of fractal encoding in fractal image compression

For each range block, R_i

- Find domain block, D_i which closely approximates R_i on application of transformation
- Offset is calculated and stored as mean difference between APV of range and transformed domain
- Code of R_i is generated using position of domain block D_i , coefficients and offset.

End

Steps of fractal decoding in fractal image compression

Do

For each fractal code FC_i in transformed image

- Domain in FC_i is contracted using scale factor, s
- Decoding coefficients are applied on contracted domain
- Offset is added to each pixel and stored as range block

End

Range image is constructed by combining all range blocks

Range image is stored in domain image for further iterations

until convergence is achieved

Range image generated after convergence is reconstructed image.

3 Proposed Methodology

Encoding process in fractal image compression accounts for high computation as large number of iterations is required in domain to range block comparison and takes long execution time. Cuttlefish optimising algorithm is used to optimise the computation and execution time of encoding process.

Adel Sabry Eesa, Adnan Mohsin Abdulazeez Brifceni, Zeynep Orman [16] proposed cuttlefish optimising algorithm in 2013, which simulates the behaviour of cuttlefish to change its colour according to surroundings. It is a type of cephalopods [25, 26] known to change its pattern either to hide itself or to showcase itself more. Its texture changing property is produced by three layers of skin [26, 27]: chromatophores, iridophore and leucophores. Chromatophores is the top most layer just under the cuttlefish skin and includes an elastic saccule which stretches or skews on muscle change. It can produce red, orange, brown and yellow colours. Iridophores is the next layer consisting of platelets, responsible in producing metallic colours. Leucophore is the last layer of the skin, responsible for white spots on animal skin. It acts like a mirror, that is, it scatters and reflects the incoming light as it is. This layer helps cuttlefish to blend in its environment. Changing appearance of cuttlefish depends upon which skin layer is affected by incident light. Reflected light can come from chromatophore; iridophore or leucophore; or combination of both.

A new meta-heuristic optimisation algorithm is proposed in [16, 14] to mimic the colour changing behaviour of cuttlefish. Algorithm is summarised in Fig. 2. It divides the population into four groups depicting six different reflected light cases, as shown in Fig. 3.

There are two main processes: reflection and visibility. Reflection depicts the reflected light denoted by any one of the six cases in Fig. 3. Visibility depicts how close we have reached to patterns in environment. Reflection and visibility are used to find new solutions as formulated in Eq. 1.

Cuttlefish algorithm [16, 14] starts with random population as almost all meta-heuristic techniques do. Algorithm in Fig. 2 is explained as follows:

- Initialise the population with random solution and store current best solution.
- Population is divided into four groups: G1, G2, G3 and G4.
- Case 1 and Case 2: For global search, G1 is used. Chromatophore and iridophores are interacted to produce new solution.
- Case 3 and Case 4: For local search, G2 is used. Solution is produced by reflected light from best solution and visibility of pattern using iridophores.
- Case 5: G3 is used for local search and new solution is generated by reflected light around the best solution and visibility of pattern using leucophores.
- Case 6: G4 is used for global search and it just reflects the incoming light using leucophores.

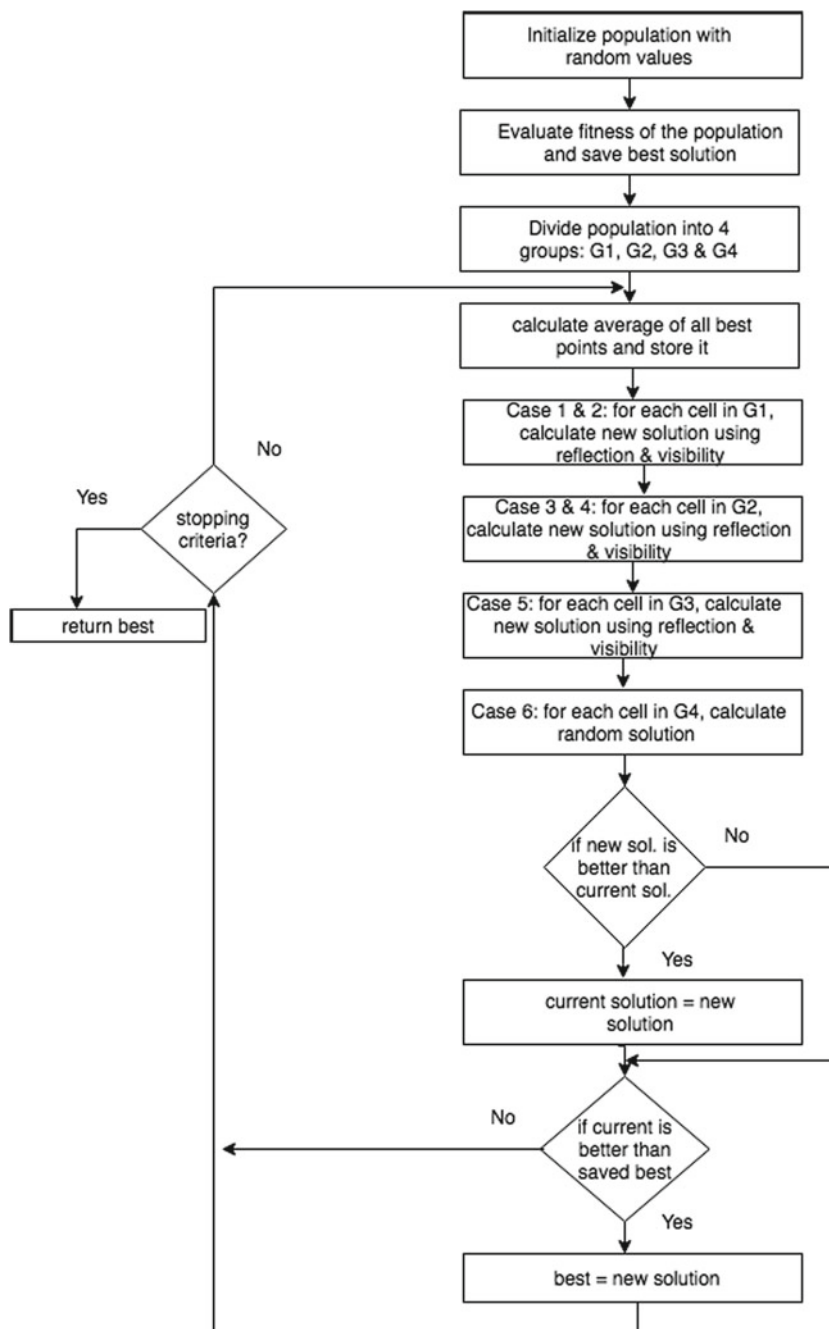
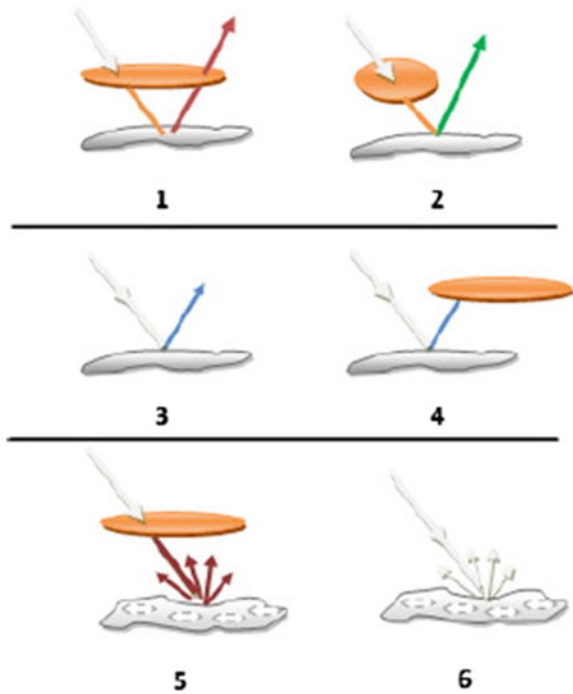


Fig. 2 Flowchart for cuttlefish optimisation algorithm

Fig. 3 Reflected light from cuttlefish skin (six cases)



3.1 Working of Case 1 and Case 2 (Group 1, Global Search)

Reflected light is generated due to interaction of chromatophores and iridophores, as shown in Fig. 3. Chromatophores will either contract or relax muscle to stretch or shrink saccule. Reflected light will pass through chromatophores or not. For group 1 to simulate case 1 and case 2, V is set to 1 and R is to be calculated. To find new solution according to Eq. 1, reflection and visibility is calculated as:

$$\begin{aligned}
 reflection_j &= R * G_1[i].Points[j] \\
 R &= random () * (r_1 - r_2) + r_2
 \end{aligned}$$

$$\begin{aligned}
 visibility_i &= V * (Best.Points[j] - G_1.Points[j]) \\
 V &= random () * (v_1 - v_2) + v_2
 \end{aligned}
 \tag{10}$$

where R = reflection degree and V = visibility factor

3.2 Working of Case 3 and Case 4 (Group 2, Local Search)

This group uses leucophores which is light reflecting layer generally used to conceal organs. It reflects the light from the environment and shows some specific colour. To find new best solution, visibility remains as in G1 but reflection is modified as Eq. 11. Here, R is set to 1 and V is calculated and new best position is generated according to Eq. 1. As it is a local search, it produces the interval around the best solution.

$$\begin{aligned} reflection_j &= R * Best.Points[j] \\ R &= 1 \end{aligned}$$

$$\begin{aligned} visibility_j &= V * (Best.Points[j] - G_1.Points[j]) \\ V &= random() * (v_1 - v_2) + v_2 \end{aligned} \quad (11)$$

3.3 Working of Case 5 (Group 3)

Group 3 simulates the working of case 5 in which leucophores acts as a mirror and reflects incoming wavelength of the light. Here, we assume that incoming colour is best solution and reflected colour would be around best. Reflection and visibility are formulated as follows keeping $R = 1$.

$$\begin{aligned} reflection_j &= R * Best.Points[j] \\ R &= 1 \end{aligned}$$

$$\begin{aligned} visibility_j &= V * (Best.Points[j] - AV_{Best}) \\ V &= random() * (v_1 - v_2) + v_2 \end{aligned} \quad (12)$$

3.4 Working of Case 6 (Group 4)

This group in population uses leucophores to reflect incoming light and try to match pattern as in surroundings.

4 Experiments and Results

Optimising image compression using fractal encoding and decoding with cuttlefish algorithm is performed on an image of size 512*512, and is implemented in MATLAB 7.0. Experiments were run on a computer with an Intel i5 processor running at 2.4 MHz using 16 GB RAM, running Microsoft Windows XP. The results of this implementation are compared with other two existing meta-heuristic techniques: one is genetic algorithm and the other is particle swarm optimisation.

Fractal image compression is applied on the image and is divided into 4*4 non-overlapping range blocks and 8*8 overlapping domain blocks. Affine transformation is applied using parameters specified in Table 1 for range-domain comparisons. Reconstructed image using fractal image compression is shown in Fig. 4. It took around 9 s to encode and approx. 1.5 s to decode.

To optimise encoding time, keeping CR and quality of an image to near original, cuttlefish optimisation algorithm is applied to enhance encoding process. Various combinations of parameter set were tested as shown in Table 2. It has been seen that choice of parameter set for r1, r2, v1 and v2, to calculate reflection degree and visibility factor, be chosen as 2, -1, 0.5 and -0.5, respectively.

As can be seen from Table 3, fractal image compression optimised with cuttlefish algorithm gives higher compression ratio with reduced encoding time and same decoding time (Fig. 5).

Table 1 Transform coefficient parameters for IFS/PIFS

	a	b	c	d	e	f
GA	0.1514	0.3863	0.2185	0.0254	0.0145	0.7911
CFA	0.3121	0.2121	0.6560	0.0555	0.0021	0.6129



Fig. 4 Original and reconstructed image using FIC

Table 2 CFA parameters

	r1	r2	v1	v2	CR
Set 1	1	- 0.5	1	- 1	31.5121
Set 2	0.4	- 0.2	1	- 1	30.2787
Set 3	1	- 0.5	0.3	- 0.3	31.4511
Set 4	2	-1	0.5	-0.5	31.7678
Set 5	0.5	- 0.2	1	- 1	31.6991

Table 3 Comparison of image compression techniques using fractals with CFA, GA and PSO

	FIC	FIC with CFA	FIC with GA	FIC with PSO
CR	31.11	31.76	30.86	29.11
Encoding time (s)	9.93	5.65	6.87	6.23
Decoding time (s)	1.46	1.51	1.48	1.23



Fig. 5 Reconstructed image using FIC and optimised with (a) cuttlefish algorithm, (b) GA and (c) PSO

5 Conclusion and Future Aspects

The need to compress image due to large image size and limited storage and transmission resources has promoted compression of image to great extent. Implementation of image compression based on fractal codes is described in this paper. We have seen the issue of high computational cost of image encoding using fractals and tried to optimise using one of the meta-heuristic techniques, that is, cuttlefish optimisation algorithm. Results show that on application of cuttlefish algorithm for fractal compression and fractal coding alone, computational time has been reduced from 9.93 to 5.65 s with an increase in CR by 1.02%, whereas fractal image compression with other meta-heuristic techniques, such as GO and PSO, gave 69 and 62.7% of reduction in coding time. In addition to this, cuttlefish algorithm in hybridisation with other meta-heuristic techniques will be explored for further optimisation.

References

1. R.C. Gonzalez, *Digital Image Processing* (Pearson Education, India, 2009)
2. S. Dhawan, A review of image compression and comparison of its algorithms. *IJECT* **2**(1) (2011)
3. B. Pancholi, R. Shah, M. Modi, Tutorial review on existing image compression techniques. *Int. J. Eng. Comput. Sci.* **3**(8), 7882–7889 (2014). ISSN:2319-7242
4. M. Devi, U. Mehta, A review on various techniques of image compression. *Int. J. Eng. Comput. Sci.* **5**(7), 17127–17129 (2016). ISSN: 2319-7242
5. M.F. Barnsley, *Fractals Everywhere*. Morgan Kaufmann Publishers (2000)
6. K. Manda et al., Population based metaheuristic techniques for solving optimization problems: a selective survey. *Int. J. Emerg. Technol. Adv. Eng.* **2**(11) (2012). ISSN 2250-2459
7. Y. Xinjie, G. Mitsuo, *Introduction to Evolutionary Algorithms* (Springer, London Limited, 2010)
8. S. Sehgal, S. Kumar, M. Hima Bindu, Remotely sensed image thresholding using OTSU and differential evolution approach, in: *2017 7th International Conference on Cloud Computing, Data Science and Engineering–Confluence*
9. P. Sindhuja, P. Ramamoorthy, M. Suresh Kumar, A brief survey on nature inspired algorithms: clever algorithms for optimization. *Asian J. Comput. Sci. Technol.* **7**(1), 27–32 (2018). ISSN: 2249-0701
10. B. Bani-Eqbal, Speeding up fractal image compression. *Proc. SPIE.* **2418**, 2/1–2/3 (1995). <https://doi.org/10.1117/12.204140>
11. D. Vidya, R. Parthasarathy, T.C. Bina, N.G. Swaroopa, Architecture for fractal image compression. *J. Syst. Archit.* **46**, 275–1291 (2000)
12. S.S. Bobde, M.V. Kulkarni, P.V. Kulkarni, Fractal image compression using genetic algorithm, in *2010 International Conference on Advances in Computer Engineering* (2010)
13. M. Omari, S. Yaichi, Image compression based on genetic algorithm optimization (IEEE, 2015). 978–1-4799-8172-4/15
14. K. Uma, P. Geetha Palanisamy, P. Geetha Poornachandran, Comparison of image compression using GA, ACO and PSO techniques, in *IEEE-International Conference on Recent Trends in Information Technology, ICRTIT* (2011)
15. G. Vahdati, H. Khodadadi, M. Yaghoobi, Fractal image compression based on spatial correlation and hybrid particle swarm optimization with genetic algorithm, in *2010 2nd International Conference on Software Technology and Engineering (ICSTE)* (2010)

16. A.S. Eesa, A.M.A. Brifcani, Z. Orman, Cuttlefish algorithm—a novel bio-inspired optimization algorithm. *Int. J. Sci. Eng. Res.* **4**(9) (2013)
17. A.S. Eesa, A.M.A Brifcani, Z. Orman, A new tool for global optimization problems- cuttlefish algorithm, in *Proceeds in International Conference on Computer Science (ICCS 2014)* (Rome, Italy, September 18–19, 2014)
18. S. Binitha, S.S. Sathya, A survey of bio inspired optimization algorithms. *Int. J. Soft Comput. Eng. (IJSCE)* **2**(2) (2012). ISSN: 2231-2307
19. M. Marcin, S. Czesãaw, Test functions for optimization needs (2005)
20. J.E. Hutchinson, Fractals and self similarity. *Indiana Univ. Math. J.* **35**(5) (1981)
21. M.F. Barnsley, A.D. Sloan, A better way to compress images. *BYTE* 215–218 (1988)
22. A.E. Jaquin, Image coding based on a fractal theory of iterated contractive image transformation. *IEEE Trans. Image Process.* **1**(1) (1992)
23. A.E Jaquin, Fractal image coding: a review. *Proc. Tile IEEE* **81**(10) (1993)
24. Y. Fisher, *Fractal Image Compression*, Siggraph 92 Course Notes
25. M.M. Lydia, J.D. Eric, T.H. Roger, N.M. Justin, Mechanisms and behavioural functions of structural coloration in cephalopods. *J. R. Soc. Interface* (2008)
26. <http://www.thecephalopodpage.org/>
27. E. Florey, Ultrastructure and function of cephalopod chromatophores. *Am. Zool.* (1969)

Chaos Theory on Generative Adversarial Networks for Encryption and Decryption of Data



Juhi Purswani, Rajesh Rajagopal, Riya Khandelwal and Anuraj Singh

Abstract Today's world involves sharing a tremendous amount of vital information and data over the web and cloud for almost everything. Any hacker or cyber-terrorist can get access to the data and hence the security of the data becomes extremely essential. Through this research, the possibilities of improving the cryptosystem has been explored by making use of generative adversarial networks in which our own shared key, which is generated with the help of chaotic generator has been incorporated. The key formed leads to the increase in randomness which in turn makes it even more difficult to crack it, thus making the system even more secure.

Keywords Cryptography · Generative adversarial networks · Neural networks · Chaotic maps · Security · Keras · Pseudo-random number generator

1 Introduction

Artificial intelligence is a field that is growing at a very fast pace. One major issue that it still faces is the security of the data shared. Datasets are an integral part of the field. Since only large companies and MNCs are privileged with these, these companies are very secretive about sharing the dataset with others.

For the growth of the company, it is genuine that it will be interested to share its data with the outsiders like data scientists so that the latter could work upon it and

J. Purswani (✉) · R. Rajagopal · R. Khandelwal · A. Singh
Atal Bihari Vajpayee Indian Institute of Information Technology and Management, Gwalior
474001, Madhya Pradesh, India
e-mail: juhi.purswani7@gmail.com

R. Rajagopal
e-mail: rajesh@iiitm.ac.in

R. Khandelwal
e-mail: riyakwl28@gmail.com

A. Singh
e-mail: anuraj@iiitm.ac.in

© Springer Nature Singapore Pte Ltd. 2020
L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_20

use it to validate the models to find a particular solution, but one thing that is still in question is how the data will be protected?

One possible solution for this problem is data anonymization, since securing or hiding the sensitive parts will lead to some security. But this also does not make the data secure due to inferences, since some part of the secured data can still be predicted with the other part that is not anonymized.

Another solution that could be thought of is with the help of homomorphic encryption. The homomorphic encryption is still new in the market but has a great potential to solve the cryptography-related problems in the field of AI. The idea behind homomorphic encryption is that the processing of the data, calculations, and so on, can be done on the encrypted texts or data itself without the need to decrypt it first as in [1]. But the technology is still very new and computationally not very efficient and not differentiable. So it is very early to think of this as a solution.

The solution that the Google Brain team came up with is with the help of generative adversarial networks as in [2]. The idea was to use neural networks to develop a cryptographic system which makes the data secure from other neural models with the help of generative adversarial neural networks. They found ways in which neural networks can learn to implement cryptography without any prior knowledge of cryptographic algorithms. The results were outstanding considering that traditionally neural networks were never thought of to be used for cryptography purposes. But still the accuracy was less, so it could not be applied in real-time scenarios.

For the security of data, encryption has also been done with the traditional cryptographic algorithms by introducing chaos in the common key to make it more random so that it cannot be broken down by the hackers [3]. The concept is called chaotic cryptography. The system proposed was for images only and cannot be applied to text or any other type of data [4].

The other solutions proposed the method of chaotic neural network for encryption for either text or image. Image encryption using chaotic neural network and text encryption through neural networks was proposed [5, 6]. But it could be applied to image or text only hence made it unsuitable for other data types like video, sound and so on. Time-delayed chaotic neural networks were also used [7] which increase the strength of encryption but there was much discrepancy in decrypted text and plain text, hence making them unsuitable to be used in real-life scenario. M. Arvandi, S. Whu and W.W. Melek also proposed the method of using recurrent neural network for data encryption [8] and their method prove to be quite robust but lack the accuracy.

To garner the problems to the above-stated issue and provide the solution, this paper proposes a framework for generative adversarial networks-based cryptographic system in which the shared key will be generated with the help of a dynamical chaotic random number generator [9, 10] to tackle the issue of accuracy. The key generated here will be more random and therefore more secure, making the system stronger in

terms of security, thus providing a way in which the data can be secured from other neural models in the real world. Also encryption and decryption will be performed on bits; hence it could be applied to any data type.

2 Related Works

Already research is going on in exploring the possibilities of implementing cryptography with the help of neural networks. All of these researches had one thing in common that is the key that was formed was from the trained weights, in which the models were then synchronized to perform encryption. This section represents key utilities achieved using generative adversarial networks and chaotic generators in the similar fields.

2.1 *Generative Adversarial Networks and Their Utilities*

A generative adversarial neural network, that is, GAN is a much more modified version of neural network having a pair of models competing with each other for the purpose of learning, analysing and capturing the dataset variations.

- GAN architecture can be well understood by keeping in mind the terminology of encryptor and decryptor. In this particular terminology the generator G can be thought of as the encryptor whose work is to encrypt the data. The discriminator D can be thought of as an expert trying to find the plain text and differentiating between the plain text formed and original text by taking in both the key and encrypted text as input. Both the models learn from each others' losses in the training phase as they are trained together. The architecture of a general GAN is shown in Fig. 1.
- The model has two components mainly:
 1. Generator: Its work is to predict the features, when labels are being fed into it.
 2. Discriminator: Its work is exactly opposite to that of generator; it predicts labels, when being fed features.
- Use of GAN for cryptography: The GANs were first used for cryptography in Google Brain research work [2]. Here the possibility of using a neural network for cryptography came into picture which traditionally was always considered a bad idea. The models were trained adversarially to learn encryption and decryption techniques; no particular algorithm was used for the purpose. The neural networks learned themselves.

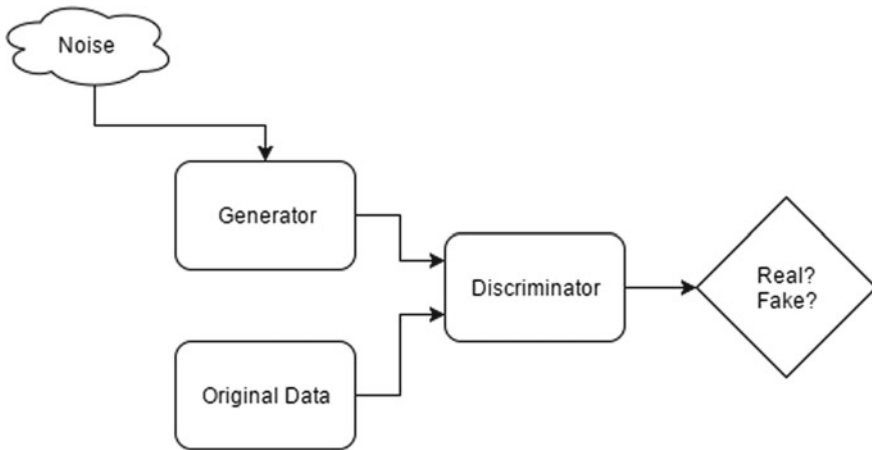


Fig. 1 General GAN architecture

2.2 Chaos Theory

Chaos theory is well defined as a nonlinear, complex dynamical system and is a branch of mathematics. This dynamical system is highly dependent on the initial properties and conditions because of which the outputs that are produced are very much random-like and highly unpredictable, but still easily reproducible and deterministic. This property becomes the main reason for the usefulness of chaotic maps in the field of secure communications as in [11].

- **Chaos Dynamics for Encryption and Decryption:** Chaotic cryptography is a field that makes use of the concepts of chaos theory and cryptography for the encryption purposes. Traditionally number theory or some algorithms have been used along with it to make encryption possible. The sensitivity of chaotic maps with their initial conditions is quite useful for the encryption purposes as in work [12].
- **Pseudo-Random Number Generator with Chaotic Properties:** One of the widely used methods for the purpose of generating pseudo-random sequences using chaos theory in a dynamical system is logistic map as in [9]. The sequence produced has been used for cryptography purposes but with traditional cryptographic algorithms only. Using it with GANs is totally a new dimension in the field of cryptography.

3 Proposed Framework

In order to use neural networks for cryptography in real-life scenarios, there should be no discrepancy between the plain text and the decrypted text. Thus the accuracy should be quite high.

The primary aim of this research is to develop a method that uses shared key with much more randomness and train a pair of neural networks, for stronger encryption of plain text and other for more accurate decryption of encrypted text by much less error in plain text and decrypted text. Our research has the following objectives:

- (a) To generate more random bit sequence to be used as shared key
- (b) To implement a GAN architecture for encryption and decryption of data in the form of bits
- (c) And finally, to integrate the above so as to increase the accuracy and strength of the encryption.

By individually studying the three components, the complete overview of the system can be understood, namely:

- (a) Generative adversarial neural network
- (b) Chaotic pseudo-random number generator
- (c) Mechanism and algorithm.

3.1 *Generative Adversarial Network*

Unlike the traditional way of training two models as generator and discriminator, here it is modified in a way that three neural networks will be used. A pair of neural network will work as a generator and third will be a discriminator in a modified way. The three neural networks will be:

- (a) *Encryptor*: The input will be the plain text and the shared key, both in the binary sequence and generate the encrypted text.
- (b) *Decryptor*: Input will be the encrypted text and the shared key to generate the decrypted text as output.
- (c) *Eavesdropper*: It will take the input as encrypted text only, that is, it will intercept the text and will decrypt it without having the shared key.

Layers used in the generative adversarial network: All the three neural networks have the same architecture comprising the following layers:

- (a) Fully connected dense layer
- (b) Convolutional layer
- (c) Flatten layer

So in total, we have used one dense layer, four convolutional layer and one flatten layer.

Replacing pooling layer with strided convolution: For down sampling the input matrix MaxPooling layer is used in between the convolution layers, which in turn decreases the total number of training parameters in network. But here instead of directly downsampling we have used strided convolutions so as to allow the network to learn its own spatial sampling.

Activation functions used: The encryption is performed on binary sequences, 0 and 1. So to normalize the output of every layer in $[-1, 1]$ we have used *tanh* activation in every layer except for the last layer where we have used *sigmoid* activation.

3.2 Chaotic Pseudo-Random Number Generator

To generate the shared key with chaotic properties, the chaos theory will be applied as chaotic mapping to generate the pseudo-random number generator. The numbers generated by this generator will then be mapped into the bit form 0 or 1.

- (a) *Logistic mapping:* It is the one-dimensional iterative mapping [9], which produces chaos in the system by generating pseudo-random numbers. To obtain the value at given time ‘ t ’, it uses ‘ $t-1$ ’ value, by the formula:

$$x_t = r \cdot x_{t-1} \cdot (1 - x_{t-1})$$

$$x_0 \in (0, 1)$$

To obtain all the future values in the range $[0, 1]$, the initial value, that is value at time $t = 0$, takes the range $(0, 1]$. Also, the property of trajectory of the function depends on the parameter ‘ r ’ and experiments have shown that trajectory is chaotic when ‘ r ’ is strictly in the range $[3.99, 4]$.

- (b) *Henon map:* The second mapping scheme that was used for the pseudo-random number generation is the Henon map, which is a two-dimensional chaotic map. It is also an iterative map. The initial conditions are represented by the equations:

$$x_{k+1} = -\alpha x_k^2 + y_k + 1$$

$$y_{k+1} = \beta x_k,$$

where α and β have values 1.4 and 0.3, respectively [10]. To form a binary sequence, the outputs of these functions are mapped to 0, 1 based on the threshold value. It was found that the pseudo-random sequences formed were having great statistical properties.

Given the initial condition, the values obtained are deterministic, but considering their high susceptibility to the provided initial conditions which are chosen randomly in the above-stated range, the iterative values become unpredictable, hence making them suitable to be used as a random number to obtain the shared binary key.

3.3 Mechanism and Algorithm

The whole system is developed by implementing the following steps:

- (a) For the given size of data, pseudo-random numbers were generated using chaotic mapping.
- (b) These numbers were then mapped to 0,1 so as to get the random binary sequence. This binary sequence is then used as shared key in the neural network.
- (c) For the neural network three architectures are laid, comprising three convolutional layers, one fully connected layer and one flatten layer with *tanh* as the activation function for all the layers except for the last layer where *sigmoid* function is used (Fig. 2).
- (d) The data are converted into binary form and then fed into the networks
- (e) The encryptor and decryptor are trained by freezing the weights of the eavesdropper. Encryptor takes the shared key and plain text, and decryptor takes the shared key and encrypted data from encryptor, all in bits.
- (f) The output is stored and the re-constructional loss between plain text and decrypted data is calculated.
- (g) This re-constructional loss is then fed into the eavesdropper for training by now freezing the weights of encryptor and decryptor both.
- (h) The discriminator loss between plain text and output of eavesdropper is calculated.
- (i) The same procedure is followed in every iteration with the goal of minimizing the re-constructional loss and maximizing the discriminator loss.

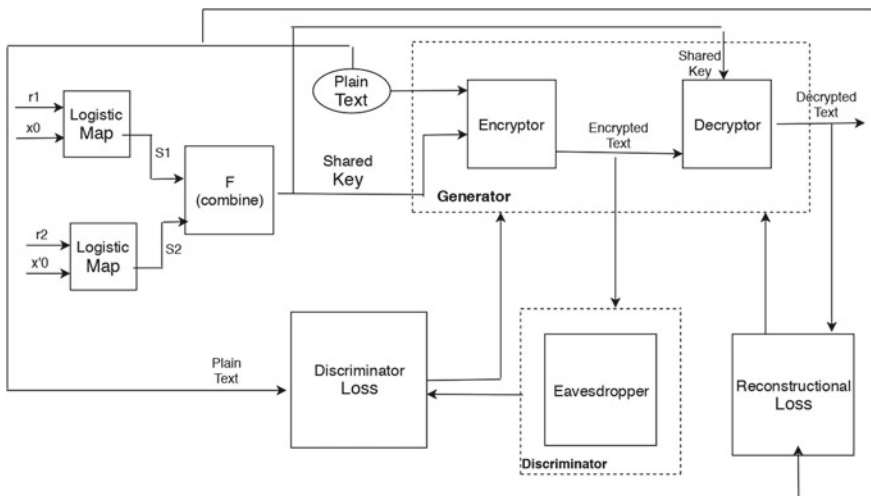


Fig. 2 System design and model overview

The pseudo-random numbers were generated with the above-stated logistic map and Henon map. So the above mechanism is followed individually with both mappings. The results obtained were then compared.

3.4 Simulation Results

The pseudo-random generator was implemented in python and the neural networks were implemented using Tensorflow keras api. Once the implementation process was completed, we took text data and tested our system. Following are the results drawn from the experiments, written according to the objectives:

- (a) *Bitwise accuracy of the decrypted text*: The primary motive of our system was to decrease the discrepancy between the decrypted text and the plain so as to make the system suitable for real life applications. As can be inferred, with the application of the chaos theory for the state-of-the-art system, the accuracy has increased.
- (b) *Strength of the encrypted data*: The primary objective of the system is to provide a mechanism for the strong encryption of data so that it could not be decrypted by any other neural network. Carrying forward from the above-stated objective, we have therefore trained the discriminator as eavesdropper which try to decrypt the data without the use of shared key. As per the analysis, it was found that the Henon key provides better security than logistic key for the given key size (Figs. 3 and 4, Tables 1 and 2).

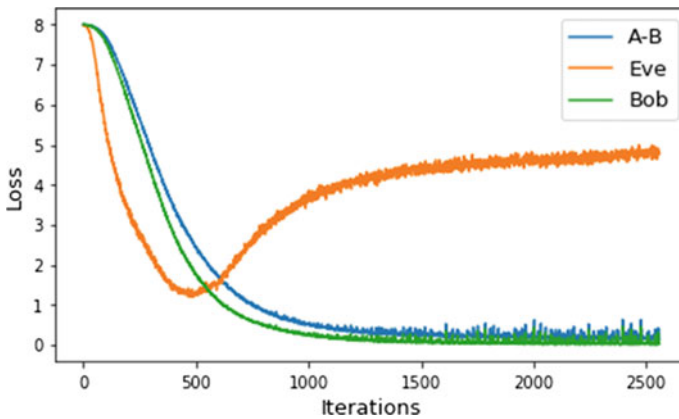


Fig. 3 Eavesdropper loss with logistic key

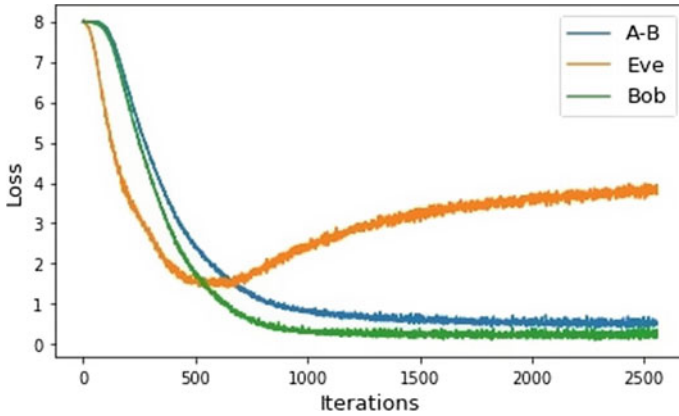


Fig. 4 Eavesdropper loss with Henon key

Table 1 Model accuracy

Sr. No.	Model	Accuracy (%)
1	Model formulated with python inbuilt random function	77.35
2	Model formulated with logistic chaotic key	86.74
3	Model implemented with Henon key	98.49

Table 2 Eavesdropper’s loss

S. No.	Model	Loss
1	Without chaotic key	2.96
2	With logistic key	3.87
3	With Henon key	4.94

Higher loss indicates greater strength of encryption

4 Conclusion

A new methodology for encryption and decryption of data was proposed, which combines the benefit of chaos theory to produce shared key and generative adversarial network to perform strong encryption of data. The logistic map and Henon map were individually used to obtain the shared key. This shared key is then fed into the combination of three neural networks which are trained in a way to produce the strongly encrypted text.

Acknowledgements Funding agency: Science and Engineering Research Board (SERB), ECR/2015/000234.

References

1. T. Graepel, K. Lauter, M. Naehrig, ML Confidential: Machine Learning on Encrypted Data, Information Security and Cryptology ICISC 2012. ICISC 2012 (2013)
2. M. Abadi, D.G. Andersen, Learning to Protect Communications with Adversarial Neural Cryptography, Google Brain, October 24, (2016)
3. J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps. *Int. J. Bifurc. Chaos*
4. A. Akhavan, A. Samsudin, A. Akhshani, A symmetric image encryption scheme based on combination of nonlinear chaotic maps. *J. Frankl. Inst.* 1797–1813 (2011)
5. M. Chauhan, R. Prajapati, Image encryption using chaotic based artificial neural network, *Int. J. Sci. Eng. Res.* (2014)
6. E. Volna, M. Kotyrba, M. Janosek, Cryptography based on neural network (Department of Informatics and Computers, 2005)
7. U. Wenwu, Z. Cao, Cryptography based on delayed chaotic neural networks (Department of Mathematics, Southeast University, 28 March 2006)
8. M. Arvandi, S. Whu, W.W. Malik, Symmetric cypher design using recurrent neural networks, in *International Joint Conference on neural networks*, vol. 21 (2006)
9. A. Kanso, N. Samaoui, Logistic Maps for Binary Number Generation, (Departments of Mathematics and Computer Science, 30 October 2007)
10. M. Suneel, Cryptographic pseudo random sequences from the chaotic Henon Map, Defence Research and Development Organization, 9 April (2009)
11. T. Godhavari, N.L. Alamelu, R. Sundararajan, Cryptography using neural network, in *IEEE Indecon 2005 Conference*, (Chennai, India, 2005)
12. W. Yu, J. Cao, Cryptography based on delayed chaotic neural network. *Phys. Lett. A* (2006)

Author Index

A

Abdul Ahad, Mohd, 223
Ahuja, Laxmi, 237
Ali Haq, Zeeshan, 223
Ansari, Iffat Rehman, 65
Ansari, Shaharyar Alam, 169
Aurobindo, K., 55

B

Bhatia, Tarunpreet, 155

C

Chakravarthi, Veena S., 183
Chaudhary, Abhishek, 91
Chhikara, Prateek, 155

D

Dalal, Sahil, 135
Dhanda, Namrata, 197
Dhar, Rudra Sankar, 83

G

Ganzha, Maria, 105
Gautam, Shweta, 25
Ghosh, Ankush, 15
Gupta, Abhishek, 55
Gupta, Kapil Kumar, 197
Gupta, Prakhar, 155
Gupta, Sheifali, 121

H

Hima Bindu, M., 237

J

Jain, Vishal, 121
Jindal, Udit, 121

K

Karthikeyan, B., 35
Kaur, Paramjot, 215
Kaur, Ramanpreet, 215
Khanam, Ruqaiya, 1
Khandelwal, Riya, 251
Khan, Ramsha, 1
Khangte, Lalthanpuui, 83
Kumar, Ankit, 83
Kumar, B. Nithish, 35
Kumar, Upendra, 197

M

Mahajan, Anurag, 55
Małkiński, Mikołaj, 105
Mamatsashvili, Givi-Giorgi, 105

N

Naaz, Sameena, 65

P

Paprzycki, Marcin, 105, 121
Parashar, Pragati, 1
Ponichtera, Konrad, 105
Purswani, Juhi, 251

R

Rajagopal, Rajesh, 251

Rao, K. Shubha, [183](#)
Rao, N. Madhusudana, [35](#)
Raza, Nida, [47](#)
Reddy, M. Rajasekhar, [35](#)

S

Saini, Prateek, [91](#)
Sana, Ehtesham, [65](#)
Sarkar, Sampreet, [15](#)
Sehgal, Smriti, [237](#)
Sen, Amlan, [15](#)
Shukla, K. K., [91](#)
Siddiqui, Farheen, [47](#)
Singh, Anuraj, [251](#)
Singh, Prabhjot, [155](#)
Singla, Manisha, [91](#)

Soni, Praveen, [25](#)
Soni, Samyak, [91](#)
Sultana, Parvin, [83](#)

T

Tripathi, Gautami, [223](#)

V

Verma, Divya, [135](#)
Vishwakarma, Virendra Prasad, [135](#)

Z

Zafar, Aasim, [169](#)