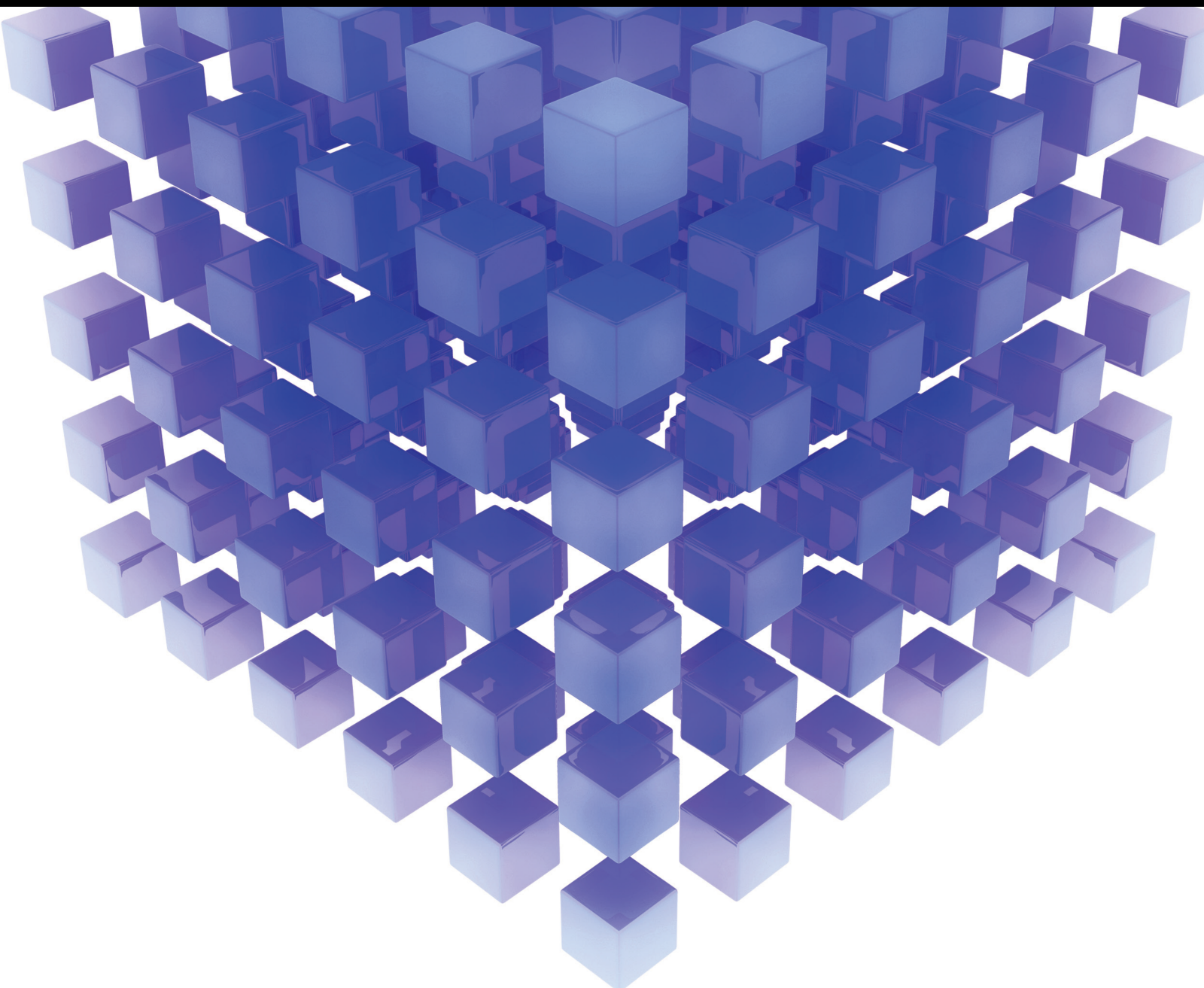


Hybrid Approaches for Image and Video Processing

Lead Guest Editor: Nouman Ali

Guest Editors: Muhammad Sajid, Bushra Zafar, Saadat Hanif Dar, and Savvas A. Chatzichristofis





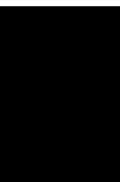
Hybrid Approaches for Image and Video Processing

Mathematical Problems in Engineering

Hybrid Approaches for Image and Video Processing

Lead Guest Editor: Nouman Ali


Guest Editors: Muhammad Sajid, Bushra Zafar,
Saadat Hanif Dar, and Savvas A. Chatzichristofis



Copyright © 2022 Hindawi Limited. All rights reserved.

This is a special issue published in “Mathematical Problems in Engineering.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Chief Editor

Guangming Xie , China

Academic Editors

Kumaravel A , India
Waqas Abbasi, Pakistan
Mohamed Abd El Aziz , Egypt
Mahmoud Abdel-Aty , Egypt
Mohammed S. Abdo, Yemen
Mohammad Yaghoub Abdollahzadeh
Jamalabadi , Republic of Korea
Rahib Abiyev , Turkey
Leonardo Acho , Spain
Daniela Addessi , Italy
Arooj Adeel , Pakistan
Waleed Adel , Egypt
Ramesh Agarwal , USA
Francesco Aggogeri , Italy
Ricardo Aguilar-Lopez , Mexico
Afaq Ahmad , Pakistan
Naveed Ahmed , Pakistan
Elias Aifantis , USA
Akif Akgul , Turkey
Tareq Al-shami , Yemen
Guido Ala, Italy
Andrea Alaimo , Italy
Reza Alam, USA
Osamah Albahri , Malaysia
Nicholas Alexander , United Kingdom
Salvatore Alfonzetti, Italy
Ghous Ali , Pakistan
Nouman Ali , Pakistan
Mohammad D. Aliyu , Canada
Juan A. Almendral , Spain
A.K. Alomari, Jordan
José Domingo Álvarez , Spain
Cláudio Alves , Portugal
Juan P. Amezcua-Sanchez, Mexico
Mukherjee Amitava, India
Lionel Amodeo, France
Sebastian Anita, Romania
Costanza Arico , Italy
Sabri Arik, Turkey
Fausto Arpino , Italy
Rashad Asharabi , Saudi Arabia
Farhad Aslani , Australia
Mohsen Asle Zaem , USA

Andrea Avanzini , Italy
Richard I. Avery , USA
Viktor Avrutin , Germany
Mohammed A. Awadallah , Malaysia
Francesco Aymerich , Italy
Sajad Azizi , Belgium
Michele Baccocchi , Italy
Seungik Baek , USA
Khaled Bahlali, France
M.V.A Raju Bahubalendruni, India
Pedro Balaguer , Spain
P. Balasubramaniam, India
Stefan Balint , Romania
Ines Tejado Balsera , Spain
Alfonso Banos , Spain
Jerzy Baranowski , Poland
Tudor Barbu , Romania
Andrzej Bartoszewicz , Poland
Sergio Baselga , Spain
S. Caglar Baslamisli , Turkey
David Bassir , France
Chiara Bedon , Italy
Azeddine Beghdadi, France
Andriette Bekker , South Africa
Francisco Beltran-Carbajal , Mexico
Abdellatif Ben Makhlof , Saudi Arabia
Denis Benasciutti , Italy
Ivano Benedetti , Italy
Rosa M. Benito , Spain
Elena Benvenuti , Italy
Giovanni Berselli, Italy
Michele Betti , Italy
Pietro Bia , Italy
Carlo Bianca , France
Simone Bianco , Italy
Vincenzo Bianco, Italy
Vittorio Bianco, Italy
David Bigaud , France
Sardar Muhammad Bilal , Pakistan
Antonio Bilotta , Italy
Sylvio R. Bistafa, Brazil
Chiara Boccaletti , Italy
Rodolfo Bontempo , Italy
Alberto Borboni , Italy
Marco Bortolini, Italy

Paolo Boscariol, Italy
Daniela Boso , Italy
Guillermo Botella-Juan, Spain
Abdesselem Boulkroune , Algeria
Boulaïd Boulkroune, Belgium
Fabio Bovenga , Italy
Francesco Braghin , Italy
Ricardo Branco, Portugal
Julien Bruchon , France
Matteo Bruggi , Italy
Michele Brun , Italy
Maria Elena Bruni, Italy
Maria Angela Butturi , Italy
Bartłomiej Błachowski , Poland
Dhanamjayulu C , India
Raquel Caballero-Águila , Spain
Filippo Cacace , Italy
Salvatore Caddemi , Italy
Zuowei Cai , China
Roberto Caldelli , Italy
Francesco Cannizzaro , Italy
Maosen Cao , China
Ana Carpio, Spain
Rodrigo Carvajal , Chile
Caterina Casavola, Italy
Sara Casciati, Italy
Federica Caselli , Italy
Carmen Castillo , Spain
Inmaculada T. Castro , Spain
Miguel Castro , Portugal
Giuseppe Catalanotti , United Kingdom
Alberto Cavallo , Italy
Gabriele Cazzulani , Italy
Fatih Vehbi Celebi, Turkey
Miguel Cerrolaza , Venezuela
Gregory Chagnon , France
Ching-Ter Chang , Taiwan
Kuei-Lun Chang , Taiwan
Qing Chang , USA
Xiaoheng Chang , China
Prasenjit Chatterjee , Lithuania
Kacem Chehdi, France
Peter N. Cheimets, USA
Chih-Chiang Chen , Taiwan
He Chen , China



































Kebing Chen , China
Mengxin Chen , China
Shyi-Ming Chen , Taiwan
Xizhong Chen , Ireland
Xue-Bo Chen , China
Zhiwen Chen , China
Qiang Cheng, USA
Zeyang Cheng, China
Luca Chiapponi , Italy
Francisco Chicano , Spain
Tirivanhu Chinyoka , South Africa
Adrian Chmielewski , Poland
Seongim Choi , USA
Gautam Choubey , India
Hung-Yuan Chung , Taiwan
Yusheng Ci, China
Simone Cinquemani , Italy
Roberto G. Citarella , Italy
Joaquim Ciurana , Spain
John D. Clayton , USA
Piero Colajanni , Italy
Giuseppina Colicchio, Italy
Vassilios Constantoudis , Greece
Enrico Conte, Italy
Alessandro Contento , USA
Mario Cools , Belgium
Gino Cortellessa, Italy
Carlo Cosentino , Italy
Paolo Crippa , Italy
Erik Cuevas , Mexico
Guozeng Cui , China
Mehmet Cunkas , Turkey
Giuseppe D'Aniello , Italy
Peter Dabnichki, Australia
Weizhong Dai , USA
Zhifeng Dai , China
Purushothaman Damodaran , USA
Sergey Dashkovskiy, Germany
Adiel T. De Almeida-Filho , Brazil
Fabio De Angelis , Italy
Samuele De Bartolo , Italy
Stefano De Miranda , Italy
Filippo De Monte , Italy

José António Fonseca De Oliveira
Correia , Portugal
Jose Renato De Sousa , Brazil
Michael Defoort, France
Alessandro Della Corte, Italy
Laurent Dewasme , Belgium
Sanku Dey , India
Gianpaolo Di Bona , Italy
Roberta Di Pace , Italy
Francesca Di Puccio , Italy
Ramón I. Diego , Spain
Yannis Dimakopoulos , Greece
Hasan Dinçer , Turkey
José M. Domínguez , Spain
Georgios Dounias, Greece
Bo Du , China
Emil Dumic, Croatia
Madalina Dumitriu , United Kingdom
Premraj Durairaj , India
Saeed Eftekhar Azam, USA
Said El Kafhali , Morocco
Antonio Elipe , Spain
R. Emre Erkmen, Canada
John Escobar , Colombia
Leandro F. F. Miguel , Brazil
FRANCESCO FOTI , Italy
Andrea L. Facci , Italy
Shahla Faisal , Pakistan
Giovanni Falsone , Italy
Hua Fan, China
Jianguang Fang, Australia
Nicholas Fantuzzi , Italy
Muhammad Shahid Farid , Pakistan
Hamed Faruqi, Iran
Yann Favennec, France
Fiorenzo A. Fazzolari , United Kingdom
Giuseppe Fedele , Italy
Roberto Fedele , Italy
Baowei Feng , China
Mohammad Ferdows , Bangladesh
Arturo J. Fernández , Spain
Jesus M. Fernandez Oro, Spain
Francesco Ferrise, Italy
Eric Feulvarch , France
Thierry Floquet, France

Eric Florentin , France
Gerardo Flores, Mexico
Antonio Forcina , Italy
Alessandro Formisano, Italy
Francesco Franco , Italy
Elisa Francomano , Italy
Juan Frausto-Solis, Mexico
Shujun Fu , China
Juan C. G. Prada , Spain
HECTOR GOMEZ , Chile
Matteo Gaeta , Italy
Mauro Gaggero , Italy
Zoran Gajic , USA
Jaime Gallardo-Alvarado , Mexico
Mosè Gallo , Italy
Akemi Gálvez , Spain
Maria L. Gandarias , Spain
Hao Gao , Hong Kong
Xingbao Gao , China
Yan Gao , China
Zhiwei Gao , United Kingdom
Giovanni Garcea , Italy
José García , Chile
Harish Garg , India
Alessandro Gasparetto , Italy
Stylianos Georgantzinou, Greece
Fotios Georgiades , India
Parviz Ghadimi , Iran
Ştefan Cristian Gherghina , Romania
Georgios I. Giannopoulos , Greece
Agathoklis Giaralis , United Kingdom
Anna M. Gil-Lafuente , Spain
Ivan Giorgio , Italy
Gaetano Giunta , Luxembourg
Jefferson L.M.A. Gomes , United Kingdom
Emilio Gómez-Déniz , Spain
Antonio M. Gonçalves de Lima , Brazil
Qunxi Gong , China
Chris Goodrich, USA
Rama S. R. Gorla, USA
Veena Goswami , India
Xunjie Gou , Spain
Jakub Grabski , Poland

Antoine Grall , France
George A. Gravvanis , Greece
Fabrizio Greco , Italy
David Greiner , Spain
Jason Gu , Canada
Federico Guarracino , Italy
Michele Guida , Italy
Muhammet Gul , Turkey
Dong-Sheng Guo , China
Hu Guo , China
Zhaoxia Guo, China
Yusuf Gurefe, Turkey
Salim HEDDAM , Algeria
ABID HUSSANAN, China
Quang Phuc Ha, Australia
Li Haitao , China
Petr Hájek , Czech Republic
Mohamed Hamdy , Egypt
Muhammad Hamid , United Kingdom
Renke Han , United Kingdom
Weimin Han , USA
Xingsi Han, China
Zhen-Lai Han , China
Thomas Hanne , Switzerland
Xinan Hao , China
Mohammad A. Hariri-Ardebili , USA
Khalid Hattaf , Morocco
Defeng He , China
Xiao-Qiao He, China
Yanchao He, China
Yu-Ling He , China
Ramdane Hedjar , Saudi Arabia
Jude Hemanth , India
Reza Hemmati, Iran
Nicolae Herisanu , Romania
Alfredo G. Hernández-Díaz , Spain
M.I. Herreros , Spain
Eckhard Hitzer , Japan
Paul Honeine , France
Jaromir Horacek , Czech Republic
Lei Hou , China
Yingkun Hou , China
Yu-Chen Hu , Taiwan
Yunfeng Hu, China
Can Huang , China
Gordon Huang , Canada
Linsheng Huo , China
Sajid Hussain, Canada
Asier Ibeas , Spain
Orest V. Iftime , The Netherlands
Przemyslaw Ignaciuk , Poland
Giacomo Innocenti , Italy
Emilio Insfran Pelozo , Spain
Azeem Irshad, Pakistan
Alessio Ishizaka, France
Benjamin Ivorra , Spain
Breno Jacob , Brazil
Reema Jain , India
Tushar Jain , India
Amin Jajarmi , Iran
Chiranjibe Jana , India
Łukasz Jankowski , Poland
Samuel N. Jator , USA
Juan Carlos Jáuregui-Correa , Mexico
Kandasamy Jayakrishna, India
Reza Jazar, Australia
Khalide Jbilou, France
Isabel S. Jesus , Portugal
Chao Ji , China
Qing-Chao Jiang , China
Peng-fei Jiao , China
Ricardo Fabricio Escobar Jiménez , Mexico
Emilio Jiménez Macías , Spain
Maolin Jin, Republic of Korea
Zhuo Jin, Australia
Ramash Kumar K , India
BHABEN KALITA , USA
MOHAMMAD REZA KHEDMATI , Iran
Viacheslav Kalashnikov , Mexico
Mathiyalagan Kalidass , India
Tamas Kalmar-Nagy , Hungary
Rajesh Kaluri , India
Jyotheeswara Reddy Kalvakurthi, India
Zhao Kang , China
Ramani Kannan , Malaysia
Tomasz Kapitaniak , Poland
Julius Kaplunov, United Kingdom
Konstantinos Karamanos, Belgium
Michal Kawulok, Poland


Irfan Kaymaz , Turkey
Vahid Kayvanfar , Qatar
Krzysztof Kecik , Poland
Mohamed Khader , Egypt
Chaudry M. Khalique , South Africa
Mukhtaj Khan , Pakistan
Shahid Khan , Pakistan
Nam-Il Kim, Republic of Korea
Philipp V. Kiryukhantsev-Korneev ,
Russia
P.V.V Kishore , India
Jan Koci , Czech Republic
Ioannis Kostavelis , Greece
Sotiris B. Kotsiantis , Greece
Frederic Kratz , France
Vamsi Krishna , India
Edyta Kucharska, Poland
Krzysztof S. Kulpa , Poland
Kamal Kumar, India
Prof. Ashwani Kumar , India
Michal Kunicki , Poland
Cedrick A. K. Kwuimy , USA
Kyandoghere Kyamakya, Austria
Ivan Kyrchei , Ukraine
Márcio J. Lacerda , Brazil
Eduardo Lalla , The Netherlands
Giovanni Lancioni , Italy
Jaroslaw Latalski , Poland
Hervé Laurent , France
Agostino Lauria , Italy
Aimé Lay-Ekuakille , Italy
Nicolas J. Leconte , France
Kun-Chou Lee , Taiwan
Dimitri Lefebvre , France
Eric Lefevre , France
Marek Lefik, Poland
Yaguo Lei , China
Kauko Leiviskä , Finland
Ervin Lenzi , Brazil
ChenFeng Li , China
Jian Li , USA
Jun Li , China
Yueyang Li , China
Zhao Li , China






























Zhen Li , China
En-Qiang Lin, USA
Jian Lin , China
Qibin Lin, China
Yao-Jin Lin, China
Zhiyun Lin , China
Bin Liu , China
Bo Liu , China
Heng Liu , China
Jianxu Liu , Thailand
Lei Liu , China
Sixin Liu , China
Wanquan Liu , China
Yu Liu , China
Yuanchang Liu , United Kingdom
Bonifacio Llamazares , Spain
Alessandro Lo Schiavo , Italy
Jean Jacques Loiseau , France
Francesco Lolli , Italy
Paolo Lonetti , Italy
António M. Lopes , Portugal
Sebastian López, Spain
Luis M. López-Ochoa , Spain
Vassilios C. Loukopoulos, Greece
Gabriele Maria Lozito , Italy
Zhiguo Luo , China
Gabriel Luque , Spain
Valentin Lychagin, Norway
YUE MEI, China
Junwei Ma , China
Xuanlong Ma , China
Antonio Madeo , Italy
Alessandro Magnani , Belgium
Toqeer Mahmood , Pakistan
Fazal M. Mahomed , South Africa
Arunava Majumder , India
Sarfranz Nawaz Malik, Pakistan
Paolo Manfredi , Italy
Adnan Maqsood , Pakistan
Muazzam Maqsood, Pakistan
Giuseppe Carlo Marano , Italy
Damijan Markovic, France
Filipe J. Marques , Portugal
Luca Martinelli , Italy
Denizar Cruz Martins, Brazil

Francisco J. Martos , Spain
Elio Masciari , Italy
Paolo Massioni , France
Alessandro Mauro , Italy
Jonathan Mayo-Maldonado , Mexico
Pier Luigi Mazzeo , Italy
Laura Mazzola, Italy
Driss Mehdi , France
Zahid Mehmood , Pakistan
Roderick Melnik , Canada
Xiangyu Meng , USA
Jose Merodio , Spain
Alessio Merola , Italy
Mahmoud Mesbah , Iran
Luciano Mescia , Italy
Laurent Mevel , France
Constantine Michailides , Cyprus
Mariusz Michta , Poland
Prankul Middha, Norway
Aki Mikkola , Finland
Giovanni Minafò , Italy
Edmondo Minisci , United Kingdom
Hiroyuki Mino , Japan
Dimitrios Mitsotakis , New Zealand
Ardashir Mohammadzadeh , Iran
Francisco J. Montáns , Spain
Francesco Montefusco , Italy
Gisele Mophou , France
Rafael Morales , Spain
Marco Morandini , Italy
Javier Moreno-Valenzuela , Mexico
Simone Morganti , Italy
Caroline Mota , Brazil
Aziz Moukrim , France
Shen Mouquan , China
Dimitris Mourtzis , Greece
Emiliano Mucchi , Italy
Taseer Muhammad, Saudi Arabia
Ghulam Muhiuddin, Saudi Arabia
Amitava Mukherjee , India
Josefa Mula , Spain
Jose J. Muñoz , Spain
Giuseppe Muscolino, Italy
Marco Mussetta , Italy

Hariharan Muthusamy, India
Alessandro Naddeo , Italy
Raj Nandkeolyar, India
Keivan Navaie , United Kingdom
Soumya Nayak, India
Adrian Neagu , USA
Erivelton Geraldo Nepomuceno , Brazil
AMA Neves, Portugal
Ha Quang Thinh Ngo , Vietnam
Nhon Nguyen-Thanh, Singapore
Papakostas Nikolaos , Ireland
Jelena Nikolic , Serbia
Tatsushi Nishi, Japan
Shanzhou Niu , China
Ben T. Nohara , Japan
Mohammed Nouari , France
Mustapha Nourelfath, Canada
Kazem Nouri , Iran
Ciro Núñez-Gutiérrez , Mexico
Włodzimierz Ogryczak, Poland
Roger Ohayon, France
Krzysztof Okarma , Poland
Mitsuhiro Okayasu, Japan
Murat Olgun , Turkey
Diego Oliva, Mexico
Alberto Olivares , Spain
Enrique Onieva , Spain
Calogero Orlando , Italy
Susana Ortega-Cisneros , Mexico
Sergio Ortobelli, Italy
Naohisa Otsuka , Japan
Sid Ahmed Ould Ahmed Mahmoud , Saudi Arabia
Taoreed Owolabi , Nigeria
EUGENIA PETROPOULOU , Greece
Arturo Pagano, Italy
Madhumangal Pal, India
Pasquale Palumbo , Italy
Dragan Pamučar, Serbia
Weifeng Pan , China
Chandan Pandey, India
Rui Pang, United Kingdom
Jürgen Pannek , Germany
Elena Panteley, France
Achille Paolone, Italy

George A. Papakostas , Greece
Xosé M. Pardo , Spain
You-Jin Park, Taiwan
Manuel Pastor, Spain
Pubudu N. Pathirana , Australia
Surajit Kumar Paul , India
Luis Payá , Spain
Igor Pažanin , Croatia
Libor Pekař , Czech Republic
Francesco Pellicano , Italy
Marcello Pellicciari , Italy
Jian Peng , China
Mingshu Peng, China
Xiang Peng , China
Xindong Peng, China
Yuexing Peng, China
Marzio Pennisi , Italy
Maria Patrizia Pera , Italy
Matjaz Perc , Slovenia
A. M. Bastos Pereira , Portugal
Wesley Peres, Brazil
F. Javier Pérez-Pinal , Mexico
Michele Perrella, Italy
Francesco Pesavento , Italy
Francesco Petrini , Italy
Hoang Vu Phan, Republic of Korea
Lukasz Pieczonka , Poland
Dario Piga , Switzerland
Marco Pizzarelli , Italy
Javier Plaza , Spain
Goutam Pohit , India
Dragan Poljak , Croatia
Jorge Pomares , Spain
Hiram Ponce , Mexico
Sébastien Poncet , Canada
Volodymyr Ponomaryov , Mexico
Jean-Christophe Ponsart , France
Mauro Pontani , Italy
Sivakumar Poruran, India
Francesc Pozo , Spain
Aditya Rio Prabowo , Indonesia
Anchasa Pramuanjaroenkij , Thailand
Leonardo Primavera , Italy
B Rajanarayan Prusty, India

Krzysztof Puszynski , Poland
Chuan Qin , China
Dongdong Qin, China
Jianlong Qiu , China
Giuseppe Quaranta , Italy
DR. RITU RAJ , India
Vitomir Racic , Italy
Carlo Rainieri , Italy
Kumbakonam Ramamani Rajagopal, USA
Ali Ramazani , USA
Angel Manuel Ramos , Spain
Higinio Ramos , Spain
Muhammad Afzal Rana , Pakistan
Muhammad Rashid, Saudi Arabia
Manoj Rastogi, India
Alessandro Rasulo , Italy
S.S. Ravindran , USA
Abdolrahman Razani , Iran
Alessandro Reali , Italy
Jose A. Reinoso , Spain
Oscar Reinoso , Spain
Haijun Ren , China
Carlo Renno , Italy
Fabrizio Renno , Italy
Shahram Rezapour , Iran
Ricardo Rianza , Spain
Francesco Riganti-Fulginei , Italy
Gerasimos Rigatos , Greece
Francesco Ripamonti , Italy
Jorge Rivera , Mexico
Eugenio Roanes-Lozano , Spain
Ana Maria A. C. Rocha , Portugal
Luigi Rodino , Italy
Francisco Rodríguez , Spain
Rosana Rodríguez López, Spain
Francisco Rossomando , Argentina
Jose de Jesus Rubio , Mexico
Weiguo Rui , China
Rubén Ruiz , Spain
Ivan D. Rukhlenko , Australia
Dr. Eswaramoorthi S. , India
Weichao SHI , United Kingdom
Chaman Lal Sabharwal , USA
Andrés Sáez , Spain

Bekir Sahin, Turkey
Laxminarayan Sahoo , India
John S. Sakellariou , Greece
Michael Sakellariou , Greece
Salvatore Salamone, USA
Jose Vicente Salcedo , Spain
Alejandro Salcido , Mexico
Alejandro Salcido, Mexico
Nunzio Salerno , Italy
Rohit Salgotra , India
Miguel A. Salido , Spain
Sinan Salih , Iraq
Alessandro Salvini , Italy
Abdus Samad , India
Sovan Samanta, India
Nikolaos Samaras , Greece
Ramon Sancibrian , Spain
Giuseppe Sanfilippo , Italy
Omar-Jacobo Santos, Mexico
J Santos-Reyes , Mexico
José A. Sanz-Herrera , Spain
Musavarah Sarwar, Pakistan
Shahzad Sarwar, Saudi Arabia
Marcelo A. Savi , Brazil
Andrey V. Savkin, Australia
Tadeusz Sawik , Poland
Roberta Sburlati, Italy
Gustavo Scaglia , Argentina
Thomas Schuster , Germany
Hamid M. Sedighi , Iran
Mijanur Rahaman Seikh, India
Tapan Senapati , China
Lotfi Senhadji , France
Junwon Seo, USA
Michele Serpilli, Italy
Silvestar Šesnić , Croatia
Gerardo Severino, Italy
Ruben Sevilla , United Kingdom
Stefano Sfarra , Italy
Dr. Ismail Shah , Pakistan
Leonid Shaikhet , Israel
Vimal Shanmuganathan , India
Prayas Sharma, India
Bo Shen , Germany
Hang Shen, China

Xin Pu Shen, China
Dimitri O. Shepelsky, Ukraine
Jian Shi , China
Amin Shokrollahi, Australia
Suzanne M. Shontz , USA
Babak Shotorban , USA
Zhan Shu , Canada
Angelo Sifaleras , Greece
Nuno Simões , Portugal
Mehakpreet Singh , Ireland
Piyush Pratap Singh , India
Rajiv Singh, India
Seralathan Sivamani , India
S. Sivasankaran , Malaysia
Christos H. Skiadas, Greece
Konstantina Skouri , Greece
Neale R. Smith , Mexico
Bogdan Smolka, Poland
Delfim Soares Jr. , Brazil
Alba Sofi , Italy
Francesco Soldovieri , Italy
Raffaele Solimene , Italy
Yang Song , Norway
Jussi Sopanen , Finland
Marco Spadini , Italy
Paolo Spagnolo , Italy
Ruben Specogna , Italy
Vasilios Spitas , Greece
Ivanka Stamova , USA
Rafał Stanisławski , Poland
Miladin Stefanović , Serbia
Salvatore Strano , Italy
Yakov Strelniker, Israel
Kangkang Sun , China
Qiuqin Sun , China
Shuaishuai Sun, Australia
Yanchao Sun , China
Zong-Yao Sun , China
Kumarasamy Suresh , India
Sergey A. Suslov , Australia
D.L. Suthar, Ethiopia
D.L. Suthar , Ethiopia
Andrzej Swierniak, Poland
Andras Szekrenyes , Hungary
Kumar K. Tamma, USA

Yong (Aaron) Tan, United Kingdom
Marco Antonio Taneco-Hernández , Mexico
Lu Tang , China
Tianyou Tao, China
Hafez Tari , USA
Alessandro Tasora , Italy
Sergio Teggi , Italy
Adriana del Carmen Téllez-Anguiano , Mexico
Ana C. Teodoro , Portugal
Efstathios E. Theotokoglou , Greece
Jing-Feng Tian, China
Alexander Timokha , Norway
Stefania Tomasiello , Italy
Gisella Tomasini , Italy
Isabella Torricollo , Italy
Francesco Tornabene , Italy
Mariano Torrisi , Italy
Thang nguyen Trung, Vietnam
George Tsiatas , Greece
Le Anh Tuan , Vietnam
Nerio Tullini , Italy
Emilio Turco , Italy
Ilhan Tuzcu , USA
Efstratios Tzirtzilakis , Greece
FRANCISCO UREÑA , Spain
Filippo Ubertini , Italy
Mohammad Uddin , Australia
Mohammad Safi Ullah , Bangladesh
Serdar Ulubeyli , Turkey
Mati Ur Rahman , Pakistan
Panayiotis Vafeas , Greece
Giuseppe Vairo , Italy
Jesus Valdez-Resendiz , Mexico
Eusebio Valero, Spain
Stefano Valvano , Italy
Carlos-Renato Vázquez , Mexico
Martin Velasco Villa , Mexico
Franck J. Vernerey, USA
Georgios Veronis , USA
Vincenzo Vespri , Italy
Renato Vidoni , Italy
Venkatesh Vijayaraghavan, Australia

Anna Vila, Spain
Francisco R. Villatoro , Spain
Francesca Vipiana , Italy
Stanislav Vitek , Czech Republic
Jan Vorel , Czech Republic
Michael Vynnycky , Sweden
Mohammad W. Alomari, Jordan
Roman Wan-Wendner , Austria
Bingchang Wang, China
C. H. Wang , Taiwan
Dagang Wang, China
Guoqiang Wang , China
Huaiyu Wang, China
Hui Wang , China
J.G. Wang, China
Ji Wang , China
Kang-Jia Wang , China
Lei Wang , China
Qiang Wang, China
Qingling Wang , China
Weiwei Wang , China
Xinyu Wang , China
Yong Wang , China
Yung-Chung Wang , Taiwan
Zhenbo Wang , USA
Zhibo Wang, China
Waldemar T. Wójcik, Poland
Chi Wu , Australia
Qihong Wu, China
Yuqiang Wu, China
Zhibin Wu , China
Zhizheng Wu , China
Michalis Xenos , Greece
Hao Xiao , China
Xiao Ping Xie , China
Qingzheng Xu , China
Binghan Xue , China
Yi Xue , China
Joseph J. Yame , France
Chuanliang Yan , China
Xinggang Yan , United Kingdom
Hongtai Yang , China
Jixiang Yang , China
Mijia Yang, USA
Ray-Yeng Yang, Taiwan

Zaoli Yang , China
Jun Ye , China
Min Ye , China
Luis J. Yebra , Spain
Peng-Yeng Yin , Taiwan
Muhammad Haroon Yousaf , Pakistan
Yuan Yuan, United Kingdom
Qin Yuming, China
Elena Zaitseva , Slovakia
Arkadiusz Zak , Poland
Mohammad Zakwan , India
Ernesto Zambrano-Serrano , Mexico
Francesco Zammori , Italy
Jessica Zangari , Italy
Rafal Zdunek , Poland
Ibrahim Zeid, USA
Nianyin Zeng , China
Junyong Zhai , China
Hao Zhang , China
Haopeng Zhang , USA
Jian Zhang , China
Kai Zhang, China
Lingfan Zhang , China
Mingjie Zhang , Norway
Qian Zhang , China
Tianwei Zhang , China
Tongqian Zhang , China
Wenyu Zhang , China
Xianming Zhang , Australia
Xuping Zhang , Denmark
Yinyan Zhang, China
Yifan Zhao , United Kingdom
Debao Zhou, USA
Heng Zhou , China
Jian G. Zhou , United Kingdom
Junyong Zhou , China
Xueqian Zhou , United Kingdom
Zhe Zhou , China
Wu-Le Zhu, China
Gaetano Zizzo , Italy
Mingcheng Zuo, China





Contents

CNN-Based Automatic Helmet Violation Detection of Motorcyclists for an Intelligent Transportation System

Tasbeeha Waris, Muhammad Asif , Maaz Bin Ahmad , Toqeer Mahmood , Sadia Zafar, Mohsin Shah, and Ahsan Ayaz






Research Article (11 pages), Article ID 8246776, Volume 2022 (2022)

Classification of Woven Fabric Faulty Images Using Convolution Neural Network

Rehan Ashraf , Yasir Ijaz, Muhammad Asif , Khurram Zeeshan Haider , Toqeer Mahmood , and Muhammad Owais

Research Article (16 pages), Article ID 2573805, Volume 2022 (2022)

Brain Tumor Detection using Decision-Based Fusion Empowered with Fuzzy Logic

Aqsa Tahir , Muhammad Asif , Maaz Bin Ahmad , Toqeer Mahmood , Muhammad Adnan Khan , and Mushtaq Ali






Research Article (13 pages), Article ID 2710285, Volume 2022 (2022)

Facial Mask Detection Using Image Processing with Deep Learning

Hongyu Ding, Muhammad Ahsan Latif , Zain Zia , Muhammad Asif Habib , Muhammad Abdul Qayum , and Quancai Jiang

Research Article (10 pages), Article ID 8220677, Volume 2022 (2022)

Hybrid Approach for Shelf Monitoring and Planogram Compliance (Hyb-SMPC) in Retails Using Deep Learning and Computer Vision

Mehwish Saqlain , Saddaf Rubab , Malik M. Khan , Nouman Ali , and Shahzeb Ali 

Research Article (18 pages), Article ID 4916818, Volume 2022 (2022)

COMSATS Face: A Dataset of Face Images with Pose Variations, Its Design, and Aspects

Mahmood Ul Haq, Muhammad Athar Javed Sethi , Rehmat Ullah , Aamir Shazhad, Laiq Hasan, and Ghulam Mohammad Karami 

Research Article (11 pages), Article ID 4589057, Volume 2022 (2022)

Image Deblurring Algorithm Based on the Gaussian-Scale Mixture Expert Field Model

Jing Zhang  and Tao Zhang



Research Article (10 pages), Article ID 5926755, Volume 2022 (2022)

An LSTM with Differential Structure and Its Application in Action Recognition

Weifeng Chen , Fei Zheng , Shanping Gao , and Kai Hu 



Research Article (14 pages), Article ID 7316396, Volume 2022 (2022)

Efficient Localization of Multitype Barcodes in High-Resolution Images

Jinwang Yi  and Yuanbiao Xiao 





Research Article (10 pages), Article ID 5256124, Volume 2022 (2022)

Video Style Transfer based on Convolutional Neural Networks





Sun Dong , Youdong Ding, Yun Qian , and Mengfan Li

Research Article (9 pages), Article ID 8918722, Volume 2022 (2022)

A Real-Time Framework for Human Face Detection and Recognition in CCTV Images

Rehmat Ullah , Hassan Hayat, Afsah Abid Siddiqui, Uzma Abid Siddiqui, Jebran Khan , Farman Ullah, Shoaib Hassan, Laiq Hasan, Waleed Albattah , Muhammad Islam, and Ghulam Mohammad Karami 
Research Article (12 pages), Article ID 3276704, Volume 2022 (2022)

A Deep Learning Framework for Leukemia Cancer Detection in Microscopic Blood Samples Using Squeeze and Excitation Learning

Maryam Bukhari , Sadaf Yasmin , Saima Sammad , and Ahmed A. Abd El-Latif 
Research Article (18 pages), Article ID 2801227, Volume 2022 (2022)

Research Article

CNN-Based Automatic Helmet Violation Detection of Motorcyclists for an Intelligent Transportation System

Tasbeeha Waris,¹ Muhammad Asif ,¹ Maaz Bin Ahmad ,² Toqeer Mahmood ,³ Sadia Zafar,¹ Mohsin Shah,⁴ and Ahsan Ayaz¹

¹Department of Computer Science, Lahore Garrison University, Lahore, Pakistan

²College of Computing and Information Science, KIET, Karachi, Pakistan

³Faculty of Computer Science, National Textile University, Faisalabad, Pakistan

⁴Department of Telecommunication, Hazara University, Mansehra, Pakistan

Correspondence should be addressed to Toqeer Mahmood; toqeer.mahmood@yahoo.com

Received 8 May 2022; Revised 17 September 2022; Accepted 27 September 2022; Published 17 October 2022

Academic Editor: Muhammad Sajid

Copyright © 2022 Tasbeeha Waris et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An intelligent transportation system (ITS) is an advanced application that supports multiple transport and traffic management modes. ITS services include calling for emergency rescue and monitoring traffic laws with the help of roadside units. It is observed that many people lose their lives in motorbike accidents mainly due to not wearing helmets. Automatic helmet violation detection of motorcyclists from real-time videos is a demanding application in ITS. It enables one to spot and penalize bikers without a helmet. So, there is a need to develop a system that automatically detects and captures motorbikers without a helmet in real time. This work proposes a system to detect helmet violations automatically from surveillance videos captured by roadside-mounted cameras. The proposed technique is based on faster region-based convolutional neural network (R-CNN) deep learning model that takes video as an input and performs helmet violation detection to take necessary actions against traffic rule violators. Experimental analysis shows that the proposed system gives an accuracy of 97.69% and supersedes its competitors.

1. Introduction

The world's population is increasing at an unprecedented rate. As per a survey report, the world population was around 600 million at the start of the eighteenth century, which has now increased up to 7.8 billion in 2020 [1]. The increasing population rate is directly proportional to an increase in the use of vehicles. In 2018, the total number of registered vehicles was 23,588,268 compared to 21,506,641 in the previous year [2]. Motorbike is cheaper and an affordable source of transportation for middle-class people. The number of registered motorbikes reached an astonishing number of 17,465,880 in the year 2018, as compared to 15,664,098 in the previous year [1, 2]. According to the stats for the year 2018, 74% of all registered vehicles were motorbikes [3]. Due to the increased number of vehicles, road congestion caused more

accidents [4]. An intelligent transportation system (ITS) is an advanced transportation system, a collection of integrated technologies like electronics, communication, sensors, cameras, and so on [5]. It aims to provide a risk-free system that saves human lives and time and keeps them informed about road conditions, like weather, construction, and other calamities [6–8]. ITS is capable of implementing a transportation system that is smart, fully functional, and based on real-time calculations. This system usually calls the helpline in case of any emergency or accident encountered by travellers. It uses surveillance cameras mounted on roads to check violations [9, 10]. It incorporates different applications from basic to advanced, i.e., navigation systems for vehicles, variable message signs, and surveillance cameras on the road are some of its applications [11–14]. Figure 1 displays some applications of ITS.

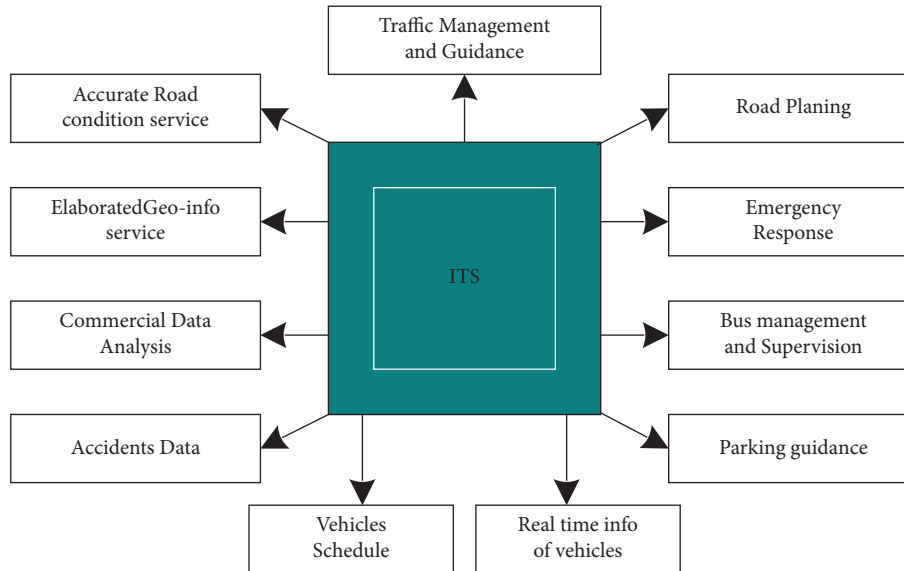


FIGURE 1: Applications of ITS.

Figure 2 shows an increase in the number of accidents in Pakistan, separating fatal and non-fatal accidents [15]. Motorbike is not only the most widely used vehicle but also the most dangerous mode of transportation [16]. According to a study conducted in Pakistan Institute of Medical Science (PIMS) during September 2015–December 2015, 709 total accidents were reported in the hospital. Out of these accidents, 71% were related to motorbikers [17]. It shows that most of the victims of traffic accidents are bike riders. So, it leads to a high causality rate in bikers during or after accidents. In those cases, riding without a helmet is the primary cause of death. According to stats, helmet reduces the death rate by 37% and the head injury rate by 69% [18]. So, it is mandatory by law to use a helmet while riding a bike [19]. Capturing all the people violating the rules for a traffic warden standing on a road is difficult. The worldwide reviews of studies proved that fatal accidents causing severe injuries had been reduced from 40% to 11% in the presence of surveillance cameras [20]. So, it is evident that there is a need to develop an intelligent system that automatically detects bikers without wearing a helmet with the help of surveillance cameras.

This paper develops a system for automatically detecting bikers without a helmet using a faster region-based convolutional neural network (R-CNN). The system takes input in the form of video and converts that into frames to perform helmet violation detection. The dataset has been collected from two sources, i.e., online repositories and self-captured videos from different locations in Lahore, Pakistan. The experimental analysis shows that the proposed system has 97.69% accuracy. It may help to take necessary actions against traffic rule violators.

The rest of the paper is organized as follows. Section 2 consists of a literature review. Section 3 contains the proposed helmet violation detection technique. Experimental analysis is performed in Section 4. Finally, Section 5 concludes the paper.

2. Literature Review

Computer vision and digital image processing are used in various applied domains such as remote sensing, pose detection, decision making, path detection, defect detection, and automatic driving [21–26]. The recent focus of research in this field is the use of deep learning models that have shown good results in various applied domains [27–29].

Many researchers have suggested different methods to solve the problem of automatic detection of helmet in real-time environment. Cheverton [30] implemented a system using support vector machine (SVM) and background subtraction techniques to identify bikers with and without a helmet. The self-generated dataset has been used for the development of the system. However, the system has two main limitations. Firstly, it examines the whole frame for helmet detection, increasing overall computational cost. Secondly, it also has an issue: it incorrectly classified the number of heads without a helmet. Silva et al. [31] introduced a hybrid descriptor model based on texture and geometric features to detect bikers without a helmet. The Hough transform (HT) and SVM are used to detect the head of the biker. The self-generated dataset has been used for the training of the algorithm. They extended their work and used a multilayer perception model to differentiate among different objects showing an accuracy of 94.23%.

Silva et al. [32] proposed a system based on HT and histogram oriented gradient (HOG) that helps extract the image's attributes. The input images are taken from the roadside cameras, and database of 255 images is established. The developed system has given accuracy of 91.37%. Waranusast et al. [33] suggested a system based on the K-nearest neighbor (KNN) classifier that helps determine and detect motorcyclists with and without helmets. The system has been tested on the self-created dataset. The input image is taken from a web camera. The experimental results showed that the system had given a correct detection for the far lane,

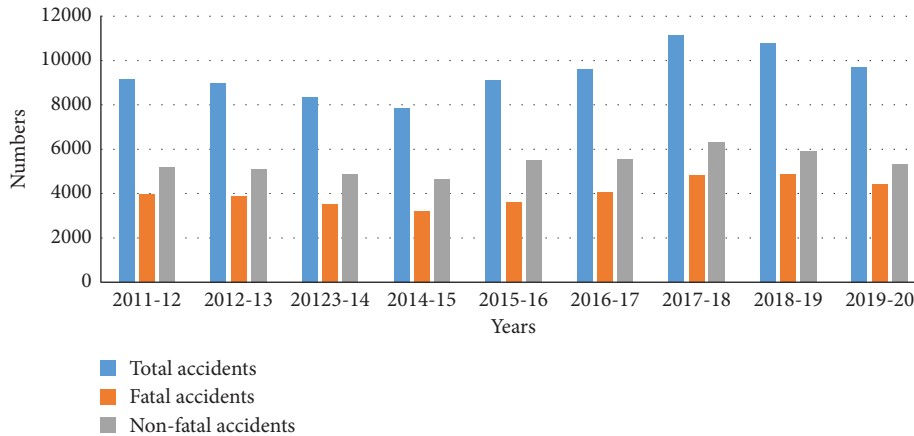


FIGURE 2: Trends of accidents in Pakistan.

near lane, and both lanes as 68%, 84%, and 74%, respectively. Dahiya et al. [34] developed a system that helps detect a motorcyclist without a helmet using HOG, SIFT, LBP, and SVM machine learning techniques. The input is taken from the camera in video and then converted to frames for further processing. They applied the background subtraction technique to select moving objects from the frames. HOG, scale invariant feature transform (SIFT), and local binary pattern (LBP) techniques are applied to extract features. If anything except a bike is detected, it has been overlooked. After that, SVM is used to classify the bikers with and without helmets. The self-generated dataset has been taken for testing purposes. The system has given an accuracy of 93.80%.

Boonsirisumpun et al. [35] deployed a convolutional neural network (CNN) system to detect bikers without a helmet. The input has been taken using cameras. The dataset of 493 images has been used for training purposes. The system used four CNN-based models, including Google Net, MobileNet, VGG19, and VGG16. The MobileNet gave the highest accuracy, which is 85.19%. Raj et al. [36] contributed to detecting bikers who have violated helmet-wearing rules based on a deep learning technique. The task of detecting motorcycles is accomplished using HOG and then selecting the region of interest. They applied CNN technology to identify bikers without helmets and to perform number plate recognition. The self-generated dataset from different sources has been used. They claimed accuracy of 94.70%. Wu et al. [37] used YOLOv3 and YOLO-dense models to detect bikers without a helmet. They collected datasets from two sources, i.e., self-generated and the Internet. The experimental results indicated that they had achieved 95.15% mAP for YOLOv3 and 97.59% for the YOLO-dense model.

Siebert and Lin [38] utilized a deep learning approach, RatinaNet50, to detect bikers without a helmet. The proposed system has used self-generated data for training. The two classes have been created, i.e. "With Helmet" and "Without Helmet." The experimental result showed that an accuracy of 72.8% has been achieved. Vishnu et al. [39] used

an adaptive search method to identify moving objects. After that, CNN on a self-generated dataset was used to identify bikers from moving objects. Finally, CNN is implemented to differentiate bikers not wearing a helmet.

Mistry et al. [40] used CNN to detect bikers without a helmet. They used YOLOv2 in 2 levels. Firstly, the system used YOLOv2 to detect different objects and motorcyclists without helmets. The COCO dataset has been used for training purposes. The experimental result gives an accuracy of 92.87%. Afzal et al. [41] used Faster R-CNN to detect bikers that have not used helmets. The system has been trained on a self-generated dataset. The experimental results gave an accuracy of 97.26%. Kharade et al. [42] introduced a system for detecting motorcyclists who are not wearing helmets through deep learning algorithms based on the YOLOv4 model. The proposed model indicates true performance in traffic motion pictures compared to current CNN-based algorithms.

The primary goal of Sridhar et al. [43] is to look at whether the person wears a helmet or not through YOLOv2. A method that uses deep convolutional neural networks (CNNs) for revealing motorcycle riders who disobey the legal guidelines has been established. It first detects the motorbike and then classifies it as with or without helmet. The proposed architecture yielded better experimental results in comparison with traditional algorithms.

Kathane et al. [44] used the YOLOv3 algorithm for implementation. Exceptional deep learning models are trained for object detection. The developed system uses three diverse deep learning models to detect these objects. The established system gives 88.5% precision for motorcycle detection and 91.8% for number plate detection. Rajalakshmi and Saravanan [45] developed a system for monitoring and handling persons breaking the guidelines through a convolutional neural network (CNN). The system performs vehicle classification, helmet detection, and mask detection through an appropriate CNN-based model. Table 1 displays the summary of the abovementioned related work.

The existing systems above can detect bikers without a helmet, but there are also some limitations. Most of the

TABLE 1: Summary of related work.

Paper	Objective and approach	Algorithm	Dataset	Accuracy	Remarks
Chiverton [30]	Background subtraction is used to identify bikes, and SVM is used to identify bikers without a helmet.	Support vector machine	Self-generated	N/A	High computation cost because the system determines the full frame for the detection of helmet.
Silva et al. [31]	A multilayer perception model is used to differentiate between different objects. The hybrid descriptor is based on the local binary operator to extract features.	Hough transforms with SVM	Self-generated	94.23%	Low-quality images are used. Descriptors give multiple attributes, which makes classification difficult.
Silva et al. [32]	Hough transformation and histogram oriented gradient help extract the image's attributes.	Hough transformation and histogram oriented gradient	Self-generated	91.37%	N/A
Waranusast et al. [33]	K-nearest neighbor classifier is applied to identify bikers not wearing a helmet.	Machine vision	Self-generated	74%	Accuracy is low.
Dahiya et al. [34]	Classification is done by using a binary classifier and visual features.	HOG, SIFT, LBP, and SVM	Self-generated	93.80	The system is not trained to classify bikers wearing the scarf instead of the helmet.
Boonsirirumpun et al. [35]	The single shot multibox detector (SSD) technique for detecting motorcyclists not wearing helmets is used.	Google Net, MobileNet, VGG19, and VGG16	Self-generated	85.19%	The system is not trained to classify bikers wearing the scarf instead of the helmet.
Raj et al. [36]	The researchers used HOG to identify bikers and CNN to detect helmet violators and number plate.	CNN	Self-generated	94.70%	N/A
Wu et al. [37]	YOLOv3 and YOLO-dense backbone are used to detect bikers without a helmet.	Deep learning	Dataset collected from two sources, i.e. self-generated and internet	YOLOv3: 95.15% and YOLO-dense backbone: 97.59%	For YOLOv3, mean average precision is 95.15%, and for the YOLO-dense backbone, it is 97.59%
Siebert and Lin [38]	A deep learning algorithm is used to detect bikers without a helmet.	ResNet50	Self-generated	72.8%	Accuracy is low, which can be improved.
Vishnu et al. [39]	CNN is used for helmet detection.	CNN	Self-generated	Accuracy is 93%	N/A
Mistry et al. [40]	YOLOv2 is used to detect bikers without a helmet.	Deep learning	COCO dataset	Accuracy is 94.7%	N/A
Afzal et al. [41]	Faster R-CNN is used to detect bikers without a helmet.	Faster R-CNN	Self-generated dataset	Accuracy is 97.26%	N/A

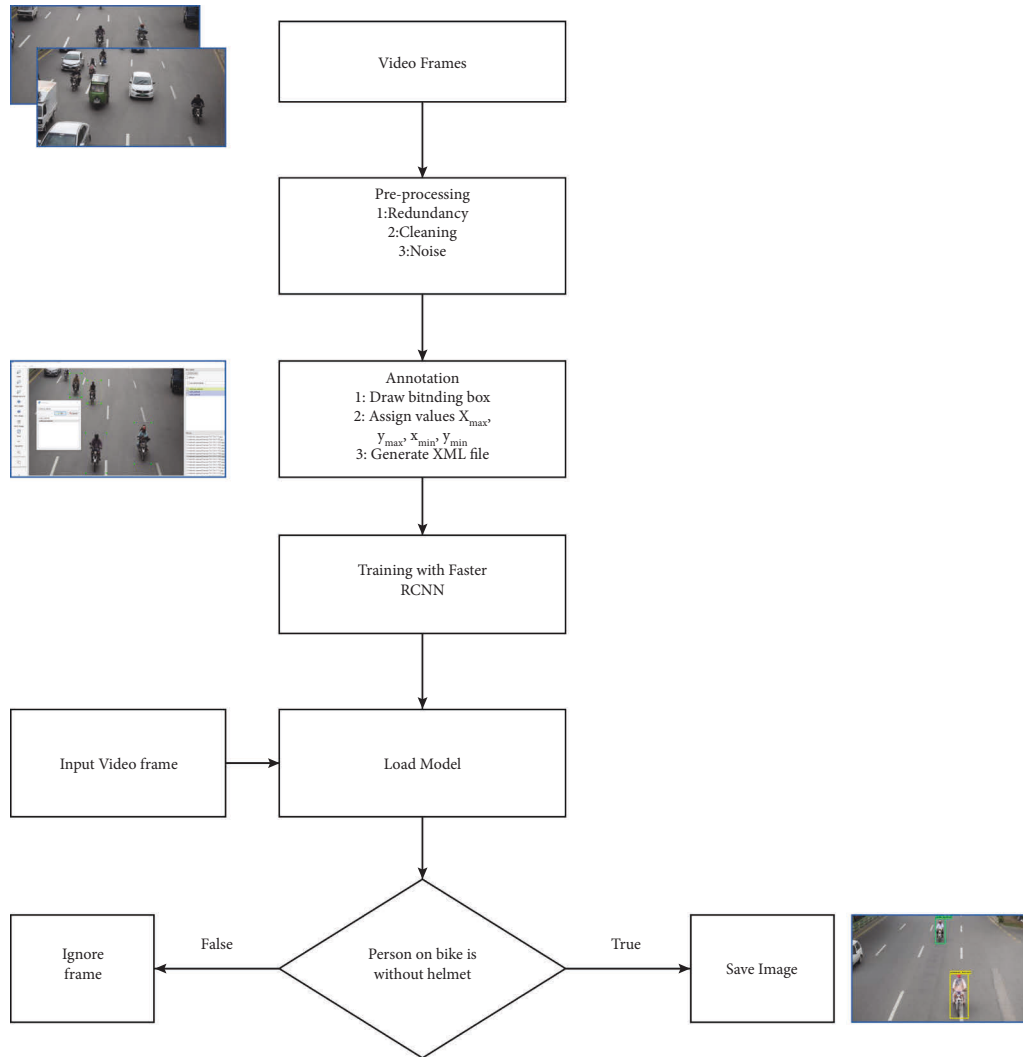


FIGURE 3: Block diagram of the proposed system.

existing systems have low accuracy. Moreover, the dataset used to develop the system is also limited. Furthermore, some of the above systems cannot differentiate between helmet and scarf. The proposed system can easily differentiate between a helmet and a scarf. The significant contribution of this work is the establishment of the dataset that consists of almost all types of bikes. In addition, the proposed technique is developed using a comprehensive dataset and achieves higher accuracy than the existing systems.

3. Proposed System

This section presents a proposed technique to automatically detect helmet violations from surveillance videos captured by roadside-mounted cameras. The proposed technique is based on Faster R-CNN deep learning model that takes video as an input and performs helmet violation detection to take necessary actions against traffic rule violators. The proposed system performs multiple operations in a

sequence. Firstly, it detects motorbikes and separates these from other vehicles. Secondly, it categorizes riders into two classes, i.e. “With Helmet” and “Without Helmet.” A deep learning algorithm, i.e., Faster R-CNN, is used to detect the bikers without helmets. Figure 3 shows the block diagram of the proposed technique. The following sections describe each component of the proposed technique.

3.1. Data Acquisition. A dataset of bikers with and without helmet is required to develop a system. For data acquisition, three sources include two datasets from existing works [41, 46] and one dataset of self-captured data to accommodate most of the motorcycles running in different countries. The second source includes the surveillance videos captured from Lahore safe city cameras mounted on different roads of Lahore, Pakistan. The captured videos consist of the frontal and back views of the motorcyclists and are converted into frames at the rate of 25 fps. Figure 4 shows sample images from the dataset.



FIGURE 4: Sample images from the dataset.

3.2. Preprocessing. The dataset should be preprocessed to get the appropriate data according to the problem. The obtained dataset contained redundant data, frames with irrelevant images, an incomplete object, etc. Manual preprocessing is done to select appropriate frames from the dataset [47]. Redundant images are removed from the dataset. A total of 23800 frames are selected after preprocessing, i.e., in which 13631 are with helmets and the remaining 10169 are without helmets.

3.3. Annotation. Annotation has been used for image labelling [48, 49]. In this work, a bounding box is drawn around the image. A total of four values are assigned to the bounding box. The label “with helmet” is assigned to images containing bikers with helmets, and the “without helmet” label is assigned to bikers without wearing a helmet. The sample annotated image is shown in Figure 5.

3.4. Faster R-CNN. This work uses Faster R-CNN [50] to detect bikers without a helmet. It is the extended version of the Fast R-CNN [51] and consists of two main modules, region proposal network (RPN) and Fast R-CNN. The RPN guides the Fast R-CNN detection module to find objects in the image [52]. The RPN generates a region proposal, and Fast R-CNN helps to perform object detection from the proposed region. The general architecture of the Faster R-CNN is shown in Figure 6.

This task is performed with the help of a fully convolutional network for sharing computation with a Fast R-CNN object detection network. The RPN takes an image as input (of any dimension) and generates a series of rectangular object proposals along with an objectless score as an output. So, the RPN does not require extra time to

generate the region proposals compared to its competitors like selective search. This sharing of convolutional layers also helps in reducing the training time.

A small window is sided over the feature map for the generation of region proposals. The RPN consists of a regressor and classifier. Classifier tells about the probability of an object at a specific location while regressor tells its coordinates. The aspect ratio and scale are critical parameters for any image, and their values are set to 3. The central part of the sliding window is known as anchor. There are a total of 9 anchors at a position by default. Each anchor is assigned a binary label telling whether an object is present or not. A positive label is assigned to the anchors that either have maximum intersection-over-union (IoU) overlap with a ground-truth box or have IoU overlap greater than 0.7 with any ground-truth box. A negative label is assigned to the anchor if its IoU is less than 0.3. Labels are assigned on two bases, i.e., “the anchors that have high intersection-over-union overlap with a ground truth box” and “the anchors with intersection-over-union overlap which are higher than 0.7.” For the training of RPNs, a loss function given in equation (1) is used [53].

$$\begin{aligned} \text{Loss}(\{a_i\}, \{b_i\}) = & \frac{1}{M_{\text{Class}}} \sum_i N_{\text{class}}(a_i, a_i^*) \\ & + \lambda \frac{1}{M_{\text{reg}}} \sum_i a_i^* N_{\text{reg}}(b_i, b_i^*), \end{aligned} \quad (1)$$

where i indicates the anchor index in a mini-batch and a_i is the probability of anchor i predicted as an object. a_i^* denotes the ground truth label, and its value is 1 or 0, depending on whether the anchor is positive or negative. The coordinates of the predicted bounding box are represented by b_i vector

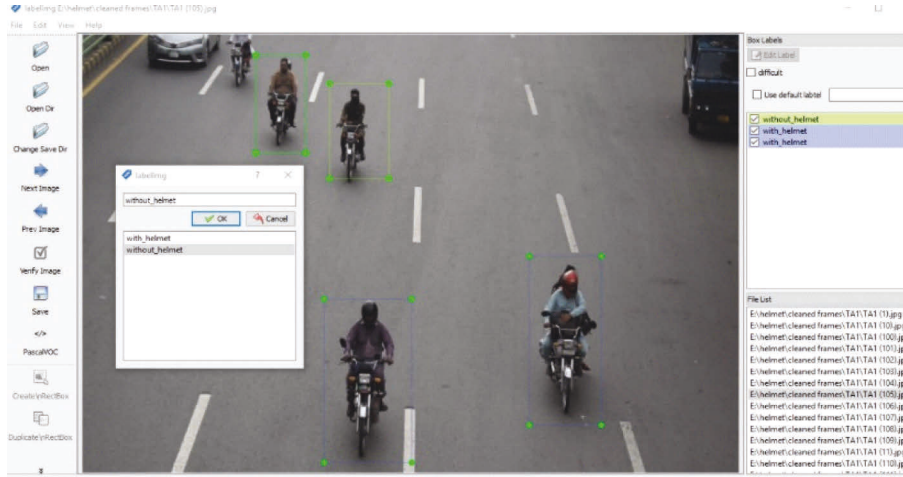


FIGURE 5: Sample annotated image.

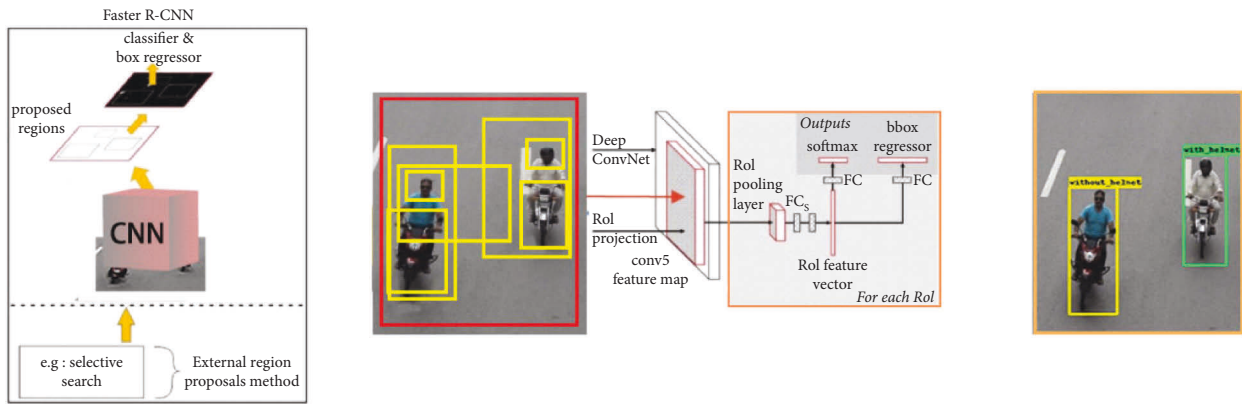


FIGURE 6: The architecture of the Faster R-CNN.

and b_i^* represents the positive anchor's ground truth label. N_{class} is referred to as classification loss. It is the log of the loss over object and non-object classes. N_{reg} is the regression loss, so the expression $a_i^* N_{\text{reg}}$ shows that regression loss has an effect only for positive anchors where $a_i^* = 1$. The classification and regression layers output comprise $\{a_i\}$ and $\{b_i\}$ that are normalized using M_{Class} and M_{reg} , respectively. λ is used as the balancing weight.

The input of the proposed model is cropped helmet image of size $224 \times 224 \times 3$. There are 8 blocks in the backbone architecture, of which 3 are connected layers, and the remaining 5 are convolutional layers. Non-linearities follow each convolutional layer as the max pooling and rectification (ReLU) layer. The outcomes of two of the three fully connected layers are 4049 dimensional. The output of the last connected layer depends on the class present in the dataset and has $N=2622$. The primary purpose of the softmax layer is to handle the un-normalized vectors. It is placed right after the 2nd connected layer. The output of all these is the prediction probability represented in the form of probabilistic scores as shown in the following equation:

$$\text{probabilistic score} = P_c = \frac{e^{P_c}}{\sum_d e^{P_d}} \text{ for all } c \in \{1, 2, \dots, n\}. \quad (2)$$

Table 2 compares Faster R-CNN with other models like Fast R-CNN and R-CNN. The comparison is performed by taking three attributes, i.e., the region proposal method, computation time, and prediction time. Faster R-CNN uses RPN for region proposal instead of a selective search method which is used in R-CNN [54] and Fast R-CNN. Moreover, the computation and prediction time of Faster R-CNN is better than its predecessor, making it appropriate to be used in this work.

4. Experiment Analysis

Core i7 system is used with 32 GB RAM and Ubuntu operating system to develop a proposed technique. For training and validation of the model, GPU GTX 1080 Ti is being used. A dataset that contains a total of 23800 images is divided into two parts, i.e., training and validation. For training and validation of the model, 70% and 30% of the data are used,

TABLE 2: Comparison of Faster R-CNN with other models.

Attributes	Faster R-CNN	Fast R-CNN	R-CNN
Region proposal method	Region proposal network	Selective search	Selective search
Computation time	0.2 seconds	2 seconds	40–50 seconds
Prediction timing	Low computation time	High computation time	High computation time

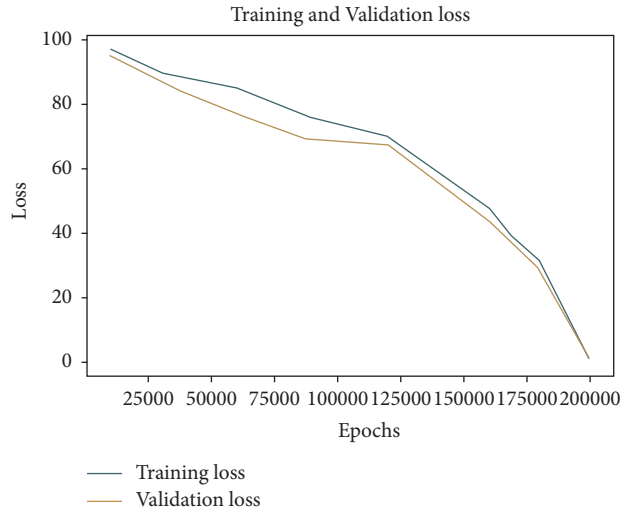


FIGURE 7: Training and validation loss.

respectively. The number of epochs is set to 200000. During the training process, an early stop function is used in which the model is trained until the convergence does not occur. Figure 7 shows the training and validation loss. It indicates that, initially, validation loss is high. But as the training continues, loss gradually decreases. At 200000 epochs, this loss decreases significantly. It is necessary to pass the detected object to the model for the classification of the object.

Figure 8 illustrates the training and validation accuracy. The system attains an accuracy of 97.69%. As training starts, accuracy is low, and the loss is high. As time increases, the accuracy is also increased. The maximum accuracy obtained at 200000 epochs is 97.69%.

The confusion matrix for the proposed technique is shown in Table 3. In this work, 7133 samples are used for validation purposes in which 4089 samples are those who have helmets, and 3044 are those who do not have helmets. For samples that have helmets, 3995 samples are predicted correctly. Only 94 samples are wrongly predicted. For the remaining 3044 samples (without helmets scenario), 71 are wrongly predicted while 2973 are predicted correctly.

Several performance metrics are computed to evaluate the proposed system. Table 4 lists the performance measure metrics and their values. It indicates that proposed system has achieved 97.67% accuracy, 97.70% precision, 97.98% F1 score, and 98.25% sensitivity.

Table 5 lists the comparative analysis of the proposed technique with the existing systems. It reflects that the proposed system gives accuracy of 97.69% and supersedes its competitor.

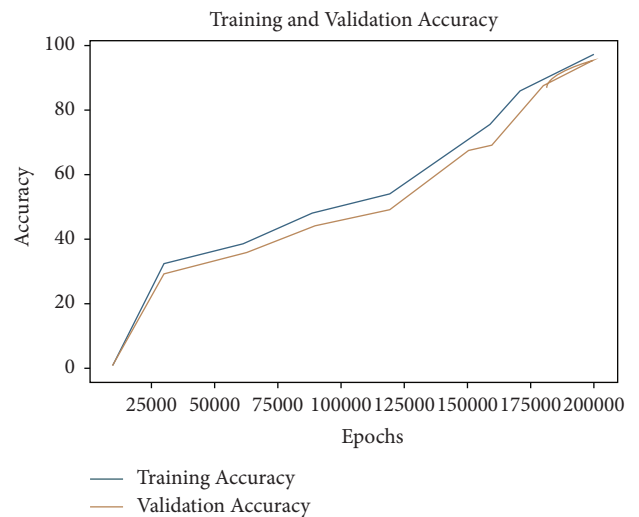


FIGURE 8: Training and validation accuracy.

TABLE 3: Confusion matrix.

$N = 7133$	With helmet	Without helmet
With helmet	3995	94
Without helmet	71	2973

Figure 9 displays some predictions made by the proposed system. The yellow bounding box indicates those motorcyclists who did not wear the helmet, whereas green

TABLE 4: Experimental results.

Performance metrics	Scores	Derivations
Sensitivity	0.9825	$TPR = TP / (TP + FN)$
Specificity	0.9694	$SPC = TN / (FP + TN)$
Precision	0.9770	$PPV = TP / (TP + FP)$
Negative predictive value	0.9767	$NPV = TN / (TN + FN)$
False positive rate	0.0306	$FPR = FP / (FP + TN)$
False discovery rate	0.0230	$FDR = FP / (FP + TP)$
False negative rate	0.0175	$FNR = FN / (FN + TP)$
Accuracy	0.9769	$ACC = (TP + TN) / (P + N)$
F1 score	0.9798	$F1 = 2TP / (2TP + FP + FN)$
Matthews correlation coefficient	0.9528	$TP \times TN - FP \times FN / \sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}$

TABLE 5: Comparison of the proposed technique with existing work.

Author	Technique	Accuracy (%)
Proposed	Faster R-CNN	97.6
Vishnu et al. [39]	CNN	92.87
Dasgupta et al. [53]	CNN	91.08
Afzal et al. [41]	Faster R-CNN	97.26

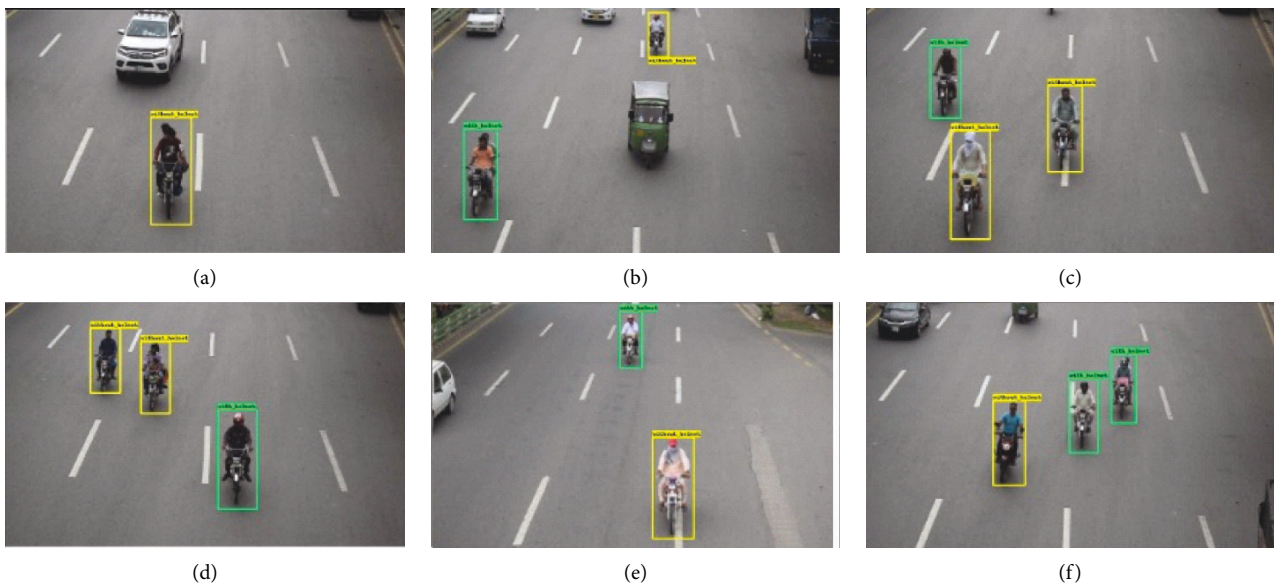


FIGURE 9: Cases predicted by the system.

bounding box represents those who have worn the helmet. In Figure 9(a), the system correctly predicted the helmet violation, and the yellow bounding box encompasses the motorcyclist who did not wear the helmet. Figure 9(b) portrays the case of correct and incorrect prediction of helmet violations. Similarly, in Figures 9(c)–9(f), the algorithm correctly predicted both kinds of motorcyclist, i.e., with and without helmet. It is evident from Figures 9(c) and 9(e) that the proposed system successfully differentiated among helmet, scarf, and cap.

5. Conclusion

Automatic helmet violation detection of motorcyclists from real-time videos is a demanding application in ITS. It enables one to spot and penalize bikers without a helmet. This work proposes an automatic helmet violation detection technique for ITS. The proposed technique is based on Faster R-CNN deep learning model that takes video as an input and performs helmet violation detection to take necessary actions against traffic rule violators. The experimental analysis

shows that the proposed technique achieved 97.6% accuracy. This work may be extended to incorporate more features, like number plate detection and other traffic violations, in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] World population, "World population," 2021, <https://www.worldometers.info/world-population/world-population-by-year/>.
- [2] Vehicle rate, "Vehicle rate," 2019, <https://profit.pakistantoday.com.pk/2019/06/16/registered-vehicles-in-pakistan-increased-by-9-6-in-2018/>.
- [3] M. S. Bajwa, S. Hamid, M. M. Iqbal, S. Tariq, K. Subhani, and L. Suhail, "Changing presentation of traumatic brain injuries in a tertiary hospital Lahore following enforcement of motorcycle helmet laws-A mixed-method study," *Annals of King Edward Medical University*, vol. 27, pp. 85–95, 2021.
- [4] M. Khan and B. Arora, "Traffic congestion reduction and accident circumvention system via incorporation of CAV and VANET," *International Journal of Ambient Computing and Intelligence*, vol. 12, no. 1, pp. 53–72, 2021.
- [5] I. Laña, J. J. Sanchez-Medina, E. I. Vlahogianni, and J. Del Ser, "From data to actions in intelligent transportation systems: a prescription of functional requirements for model actionability," *Sensors*, vol. 21, no. 4, p. 1121, 2021.
- [6] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1840–1852, 2021.
- [7] S. Kaffash, A. T. Nguyen, and J. Zhu, "Big data algorithms and applications in intelligent transportation system: a review and bibliometric analysis," *International Journal of Production Economics*, vol. 231, Article ID 107868, 2021.
- [8] P. Sun and A. Boukerche, "AI-Assisted data dissemination methods for supporting intelligent transportation systems," *Internet Technology Letters*, vol. 4, no. 1, p. e169, 2021.
- [9] D. Gura, I. Markovskii, N. Khusht, I. Rak, and S. Pshidatok, "A complex for monitoring transport infrastructure facilities based on video surveillance cameras and laser scanners," *Transportation Research Procedia*, vol. 54, pp. 775–782, 2021.
- [10] C. Li and P. Xu, "Application on traffic flow prediction of machine learning in intelligent transportation," *Neural Computing & Applications*, vol. 33, no. 2, pp. 613–624, 2021.
- [11] A. Khadhir, B. Anil Kumar, and L. D. Vanajakshi, "Analysis of global positioning system based bus travel time data and its use for advanced public transportation system applications," *Journal of Intelligent Transportation Systems*, vol. 25, no. 1, pp. 58–76, 2021.
- [12] T. Garg and G. Kaur, "A systematic review on intelligent transport systems," *Journal of Computational and Cognitive Engineering*, 2022.
- [13] M. Rizwan, M. Asif, M. B. Ahmad, and K. Masood, "Park my ride: your true parking companion," in *Proceedings of the International Conference on Intelligent Technologies and Applications*, pp. 695–708, Bahawalpur, Pakistan, November 2019.
- [14] A. Hassan, M. S. Abbas, M. Asif, M. B. Ahmad, and M. Z. Tariq, "An automatic accident detection system: a hybrid solution," in *Proceedings of the 2019 4th International Conference on Information Systems Engineering (ICISE)*, pp. 53–57, Shanghai, China, May 2019.
- [15] Traffic accidents, "Traffic accidents," 2012, <https://www.pbs.gov.pk/node/1124>.
- [16] S. Agondeze, S. S. Kizza, P. Vuzi, and C. Ddamulira, "Occupational hazards among laboratory hub riders in selected health centres in central region of Uganda," *Direct Research Journal of Public Health and Environmental Technology*, vol. 6, no. 2, pp. 6–20, 2021.
- [17] N. Akhtar and K. Pathak, "Carbon nanotubes in the treatment of skin cancers: safety and toxicological aspects," *Pharmaceutical Nanotechnology*, vol. 5, no. 2, pp. 95–110, 2017.
- [18] B. C. Liu, R. Ivers, R. Norton, S. Boufous, S. Blows, and S. K. Lo, "Helmets for Preventing Injury in Motorcycle Riders," *Cochrane database of systematic reviews*, vol. 23, 2008.
- [19] C. S. Olsen, A. M. Thomas, M. Singleton et al., "Motorcycle helmet effectiveness in reducing head, face and brain injuries by state and helmet law," *Injury epidemiology*, vol. 3, pp. 8–11, 2016.
- [20] C. Wilson, C. Willis, J. K. Hendrikz, R. Le Brocque, and N. Bellamy, "Speed Cameras for the Prevention of Road Traffic Injuries and Deaths," *Cochrane database of systematic reviews*, vol. 6, 2010.
- [21] M. Mehmood, A. Shahzad, B. Zafar, A. Shabbir, and N. Ali, "Remote sensing image classification: a comprehensive review and applications," *Mathematical Problems in Engineering*, vol. 2022, Article ID 5880959, 24 pages, 2022.
- [22] M. Yang, "Research on vehicle automatic driving target perception technology based on improved MSRPN algorithm," *Journal of Computational and Cognitive Engineering*, vol. 1, pp. 147–151, 2022.
- [23] A. Shahzad, B. Zafar, N. Ali et al., "COVID-19 vaccines related user's response categorization using machine learning techniques," *Computation*, vol. 10, no. 8, p. 141, 2022.
- [24] M. Ali, M. Z. Asghar, M. Shah, and T. Mahmood, "A simple and effective sub-image separation method," *Multimedia Tools and Applications*, vol. 81, no. 11, pp. 14893–14910, 2022.
- [25] F. Ashiq, M. Asif, M. B. Ahmad et al., "CNN-based object recognition and tracking system to assist visually impaired people," *IEEE Access*, vol. 10, pp. 14819–14834, 2022.
- [26] T. Mahmood, R. Ashraf, and C. N. Faisal, "An Efficient Scheme for the Detection of Defective Parts in Fabric Images Using Image Processing," *The Journal of The Textile Institute*, pp. 1–9, 2022.
- [27] J. Meng, Y. Li, H. Liang, and Y. Ma, "Single-image dehazing based on two-stream convolutional neural network," *Journal of Artificial Intelligence and Technology*, vol. 2, pp. 100–110, 2022.
- [28] Y. Yang and X. Song, "Research on face intelligent perception technology integrating deep learning under different illumination intensities," *Journal of Computational and Cognitive Engineering*, vol. 1, no. 1, pp. 32–36, 2022.
- [29] H. B. Ul Haq, M. Asif, M. B. Ahmad, R. Ashraf, and T. Mahmood, "An effective video summarization framework based on the object of interest using deep learning," *Mathematical Problems in Engineering*, vol. 2022, Article ID 7453744, 25 pages, 2022.

- [30] J. Chiverton, "Helmet presence classification with motorcycle detection and tracking," *IET Intelligent Transport Systems*, vol. 6, no. 3, pp. 259–269, 2012.
- [31] R. Silva, K. Aires, T. Santos, K. Abdala, R. Veras, and A. Soares, "Automatic detection of motorcyclists without helmet," in *Proceedings of the 2013 XXXIX Latin American Computing Conference (CLEI)*, pp. 1–7, Caracas, Venezuela, October 2013.
- [32] R. R. V. E. Silva, K. R. T. Aires, and R. d. M. S. Veras, "Helmet detection on motorcyclists using image descriptors and classifiers," in *Proceedings of the 2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 141–148, Rio de Janeiro, Brazil, October 2014.
- [33] R. Waranusast, N. Bundon, V. Timtong, C. Tangnoi, and P. Pattanathaburt, "Machine vision techniques for motorcycle safety helmet detection," in *Proceedings of the 2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013)*, pp. 35–40, Wellington, New Zealand, November 2013.
- [34] K. Dahiya, D. Singh, and C. K. Mohan, "Automatic detection of bike-riders without helmet using surveillance videos in real-time," in *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 3046–3051, Vancouver, Canada, July 2016.
- [35] N. Boonsirisumpun, W. Puarungroj, and P. Wairochanaphuttha, "Automatic detector for bikers with no helmet using deep learning," in *Proceedings of the 2018 22nd International Computer Science and Engineering Conference (ICSEC)*, pp. 1–4, Chiang Mai, Thailand, November 2018.
- [36] K. D. Raj, A. Chairat, V. Timtong, M. N. Dailey, and M. Ekpanyapong, "Helmet violation processing using deep learning," in *Proceedings of the 2018 International Workshop on Advanced Image Technology (IWAIT)*, pp. 1–4, Chiang Mai, Thailand, January 2018.
- [37] F. Wu, G. Jin, M. Gao, H. Zhiwei, and Y. Yang, "Helmet detection based on improved YOLO V3 deep model," in *Proceedings of the 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*, pp. 363–368, Banff, Canada, May 2019.
- [38] F. W. Siebert and H. Lin, "Detecting motorcycle helmet use with deep learning," *Accident Analysis & Prevention*, vol. 134, Article ID 105319, 2020.
- [39] C. Vishnu, D. Singh, C. K. Mohan, and S. Babu, "Detection of motorcyclists without helmet in videos using convolutional neural network," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 3036–3041, Anchorage, AK, USA, May 2017.
- [40] J. Mistry, A. K. Misraa, M. Agarwal, A. Vyas, V. M. Chudasama, and K. P. Upla, "An automatic detection of helmeted and non-helmeted motorcyclist with license plate extraction using convolutional neural network," in *Proceedings of the 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1–6, Montreal, Canada, November 2017.
- [41] A. Afzal, H. U. Draz, M. Z. Khan, and M. U. G. Khan, "Automatic helmet violation detection of motorcyclists from surveillance videos using deep learning approaches of computer vision," in *Proceedings of the 2021 International Conference on Artificial Intelligence (ICAI)*, pp. 252–257, Islamabad, Pakistan, April 2021.
- [42] N. Kharade, S. Mane, J. Raghav, N. Alle, A. Khatavkar, and G. Navale, "Deep-learning based helmet violation detection system," in *Proceedings of the 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV)*, pp. 1–4, Gandhinagar, India, September 2021.
- [43] P. Sridhar, M. Jagadeeswari, S. H. Sri, N. Akshaya, and J. Haritha, "Helmet violation detection using YOLO v2 deep learning framework," in *Proceedings of the 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 1207–1212, Tirunelveli, India, April 2022.
- [44] M. Kathane, S. Abhang, A. Jadhavar, A. D. Joshi, and S. T. Sawant, "Traffic rule violation detection system: deep learning approach," in *Advanced Machine Intelligence and Signal Processing*, pp. 191–201, Springer, Singapore, Asia, 2022.
- [45] N. Rajalakshmi and K. Saravanan, "Traffic violation investigation using transfer learning," in *Proceedings of the 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC)*, pp. 286–292, Chennai, India, April 2022.
- [46] J. E. Espinosa-Oviedo, S. A. Velastín, and J. W. Branch-Bedoya, "EspiNet V2: a region based deep learning model for detecting motorcycles in urban scenarios," *Dyna*, vol. 86, no. 211, pp. 317–326, 2019.
- [47] W. Lotter, A. R. Diab, B. Haslam et al., "Robust breast cancer detection in mammography and digital breast tomosynthesis using an annotation-efficient deep learning approach," *Nature Medicine*, vol. 27, no. 2, pp. 244–249, 2021.
- [48] X. Sun, H. Hao, Y. Liu, Y. Zhao, Y. Wang, and Y. Du, "Research on the application of YOLOv4 in power inspection," *IOP Conference Series: Earth and Environmental Science*, vol. 693, Article ID 012038, 2021.
- [49] A. L. da Costa Oliveira, A. B. de Carvalho, and D. O. Dantas, "Faster R-CNN approach for diabetic foot ulcer detection," in *VISIGRAPP*, vol. 4, pp. 677–684, VISAPP, 2021.
- [50] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.
- [51] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Santiago, Chile, December 2015.
- [52] T. Nazir, A. Irtaza, J. Rashid, M. Nawaz, and T. Mehmood, "Diabetic retinopathy lesions detection using faster-RCNN from retinal images," in *Proceedings of the 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, pp. 38–42, Riyadh, Saudi Arabia, November 2020.
- [53] M. Dasgupta, O. Bandyopadhyay, and S. Chatterji, "Automated Helmet Detection for Multiple Motorcycle Riders Using CNN," in *Proceedings of the 2019 IEEE Conference on Information and Communication Technology*, pp. 1–4, Allahabad, India, December 2019.
- [54] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

Research Article

Classification of Woven Fabric Faulty Images Using Convolution Neural Network

Rehan Ashraf ¹, **Yasir Ijaz**,¹ **Muhammad Asif** ¹, **Khurram Zeeshan Haider** ²,
Toqeer Mahmood ¹ and **Muhammad Owais**¹

¹Department of Computer Science, National Textile University, Faisalabad 37610, Pakistan

²Department of Software Engineering, Government College University, Faisalabad 37610, Pakistan

Correspondence should be addressed to Toqeer Mahmood; toqeer.mahmood@yahoo.com

Received 31 October 2021; Revised 30 April 2022; Accepted 27 July 2022; Published 27 August 2022

Academic Editor: Ardashir Mohammadzadeh

Copyright © 2022 Rehan Ashraf et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Convolution neural network (CNN) is one of the most popular machine learning techniques that is being used in many applications like image classification, image analysis, textile archives, object recognition, and many more. In the textile industry, the classification of defective and nondefective fabric is an essential and necessary step to control the quality of fabric. Traditionally, a user physically inspects and classifies the fabric, which is an ineffective and tedious activity. Therefore, it is desirable to have an automated system for detecting defects in the fabric. To address these issues, this research proposes a solution for classifying defective and nondefective fabric using deep learning-based framework. Therefore, in this research, an image processing technique with CNN-based GoogleNet is presented to classify defective and nondefective fabric. To achieve the purpose, the system is trained using different kinds of fabric defects. The performance of the proposed approach was evaluated on the textile texture TILDA dataset, and achieved a classification accuracy of 94.46%. The classification results show that the proposed approach for classifying defective and nondefective fabric is better as compared to other state-of-the-art approaches such as Bayesian, BPNN, and SVM.

1. Introduction

Fabric texture refers to how the fabric surface feels [1]. Raw material is important to achieve the high quality of the fabrics. This can be achieved with concentration to remove all faults that are on fabrics such as missing needles, dirt spots, hooks, crack points, holes, scratches, fly, color bleedings, oil spots, broken, lack, or any other [2, 3]; some of the fabric defects are shown in Figure 1. There are different competitors in the marketplace; for survival, it should be the ultimate priority for the textile industry to maintain its quality [4]. After manufacturing the fabric, it is categorized into different types, the first category of the fabric is 100% defect-free. Second, the fabric contains some kind of defect on the fabric surface. The defective fabric is sold in 45% to 65% of the first category, and it represents a major loss for any textile industry [1, 5]. However, the quality of the fabric can be improved by applying the latest technologies during

the manufacturing because customer expectations vary with the quality [6]. Therefore, the fabric inspection has a significant role in controlling the fabric quality for any textile industry; without controlling the quality and missing the monitoring of the fabric structure, a manufacturer bears the main loss that results in a downfall in the market as well. In this regard, there are two techniques to improve the quality of the fabric, one is the fabric quality inspection by the human, which is called manual inspection, which is an old strategy to control the fabric quality and has various drawbacks and limitations; the other one is the fabric quality monitoring by an automatic system that overcomes several drawbacks of the manual inspection method [7, 8]. The fabric surface may be velvety smooth, rough, silky, or any other [1]. The texture features depend upon the weaving machine used in the textile industry. The texture is very important for any type of cloth like cotton, silk, leather, wool, flax, or any other. Therefore, there is a slight difference



FIGURE 1: Some fabric defects which occur during manufacturing processing: (a) end-out, (b) soiled-filling, (c) sloughed-filling, (d) mispick, (e) soiled-end, (f) warp-slab, (g) knot, (h) oil spot, (i) end-out, (j) missing-yarn (k) slub (l) hole.

or damaged area on the fabric surface that may create a significant loss for manufacturers [8, 9]. To categorize the fabric, a 4-point system is used in the textile industry; the 4-point system categorizes the fabric on the basis of significance, defect size, and type of defect, as given in Table 1.

In woven fabric, the fabric yarn is formed in the horizontal and vertical directions: the horizontal direction is known as the warp direction, and the vertical direction is known as the weft direction as shown in Figure 2. In woven fabric, the defects appeared in a longitudinal direction (warp direction) or in a horizontal direction (weft direction) these defects appear due to missing yarn. The yarn represents whether the fabric is defective or not, where the defect occurs due to yarn presence or absence, such as end-out, miss-end, and broken-end or picks. The other yarn defects, such as waste or contamination, slubs, and trapped, occur during the weaving process. Some other machine-related defects exhibit structural change (holes or tears) or machine residue (specks of dirt or oil spots). The number of defects and their source of occurrence have been discussed previously. Therefore, it is of high importance to study and propose an automated

TABLE 1: Four-point system.

Defect size	Allocated point
Up to 3 inch defect	Point 1
3 to 6 inch defect	Point 2
6 to 9 inch defect	Point 3
Over 9 inch	Point 4

solution to handle fabric defects that the textile industry is facing that will help to increase revenue as well.

Faulty fabric shackles the overall quality of woven garments such as jackets, trousers, pants, and shirts [8]. In the fashion market, woven fabric defects such as a loose wrap, double end, tight end, hole thick, and thin place [10] are classified as the major defect. In recent years, researchers are proposing deep learning-based frameworks to overcome the challenges of traditional fabric inspection [11, 12].

There are three main challenges to pointing out fabric faults and classifying them. First, there is a lot of fabric, and their characteristics vary; the fabrics can be classified into 17 groups such as pm, pg, p1, p2, p4, p31m, p3, p3m1, p4m,

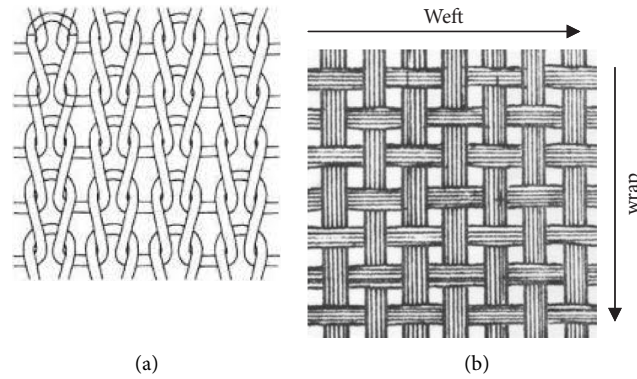


FIGURE 2: Structure of woven fabric: (a) knit fabric and (b) woven fabric.

p4g, pmg, pgg, p6, p6m, cm, cmm, and pmm, and these fabrics or organized repetitively along square, rectangular, parallelogram, hexagonal, and rhombic shapes [13, 14].

In Figure 3, we show some fabric samples with 2D textures. Second, the defects of the fabric also varied, the fabric faults are classified into different 70 categories in the textile industry that are occurred due to different factors such as machine failure, oil spot, and needle problems [13, 15]. Thirdly, to conduct the experiment, there are rarely fabric faulty samples available. There is a wide variety of fabrics, and their faults also vary, and it is a very difficult task for a single system to point out all types of defects, such as end-out, soiled filling, sloughed-filling, misspick, soiled-end, wrap-slab, knot, oil Spot, missing-yarn, slub, and hole as shown in Figure 1. To conduct experiment, we use MATLAB 2018a for this purpose and point out these defects on the surface of the fabric.

Based on the abovementioned discussion, the main contributions of the proposed research work are as follows:

- (i) Accurate classification of fabric faults using GoogleNet.
- (ii) Efficient classification of fabric faults using the correlation factor to deal with the overfitted training data.
- (iii) To the best of our knowledge, it is the first time in textile analysis that GoogleNet has been employed for defective and nondefective fabric classification. Reported results exhibit the efficacy of GoogleNet to classify the fabric faults and computing of a deep and discriminative set of features with improved performance.
- (iv) But with various numbers of patterns, it is hard for people who are unfamiliar with the local fabric faults to remember their details. However, we can define this issue as a computer image recognition problem and use machine learning techniques to help us solve the problem.
- (v) It consists of 22 layers neural network with the combination of layers of convolution, max pooling, softmax, and a new idea of inception module. The proposed inception layer is to find the optimal local construction in each layer and repeat it spatially.

Each “inception” module is the construction of the different sizes for each convolution node (1×1 , 3×3 , and 5×5) and 3×3 max pooling node (see Figure 4).

2. Literature Review

Deep learning- and computer vision-based techniques are used in various applications such as medical image analysis, objection detection, and action recognition [16–20]. Recent research is focused on the use of midlevel features and deep learning models to build robust decision support systems and IoT applications [21–24]. Moreover, the researchers are applying these methods for the classification and detection of fabric faults.

Tong et al. [25] demonstrated that the Gabor filter can be used for fabric inspection. In the proposed scheme, Gabor filters are used for linear filtering to analyze whether the fabric region is affected or not. For segmentation purposes, the author used a threshold value, and the experiment was conducted on TILDA dataset which includes 50 non-defective samples and 300 defective samples. In the experiments, 90.0% sensitivity and 87.1% accuracy are achieved. The authors pointed out the three main defects of the fabric. The first two are structural defects that are related to weaving texture, and the third one is the tonal defect. The tonal defect changes the local intensity value. Kaur et al. [26] projected the Gabor technique to address the faulted texture by using digital image processing techniques. Colin Sc Tsang and his team represent a novel Elo rating method for fabric inspection from the uniform background of the fabric. Colin Sc Tsang et al. used a novel Elo rating (ER) method to identify the faulted fabric from the uniform background. The purpose of this inspection is to detect, identify, and locate any defect in the fabric to maintain its quality in the manufacturing industry. Anandan et al. [8] used different techniques for detecting fabric inspection combining aspects such as GLCM (Gray-level co-occurrence matrix) and CT (curvelet transform). In this work, three main flaws of fabric are pointed out, such as holes, spots, and lines on the fabric surface. For the feature extraction (spots) from the fabric surface, the author used the blob algorithm, and to point out the holes on the fabric surface, Peng et al. [27] used the

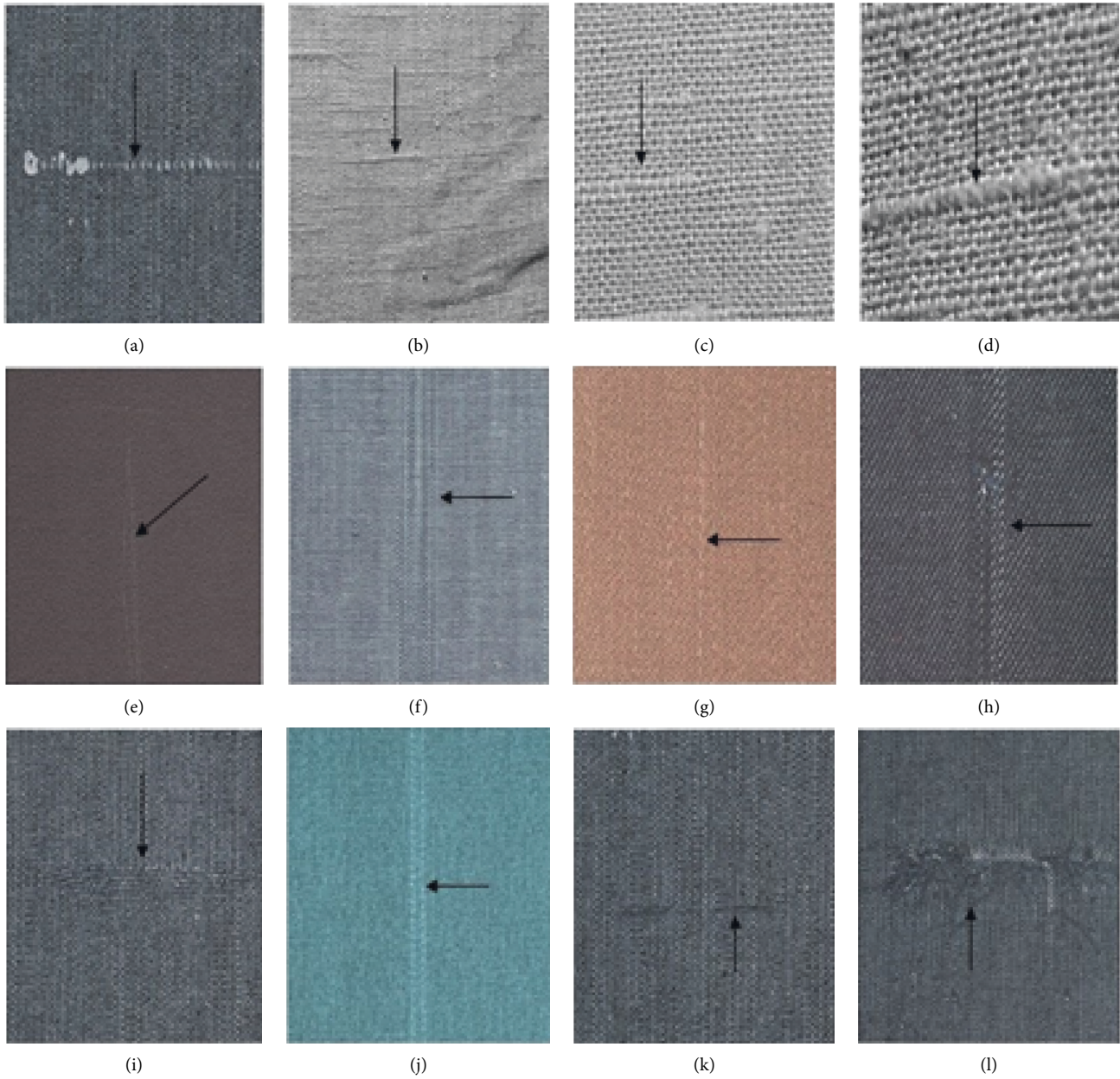


FIGURE 3: Defective fabric samples with complex patterned textures.

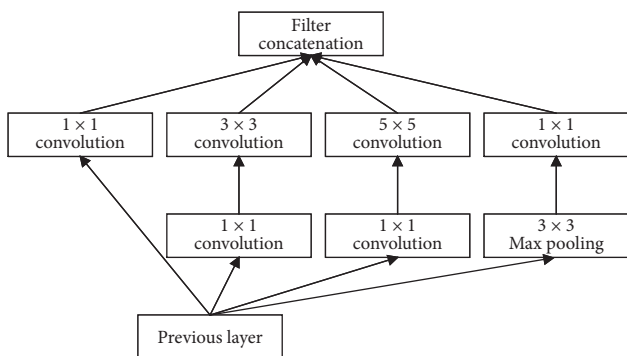


FIGURE 4: Structure of the inception module from the GoogleNet network.

Canny feature extraction algorithm, and the Gary feature extraction method was used to point out the lines on the fabric surface. Selvi et al. [28] demonstrated that fabric fault detection using image processing techniques and ANN is unique and prominent compared to the other ones.

Zhang et al. demonstrate the Euclidean distance for color dissimilarity with KNN (K-nearest neighbor) techniques used to separate the defective region from the nondefective region by using digital image processing techniques. Hanmandlu et al. [29] demonstrate the FDT fuzzy decision tree used to address the fabric inspection; in this scheme, four algorithms are used to extract features such as LBP (local binary patterns), SIFT (scale-invariant feature transform), LDP (local directional patterns), and SURF (speeded up robust fractures). For the classification purpose, the authors

used both fuzzy Shannon entropy and fuzzy Gini index. For classification purposes, Hanmandlu et al. [29] used a decision tree classifier and fuzzy decision tree. By using these classifiers, they got a 91.5% result. The dataset contains defective and nondefective silk samples, which contains a total of 250 silk samples that were divided into 50 classes, and each class contains five silk samples where 40 classes contain defective, and ten classes contain nondefective silk samples. Based on SIFT and SURF, they got a maximum of 100% results. The results of LBP features using linear kernel, and a maximum of 96% accuracy is achieved. Aldemir et al. [30] demonstrated the linear and nonlinear techniques for fabric inspection using Gabor filtering. Wang et al. [31] used to address whether the fabric is faulty or not, namely, gray-level statistical and morphological methods. Dhawas et al. [32] demonstrate fabric inspection can be categorized into three techniques such as statistical, spectral, and model based. Prajakta et al. [33] combined computer vision methodology with neural networks to identify the classification of textile defects. Karunamoorthy et al. applied artificial neural network to classifiers to separate the fabric faults from the uniform background. Classification of the fabric inspection using the structural approach are classified into three categories, namely, statically, spectral and model approach. The structural approaches point out the individual pixel from the uniform background of the fabric surface. According to Nasira [28], the structural approaches were not successful for the fabric inspection due to the stochastic variations in the fabric layout. The first statistical approaches are used for the distribution of pixel value. The main objective of this approach is to classify the defective region from the defect-free region with distinct statistical behavior. The second spectral approach is applied only to uniform textured materials like fabric due to the high degree of periodicity [7]. Therefore, the spectral approach is used to extract the feature, which is less sensitive to noise. The third approach which is model-based is used to extract the features from the faint aligned region. There are several techniques used for automated fabric defect detection. Among them, namely, clustering, SVM (support vector machine), neural network, and statistical are more useful among others [34].

3. Proposed Methodology

It is desirable to have such a system and classification for fabric inspection that should be able to cope with the other various types of fabric defect detection methods that are highlighted in the literature. The fabric inspection means extracting the texture to demonstrate whether the fabric is defective or defect-free. The fabric defects are detected on the basis of calculated fabric features.

Due to defects, the fabric structure differs from the uniform background. Therefore, the fabric inspection is performed by monitoring the fabric structure. The surface of the fabric may contain different types of flaws that occur during the manufacturing process; therefore, it is very important to measure the fabric quality. Typically, it is a critical need within the textile industry to point out the fabric

defects and classify them before delivering them to the end user. First, there is the training phase in which the defective and defect-free formations are used as reference for the base features, and then the convolution neural network is applied to save the network parameters with the feature vector. Second, there is the defect testing phase, in which the fabric is labeled and classified into categories on the basis of certain features. There are two most important concepts, that is, correlation and convolution, used to extract the information from the image. In correlation, the matching of the neighboring patterns or masks is performed; it checks the similarity between two signals or sequences. Besides, the convolution method is used to measure the effect of one signal on other. The block diagram of the proposed scheme is presented in Figure 5.

3.1. Image Filtering and Enhancement. In preprocessing after image acquisition, we applied the image enhancement techniques. Sometimes, we needed to remove the noise or filter the image before processing it. The other terms, such as filtering, conditioning, or enhancement, were used for the same purpose. The image contains the structure or signal extracts to differentiate the interesting and uninteresting region by monitoring the pixel or its local neighborhood. Image processing contains several methods and theories to enhance the image and present the significant notation of the image.

3.1.1. Histogram Visualization. The histogram equalization approach is used to improve the demarcation in image. The histogram visualization formula calculates and displays the frequencies of values in the image dataset as shown in Figure 6. The target output image uses all gray values such as $z = z_1, z = z_2, \dots, z = z_n$. Each gray level uses approximately $q = (R_o * C_o) / n$ time, where R_o and C_o are used for rows and columns of the image. $H_i n[i]$ is used for stretching function; $H_i n[k]$ is used for the pixel, which has gray level z_i . t_1 is used for the first gray level threshold, and q_1 is used for all pixels of the input image. The $H_i n$ is used for stretching function f .

$$\sum_{i=1}^{t_1-1} H_i n[i] \leq q_1 < \sum_{i=0}^{t_1} [H_i n][i], \quad (1)$$

$$\sum_{i=1}^{t_k-1} H_i n[k] \leq (q_1 + q_2 + q_3 + \dots + q_k), \quad (2)$$

$$< \sum_{i=0}^{t_1} [H_i n][i].$$

According to equation (1), t_1 is the smallest gray level. The original histogram contains only less than or equal to the gray level value. In equation (2), t_k contains the k th threshold.

3.1.2. Gaussian Filtration. The filtration process is used for the sake of blurring the image and removing the noise. Mathematically, Gaussian filtering modifies the input signal by

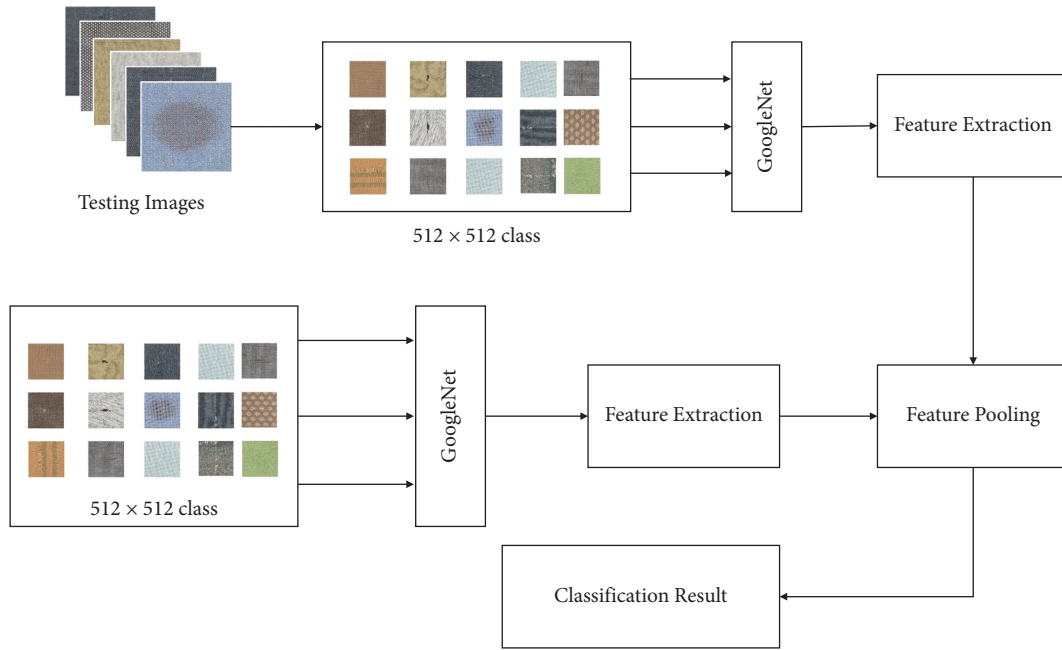


FIGURE 5: A systematic view of the proposed deep convolution neural network framework.

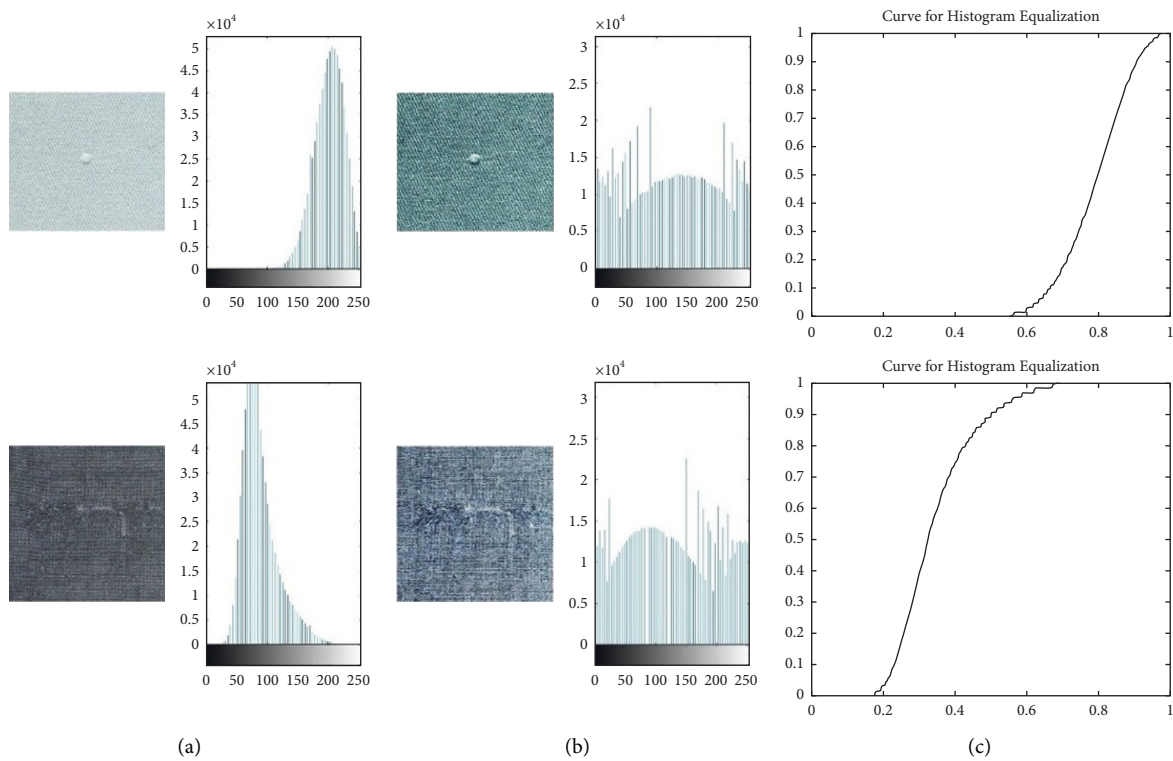


FIGURE 6: Result of histogram visualization: (a) native image, (b) enhanced image, and (c) curve for histogram equalization.

convolution with Gaussian function. During the experiment, we applied the Gaussian low-pass filter and Gaussian high-pass results to produce a significant result, as shown in Figure 7 and

its graph in Figure 8. Equation (3) works for one-dimensional Gaussian function, and equation (4) works for two dimensions. The standard deviation of the distribution is assumed to be zero.

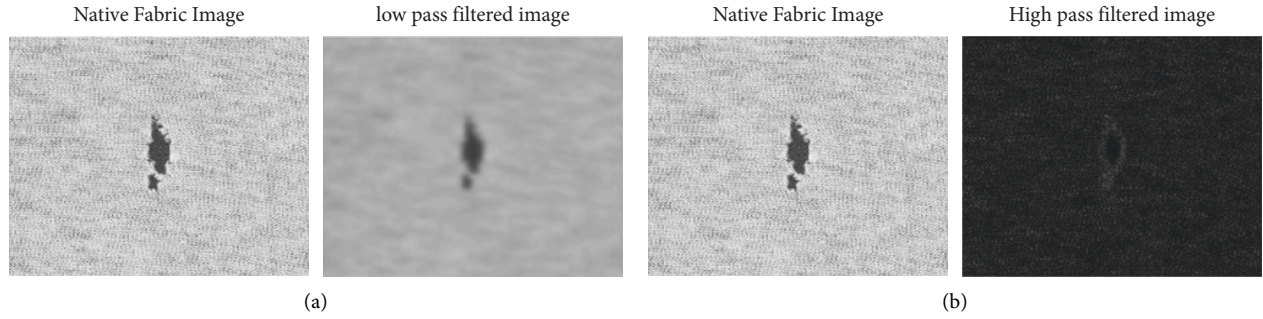


FIGURE 7: Defective fabric result of the Gaussian filter: (a) low-pass filter and (b) high-pass filter.

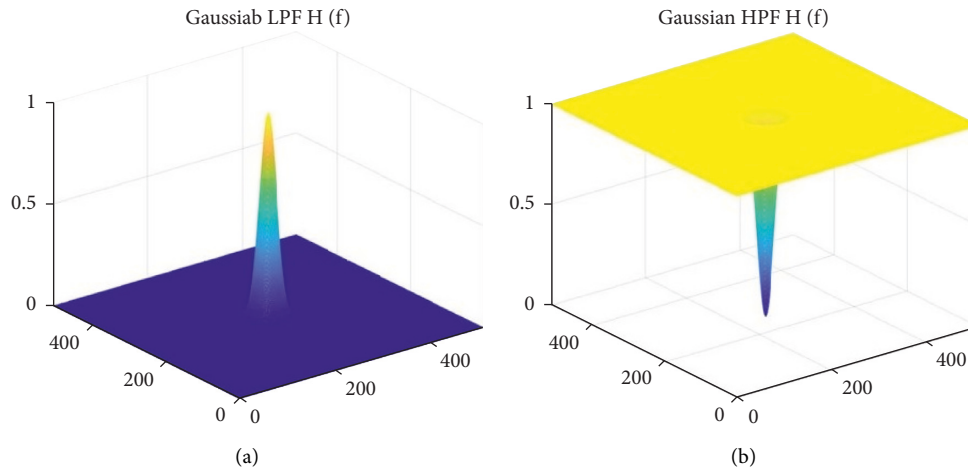


FIGURE 8: (a) Gaussian low-pass filter and (b) Gaussian high-pass filter.

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \varepsilon - \frac{x^2}{2\sigma^2}, \quad (3)$$

$$G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} \varepsilon - \frac{x^2 + y^2}{2\sigma^2}. \quad (4)$$

3.1.3. *Fourier Transform Method.* The Fourier transform method describes that any signal can be represented by the sum of sine and cosine waves with various frequencies and amplitudes. The two-dimensional Fourier transforms can show the relationship between the uniform fabric structure, regular structure, and repetition in the image space and its spectrum. The Fourier transform function (FTF) is used to monitor and describe the relationship between the regular structure of the fabric and its Fourier spectrum; the faults on the fabric surface can be pointed out if the periodic structure has changed on the Fourier spectrum. Notably, the aforementioned methods are used to analyze the fabric structure in the spectrum domain. The cross-sectional and FFT are utilized to analyze the fabric structure: the wrapped yarn of the fabric appears in the vertical direction and stores the information about its feature as f_{y1} , and the horizontal direction or weft, as f_{x1} . Therefore,

$$\begin{aligned} K1 &= |F(0, 0)|, \\ K2 &= |F(f_{x1}, 0)|, \\ K3 &= f_{x1}, \\ K4 &= \sum_{f_{xi}=0}^{f_{x1}} |F(f_{xi}, 0)|, \\ K5 &= |F(0, f_{y1})|, \\ K6 &= f_{y1}, \\ K7 &= \sum_{f_{yi}=0}^{f_{y1}} |F(0, f_{yi})|. \end{aligned} \quad (5)$$

Where the feature $k1$ represents the characteristics of the fabric structure irregularity. Features $K2$, $K3$, and $K4$ are used to detect the change in the wrap or vertical direction, whereas $K5$, $K6$, and $K7$ detect the change in the weft or horizontal direction.

The computational time for the FT is generally long: the discrete Fourier transform (DFT) for the two-dimensional is proportional to the second-order of the image. Generally, the fast Fourier transform (FFT) is used to reduce the size of the Fourier transform. If the FFT is one-dimensional, then its computation time is $N \log_2 N$. For the two-dimensional FFT, the computation time is $2N^2 \log_2 2N$. In the 2-


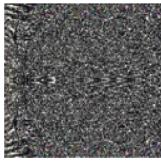
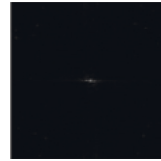

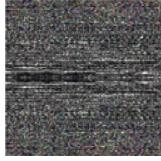
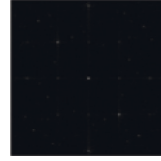

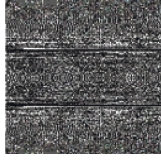
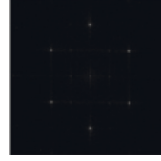

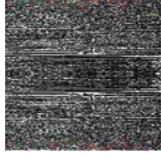

Defect Type	Defective sample	FFT Result	Fourier Spectrum
Hole			
spot			
Thread Defect			
Broken End			

FIGURE 9: Standard fabric defects and their FFT and Fourier spectrum results.

dimensional image, FFT performs 1D for rows, which converts each row of the image and 1D for columns that convert each column. In Figure 9, the results of FFT in different phases are displayed, and its phase spectrum results are presented in Figure 10.

3.2. Edge Detection Techniques. In image edge detection, the defect boundaries or discontinuities within the image are detected by computing the difference in the local image region. First, we implement the detection on one-dimensional: the one-dimensional signal only contains the rows or columns, and in 2D, the rows and columns are computed one by one by first calculating the rows and second the columns of the two-dimensional images. In image processing, edge detection techniques are used to address the target line and ignore the irrelevant information from the image.

For this purpose, the image segmentation techniques are used to partition the image into multiple segments. Actually, the object is highlighted in the image when it has a texture or color different from that of the uniform background. The image consists of the number of pixels that have multiple colors in RGB, and the adjacent pixel is different from the other. The edge is detected in those pixels that are significantly different from the others. In image processing, the prominent task to detect the specific region that is different from the background is called image segmentation. There are multiple edge detection techniques in image processing, such as Prewitt edge detector, Sobel edge detector, Canny

edge detector, Kirch's, and log edge detector. The edge detection techniques are used to highlight the defective region. During the experiment, we applied Sobel, Canny, and Prewitt edge detector to point out the fabric defects, and the results are shown in Figure 11.

The process for classifying the woven fabric fault using digital images through the proposed framework is depicted in Algorithm 1.

4. Experimental Results

4.1. Dataset. For the evaluation of the results, we utilized the TILDA dataset. The entire dataset consists of 3200 images. The dataset is composed of different types of fabric and their defects. The fabric patterned texture is different in this dataset; we evaluated our results using TILDA. The dataset contained several types of flaws, but we considered only some of them as shown in Figure 12. The standard TILDA dataset is composed of 24 defects according to the Ministry of Textiles. There was a total of 1550 images of the fabric that were considered. In addition, during the experiment, we split the dataset into 80% and 20% ratios for training and testing. The major ratio was used to train the model, and the remaining dataset was used for testing. However, during the experiment, we change the ratio for training and testing, but using this ratio, we obtained significant results.

4.2. Experimental Framework. There are several techniques that were used to check similarities and for classification purposes in the past decades; there are two major techniques used for classification purposes in image processing: the first is to calculate the features and then apply the machine learning techniques that is the domain-based approach that degrades the results when the number of classes increase or the domain changes. The second one is the convolution neural networks (CNNs) that show remarkable performance in the field of image processing [35]. A typical CNN has several building blocks, namely, convolutional, pooling, and fully connected layers. CNN extracts the features automatically instead of relying on the handcrafted features, which used the weights of the network learned from ImageNet. The architecture of the CNN is shown in Figure 13.

4.2.1. GoogleNet. GoogleNet is architecture of a convolution neural network; GoogleNet is a pretrained deep neural network that has 22 layers, and the inception network is pretrained which can classify the fabric samples into their categories such as defective or defect-free, as well as also label the fabric defect type. In MATLAB, we trained the inception network with several types of defective and defect-free fabric samples. The inspection network for CNN is also called Google brain. GoogleNet is a deep convolution neural network; codename is the inspection network that is state-of-the-art for classification in ImageNet large scale visual recognition challenges 2014 (ILSVRC14). During the training phase, GoogleNet learns rich features and then takes an input image and addresses it to cross-pounding

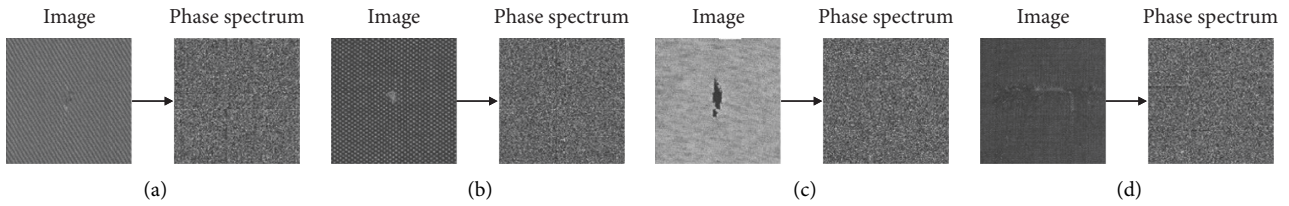


FIGURE 10: Fabric spectrum results after applying fast Fourier transform.

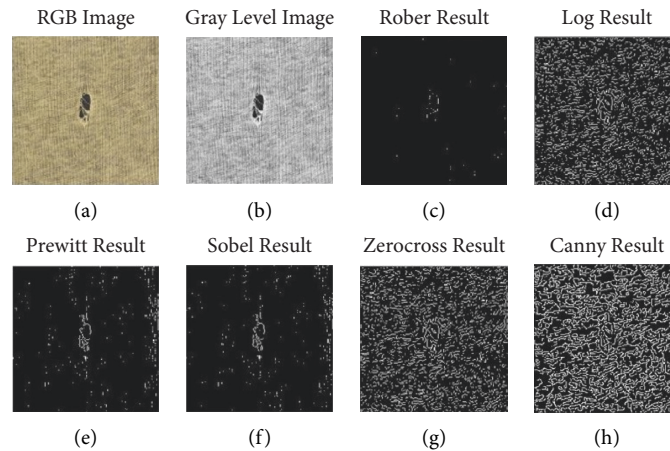


FIGURE 11: Edge detection by different techniques: (a) RGB real image, (b) Gray level image, (c) Rober results, (d) log results, (e) Prewitt results, (f) Sobel results, (g) zero cross results, and (h) Canny results.

Input: Fabric images (FI) $M [r, c]$: set of features.
Output: Classification of woven fabric images $N [r, c]$ as defective and nondefective.

- (1) Deep learning-based features are extracted.
- (2) The classifier is trained with deep learning-based features.

Begin

- (3) Computer histogram equalization.
- (4) Computer Gaussian filtering.
- (5) Computer Fourier transform.
- (6) Computer edges of the fabric faults after applying the inverse Fourier transform.
- (7) Computer deep learning features
- (8) for training samples (TestSi, TrainSi) do
 Train the classifier
- (9) end for
- (10) The classification results

End

ALGORITHM 1: Process for woven fabric fault classification.

categories. GoogleNet is the deeper network with computational efficiency, which is the ILSVRC 14 classification winner; GoogleNet works with 22 layers that are not fully connected.

The proposed model requires less space and provides significant results for classification compared to other state-of-the-art schemes such as VGG and Alexnet. The architecture of GoogleNet is shown in Figure 14; GoogleNet requires 5 million parameters, while Alexnet requires 16 million parameters. In this network, three types of filter work such as 1×1 , 3×3 and 5×5 , and 1×1 are used to reduce the size. The

inspection network incorporates the same concept in layers as Alexnet and VGG since every layer has all possible filter sets such as 1×1 , 3×3 , 5×5 , and a full convolution network as shown in Figure 4. Therefore, the system has multiple filter sets; the learning of each layer by the back prop is updated on the basis of objective functions; layers of GoogleNet after transfer learning are shown in Figure 15.

Nowadays, the research trends for classification have shifted toward CNN. In the deep learning framework, the activation function determines the output of a deep learning method that can be expressed as:

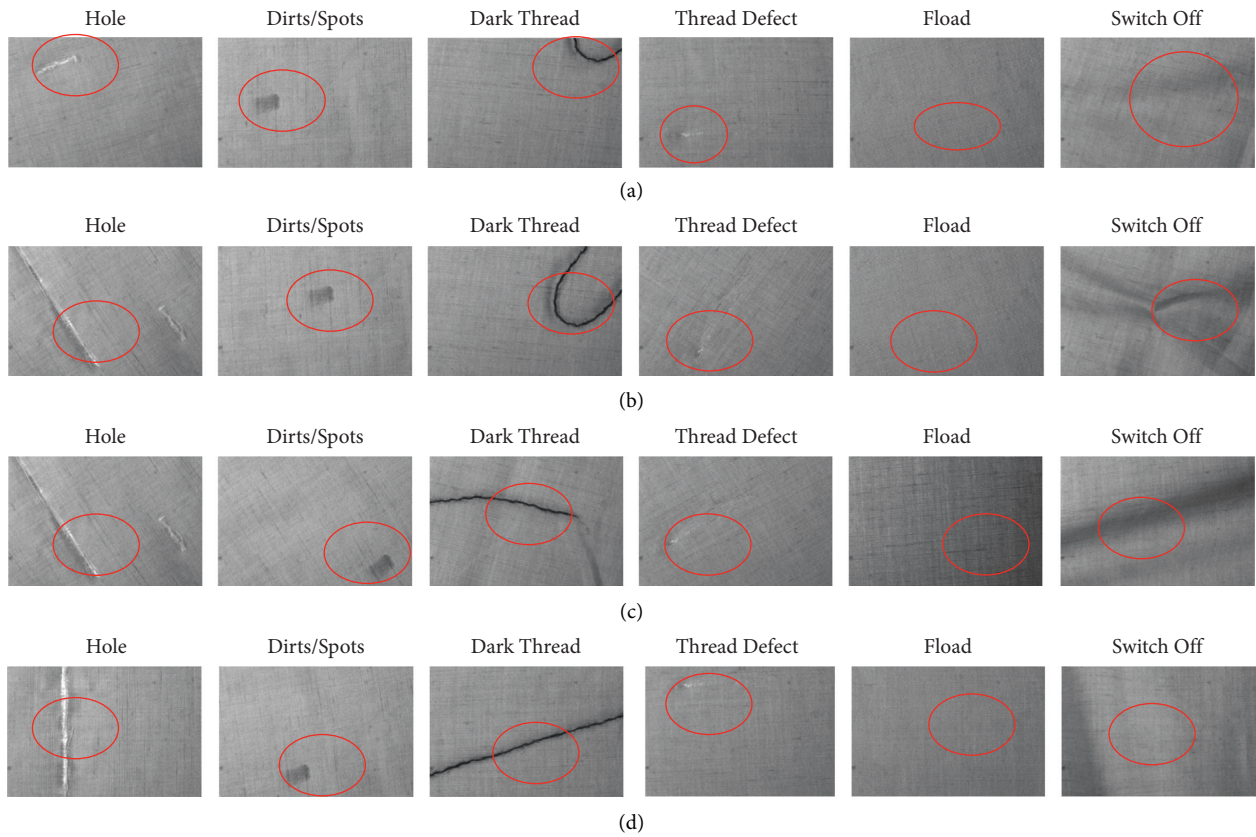


FIGURE 12: Six standard defective fabric samples that are considered: (a) hole, (b) spots/dirt, (c) thread defects, (d) darks threads, (e) flood, and (f) switch off.

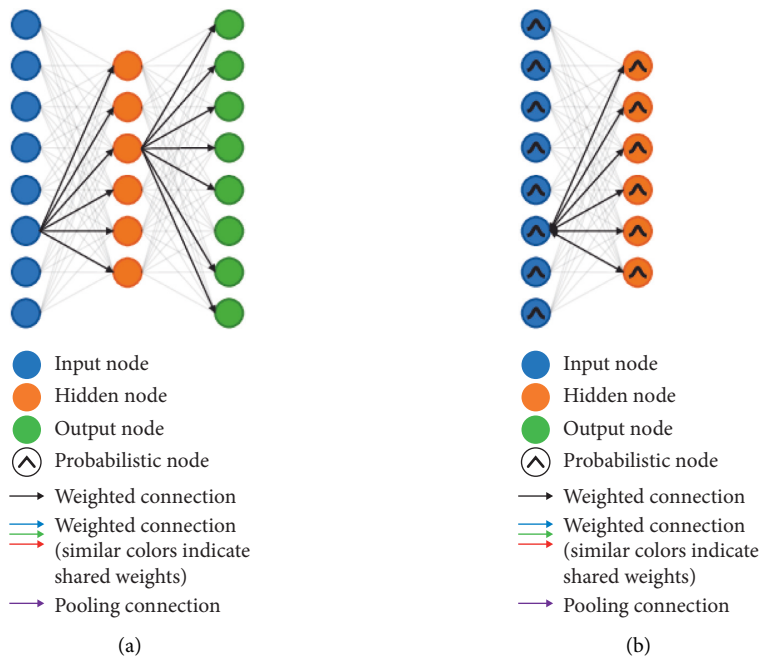


FIGURE 13: Continued.

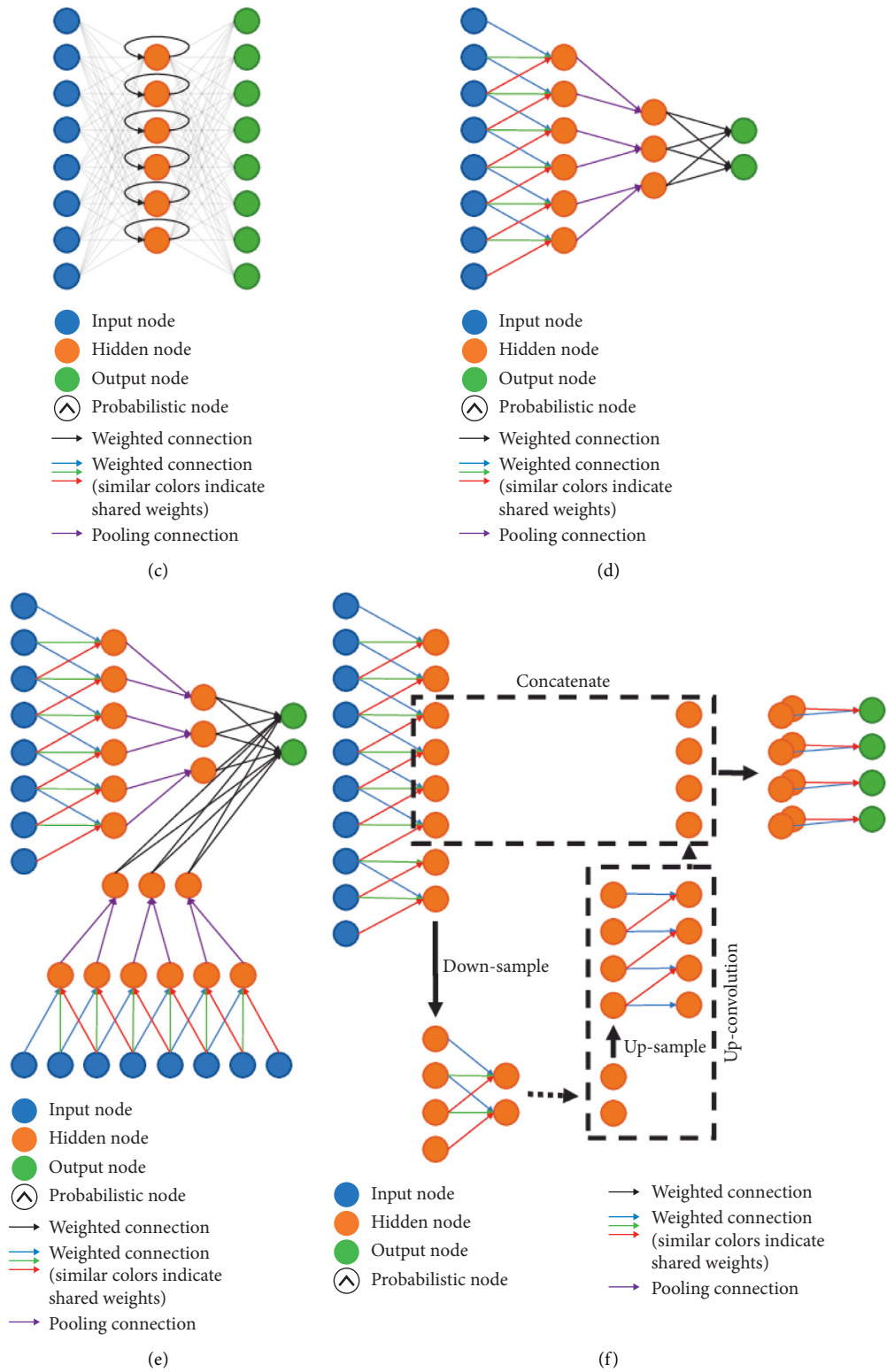


FIGURE 13: 1D NN architecture: (a) auto encode, (b) Boltzmann machine, (c) recurrent NN, (d) CNN, (e) multistream CNN, and (f) DCNN.

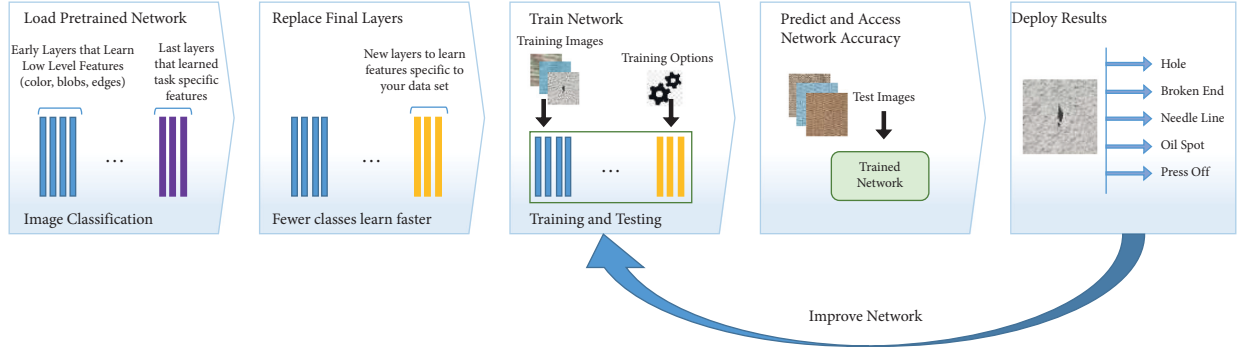


FIGURE 14: GoogleNet neural network architecture.

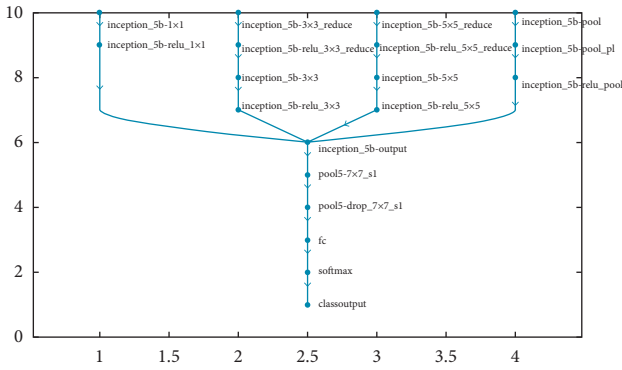


FIGURE 15: Layers of the GoogleNet method after transfer learning.

$$F(b | a; \theta) = \text{softmax}(a; \theta) = \frac{e^{(Z_i^t)^T x + y_i}}{\sum_{m=1}^m e^{(Z_k^F)^T x + y_i}}. \quad (6)$$

Here, Z_F shows the weights of the output layer. To calculate the value of θ , stochastic gradient is applied. The clustering value of biases and weight with Fx in the $i \times j$ dimension. One-dimension CNN with all layers of the network by using different kernels are shown in Figure 13. The weight and biases of CNN are mathematically expressed as:

$$\begin{aligned} W &= \{w_1, w_1, w_1 \dots w_k\}, \\ B &= \{b_1, b_1, b_1 \dots b_k\}, \\ W' &= \text{argMax}M(x) + B'E(x), \\ B' &= \text{argMax}M(x) + B'W'K(x). \end{aligned} \quad (7)$$

$M(x)$ is used to normalize the preprocessing and formation of the feature vector with the equations (2) and (3). The needed features are extracted by using the proposed deep CNN feature extractor for training and testing purposes.

We trained the network with the input images having dimensions 512×512 . The inception network extracts the features and performs other filters to conclude the result of whether the fabric has defects or not. To check the similarities and measure the distance, we used the Euclidean and Manhattan distance formulas. The classification network is

modeled with GoogleNet-based convolution neural network architecture to learn the structural fabric features; the systematic view of the proposed deep convolution neural network framework is shown in Figure 6. The results are verified by another prominent classifier such as a support vector machine (SVM) or back propagation neural network (BPNN); the results of BPNN and SVM on the same defects are given in Table 2.

4.3. Evaluation Metrics. There are different metrics that are used for defect inspection, detection rate (DR), detection accuracy (Dacc), false alarm rate (FR), recall (R), and precision (P). For the evaluation of the classification, problems normally used accuracy. It is the ratio of the accurate prediction and total prediction by the system. The obtained quantitative result is given in Table 3 and their graphical representation is presented in Figure 16, to compute the Dr, Fr, and Dacc, we follow the equations (8)–(10).

$$D_R = \frac{TP}{N_{def}} * 100\%, \quad (8)$$

$$F_R = \frac{FP}{N_{free}} * 100\%, \quad (9)$$

$$D_{acc} = \frac{TP + TN}{TP + TN + FP + FN} * 100\%. \quad (10)$$

In equation (10), N_{def} refers to the number of defective samples and N_{free} refers to the nondefective samples, and the TP and FP are the ratios of defective samples that are detected as defective or defect-free. The TN and FN are the ratios of nondefective samples that are labeled as defect-free after the evaluation. Pixel-level metric evaluates the inspection accuracy to predict the accuracy by measuring the predicted pixel. TP true positive refers to the foreground defective segmented area, and FP false positive background area refers to areas that were defective but not detected as shown in Figure 17. To calculate the precision, recall, and metrics measure, we used the equations (11)–(13) [41, 42], and their obtained results are given in Table 4, and graphical representation is given in Figure 16. The result compared with other techniques is given in Table 3, and graphical representation is given in Figure 18.

TABLE 2: Comparison performance index of the proposed technique with another classifier.

Classifier	Hole (%)	Spot (%)	Thread defect (%)	Dark thread (%)	Flood (%)	Switch off (%)	Average (%)
SVM	92.3	92.8	93.2	91.06	93.06	93.05	92.5
BPNN	90.08	88.05	93.05	91.06	87.08	86.9	89.4
Proposed deep CNN	93.36	93.02	94.35	93.69	96.35	96.01	94.46

TABLE 3: Classification performance of the comparative methods.

Schemes	Precision	Recall	F1 measure	Accuracy
Hog-based KNN [36]	74.61	74.10	74.12	74.10
Walwet-based BPNN [37]	86.72	86.00	85.98	81.97
Kumar et al. [2]	79.3	79.1	80.2	79.7
Hu et al. [38]	87.4	87.9	83.5	85.7
Mak et al. [39]	82.6	78.0	83.5	80.8
Hu et al. [40]	75.5	71.4	87.9	79.7
Purposed CNN	83.66	83.5	83.56	94.46

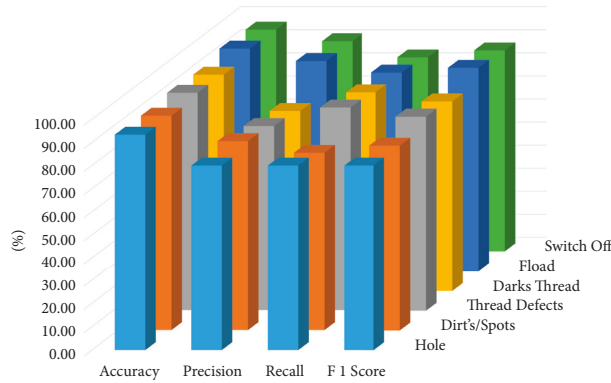


FIGURE 16: Classwise comparison with respect to precision, recall, and F score.

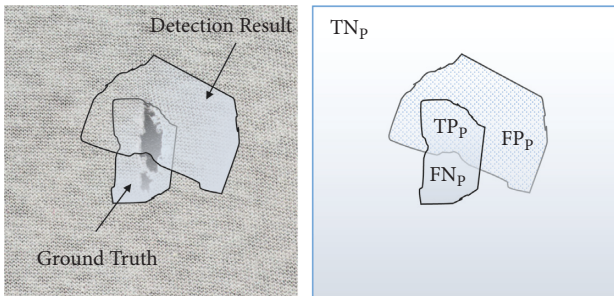


FIGURE 17: TN_p , FN_p , TP_p , and FP_p indicators.

$$R = \frac{TP_p}{TP_p + FN_p} * 100\%, \tag{11}$$

$$P = \frac{TP_p}{TP_p + FP_p} * 100\%, \tag{12}$$

$$\text{Metrics Measure} = 2 \frac{P.R}{P + R} * 100\%. \tag{13}$$

Here, R is recall, P is precision, and the metrics measure indicates F1 measure.

In this work, we also compared the results with the SVM and BPNN models and its results in Table 2, and the graph is

presented in Figure 18. A significant difference is observed by using the proposed model. As given in Tables 3 and 2, we can conclude that the proposed method only obtained a classification accuracy of 94.46%, while the other relevant schemes obtain less than that of the proposed scheme. Therefore, on the basis of obtained results, we can say that the proposed model based on GoogleNet architecture is a robust woven fabric classification CNN that is able to extract texture features for recognition and classification.

5. Discussion

The GoogleNet pretrained CNN architecture is used for the classification of the defective woven fabric images, divided into six classes such as hole, spot, dark thread, thread defect, flood, and switch off. In this work, we used 80% data for the training of the model and 20% for the validation of the model. Among the several types of defects, six major defects in fabrics are considered, as shown in Figure 12. The experiments exhibit that the classifier performed well for distinguishing the defective and nondefective fabric; the obtained accuracy is given in Table 4 and the graphical representation is given in Figure 16. The presented technique is also compared with other techniques in the literature for fabric defect classification. The obtained results of the mentioned defects and their accuracy, precision, recall, and

TABLE 4: Performance evaluation of our proposed model.

Classes	Defects	Accuracy (%)	Precision	Recall	F1 score
1	Hole	93.36	0.80	0.80	0.80
2	Spot	93.02	0.82	0.77	0.80
3	Thread defects	94.35	0.80	0.88	0.84
4	Dark thread	93.69	0.78	0.86	0.82
5	Flood	96.35	0.91	0.86	0.88
6	Switch off	96.01	0.91	0.84	0.87

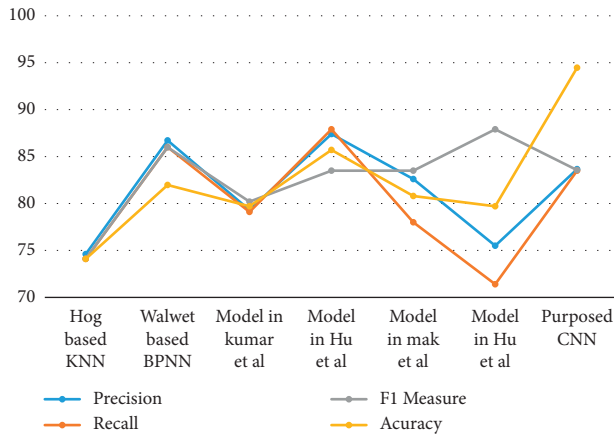


FIGURE 18: Overall comparison with respect to precision, recall, F1 measure, and accuracy with other methods.

F1 measure are given in Table 3, and the graphical representation is given in Figure 18. After comparing the overall results, it is concluded that the purposed technique utilizing a deep neural network provides accurate results as compared to other schemes, with an overall average accuracy of 94.46%.

6. Conclusion

In this article, we have presented computer-based fabric classification to address whether the fabric is defective or nondefective on the basis of fabric features of various shapes, sizes, and locations. Major woven fabric has been used for fabric material. The proposed deep neural network architecture utilized supervised learning to address the defective and defect-free fabrics. It is a deep neural architecture-based system that is trained by getting the fabric features to classify a large amount of fabric, such as woven fabric, single warp, double wrap, and double knit fabrics. Our proposed network does the following: (1) provides a more accurate result and has a low false alarm than other state-of-the-art schemes. (2) According to the experiment result for the complex patterned fabric we achieved an average accuracy of 94.46%. (3) The proposed scheme shows more robustness for different types of patterned fabrics. (4) The convolution neural network significantly differentiates between the defective or nondefective fabric. In future work, we intend to analyze fabric defects using fuzzy-based algorithms in combination with deep neural networks. The fuzzy analysis represents methods for solving problems related to uncertainty and vagueness. To improve the results, it has been employed in multiple applications of science and engineering [43–51].

Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] D. Yapi, M. S. Allili, and N. Baaziz, "Automatic fabric defect detection using learning-based local textural distributions in the contourlet domain," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1014–1026, 2018.
- [2] A. Kumar and G. K. Pang, "Defect detection in textured materials using Gabor filters," *IEEE Transactions on Industry Applications*, vol. 38, no. 2, pp. 425–440, 2002.
- [3] T. Nishimatsu, E. Toba, and T. Sakai, "Difference of eye-movements between experts and non-experts in fabric inspection," *Journal of the Textile Machinery Society of Japan*, vol. 41, no. 4, pp. 104–108, 1995.
- [4] E. Paladini, "An expert system approach to quality control," *Expert Systems with Applications*, vol. 18, no. 2, pp. 133–151, 2000.
- [5] T. Mahmood, R. Ashraf, and C. M. Nadeem Faisal, "An Efficient Scheme for the Detection of Defective Parts in Fabric Images Using Image Processing," *The Journal of the Textile Institute*, vol. 23, no. 7, pp. 1–9, 2022.
- [6] P. R. Jeyaraj and E. R. S. Nadar, "Effective textile quality processing and an accurate inspection system using the advanced deep learning technique," *Textile Research Journal*, no. SAGE Publications Sage UK: London, England, vol. 90, pp. 971–980, 2020.
- [7] M. Boluki and F. Mohanna, "Inspection of textile fabrics based on the optimal Gabor filter," *Signal, Image and Video Processing*, vol. 15, no. 7, pp. 1617–1625, 2021.
- [8] P. Anandan and R. S. Sabeenian, "Fabric Defect Detection Using Discrete Curvelet Transform," *Procedia Computer Science*, vol. 133, pp. 1056–1065, 2018.
- [9] X. Zhao, M. Zhang, and J. Zhang, "Ensemble learning-based CNN for textile fabric defects classification," *International Journal of Clothing Science & Technology*, vol. 33, no. 4, pp. 664–678, 2021.
- [10] S. L. Bangare, N. B. Dhawas, V. S. Taware, S. K. Dighe, and P. S. Bagmare, "fabric fault detection using image processing method," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 6, no. 4, pp. 405–409, 2017.
- [11] A. Rasheed, B. Zafar, A. Rasheed et al., "Fabric defect detection using computer vision techniques: a comprehensive review," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–24, 2020.

- [12] A. Conci and C. B. Proenca, "A comparison between image-processing approaches to textile inspection," *Journal of the Textile Institute*, vol. 91, no. 2, pp. 317–323, 2000.
- [13] D. Joyce, "Wallpaper group," "online," https://en.wikipedia.org/wiki/Wallpaper_group.
- [14] H. Y. Ngan, G. K. H. Pang, and N. H. C. Yung, "Automated fabric defect detection—a review," *Image and Vision Computing*, vol. 29, no. 7, pp. 442–458, 2011.
- [15] L. Amanuel, "Woven fabric defect control methods in shuttle loom," *Journal of Engineered Fibers and Fabrics*, vol. 16, pp. 155892502110141–155892502110147, 2021.
- [16] R. Ashraf, K. B. Bajwa, and T. Mahmood, "Content-based image retrieval by exploring bandletized regions through support vector machines," *Journal of Information Science and Engineering*, vol. 32, pp. 245–269, 2016.
- [17] M. Masood, T. Nazir, M. Nawaz et al., "A novel deep learning method for recognition and classification of brain tumors from MRI images," *Diagnostics*, vol. 11, no. 5, pp. 1–18, 2021.
- [18] Q. Qin, G. Ye, Z. Tu et al., "A spatial attentive and temporal dilated (SATD) GCN for skeleton-based action recognition," *CAAI Transactions on Intelligence Technology*, vol. 7, no. 1, pp. 46–55, 2022.
- [19] S. Karimi Jafarbigloo and H. Danyali, "Nuclear atypia grading in breast cancer histopathological images based on CNN feature extraction and LSTM classification," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 4, pp. 426–439, 2021.
- [20] X. Zhang and G. Wang, "Stud pose detection based on photometric stereo and lightweight YOLOv4," *Journal of Artificial Intelligence and Technology*, vol. 2, no. 1, pp. 32–37, 2022.
- [21] S. Fatima, N. Aiman Aslam, I. Tariq, and N. Ali, "Home security and automation based on internet of things: a comprehensive review," *IOP Conference Series: Materials Science and Engineering*, vol. 899, no. 1, pp. 012011–012012, 2020.
- [22] Q. Zou, K. Xiong, Q. Fang, and B. Jiang, "Deep imitation reinforcement learning for self-driving by vision," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 4, pp. 493–503, 2021.
- [23] M. A. Aslam, M. N. Salik, F. Chughtai, N. Ali, S. H. Dar, and T. Khalil, "Image classification based on mid-level feature fusion," in *Proceedings of the 2019 15th International Conference on Emerging Technologies (ICET)*, pp. 1–6, IEEE, Islamabad, Pakistan, 2019.
- [24] M. Asif, M. Bin Ahmad, S. Mushtaq, K. Masood, T. Mahmood, and A. Ali Nagra, "Long multi-digit number recognition from images empowered by deep convolutional neural networks," *The Computer Journal*, vol. 17, 2021.
- [25] L. Tong, X. Zhou, J. Wen, and C. Gao, "Optimal gabor filtering for the inspection of striped fabric," in *Proceedings of the International Conference on Artificial Intelligence on Textile and Apparel*, pp. 291–297, Springer, Cham, 2018.
- [26] N. Kaur and M. Dalal, "Application of machine vision techniques in textile (fabric) quality analysis," *IOSR Journal of Engineering*, vol. 02, no. 04, pp. 582–584, 2012.
- [27] D. Peng, G. Zhong, Z. Rao, T. Shen, Y. Chang, and M. Wang, "A fast detection scheme for original fabric based on blob, Canny and rotating integral algorithm," in *Proceedings of the 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, pp. 113–118, IEEE, Chongqing, China, 2018.
- [28] S. S. T. Selvi and G. M. Nasira, "An effective automatic fabric defect detection system using digital image processing," *J. Environ. Nanotechnol.*, vol. 6, pp. 79–85, 2017.
- [29] M. Hanmandlu, D. K. Choudhury, and S. Dash, "Detection of fabric defects using fuzzy decision tree," *International Journal of Signal and Imaging Systems Engineering*, vol. 9, no. 3, pp. 184–198, 2016.
- [30] E. Aldemir, H. Özdemir, and Z. Sarı, "An improved gray line profile method to inspect the warp–weft density of fabrics," *Journal of the Textile Institute*, vol. 110, no. 1, pp. 105–116, 2019.
- [31] G. Wang, H. Chen, S. Lv, R. Bikram Maharjan, and X. Zhao, "Research on permeability of satin fabrics based on fractal theory," *Journal of Reinforced Plastics and Composites*, vol. 34, no. 5, pp. 377–387, 2015.
- [32] S. L. Bangare, N. B. Dhawas, V. S. Taware, S. K. Dighe, and P. S. Bagmare, "Implementation of fabric fault detection system using image processing," *International Journal of Research in Advent Technology*, vol. 5, pp. 115–119, 2017.
- [33] P. A. Jadhav and P. Biradar, "Wavelet based features for defect detection in fabric using genetic algorithm," *IOSR Journal of Computer Engineering*, vol. 16, no. 3, pp. 116–120, 2014.
- [34] V. Dharmistha and D. Vishwakarma, "Analysis of Fabric Properties Using Digital Fabric Simulator," *International Journal of Engineering Research and Development*, vol. 66, no. 3, pp. 44–47, 2012.
- [35] C.-F. J. Kuo and C.-J. Lee, "A back-propagation neural network for recognizing fabric defects," *Textile Research Journal*, vol. 73, no. 2, pp. 147–151, 2003.
- [36] K. Yıldız, A. Buldu, and M. Demetgul, "A thermal-based defect classification method in textile fabrics with k-nearest neighbor algorithm," *Journal of Industrial Textiles*, vol. 45, no. 5, pp. 780–795, 2016.
- [37] T.-L. Su, H.-W. Chen, G.-B. Hong, and C.-M. Ma, "Automatic inspection system for defects classification of stretch knitted fabrics," in *Proceedings of the 2010 International Conference on Wavelet Analysis and Pattern Recognition*, pp. 125–129, IEEE, Qingdao, China, 2010.
- [38] G. H. Hu, "Optimal ring Gabor filter design for texture defect detection using a simulated annealing algorithm," in *Proceedings of the 2014 International Conference on Information Science, Electronics and Electrical Engineering (ISEEE)*, pp. 860–864, IEEE, Sapporo, Japan, 2014.
- [39] K. L. Mak, P. Peng, and K. F. Cedric Yiu, "Fabric defect detection using multi-level tunedmatched Gabor filters," *Journal of Industrial and Management Optimization*, vol. 8, no. 2, pp. 325–341, 2012.
- [40] G. H. Hu, Q. H. Wang, and G. H. Zhang, "Unsupervised defect detection in textiles based on Fourier analysis and wavelet shrinkage," *Applied Optics*, vol. 54, no. 10, pp. 2963–2980, 2015.
- [41] M. Alyas Khan, M. Ali, M. Shah et al., "Machine learning-based detection and classification of walnut fungi diseases," *Intelligent Automation & Soft Computing*, vol. 30, no. 3, pp. 771–785, 2021.
- [42] T. Mahmood, M. Shah, J. Rashid, T. Saba, M. W. Nisar, and M. Asif, "A passive technique for detecting copy-move forgeries by image feature matching," *Multimedia Tools and Applications*, vol. 79, no. 43–44, pp. 31759–31782, 2020.
- [43] S. N. Qase, A. Ahmadian, A. Mohammadzadeh, S. Rathinasamy, and B. Pahlevanzadeh, "A type-3 logic fuzzy system: Optimized by a correntropy based Kalman filter with adaptive fuzzy kernel size," *Information Sciences*, vol. 572, pp. 424–443, 2021.

- [44] J. H. Wang, J. Tavoosi, A. Mohammadzadeh et al., "Non-singleton type-3 fuzzy approach for flowmeter fault detection: experimental study in a gas industry," *Sensors*, vol. 21, pp. 7419–7423, 2021.
- [45] O. Castillo, J. R. Castro, and P. Melin, "Interval type-3 fuzzy systems: theory and design," *Studies in Fuzziness and Soft Computing*, vol. 418, pp. 1–100, 2022.
- [46] R. G. Saeidi, M. Latifi, S. S. Najjar, and A. G. Saeidi, "Computer vision-aided fabric inspection system for on-circular knitting machine," *Textile Research Journal*, vol. 75, no. 6, pp. 492–497, 2005.
- [47] P. Li, H. Zhang, J. Jing, R. Li, and J. Zhao, "Fabric defect detection based on multi-scale wavelet transform and Gaussian mixture model method," *Journal of the Textile Institute*, vol. 106, no. 6, pp. 587–592, 2014.
- [48] L. Jianli and Z. Baoqi, "Identification of fabric defects based on discrete wavelet transform and back-propagation neural network," *Journal of the Textile Institute*, vol. 98, no. 4, pp. 355–362, 2007.
- [49] L. Bissi, G. Baruffa, P. Placidi, E. Ricci, A. Scorzoni, and P. Valigi, "Automated defect detection in uniform and structured fabrics using Gabor filters and PCA," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 838–845, 2013.
- [50] W. Lia and L. Cheng, "Yarn-dyed Woven Defect Characterization and Classification Using Combined Features and Support Vector Machine," *The Journal of The Textile Institute*, vol. 113, no. 8, 2014.
- [51] M. E. Stivanello, S. Vargas, M. L. Roloff, and M. R. Stemmer, "Automatic detection and classification of defects in knitted fabrics," *IEEE Latin America Transactions*, vol. 14, no. 7, pp. 3065–3073, 2016.

Research Article

Brain Tumor Detection using Decision-Based Fusion Empowered with Fuzzy Logic

Aqsa Tahir ¹, **Muhammad Asif** ¹, **Maaz Bin Ahmad** ², **Toqeer Mahmood** ³,
Muhammad Adnan Khan ^{4,5} and **Mushtaq Ali**⁶

¹Department of Computer Science, Lahore Garrison University, Lahore 54000, Pakistan

²College of Computing and Information Sciences, KIET, Karachi 75190, Pakistan

³Department of Computer Science, National Textile University, Faisalabad 37610, Pakistan

⁴Pattern Recognition and Machine Learning Lab, Department of Software Gachon University, Seongnam 13557, Republic of Korea

⁵Riphah School of Computing & Innovation, Faculty of Computing, Riphah International University Lahore Campus, Lahore 54000, Pakistan

⁶Department of Computer Science and Information Technology, Hazara University, Mansehra 21300, Pakistan

Correspondence should be addressed to Toqeer Mahmood; toqeer.mahmood@yahoo.com

Received 25 February 2022; Revised 25 June 2022; Accepted 4 July 2022; Published 21 August 2022

Academic Editor: Muhammad Sajid

Copyright © 2022 Aqsa Tahir et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Brain tumor is regarded as one of the fatal and dangerous diseases on the planet. It is present in the form of uncontrolled and irregular cells in the brain of an infected individual. Around 60% of glioblastomas turn into large tumors if it is not diagnosed earlier. Some valuable literature is available on tumor diagnosis, but there is room for improvement in overall performance. Machine Learning (ML)-based techniques have been widely used in the medical domain for early diagnostic diseases. The use of ML techniques in conjunction with improved image-guided technology may help in improving the performance of the brain tumor detection process. In this work, an ML-based brain tumor detection technique is presented. Adaptive Back Propagation Neural Network (ABPNN) and Support Vector Machine (SVM) algorithms are used along with fuzzy logic. The fuzzy logic is used to fuse the result of ABPNN and SVM. The proposed technique is developed using the BRATS dataset. Experimental results reveal that the ABPNN model achieved 98.67% accuracy in the training phase and 96.72% accuracy in the testing phase. On the other hand, the SVM model has attained 98.48% and 97.70% accuracy during the training and testing phases. After applying fuzzy logic for decision-based fusion, the overall accuracy of the proposed technique reaches 98.79% and 97.81% for the training and the testing phases, respectively. The comparative analysis with existing techniques shows the supremacy of the proposed technique.

1. Introduction

The term “tumor” refers to a disease that causes swelling or corpus in the body. It can be related to any pathological process. Tumors constitute a significant demonstration of a massive and diverse clutch of ailments known as cancers or usually neoplasms [1]. The brain tumor is one of the fatal and complex types of tumor. It is formed because of a remarkable and aberrant increase in the cells inside the human brain. In ordinary circumstances, the development of a tumor initiates from the blood vessels, cells of the brain, and nerves

imminent out of the brain. Over time, the brain tumor has become a significant cause of disabilities and deaths worldwide [2, 3]. Brain tumor location and its capability to feast rapidly make treatment with radiations or surgery alike fighting an opponent hiding amongst caves and minefields. Inappropriately, many safer and easier ways to eliminate a small tumor than a large one are available [4]. About 60% of glioblastomas start as lower small tumors and, over time, become giant tumors.

According to the United States (US), National Cancer Institute estimated new brain tumor cases in the year 2022

are 25,050 (14,170 men and 10,880 women), and estimated deaths caused by brain tumors will be 18,280 [5]. It is also expected that 4,170 children (less than 15 years) will also be affected by a brain tumor. Worldwide, an estimated 308,102 primary brain or spinal cord tumor cases will be reported in 2020. Figure 1 shows the rate of new cases and death rate due to brain tumors in the US.

Figure 2 shows the overall age groupwise number of cases. As it shows, brain tumor cases are high in people aged 60–75. These are moderate in people aged 45–60 and 75–80. Moreover, these are minor in people under 45 and major in people above 80.

In medical science, technology helps scientists examine diseases on a cellular level. It provides antibodies against them in the early stage, which will help to save thousands of lives all-round the globe. Early detection of a brain tumor may help to reduce the casualty rate of brain tumor patients. The brain tumor manual diagnostic procedure is done with the help of domain specialists, which is an extraordinary time taking task. The detection accuracy is highly dependent on the expertise of the domain specialist. Artificial intelligence has brought a revolution in the medical diagnostic domain, improving efficiency and accuracy. The use of ML-based techniques for brain tumor detection may help to speed up the diagnosis process and reduce the death rate. There are some valuable ML-based techniques in the literature for brain tumor detection, but there is room to improve the overall accuracy of these techniques.

This paper presents a brain tumor detection technique in which Adaptive Back Propagation Neural Network (ABPNN) and Support Vector Machine (SVM) algorithms are used along with fuzzy logic. The fuzzy logic is used to fuse the result of ABPNN and SVM, which may help to reduce the false diagnosis. The dataset used in this work to develop the technique is taken from the Kaggle website [7]. It contains Computed tomography (CT) scan details of 3762 patients. It comprises 17 input parameters and one output parameter [7]. The experimental results show that the overall accuracy of the proposed technique is 98.79% and 97.81% for the training and the testing phases, respectively.

The rest of the article is organized as follows. Section 2 presents the literature review in which different methodologies and results are discussed. In Section 3, the proposed methodology is explained. Section 4 describes the experimental results and comparative analysis. Lastly, the paper is concluded in Section 5.

2. Literature Review

Digital image processing and computer vision are playing a vital role in many applications such as remote sensing, autonomous driving, medical image analysis, pose detection, security-based applications, and automated disease detection [8–12]. Recent focus of computer vision community is the use of deep-learning model [13–15] that are computationally expensive. However, at the same time, the research community is still widely presenting machine learning (ML)-based solutions [16–18].

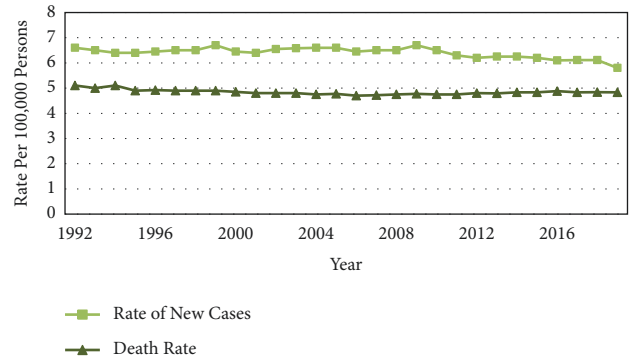


FIGURE 1: Number of brain tumor cases per year in the US [5].

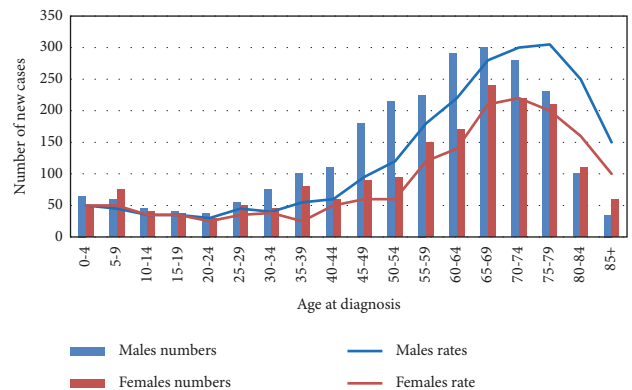


FIGURE 2: Number of brain tumor cases per age group [6].

In the literature, several attempts have been made to diagnose brain tumors using various ML techniques. Babu et al. [19] have presented a fusion-based brain tumor segmentation technique in which a convolutional neural network (CNN) is used for the fusion of Chan-Vese and level set segmentation methods. They also performed a comparative analysis of fusion-based and clustered-based segmentation techniques to identify the tumor. They claimed that CNN fusion-based segmentation outperforms the clustered-based segmentation technique in terms of segmentation error and minimal loss of information. Abbas et al. [20] have explained Local Independent Projection-based Classification (LIPC) for tumor segmentation using Principal Component Analysis (PCA). Image enhancement and noise removal are done using image preprocessing. To achieve an enhanced and efficient classification score, different textural features are considered and condensed using PCA. The segmentation results demonstrated a 0.95 Dice Score (DS) and 0.72 precision.

Rajan & Sundar [21] have proposed a hybrid-energy-efficient technique for automatic brain tumor segmentation and detection. They used Support Vector Machine (SVM) for brain tumor detection and K-means clustering with Fuzzy C-Means and active contours to perform brain tumor segmentation. They have attained an accuracy of 97.73%. The main limitation of their model is its high computational time because of the numerous techniques involved. Ullah et al. [22] have proposed a brain MRI image classification

technique that classifies images into abnormal and normal classes. After performing several preprocessing steps, they used Discrete Wavelet Transform (DWT) for feature extraction. Finally, they used an advanced Deep Neural Network (DNN) to classify whether the brain MRI image is normal or abnormal. They have achieved 95.8% accuracy. Josephine & Murugan [23] have proposed a method for detecting brain cancer utilizing Artificial Neural networks (ANN). They used Gabor features, Gray Level Co-occurrence Matrix (GLCM), and associated texture feature for brain tumor detection. They achieved 96% accuracy on a dataset of 30 MRI images. Ahmmed et al. [24] have proposed a technique for a brain tumor and its stages classification based on SVM and ANN. They used Temper-based K-means and modified Fuzzy C-means (TKFCM) clustering algorithm for segmentation of MRI images. Region property-based features and first-order statistics are extracted from segmented images. The first-order statistic is used to detect tumors from MRI images with the help of SVM. The contrast, the second type of feature helps to detect the stage of the tumor using the ANN. They have achieved an accuracy of 97.37% with a Bit Error Rate (BER) of 0.0294.

Mehmood et al. [6] have proposed a system to assist medical specialists that have the capabilities to perform brain tumor detection, segmentation, and 3D visualization from MRI images. For segmentation, they have used semiautomatic and adaptive threshold selection procedures. To classify a tumor into benign and malignant, the SVM classification model is used. Lastly, the volume marching cube algorithm is used for 3D visualization of the brain and tumors. They have achieved 99% accuracy. Dutta & Bandyopadhyay [25] have proposed a brain tumor detection technique using NGBoost classifier. The authors claimed an accuracy of 98.54%. Dutta & Bandyopadhyay [26] have proposed a technique for brain tumor detection using AdaBoost classifier. They have attained an accuracy of 98.97%. Tahir et al. [27] have proposed a technique for brain tumor detection. They have attained an accuracy of 87%. Munajat & Utaminigrum [28] have presented a GLCM and Back-Propagation Neural Network (BPNN)-based technique for brain tumor detection. They attained an accuracy of 88.03% with an average computation time of 0.601 sec. Ismael & Abdel-Qader [29] have presented a brain tumor detection framework that uses statistical features along with a neural network algorithm. To compute the statistical features, the 2D Gabor filter and 2D DWT are used. The authors claimed 91.9% accuracy for all types of tumors and a specificity of 96% for Meningioma, 96.29% for Glioma, and 96.29% for Pituitary tumors.

Amin et al. [30] have developed an unsupervised clustering method for the segmentation of tumors. A Fused Feature Vector (FFV) is used which is a combination of the Local Binary Pattern (LBP), Gabor Wavelet Features (GWF), segmentation-based fractal texture analysis (SFTA) components, and the histogram of oriented gradients. The classification of tumors among three subtumoral regions is done using Random Forest (RF) classifier. To avoid the overfitting problem, 0.5 holdout cross-validation and five-fold methodologies are applied and detected tumors with reasonable

confidence having 100% sensitivity. Ibrahim et al. [31] have developed a neural network-based technique for brain tumor detection through MRI images. It consists of three phases including preprocessing, dimensionality reduction, and classification. The experimental analysis shows that they attained an accuracy of 96.33%. Othman & Basri [32] have designed an automated brain tumor classification technique using PCA and Probabilistic Neural Network (PNN). They used PCA for dimensionality reduction and PNN for classification. The outcomes displayed that the proposed framework accomplished 73% correctness. Najadat et al. [33] have developed a decision tree classifier to recognize anomalies in CT brain pictures. They have achieved an accuracy of 88% on the training set and 58% on 2-fold validation. Balafar et al. [34] have presented a review of brain tumor segmentation techniques. They covered imaging modalities, noise reduction techniques, inhomogeneity correction, magnetic resonance imaging, and segmentation.

Although several valuable studies on brain tumor diagnosis and segmentation using different ML techniques have been proposed, most of these are developed using a limited number of images and have room for improvement in overall performance as explained in Table 1. Therefore, an efficient and accurate technique needs to be developed on a large dataset for diagnosing brain tumors.

3. Proposed Method

This work uses ABPNN and SVM techniques along with fuzzy logic to develop a brain tumor diagnosis system. Figure 3 shows a block diagram of the proposed system. It consists of training and validation phases. The training phase is divided into three layers; data acquisition, preprocessing, and application. In the data acquisition layer, the BRATS dataset is taken from the Kaggle website [7]. It contains Computed Tomography (CT) scan details of 3762 patients. It comprises 17 input parameters and one output parameter that indicates an abnormal or healthy person [7]. Table 2 lists the attributes of the dataset.

In preprocessing layer, data normalization along with missing value handling is performed. Noisy data is dealt with the normalization technique. On the other hand, missing values are resolved using the mean and moving average of the existing values [35]. In the application training layer, two ML algorithms, ABPNN and SVM, are trained using pre-processed data.

The output of ABPNN and SVM is given to the evaluation layer, where miss rate, accuracy, and Mean-Squared Error (MSE) are investigated. Then, an evaluation is done to find whether the Learning Criteria (LC) are met. If LC is met, it passes that data into the cloud. Otherwise, it must be retrained [36].

The next step is to apply fuzzy logic to fuse the results of both techniques to improve the overall performance of the proposed technique. For testing purposes, the extracted attributes from CT scan images of the patient are fed to the fusion-based trained model that predicts whether the patient has a brain tumor or not. When LC is satisfied, the fusion-based trained model is stored on a central server [37].

TABLE 1: Summary of existing work.

Authors	Techniques	Dataset	Remarks
Babu et al. [19]	CNN fusion followed by Chan-Vese active contour-based segmentation	DS 1-BRATS 2015 and DS 2-brain web	The CNN fusion-based segmentation is better than clustered-based segmentation for brain tumor detection in terms of segmentation error and minimal loss of information
Abbas et al. [20]	PCA and LIPC	MICCAI dataset 30 images	They have performed segmentation with 0.95 DS and 0.72 precision
Rajan & Sundar [21]	SVM, K-means clustering with fuzzy C-means and active contours	41 images	They have attained an accuracy of 97.73%.
Ullah et al. [22]	DWT and DNN	71 images	The main limitation of this model is high computational complexity
Josephine & Murugan [23]	Artificial neural network	30 images	They have attained an accuracy of 95.8%
Ahmmmed et al. [24]	TKFCM, SVM, ANN	46 images	They have achieved 96% accuracy
Mehmood et al. [6]	SVM and volume marching cube algorithm	256 images	They have achieved 97.37% accuracy with 0.0294 BER
Dutta & Bandyopadhyay [25]	NGBoost	1644 images	They have achieved 99% accuracy
Dutta & Bandyopadhyay [26]	AdaBoost	1644 images	They have attained an accuracy of 98.54%
Munajat & Utaminigrum [28]	BPNN	3762 images	They have attained an accuracy of 98.97%
Ismael & Abdel-Qader [29]	2D DWT, 2D gabor filter and back-propagation neural network	3064 slices	They have attained an accuracy of 88.03%
Amin et al. [30]	RF along with GWF HOG, LBP, and SFTA features	531 images	They have obtained 91.9% accuracy for all types of tumors and a specificity of 96% for meningioma, 96.29% for glioma and 96.29% for pituitary tumors correspondingly
Ibrahim et al. [31]	PCA and BPNN	174 images	They have claimed 100% sensitivity
Othman & Basri [32]	PCA and PNN	35 images	They have attained 96.33% accuracy
Najadat et al. [33]	Precision tree classifier and ABPNN	25 images	They have obtained 73% accuracy
			They have achieved 59% accuracy

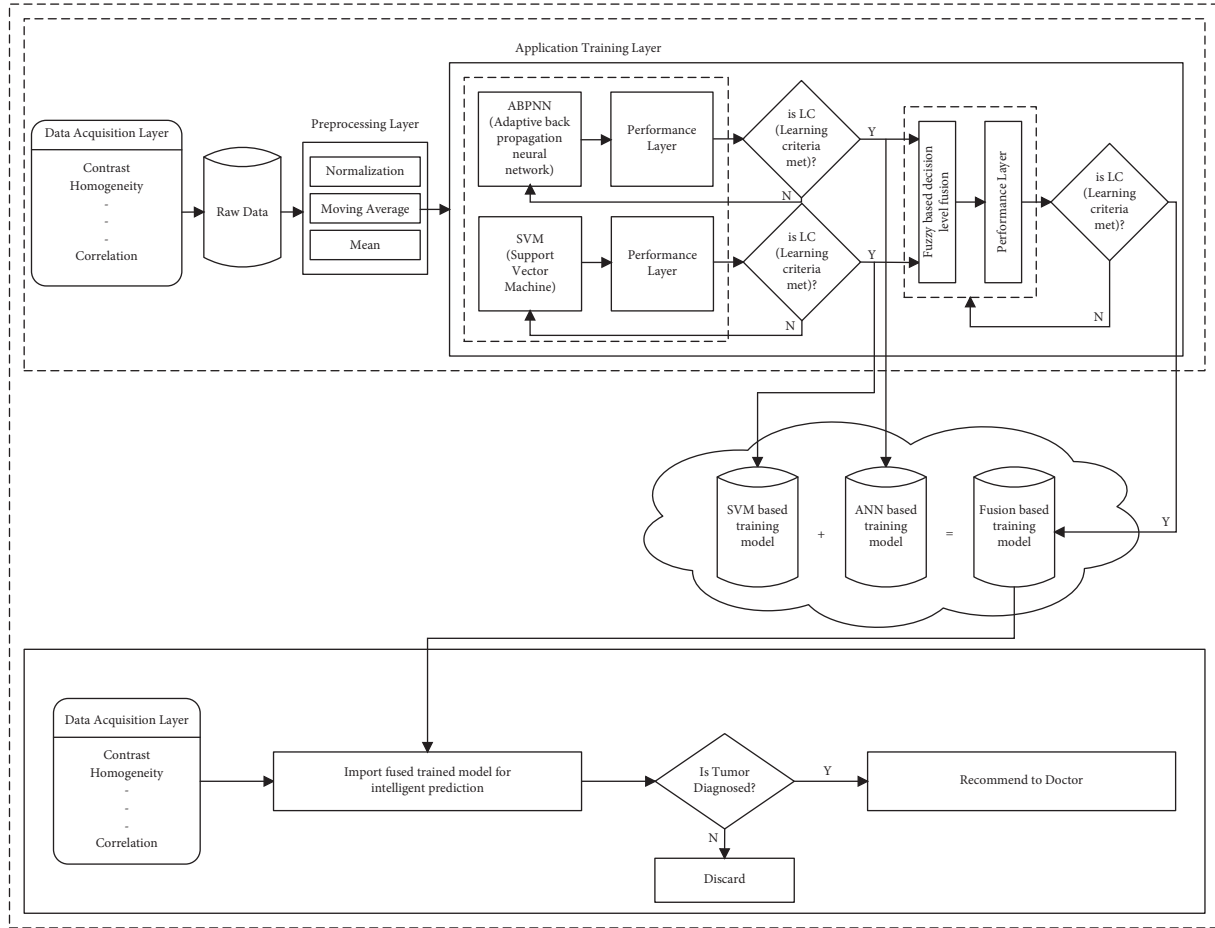


FIGURE 3: Block diagram of the proposed system.

TABLE 2: Input and output variables of the proposed DBFEFL system model.

Sr. No	Input/output variable name
Inp-1	Mean
Inp-2	Variance
Inp-3	Standard deviation
Inp-4	Skewness
Inp-5	Kurtosis
Inp-6	Contrast
Inp-7	Energy (ASM)
Inp-8	Entropy
Inp-9	Homogeneity
Inp-10	Dissimilarity
Inp-11	Correlation
Inp-12	Coarseness
Inp-13	(PSNR)
Inp-14	(SSIM) Index
Inp-15	(MSE)
Inp-16	(DC)
Inp-17	Target (0/1)
Outp-1	

3.1. ABPNN. ABPNN consists of the input, output, hidden layers, and the arrangement made from the back-propagation of error and feedforward propagation [38]. In

forward propagation, data is composed of the input layer towards the hidden layer, eventually transferred to the output layer. The output layer is then directed in reverse to the procedure of back-propagation error if it is not accepted. Inconsistent weight figures are balanced to limit error and moved towards feedforward [39].

Within the examination of the tumor, the input, output, and hidden layers are being utilized in ABPNN engineering with the feedforward algorithm using bit per data rate and conjunction [40]. In the current algorithm, distinct steps are associated. In the hidden layer, each neuron has an instigation work, e.g., $f(x) = \text{Sigmoid}(x)$. Input capacity for the sigmoid function is presented in equation (1), and the sigmoid function in the hidden layer of the proposed system is composed as presented in equation(2).

$$net_j = \sum_{i=1}^a (\mu_{ij} * INP_i) + \beta_i. \quad (1)$$

$$Out p_j = \frac{2}{1 + e^{-net_j}} \quad (2)$$

where $j = 1, 2, 3 \dots, n$.

The input parameter is taken from the output layer, as shown as follows:

$$\epsilon_k = \beta_2 + \sum_{i=1}^n (\delta_{jk*Outp_j}). \quad (3)$$

The activation function of the output layer, as shown as follows:

$$Outp_k = \frac{2}{1 + e^{-\epsilon_k}} \quad (4)$$

where $l = 1, 2, 3 \dots, r$.

The per output neuron error is calculated with the help of the squared-error function and the sum of each of these to find the total error in (5)

$$\rho = \frac{1}{2} \sum_k (\epsilon_k - Outp_k)^2. \quad (5)$$

where the desired output is represented by ϵ_k and calculated output as $Outp_k$. In (6), the output layer with the rate of weight change is written as

$$\Delta W \propto \frac{\partial \rho}{\partial w}, \quad (6)$$

$$\Delta \delta_{j,k} = -\epsilon \frac{\partial \rho}{\partial \delta_{j,k}},$$

$$\Delta \delta_{j,k} = -\epsilon \frac{\partial \rho}{\partial Outp_k} * \frac{\partial Outp_k}{\partial \epsilon_k} * \frac{\partial \epsilon_k}{\partial \delta_{j,k}}, \quad (7)$$

$$\Delta \delta_{j,k} = \epsilon (\epsilon_k - Outp_k) * Outp_k (1 - Outp_k) * Outp_j,$$

$$\Delta \delta_{j,k} = \epsilon \xi_k Outp_j,$$

$$\xi_k = (\epsilon_k - Outp_k) * Outp_k (1 - Outp_k),$$

$$\Delta \mu_{ij} \propto - \left[\sum_k \frac{\partial \rho}{\partial Outp_k} * \frac{\partial Outp_k}{\partial \epsilon_k} * \frac{\partial \epsilon_k}{\partial Outp_j} \right] * \frac{\partial Outp_j}{\partial net_j} * \frac{\partial net_j}{\partial \mu_{ij}},$$

$$\Delta \mu_{ij} = -\epsilon \left[\sum_k \frac{\partial \rho}{\partial Outp_k} * \frac{\partial Outp_k}{\partial \epsilon_k} * \frac{\partial \epsilon_k}{\partial Outp_j} \right] * \frac{\partial Outp_j}{\partial net_j} * \frac{\partial net_j}{\partial \mu_{ij}},$$

$$\Delta \mu_{ij} = -\epsilon \left[\sum_k \frac{\partial \rho}{\partial Outp_k} * \frac{\partial Outp_k}{\partial \epsilon_k} * \frac{\partial \epsilon_k}{\partial Outp_j} \right] * \frac{\partial Outp_j}{\partial net_j} * \frac{\partial net_j}{\partial \mu_{ij}}, \quad (8)$$

$$\Delta \mu_{ij} = -\epsilon \left[\sum_k (\epsilon_k - Outp_k) * Outp_k (1 - Outp_k) * \delta_{j,k} \right] * Outp_k (1 - Outp_k) * INP_i,$$

$$\Delta \mu_{ij} = -\epsilon \left[\sum_k (\epsilon_k - Outp_k) * Outp_k (1 - Outp_k) * \delta_{j,k} \right] * Outp_j (1 - Outp_k) * INP_i,$$

$$\Delta \mu_{ij} = -\epsilon \left[\sum_k (\epsilon_k - Outp_k) * Outp_k (1 - Outp_k) * \delta_{j,k} \right] * Outp_j (1 - Outp_k) * INP_i,$$

$$\xi_j = -\epsilon \left[\sum_k \xi_j \delta_{j,k} \right] * Outp_j (1 - Outp_j) * INP_i.$$

The value of changed weight will be calculated by switching the values in (7) as intimated in (8), where, $Outp_k$

$$\Delta \mu_{ij} = \epsilon \xi_j INP_i, \quad (9)$$

where

$$\xi_j = \left[\sum_k \xi_j \delta_{j,k} \right] * Outp_j (1 - Outp_j). \quad (10)$$

The hidden and output layers are shown in (11), updating the bias and weight between them.

$$\delta_{j,k}(t+1) = \delta_{j,k}(t) + \lambda \Delta \delta_{j,k}. \quad (11)$$

The updating values of bias and weight among the input layer and the hidden layer are exhibited as follows:

$$\mu_{ij}(t+1) = \mu_{ij}(t) + \lambda \Delta \mu_{ij}. \quad (12)$$

The learning rate of the brain tumor system model is represented by “ λ .”

3.2. SVM. SVM is defined as a supervised ML algorithm that can either be used for regression or classification. Though, it is more commonly used in classification problems. Each data item is plotted in N-dimensional space (N represents total features) per feature’s amount of a specific coordinate in the SVM algorithm [41, 42].

As the line equation is

$$\chi_2 = a\chi_1 + b, \quad (13)$$

where “ a ” is the line slope and “ b ” is the intercept of the line.

$$a\chi_1 - \chi_2 + b = 0. \quad (14)$$

Let suppose $\vec{x} = (\chi_1, \chi_2)^T$ & $\bar{\omega} = a - 1$. Now the beyond equation could be narrated as

$$\vec{\omega} \cdot \vec{x} + b = 0. \quad (15)$$

The following equation is the resultant of 2-dimensional vectors. Equation (13) is also referred to as a hyperplane equation. The vector’s direction $\vec{x} = (\chi_1, \chi_2)$ is symbolized as $\bar{\omega}$.

$$\omega = \frac{x_1}{\|x\|} + \frac{x_2}{\|x\|}, \quad (16)$$

where

$$\|x\| = \sqrt{x_1^2 + x_2^2 + x_3^2 + \dots + x_n^2}. \quad (17)$$

As we discern,

$$\cos(\sqsupset) = \frac{x_1}{\|x\|}, \quad (18)$$

$$\cos(\rho) = \frac{x_2}{\|x\|}.$$

Equation (16) can be inscribed as

$$\omega = (\cos(\sqsupset), \cos(\rho)) \quad (19)$$

$$\vec{\omega} \cdot \vec{x} = \|\omega\| \|x\| \cos(\sqsupset).$$

As $\sqsupset = \vartheta - \rho$, then

$$\cos(\sqsupset) = \cos(\vartheta) - \cos(\rho) \quad (20)$$

$$\cos(\sqsupset) = \cos(\vartheta)\cos(\rho) - \sin(\vartheta)\sin(\rho).$$

$\cos(\sqsupset)$ can also be written as

$$\cos(\sqsupset) = \frac{\omega_1}{\|\omega\|} \frac{x_1}{\|x\|} + \frac{\omega_2}{\|\omega\|} \frac{x_2}{\|x\|}. \quad (21)$$

By simplifying the above equation

$$\cos(\sqsupset) = \frac{\omega_1 x_1 + \omega_2 x_2}{\|\omega\| \|x\|}. \quad (22)$$

Put the value of $\cos(\sqsupset)$ is (19)

$$\vec{\omega} \cdot \vec{x} = \|\omega\| \|x\| \frac{\omega_1 x_1 + \omega_2 x_2}{\|\omega\| \|x\|}. \quad (23)$$

As the above equation explains the 2-dimensional vectors, for the n -dimensional vector, it can be written as shown in the following equation:

$$\vec{\omega} \cdot \vec{x} = \sum_{i=1}^n \omega_i x_i \quad (24)$$

where $i = 1, 2, \dots, n$.

The above equation is used to validate the correctly classifying the data

$$D = \ddot{y} (\omega \cdot x + b). \quad (25)$$

“ d ” is called the functional margin of the dataset and is written as

$$d = \min_{i=1 \dots m} D_i. \quad (26)$$

The hyperplane is selected as favorable, which has the largest value, where d is called the geometric margin of the dataset and we find out the optimal hyperplane in this article. To find out the optimal hyperplane, use the Lagrangian function, i.e., [43].

$$\gamma(\omega, b, \rho) = \frac{1}{2} \omega \cdot \omega - \sum_{i=1}^m \rho_i [y_i (\omega \cdot x + b) - 1], \quad (27)$$

$$\nabla_{\omega} \gamma(\omega, b, \rho) = \omega - \sum_{i=1}^m \rho_i y_i x_i = 0.$$

$$\nabla_b \gamma(\omega, b, \rho) = - \sum_{i=1}^m \rho_i y_i = 0. \quad (28)$$

Obtaining from (27) and (28), we can write equation (18).

$$\omega = \sum_{i=1}^m \rho_i y_i x_i \text{ and } \sum_{i=1}^m \rho_i y_i = 0. \quad (29)$$

By substituting the Lagrangian function Υ

$$\omega(\rho, b) = \sum_{i=1}^m \rho_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \rho_i \rho_j y_i y_j x_i x_j. \quad (30)$$

Thus, the above equation can also be defined in equation (19).

$$\max_{\rho} \sum_{i=1}^m \rho_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \rho_i \rho_j y_i y_j x_i x_j. \quad (31)$$

where $i = 1, 2, 3, \dots, m$.

Because of inequalities in constraints, the “ L ” multiplier method is spread to the Karush–Kuhn–Tucker (KKT) conditions. KKT complementary condition states that

$$\rho_i [y_i (\omega_i \cdot x^* + b) - 1] = 0. \quad (32)$$

In the above equation, x^* is the optimal point and b is the positive value, and for other points, its values are nearly equal to zero. So, we can write as in equation (20)

$$y_i(\omega_i \cdot x^* + b) - 1 = 0. \quad (33)$$

These are the closest points to the hyperplane, also known as support vectors. According to (33),

$$\omega - \sum_{i=1}^m \rho_i y_i x_i = 0. \quad (34)$$

It can also be written as

$$\omega = \sum_{i=1}^m \rho_i y_i x_i. \quad (35)$$

Equation (35) gets when we compute the value of b

$$y_i((\omega_i \cdot x^* + b) - 1) = 0. \quad (36)$$

Multiply both sides with y_i

$$y_i^2((\omega_i \cdot x^* + b) - 1) = 0. \quad (37)$$

As we know y^2/i is equal to 1

$$b = y_i - \omega_i \cdot x^*, \quad (38)$$

$$b = \frac{1}{S} \sum_{i=1}^S (y_i - \omega \cdot x). \quad (39)$$

In equation (39), S is the number of support vectors, and on the hyperplane, we make the predictions.

The hypothesis function is described in (40)

$$U_{SVM} = H(\omega_i) = \begin{cases} +1 & \text{if } \omega \cdot x + b \geq 0, \\ -1 & \text{if } \omega \cdot x + b < 0. \end{cases} \quad (40)$$

Class +1 will be categorized as an above point in the hyperplane, whereas -1 will be below the hyperplane (congestion not found). So, fundamentally, the main objective of the SVM algorithm is to calculate a hyperplane. It will distinguish the data correctly, and an optimal hyperplane is considered the best [1].

3.3. Decision-Based Fusion Empowered by Fuzzy Logic (DBFEFL). Fusion of data and information can be considered into three levels of abstraction: feature fusion, classifier fusion (also classified as decision-based fusion), and data fusion [44]. Decision fusion is considered a form of data fusion that combines the decisions of multiple classifiers into a mutual decision. It furthermore provides the benefit of recompensing for the insufficiencies of the specific sensor by using one or more than one added sensor [45].

The proposed DBFEFL is all about capability, intelligence, and logic. Fuzzy logic tries to handle problems with an imprecise and open set of data, sorting its chances of getting a flawless result [46]. The proposed DBFEFL for brain tumor diagnosis can be mathematically written as

$$\begin{aligned} \mu_{ABPNN} \cap \mu_{SVM} (ABPNN, SVM) \\ = \min[\mu_{ABPNN} (ABPNN), \mu_{SVM} (SVM)] \end{aligned} \quad (41)$$

According to output parameters, ABPNN's possible outcomes can be 0 or 1. Similarly, SVM's possible outcomes can either be 0 or 1. So, according to fuzzy logic, we have four fuzzy rules.

R_1 = If ABPNN outcome is 1 and SVM outcome is 1, a brain tumor is detected.

R_2 = If ABPNN outcome is 0 and SVM outcome is 0, brain tumor is not detected.

R_3 = If ABPNN outcome is 1 and SVM outcome is 0, a brain tumor is detected.

R_4 = If ABPNN outcome is 0 and SVM outcome is 1, a brain tumor is detected.

These rules are shown in the lookup diagram in Figure 4.

Figure 5 describes the surface viewer of the rules, that if ABPNN and SVM are from 0 to 40, the fuzzy decision is 0, which means a brain tumor is not detected. When the value is increased from 40 to 60, fuzzy is between 0-1, which means it can be a tumor. But when the value is greater than 60, the fuzzy decision is 1, which means a brain tumor is detected [47].

Membership function:

$$\begin{aligned} \text{Detection} = D \quad (\mu_{D, \text{No}}(d)) &= \begin{cases} 1 & 0 \leq d \leq 40, \\ \frac{60-d}{20} & 40 \leq d \leq 60, \\ 0 & 40 \leq d, \end{cases} \\ \text{Detection} = D \quad (\mu_{D, \text{yes}}(d)) &= \begin{cases} 0 & d \leq 40, \\ \frac{d-40}{20} & 40 \leq d \leq 60, \\ 1 & 60 \leq d \leq 100. \end{cases} \end{aligned} \quad (42)$$

4. Experimental Analysis

The proposed system is developed using MATLAB 2017. For experimental analysis, the dataset is divided into training and testing phases. 2634 samples are used for training, which is 70% of the data sample. 1128 samples are used for testing, i.e., 30% of the data samples [47, 48]. To evaluate the proposed system, several performance measure metrics are used that are computed with the help of equations (27) to (36) [49].

$$\text{Miss rate} = \frac{(O_1/T_0) + (O_0/T_1)}{T_0 + T_1}, \quad (43)$$

$$\text{Accuracy} = \frac{(O_0/T_0) + (O_1/T_1)}{T_0 + T_1}, \quad (44)$$

$$\text{Positive prediction value} = \frac{(O_{10}/T_1)}{(O_0/T_1) + (O_1/T_1)}, \quad (45)$$

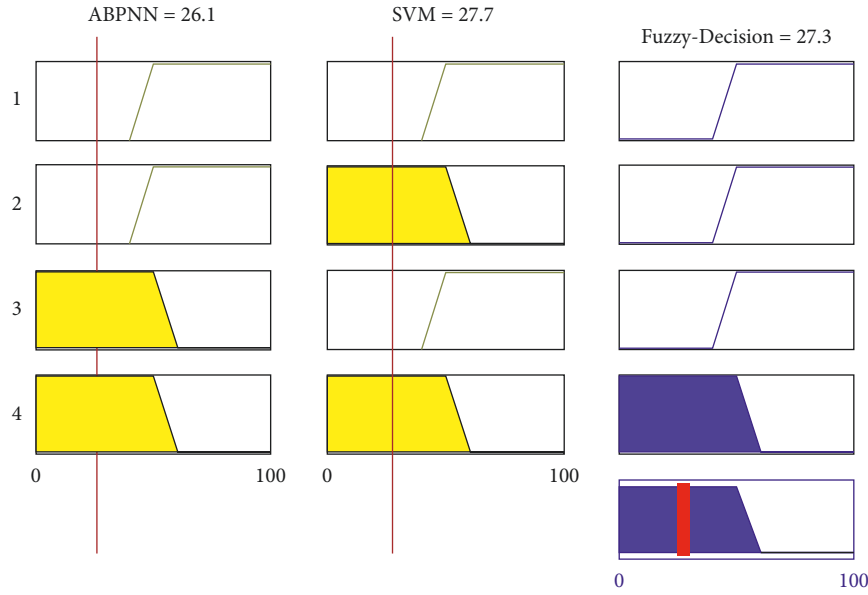


FIGURE 4: Fuzzy rules lookup diagram.

$$\text{Negative prediction value} = \frac{(O_0/T_0)}{(O_0/T_1) + (O_1/T_1)}, \quad (46)$$

$$\text{Specificity} = \frac{(O_0/T_0)}{(O_0/T_0) + (O_0/T_1)}, \quad (47)$$

$$\text{Sensitivity} = \frac{(O_1/T_1)}{(O_1/T_0) + (O_1/T_1)}, \quad (48)$$

$$\text{False_positive_ratio} = 1 - \text{specificity}, \quad (49)$$

$$\text{False_positive_ratio} = 1 - \text{Sensitivity}, \quad (50)$$

$$\text{Likelihood_ratio_positive} = \frac{\text{Sensitivity}}{1 - \text{specificity}}, \quad (51)$$

$$\text{Likelihood_ratio_negative} = \frac{1 - \text{Sensitivity}}{\text{specificity}}. \quad (52)$$

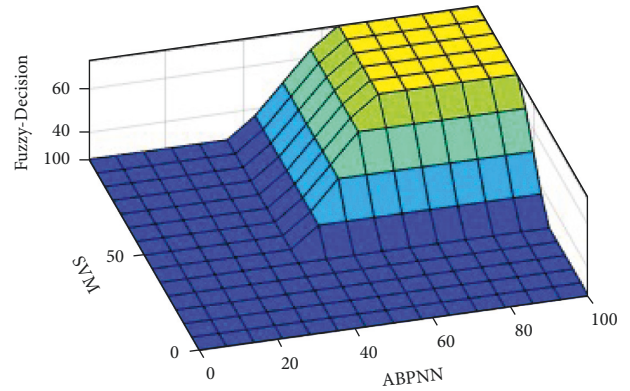


FIGURE 5: Fuzzy rules surface viewer.

The input parameters for the ABPNN and SVM algorithms are listed in Tables 3 and 4, respectively [50–52]. Tables 5 and 6 show the confusion matrix of ABPNN during the training and testing phase, respectively.

Tables 7 and 8 show the confusion matrix of SVM during the training and testing phase, respectively. Tables 9 and 10 show the confusion matrix of DBFEFL during the training and testing phase, respectively.

Table 11 lists the experimental results of the proposed brain tumor detection system at each stage in terms of several performance evaluation metrics [53]. During the testing phase, there is a 97.81% accuracy and a 2.19% miss rate. For the ABPNN model, the accuracy is 98.67% and the miss rate is 1.33% in the training phase. The accuracy and miss rates are 96.72% and 3.28% in the testing phase, respectively. In the SVM model, attained accuracy is 98.48%

TABLE 3: Input parameters for ABPNN.

Hyper-parameters	Value
Algorithm	Scaled conjugate gradient
Hidden layers	22
Epochs	11
Momentum	32
Cross-validation	5

TABLE 4: Input parameters for SVM.

Hyper-parameters	Value
Cross validation	5
Penalty	L0, L1
Loss	Hinge
Kernel	Linear

and the miss rate is 1.52% in the training phase. On the other hand, 97.70% accuracy and 2.3% miss rate are achieved in the testing phase. It can be seen that DBFEFL has an accuracy of 98.79% and a miss rate of 1.21% in the training phase.

TABLE 5: Confusion matrix of ABPNN (training phase).

N = 2634 (no. of samples)		Result (output) ($\mathbf{O}_0, \mathbf{O}_1$)	
Input	Expected output (T_0, T_1)	O_0 (0)	O_1 (1)
	$T_0 = 1682$ (0)	1672	10
	$T_1 = 952$ (1)	25	927

TABLE 6: Confusion matrix of ABPNN (testing phase).

N = 1128 (no. of samples)		Result (output) ($\mathbf{O}_0, \mathbf{O}_1$)	
Input	Expected output (T_0, T_1)	O_0 (0)	O_1 (1)
	$T_0 = 397$ (0)	393	4
	$T_1 = 731$ (1)	33	698

TABLE 7: Confusion matrix of SVM (training phase).

N = 2634 (no of samples)		Result (output) ($\mathbf{O}_0, \mathbf{O}_1$)	
Input	Expected output (T_0, T_1)	O_0 (0)	O_1 (1)
	$T_0 = 1682$ (0)	1675	7
	$T_1 = 952$ (1)	33	919

TABLE 8: Confusion matrix of SVM (testing phase).

N = 1128 (no of samples)		Result (output) ($\mathbf{O}_0, \mathbf{O}_1$)	
Input	Expected output (T_0, T_1)	O_0 (0)	O_1 (1)
	$T_0 = 397$ (0)	389	8
	$T_1 = 731$ (1)	18	713

TABLE 9: Confusion matrix of DBFEFL (training phase).

N = 2634 (no of samples)		Result (output) ($\mathbf{O}_0, \mathbf{O}_1$)	
Input	Expected output (T_0, T_1)	O_0 (0)	O_1 (1)
	$T_0 = 1682$ (0)	1675	7
	$T_1 = 952$ (1)	25	927

TABLE 10: Confusion matrix of DBFEFL (testing phase).

N = 1128 (no of samples)		Result (output) ($\mathbf{O}_0, \mathbf{O}_1$)	
Input	Expected output (T_0, T_1)	O_0 (0)	O_1 (1)
	$T_0 = 397$ (0)	390	7
	$T_1 = 731$ (1)	19	712

TABLE 11: Experimental results of the proposed system.

Measures	ABPNN (training)	ABPNN (testing)	SVM (training)	SVM (testing)	DBFEFL (training)	DBFEFL (testing)
Accuracy	98.67%	96.72%	98.48%	97.70%	98.79%	97.81%
Miss rate	1.33%	3.28%	1.52%	2.3%	1.21%	2.19%
Sensitivity	98.93%	99.43%	99.24%	98.89%	99.25%	99.03%
Specificity	98.53%	92.25%	98.07%	95.52%	98.53%	95.35%
Precision	97.37%	95.49%	96.54%	97.54%	97.37%	97.4%
Negative predictive value	99.41%	98.99%	99.58%	97.89%	99.58%	98.24%
False positive rate	1.47	7.75	1.93	4.42	1.47	4.65
False negative rate	1.07	0.57	0.76	1.11	0.75	0.97

Table 12 shows a comparative analysis of the proposed system with existing methods using the same dataset taken from the Kaggle website [7]. The experimental results revealed that the proposed method DBFEFL has achieved the highest accuracy with better performance using the latest

dataset with the maximum number of samples. The accuracy of the proposed method DBFEFL is 97.81% with the latest dataset of 3762 samples. Whereas the maximum accuracy attained using the Cross-Validated NGBoost Classifier [25] is 98.54%, they use 1644 images. Similarly, the maximum

TABLE 12: Comparative analysis between the proposed and existing methods.

Author	Technique	BRATS dataset
Proposed	DBFEFL (97.81%)	3672 images
Dutta & Bandyopadhyay [25]	Cross-validated NGBoost classifier (98.54%)	1644 images
	Gradient boost (97.37%)	
	AdaBoost (98.18%)	
	Random forest (97.98%)	
	Extra trees (94.13%)	
Dutta & Bandyopadhyay [26]	Cross-validated AdaBoost classifier (98.97%)	1644 images
	Gradient boost (90.69%)	
	Random forest (98.18%)	
	Extra trees (94.33%)	
Munajat & Utamingrum [28]	BPNN (87.01%)	3762 images

accuracy attained by using the Cross-Validated AdaBoost Classifier [26] is 98.97%, but similarly, they are using 1644 images. The accuracy using BPNN [28] is 87.01% using the same number of samples.

5. Conclusion

Early detection of brain tumors helps to decrease the casualty rate of brain tumor patients. The brain tumor manual diagnostic procedure is done with the help of domain specialists, which is an extraordinary time taking task. To automate this process, this paper presented a system for brain tumor detection that exploited ABPNN, SVM, and fuzzy logic to achieve the desired results. The outcomes of ABPNN and SVM are fused using fuzzy logic to increase the system's overall accuracy. The experimental results showed an accuracy of 98.30% and a miss rate of 1.7%. This research will be helpful in the medical science field. It can be deployed in OPD for brain tumor detection. It can be transformed into an interactive app that will take CT-scan images as an input parameter and categorize the patient as infected or normal. It may be helpful for doctors as an assistant hand for them that may strengthen their opinion regarding the diagnosis of the brain tumor patient.

Data Availability

The data used to support the study's findings are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] P. Tanwar, V. Jain, C.-M. Liu, and V. Goyal, *Big Data Analytics and Intelligence: A Perspective for Health Care*, Emerald Publishing Limited, West Yorkshire, England, 2020.
- [2] P. Jegannathan, "Brain tumor detection using convolutional neural network," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, pp. 686–692, 2021.
- [3] M. Masood, T. Nazir, M. Nawaz et al., "A novel deep learning method for recognition and classification of brain tumors from MRI images," *Diagnostics*, vol. 11, no. 5, p. 744, 2021.
- [4] D. Febrianto, I. Soesanti, and H. Nugroho, "Convolutional neural network for brain tumor detection," *IOP Conference Series: Materials Science and Engineering*, vol. 771, no. (1), Article ID 012031, 2020.
- [5] A. Noone, N. Howlader, M. Krapcho, D. Miller, A. Brest, and M. Yu, "Cancer stat facts: brain and other nervous system cancer," *SEER cancer statistics review*, vol. 2015, 1975, <https://seer.cancer.gov/statfacts/html/childbrain.html>.
- [6] I. Mehmood, M. Sajjad, K. Muhammad et al., "An efficient computerized decision support system for the analysis and 3D visualization of brain tumor," *Multimedia Tools and Applications*, vol. 78, no. 10, pp. 12723–12748, 2019.
- [7] J. Bohaju, "Brain tumor," 2020, <https://www.kaggle.com/dsv/1370629>.
- [8] Q. Zou, K. Xiong, Q. Fang, and B. Jiang, "Deep imitation reinforcement learning for self-driving by vision," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 4, pp. 493–503, 2021.
- [9] J. Zhang, G. Ye, Z. Tu et al., "A spatial attentive and temporal dilated (SATD) GCN for skeleton-based action recognition," *CAAI Transactions on Intelligence Technology*, vol. 7, no. 1, pp. 46–55, 2022.
- [10] A. Shabbir, A. Rasheed, H. Shehraz et al., "Detection of glaucoma using retinal fundus images: a comprehensive review," *Mathematical Biosciences and Engineering*, vol. 18, no. 3, pp. 2033–2076, 2021.
- [11] A. Shabbir, N. Ali, J. Ahmed et al., "Satellite and scene image classification based on transfer learning and fine tuning of ResNet50," *Mathematical Problems in Engineering*, vol. 2021, pp. 2021–2118.
- [12] A. Rehman, N. Abbas, T. Saba, T. Mahmood, and H. Kolivand, "Rouleaux red blood cells splitting in microscopic thin blood smear images via local maxima, circles drawing, and mapping with original RBCs," *Microscopy Research and Technique*, vol. 81, no. 7, pp. 737–744, 2018.
- [13] S. Fatima, N. A. Aslam, I. Tariq, and N. Ali, "Home security and automation based on internet of things: a comprehensive review," *IOP Conference Series: Materials Science and Engineering*, vol. 899, no. (1), Article ID 012011, 2020.
- [14] X. Zhang and G. Wang, "Stud pose detection based on photometric stereo and lightweight YOLOv4," *Journal of Artificial Intelligence and Technology*, vol. 2, pp. 32–37, 2022.
- [15] M. Asif, M. Bin Ahmad, S. Mushtaq, K. Masood, T. Mahmood, and A. Ali Nagra, "Long multi-digit number recognition from images empowered by deep convolutional neural networks," *The Computer Journal*, 2021.

- [16] M. Tahir, I. A. Taj, P. A. Assuncao, and M. Asif, "Fast video encoding based on random forests," *Journal of Real-Time Image Processing*, vol. 17, no. 4, pp. 1029–1049, 2020.
- [17] M. Asif, A. A. Nagra, M. B. Ahmad, and K. Masood, "Feature selection empowered by self-inertia weight adaptive particle swarm optimization for text classification," *Applied Artificial Intelligence*, vol. 36, no. 1, Article ID 2004345, 2022.
- [18] R. Ashraf, K. B. Bajwa, and T. Mahmood, "Content-based image retrieval by exploring bandletized regions through support vector machines," *Journal of Information Science and Engineering*, vol. 32, pp. 245–269, 2016.
- [19] K. R. Babu, P. Nagajaneyulu, and K. S. Prasad, "Performance analysis of CNN fusion based brain tumour detection using Chan-Vese and level set segmentation algorithms," *International Journal of Signal and Imaging Systems Engineering*, vol. 12, no. 1/2, p. 62, 2020.
- [20] K. Abbas, P. W. Khan, K. T. Ahmed, and W.-C. Song, "Automatic brain tumor detection in medical imaging using machine learning," in *Proceedings of the 2019 International Conference on Information and Communication Technology Convergence*, pp. 531–536, Jeju, Korea (South), October 2019.
- [21] P. G. Rajan and C. Sundar, "Brain tumor detection and segmentation by intensity adjustment," *Journal of Medical Systems*, vol. 43, no. 8, p. 282, 2019.
- [22] Z. Ullah, M. U. Farooq, S.-H. Lee, and D. An, "A hybrid image enhancement based brain MRI images classification technique," *Medical Hypotheses*, vol. 143, Article ID 109922, 2020.
- [23] S. Josephine and S. Murugan, "Brain tumor grade detection by using ANN," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, pp. 4175–4178, 2019.
- [24] R. Ahmmed, A. S. Swakshar, M. F. Hossain, and M. A. Rafiq, "Classification of tumors and it stages in brain MRI using support vector machine and artificial neural network," in *Proceedings of the 2017 International Conference on Electrical, Computer and Communication Engineering*, pp. 229–234, Cox's Bazar, Bangladesh, Feb 2017.
- [25] S. Dutta and S. K. Bandyopadhyay, "Revealing brain tumor using cross-validated NGBoost classifier," *International Journal of Machine Learning and Networked Collaborative Engineering*, vol. 4, no. 1, pp. 12–20, 2020.
- [26] S. Dutta and S. Bandyopadhyay, "Cross-validated AdaBoost classifier used for brain tumor detection," *EC Neurology*, vol. 12, pp. 50–57, 2020.
- [27] B. Tahir, S. Iqbal, M. Usman Ghani Khan et al., "Feature enhancement framework for brain tumor segmentation and classification," *Microscopy Research and Technique*, vol. 82, no. 6, pp. 803–811, 2019.
- [28] A. R. Munajat and F. Utaminigrum, "Brain tumor detection system based on sending email using Gray level Co-occurrence matrix and back-propagation neural network," in *Proceedings of the 6th International Conference on Sustainable Information Engineering and Technology*, vol. 2021, pp. 321–326, 2021.
- [29] M. R. Ismael and I. Abdel-Qader, "Brain Tumor Classification via Statistical Features and Back-Propagation Neural Network," in *Proceedings of the 2018 IEEE international conference on electro/information technology (EIT)*, pp. 0252–0257, Rochester, MI, USA, Oct 2018.
- [30] J. Amin, M. Sharif, M. Raza, and M. Yasmin, "Detection of brain tumor based on features fusion and machine learning," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–17, 2018.
- [31] W. H. Ibrahim, A. A. A. Osman, and Y. I. Mohamed, "MRI brain image classification using neural networks," in *Proceedings of the 2013 International Conference on Computing, Electrical and Electronic Engineering*, pp. 253–258, Khartoum Sudan, August 2013.
- [32] M. F. Othman and M. A. M. Basri, "Probabilistic neural network for brain tumor classification," in *Proceedings of the 2011 Second International Conference on Intelligent Systems, Modelling and Simulation*, pp. 136–138, Phnom Penh, Cambodia, January 2011.
- [33] H. Najadat, Y. Jaffal, O. Darwish, and N. Yasser, "A classifier to detect abnormality in CT brain images," in *Proceedings of the The 2011 IAENG International Conference on Data Mining and Applications*, pp. 374–377, 2011.
- [34] M. A. Balafar, A. R. Ramli, M. I. Saripan, and S. Mashohor, "Review of brain MRI image segmentation methods," *Artificial Intelligence Review*, vol. 33, no. 3, pp. 261–274, 2010.
- [35] P. Ambily, S. P. James, and R. R. Mohan, "Brain tumor detection using image fusion and neural network," *International Journal of Engineering Research and General Science*, vol. 3, pp. 1383–1388, 2015.
- [36] A. Ata, M. A. Khan, S. Abbas, M. S. Khan, and G. Ahmad, "Adaptive IoT empowered smart road traffic congestion control system using supervised machine learning algorithm," *The Computer Journal*, vol. 64, no. 11, pp. 1672–1679, 2021.
- [37] T. Ali, K. Masood, M. Irfan et al., "Multistage segmentation of prostate cancer tissues using sample entropy texture analysis," *Entropy*, vol. 22, no. 12, p. 1370, 2020.
- [38] V. S. Selvam and S. Shenbagadevi, "Brain tumor detection using scalp EEG with modified wavelet-ICA and multi layer feed forward neural network," in *Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6104–6109, MA, USA, August 2011.
- [39] C. Megha and J. Sushma, "Detection of brain tumor using machine learning approach," in *Proceedings of the International Conference on Advances in Computing and Data Sciences*, pp. 188–196, Ghaziabad, India, April 2019.
- [40] J. S. Paul, A. J. Plassard, B. A. Landman, and D. Fabbri, "Deep learning for brain tumor classification," *Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging*, vol. 10137, pp. 253–268, 2017.
- [41] R. M. Adnan, X. Yuan, O. Kisi, and Y. Yuan, "Streamflow forecasting using artificial neural network and support vector machine models," *American Academic Scientific Research Journal for Engineering, Technology, and Sciences*, vol. 29, pp. 286–294, 2017.
- [42] M. Alyas Khan, M. Ali, M. Shah et al., "Machine learning-based detection and classification of walnut fungi diseases," *Intelligent Automation & Soft Computing*, vol. 30, no. 3, pp. 771–785, 2021.
- [43] N. Abdullah, U. K. Ngah, and S. A. Aziz, "Image classification of brain MRI using support vector machine," in *Proceedings of the 2011 IEEE International Conference on Imaging Systems and Techniques*, pp. 242–247, Batu Ferringhi, Malaysia, May 2011.
- [44] P. R. Kshirsagar, A. N. Rakhonde, and P. Chippalkatti, "MRI image based brain tumor detection using machine learning," *Test Engineering and Management*, vol. 81, pp. 3672–3680, 2020.
- [45] J. Amin, M. Sharif, A. Haldorai, M. Yasmin, and R. S. Nayak, *Brain Tumor Detection and Classification Using Machine Learning: A Comprehensive Survey*, pp. 1–23, Complex & Intelligent Systems, Saudi Arabia, 2021.
- [46] G. Ramkumar, R. Thandaiah Prabu, N. Phalguni Singh, and U. Maheswaran, "Experimental analysis of brain tumor

- detection system using Machine learning approach,” *Materials Today Proceedings*, 2021.
- [47] A. Keerthana, B. Kavin Kumar, K. Akshaya, and S. Kamalraj, “Brain tumour detection using machine learning algorithm,” *Journal of Physics: Conference Series*, vol. 1937, no. 1, Article ID 012008, 2021.
- [48] G. Manogaran, P. M. Shakeel, A. S. Hassanein, P. Malarvizhi Kumar, and G. Chandra Babu, “Machine learning approach-based gamma distribution for brain tumor detection and data sample imbalance analysis,” *IEEE Access*, vol. 7, pp. 12–19, 2019.
- [49] J. Amin, M. Sharif, M. Raza, T. Saba, and M. A. Anjum, “Brain tumor detection using statistical and machine learning method,” *Computer Methods and Programs in Biomedicine*, vol. 177, pp. 69–79, 2019.
- [50] S. H. Javed, M. B. Ahmad, M. Asif, S. H. Almotiri, K. Masood, and M. A. A. Ghamdi, “An intelligent system to detect advanced persistent threats in industrial internet of things (IIoT),” *Electronics*, vol. 11, no. 5, p. 742, 2022.
- [51] A. Hussain, M. Asif, M. B. Ahmad, T. Mahmood, and M. A. Raza, “Malware detection using machine learning algorithms for windows platform,” in *Proceedings of International Conference on Information Technology and Applications*, pp. 619–632, Mragowo, Poland, Oct2022.
- [52] D. Jude Hemanth and J. Anitha, “Modified genetic algorithm approaches for classification of abnormal magnetic resonance brain tumour images,” *Applied Soft Computing*, vol. 75, pp. 21–28, 2019.
- [53] K. Sharma, A. Kaur, and S. Gujral, “Brain tumor detection based on machine learning algorithms,” *International Journal of Computer Application*, vol. 103, no. 1, pp. 7–11, 2014.

Research Article

Facial Mask Detection Using Image Processing with Deep Learning

**Hongyu Ding,¹ Muhammad Ahsan Latif², Zain Zia², Muhammad Asif Habib³,
Muhammad Abdul Qayum³ and Quancai Jiang¹**

¹College of Electrical Engineering and New Energy, China Three Gorges University, Yichang 443000, China

²Department of Computer Science, University of Agriculture, Faisalabad 38040, Pakistan

³Department of Computer Science, National Textile University, Faisalabad 37610, Pakistan

Correspondence should be addressed to Muhammad Asif Habib; drasif@ntu.edu.pk

Received 26 February 2022; Revised 21 April 2022; Accepted 11 May 2022; Published 12 August 2022

Academic Editor: Nouman Ali

Copyright © 2022 Hongyu Ding et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Coronavirus disease 2019 (COVID-19) has a significant impact on human life. The novel pandemic forced humans to change their lifestyles. Scientists have broken through the vaccine in many countries, but the face mask is the only protection for public interaction. In this study, deep neural networks (DNN) have been employed to determine the persons wearing masks correctly. The faster region-based convolutional neural networks (RCNN) model has been used to train the data using graphics processing unit (GPU) device. To achieve our goals, we used a multiphase detection model: first, to label the face mask, and second to detect the edge and compute edge projection for the chosen face region within the face mask. The current findings revealed that faster RCNN was efficient and precise, giving 97% accuracy. The overall loss after 200,000 epochs is 0.0503, with a trend to decrease. While the loss is falling, we are getting more accurate results. As a result, the faster RCNN technique effectively identifies whether a person is wearing face masks or not, and the training period was decreased with better accuracy. In the future, Deep Neural Network (DNN) might be used first to train the data and then compress the dimensions of the input to run it on low-powered devices, resulting in a lower computational cost. Our proposed system can achieve high face detection accuracy and coarsely obtain face posture estimation based on the specified rule. The faster RCNN learning algorithm returns high precision, and the model's lower computational cost is achieved on GPU. We use the "label-image" application to label the photographs extracted from the dataset and apply Inception V2 of faster RCNN for face mask detection and classification.

1. Introduction

COVID-19, a novel pandemic, posed a severe human pandemic threat to the entire world. This pandemic started in China at the end of 2019 and is probably one of the world's most significant health challenges this century. By January 2022, over 300 million confirmed COVID-19 cases were recorded, with almost 5.47 million deaths worldwide. Some countries' scientists enabled breaking through the vaccine in time. The research is ongoing by thousands of scientists worldwide to better understand how new virus mutations and variants like alpha, beta, gamma, delta, and omicron affect the effectiveness of the different COVID-19 vaccines. The COVID-19 pandemic has altered people's entire lifestyles. The patient has an acute lung infection for which there is currently no high-performance vaccine or treatment.

Deaths are rapidly increasing, putting strain on each nation's healthcare systems [1].

Face masks have become an essential part of human life, and the only protection is the use of face masks while interacting with public people. In this challenging pandemic condition, numerous countries have mandated that citizens wear face masks when visiting any intense public spot like shopping malls taking a meal. The customer care service provider provides services to only those customers who wear a face mask correctly. The World Health Organization (WHO) has released recommendations and guidelines for the general public and healthcare professionals who use face masks. Face masks, according to medical officers, give sufficient protection against respiratory diseases. Nurses and doctors widely used face masks as the central part of droplet safety measures and precautions. Some argue that wearing a

face mask will not protect you from COVID-19 infection. However, there is a significant distinction between absence and the absence of evidence [2].

We presented an accurate and efficient face mask detector algorithm in the present study. Our main task is to check whether or not people wear masks and stay away from public places using the proposed algorithm. It is a classification or object detection problem of two different classes wearing masks and not proper masks. There are several methods to detect objects, but we chose to utilize faster RCNN because of its fast, simple, accurate, and precise algorithm. We need to develop a system that could detect faces in this real world and recognize whether or not the detected faces have masks.

The paper is organized in the following fashion: Section 2 presents the literature review, and Section 3 represents the method's description. Section 4 is spare for the results we achieve in the present study, and the conclusion is given in Section 5.

2. Literature Review

Presently, the issue is connected to the general recognition of objects using deep learning and the detection of object classes [3]. A few researchers have been found to detect facial masks based on image analysis in the literature. Detection of the face or mask is one of the classes or groups of objects [4]. Detectors depend on deep learning structures rather than handcrafted features and have, in recent years, had outstanding performance due to their exceptional extraction robustness and capability. Face and object detection applications are utilized in education surveillance, autonomous driving, and several other fields [5, 6]. A surveillance camera's image processing technology could detect a person's face when not wearing a face mask.

Schneiderman and Kanade [7] proposed face scanners with characteristics shaped by a series of feature vectors trained using a view-based approach. The structure has been disclosed to enhance profile face detection accuracy. A detailed collection of features, similar to Haar, was projected, with rectangular features rotated to 45 degrees by Lienhart and Maydt [8]. He added another wing with Haar-like elements and a flexible span spatially separated the rectangles. Two separate neural networks were used to detect faces within plane rotations, as suggested by Torralba et al. [9]. Hotta [10] showed a support vector machine (SVM) method, local kernel-based, for face recognition, which was superior to global kernel-based SVM in recognizing impeded frontal faces. Felzenszwalb and Huttenlocher [11] presented a deformable design incorporating several object components, as Fischler and Elschlager's visual structure illustration indicated. Lin and Liu [12] proposed that the multiview face identifier be learned as a single tumble classifier. They built MBH Boost, a multiclass boost-up algorithm, distributing features into several classes.

Goldmann et al. [13] used the qualified detector divided into subclassifiers connected to several predefined image regions. The inputs of subclassifiers were fused, resulting in an updated Viola-Jones detection algorithm. Yang et al. [14]

used the first few cascade levels, including all face markers, to estimate the pose for expediency in multiview face recognition, where all the face identifiers modified to different visions have to be measured for each scan window; they used the first few cascade levels along with all face identifiers to approximate the pose for expediency. A quick bounding box estimation method for face recognition proposed by Subburaman predicts the bounding box using a small patch-based local search [15]. Mesphil et al. [16] proposed convolutional networks. These were neural networks with at least one layer that uses convolution instead of general matrix multiplication. Zhu and Ramanan [17] presented an idea to use the deformable parts-based template to detect a face jointly, measure an estimated pose, and then localize a face sign in the wild, which was later improved to coalesce the landmark approximation and image recognition tasks in a shared supervised way to enhance face recognition through unique landmark detections.

Yang et al. [18] explored channel features to recognize faces that perform well. Despite the ease of using these techniques for unregulated face recognition, the accuracy rate is still inadequate, particularly when the identifier must account for minor false alarms. Girshick et al. [19] proposed a work of inspiration RCNN, a convolutional neural network that performs a selective search to identify candidate regions containing items. The system aims to identify healthcare workers losing their surgical masks in the operating room. Ren et al. [20] have developed a real-time face recognition and monitoring technique.

Farfate et al. [21] have developed a deep learning-based face detection technique known as deep dense face recognition technology. The method does not require any clarification of landmarks or poses, and it can detect faces in a wide variety of orientations with only one model. Zhu and Ramanan [22] gave a method for dealing with occlusions and arbitrary pose changes in direct face detection. There is a new factor called normalized pixel difference. Machine learning approaches were used to create a deep transfer, hybrid direct instruction for face mask identification by Redmon and Farhadi [23]. Dong et al. [24] have proposed a deep cascaded region detection that investigates its bounding box decrease, a localization method, to achieve image recognition of potential countenances.

Sun et al. [25] enhanced the quicker RCNN technique via profoundly learning face detection algorithms. They utilized numerous strategies, including a pretraining model, multiband training, passive extraction, accurate calibration of primary parameters, and job grouping. Zhao et al. [3] proposed the surgical mask presence or absence monitoring device in the operating room. In the linguistic image segmentation for facial mask detection, gradient descent is used for preparation, while multiple linear regression cross-entropy is used for neural networks. Ejaz et al. [26] built a new method for detecting the existence of a face mask. They classified three different types of face mask usage: proper face mask-wearing, wrong face mask-wearing, and no mask.

The two most well-known classes, two-stage human detectors, and single-stage human detectors were recently used [27]. Fan and Jiang [28] suggested the inception

network that helps to find out which kernel combination is the best. The Residual Network (ResNet) trains even deeper neural networks to learn an identity function from the preceding stage.

Most recently, the RetinaFaceMask one-stage detector approach has been studied by Fan and Jiang [28]. They fused high-level semantic fusion using multiple feature maps like Feature Pyramid Network (FPN). The proposed algorithm rejects the low confidence predictions and the high intersection of the union.

The deep learning-based approaches have an inherently high degree of accuracy as compared to the other machine learning-based techniques, especially for classification and clustering. The multilayer structure in the network helps to process different tasks at different layers exclusively.

3. Description of the Method

3.1. Deep Neural Network Algorithm. DNN algorithm moves data through a sequence of “layers” of neural network models, with each layer passing a simplified summary of the data to the next layer. Several computer vision algorithms work well on datasets with a few hundred features or columns. An unstructured dataset, such as one extracted from an image, on the other hand, contains so many features that this method becomes inefficient or impossible. Traditional machine learning algorithms cannot handle 2.4 million parts in a single 800×1000 pixel RGB color image.

As the image passes through each neural network layer, DNN algorithms learn more about it. Initial layers learn how to detect low-level features such as edges, and later layers incorporate these features into a more comprehensive representation. For instance, a middle layer would detect edges to detect parts of an object in an image, such as a leg or a branch, while a deep layer might identify the entire object, such as a dog or a tree. You gather data from observations and integrate it into a single layer. The layer produces an output, which becomes the input for the following layer, and so on. This loops until the final output signal is received [29].

3.2. Types of Algorithms. There are several different types of feature extraction algorithms, which can be classified into two categories.

3.2.1. Algorithms That Rely on Classification. The regions of interest are chosen in the first stage. After that, convolutional neural networks (CNN) are used to categorize specific areas. Since prediction must be run for each selected field, this solution may be prolonged. This group includes algorithms such as the fast RCNN and faster RCNN which are enhanced variants of the region-based Convolutional Neural Network (RCNN) [30].

3.2.2. Algorithms That Rely on Regression. In contrast to the previous approach, algorithms in this category predict the class probability and define the bounding boxes surrounding the object of interest in a single run from the entire image

point of view. This group includes algorithms like You Just Look Once (YOLO) and Single Shot Multibox Detector (SSD) [30]. Deep learning and computer vision are used in various applications such as objection detection, medical image analysis, and action recognition [31, 32]. Recent research is focused on the use of mid-level features and deep learning models to build robust decision support systems and IoT applications [33–35].

3.3. Faster RCNN. In object classification and recognition, a deep learning technique known as area of interest polling is gaining much attention. Detecting objects from an image scene containing several things is one example. The goal is to extract fixed-size feature maps using maximum pooling on the entire picture as reflected in Figure 1. The object detection technique used by faster RCNN is divided into three stages.

3.3.1. Region Proposal Network. Finding the spaces in the given input image where there is a possibility of finding an object is straightforward. The position of an entity in an image can be determined. The area where there is a possibility of finding an object is surrounded by the Region of Interest (ROI).

3.3.2. Classification. The next step is to assign corresponding classes to the regions of interest defined in the previous actions. Here, the CNN approach is used (Figure 2). The proposed approach includes a detailed process for identifying all spaces of object location in an image. If no regions are placed in the first stage of the algorithm, there is no need to move on to the second step. In 2015, Girschik [36] proposed the Region Proposal Network (RPN) and ROI pooling as a DLA-based object detection solution. ROI can achieve speed and usability for both training and research performance. The ROI layers take a feature map as input, which is the output of a convolution neural network with multiple convolution layers and max-pooling layers.

An $N \times N$ matrix is generated by dividing the function map space into regions of interest. The ROI is denoted by the letter N . The first column represents the image’s index. In contrast, the second column, which ranges from the upper left-most coordinate to the bottom-most coordinate, represents the ROI coordinates. Region Proposal refers to the determined area of interest space. The system divides the area proposal’s entire room into equal-sized partitions. The number of sections in which the whole area proposal is divided must equal the output dimension. The maximum value of each divided subregion is estimated. The maximum values are copied to the output buffer.

In RPN, the image is first fed into the convolution neural network. The input image is passed through a series of convolution layers before being sent to the final layer, which creates feature maps. Every portion of the function maps includes a sliding window. The mask size for a sliding window mask is typical. The anchors for each sliding window are created. Let it be the exact center for these

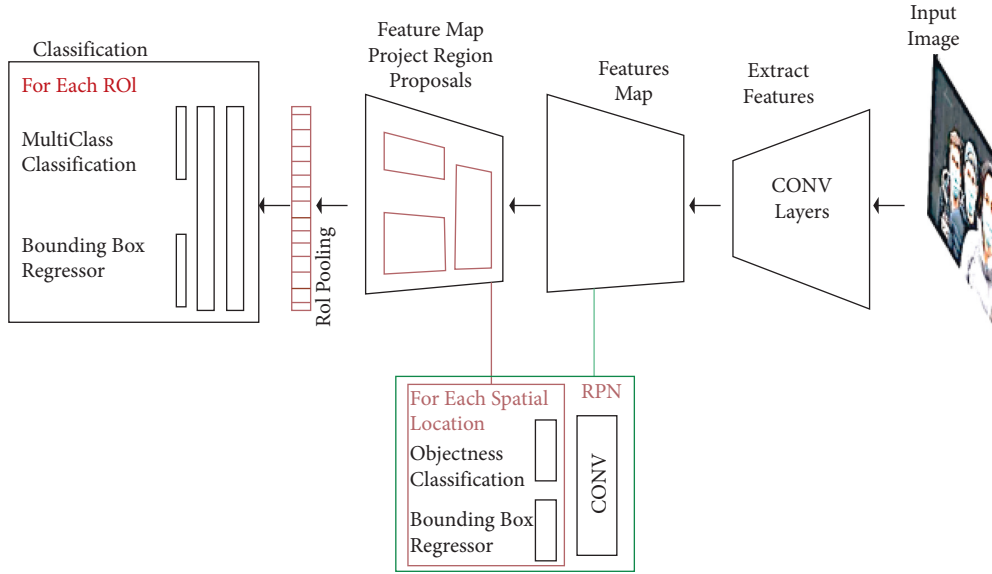


FIGURE 1: Faster RCNN.

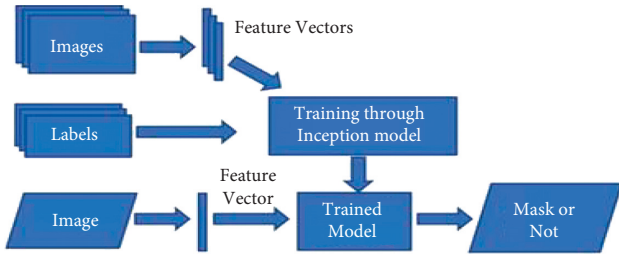


FIGURE 2: Model architecture.

anchors (x, c, y, c) . However, the aspect ratios and scaling factors of the anchors produced will differ.

In addition, a value q is determined for each of these anchors, representing the likelihood of the anchors overlapping the region's boundary surrounding the objects. A region boundary with loc coordinates is the regressor's production. The classification shows whether the area contains an object or not by a probability of 0 or 1.

$$q^* = 1, \text{ when } IoU > 0.7,$$

$$q^* = -1, \text{ when } IoU < 0.7,$$

$$q^* = 0, \text{ otherwise,}$$

$$t = \left[\frac{(x - x^a)}{w^a}, \frac{(y^* - y_a)}{h_a}, \frac{\log w}{w_a}, \frac{\log h^*}{h_a} \right], \quad (1)$$

$$t^* = \left[\frac{(x - x^a)}{w^a}, \frac{(y^* - y_a)}{h_a}, \frac{\log w^*}{w_a}, \frac{\log h^*}{h_a} \right].$$

w_a, h_a, x_a, y_a are the widths, height, and center of anchor, and h^*, w^*, x^*, y^* are the ground truth bounding box height, width, and center. Over the performance from the classification and regression networks, the loss function is established.

$$L(\{q^i\}, \{t^i\}) = \frac{1}{N_{cls} L_{cls}(q_i, q_i^*)} + \frac{\lambda 1}{N_{reg} L_{reg}(t_i, t_i^*)}. \quad (2)$$

Finally, the size 3×3 final features are extracted and fed into the networks for regression and classification.

Follow these steps to build your object detection classifier.

3.4. Inception V2 of Faster RCNN. For object detection, the faster RCNN network is a single, centralized system. It employs the area proposal network (RPN) module. It directs the unified network's quest. On the other hand, inception comprises a 22-layer inception module with no ultimately linked layers. The main advantage of this model is that it allows better use of the computational resources available on the network. The inception module functions as a network within a network, piling modules on top of one another. It has 5 million parameters, which is a factor of 12 less than AlexNet. The combination of faster RCNN and inception V2 is computationally expensive, but the results in object detection are more reliable [37].

3.4.1. Compute Unified Device Architecture (CUDA). CUDA is an NVidia technology that can perform a variety of challenging computations on the GPU. Every thread in CUDA uses kernels executed n times, and a unique number marks each line. CUDA's architecture comprises grids that are subdivided into smaller units called blocks. Each block is assigned to a multiprocessor by the hardware, which has a group of multiprocessors. Finally, threads make up blocks. These tiniest units can be synchronized together in a single block [38].

In general, the CUDA program begins with computer memory allocation while data on the host is being prepared. The data is then moved from the host to the computer. Since copying data from the host to the computer and the device takes time, it is essential to restrict the amount of data sent.

It is possible to launch kernels after the data on the system has been prepared. The results are copied back to the host after the calculation is completed. Finally, the results can be viewed, and the reserved memory can be freed.

The GPU implementation resembles that of a multi-threaded CPU program. The concept remains the same. We only copy to the system what is required, such as integral images, qualified classifiers, and detection windows. A CUDA kernel is run for each detection window's size. The program's first version calculates the locations of detection windows in the client framework. A set of identically sized windows is computed. Then it is sent to the computer, where the current window detection process will begin. Of course, the detection window is the same size, but the thread index determines its location. The findings are sent back to the host, and information about new detection windows is prepared after the last detection window in the kernel is checked. This procedure is repeated until the scale achieves the desired outcome. The transmission between the client and the computer is sluggish, suggesting a minor change. A count of detection windows and their size is computed for the next iteration on the client-side.



This data is transmitted to the computer. Based on the data obtained from the client and the thread index, it is now possible to calculate the location of a current detection window. This adjustment resulted in a 15-fold increase in detection speed [38].

3.4.2. CUDA Deep Neural Network (CuDNN). CuDNN is a GPU-based deep learning library created by NVIDIA. CuDNN is used by many machine learning systems, including Caffe, Tensor-Flow, and Chainer, to boost performance. We assume that the program is written in C++ and that it calls cuDNN and CUDA library functions directly in this study. For CNN computation, cuDNN includes several library functions. The cuDNN Convolution Forward function, for example, performs convolution, the cuDNN. Add Tensor function introduces biases, and the cuDNN Activation Forward function triggers sheet. CuDNN parts may only use data from GPU memory for input and output. To use cuDNN, all data, such as feature maps and weight filters, must first be loaded into GPU memory. Make sure CUDA and cuDNN versions are compatible with our Tensor-Flow edition. We should not have to worry about it because Anaconda will install the required versions of CUDA and cuDNN for the Tensor-Flow version you are using [39].

3.5. Dataset. There were a total of 3694 photos in the dataset. Another choice is to save 20% of the images (730 images) in the test folder and 80% of the images (2964 images) in the train folder, as displayed in Table 1.

3.6. Generate Training Data. Tensor-Flow needs hundreds of images of an object to train a successful detection classifier. The images used in training for a robust classifier should include random items and the target objects and several backgrounds and lighting conditions. Some photographs

TABLE 1: Facial dataset of people with/without wearing a mask.

Source name	Class name	Amount of data	Images
Github	With mask	1847	
Github	Without mask	1847	

should have the target object partly blurred, overlapped with something else, or just halfway visible.

After we have collected all of the images, it is time to mark the items in each one (Figure 3). The tool "LabelImg" is used to mark files. Each image's label data is saved in .xml format by labeling. These .xml files will be used to generate TFRecords, a Tensor-Flow input. Once you have labeled and saved each image, there will be one .xml file in the test and train directories for each.

Now that the images have been called, it is time to make the TFRecords that will serve as input data for the Tensor-Flow training model. The image.xml data will be used to create .csv files containing all the data for the train and test images. Type the following in the Anaconda command prompt. Train record and test record are the two files used to train the current object detection classifier.

The label map is class names to class ID numbers that tell the trainer what kind of object they are dealing with. In a text editor, create a new file named labelmap.pbtxt and save it in the training folder. The numbers in the label map IDs must match those in the generate tfrecord.py format.

The object detection training pipeline, last but not least, must be set up. The training process decides the model and parameters that will be used. This is the final step before starting running training. The faster RCNN inception v2 pets. Config file has been improved with the addition of file paths to the training data and an increase in the number of classes and examples. Save the file after you have made your changes. The training role has been developed and is ready to begin!

3.7. Execute the Training. If all is set up correctly, Tensor-Flow will begin the training. The initialization process will take up to 30 seconds before the actual training begins. It will look like this when you first start training.

The loss is recorded at each stage of training. It will begin high and progressively decrease as training progresses. Our faster RCNN inception V2 model training started at around 3.0 and quickly dropped below 0.8. Enable the model to train

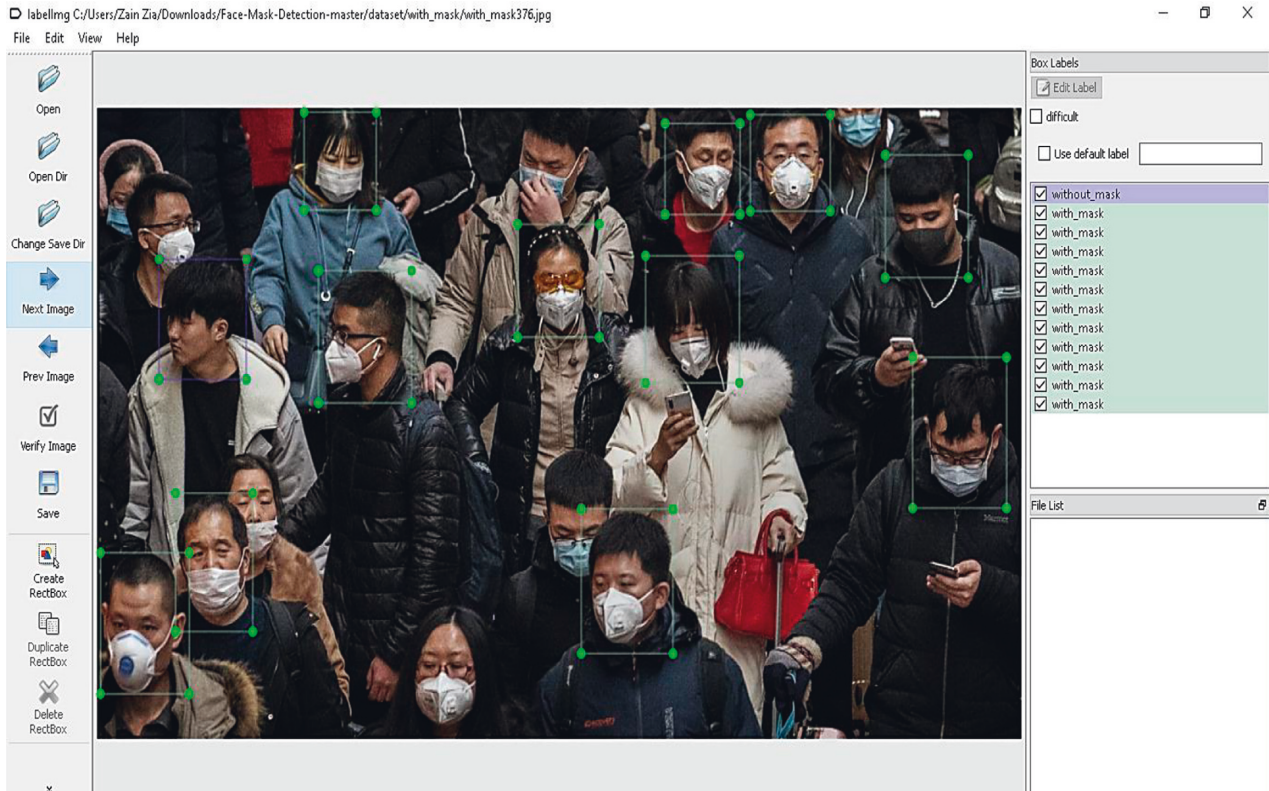


FIGURE 3: Labeled pictures.

for about 40,000 steps or two hours or until the loss is consistently less than 0.05, depending on CPU and GPU.

The algorithm is similar to the RCNN algorithm. Since you do not have to feed the convolutional neural network 2000 region proposals every time, “fast RCNN” is faster than RCNN. Instead, the convolution operation, performed only once per image, produces a feature map.

4. Results and Discussion

4.1. Training Results. When training is completed, TensorBoard keeps track of this operation. Using the tool made it possible to decide whether the model is ready for deployment or requires additional training or other modifications. It was possible to visualize the model’s learning curve using graphs such as total loss. For example, suppose the error rate remains high and constant over time. Either the model’s configuration or the data itself should be updated and corrected, and the training should be terminated.

4.2. Tensor-Board. Through Tensor-Board, we will see how the training has progressed. Tensor-Board is responsible for the visualization graphs. One of the essential graphs is the losses graph, which depicts the cumulative losses of the trained model over time during training. For a GPU-enabled OS, model training took three days. For our faster RCNN inception V2 Coco API training, the loss of the neural network (net) started at five and quickly fell below. Tensorboard is the user interface for visualizing the graph and other

resources for debugging, optimizing, and understanding the model. The number of epochs is represented by the x -axis, while the y -axis represents the time. The recognition rate is calculated in real time as part of our model training. After 200 k epochs, we can see that we have achieved our desired accuracy. The model’s accuracy was improved by data or image augmentation.

The panel has several tabs, each corresponding to the level of data you enter while running the model.

Scalars: During the model training, show a variety of valuable data.

Graphs: Display the model.

Histogram: A histogram can be used to show the weights.

Distribution: Show how the weights are distributed.

Projector: Show T-SNE algorithm and Principal Component Analysis. For dimensionality reduction, this technique is used.

It assists in understanding the interdependencies between operations, how weights are calculated, displaying the loss function, and much more. When you combine all of these pieces of knowledge, you have a powerful tool for debugging and improving the model. A vital graph is the loss graph, which depicts the classifier’s overall loss over time.

4.3. Tensor-Board Losses Graph. While the curve continues to get closer to zero as time goes by, it will never hit that point because nothing is perfect or faultless. A total loss

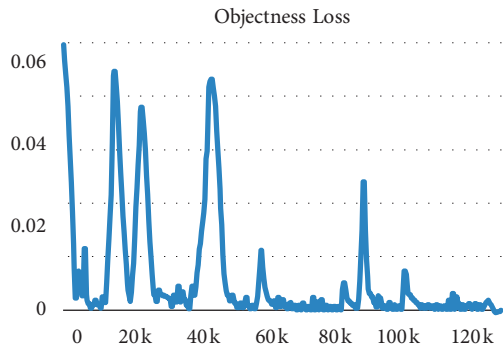


FIGURE 4: Objectness loss.

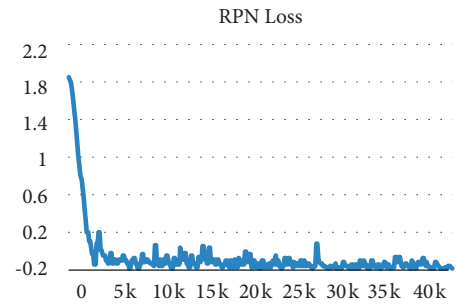


FIGURE 7: RPN loss.

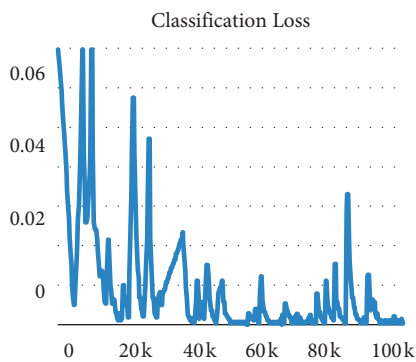


FIGURE 5: Classification loss.

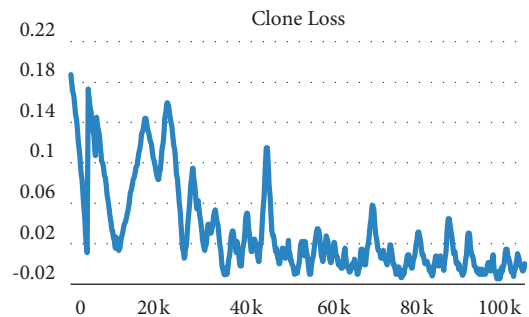


FIGURE 8: Clone loss.

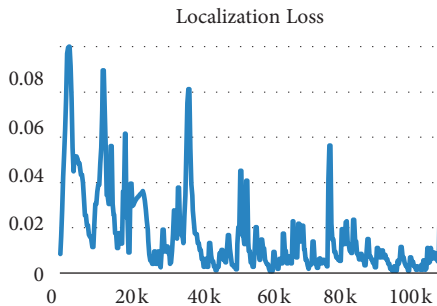


FIGURE 6: Localization loss.

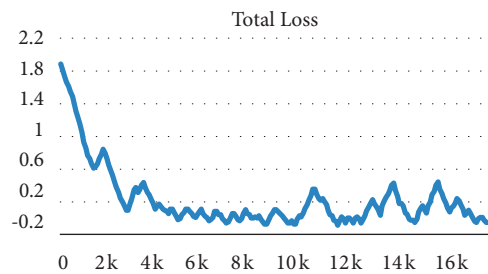


FIGURE 9: Total loss.

value of less than 2.5 is generally considered reliable, but it also indicates that the model may be improved by adjusting parameters or having a better dataset. After 200,000 epochs, the total loss is 0.0503. It tends to decrease. However, the map (mean average precision) does not increase as the loss decreases. When the number of epochs is 200,000, the map is 0.0502. The whole training process is reflected below through key graphs acquired after training. The objectness loss was found zero asymptotically after 120 k epochs (Figure 4). The classification loss was notably substantial during the early phase of training but ultimately comes close to zero asymptotically after 100 k epochs (Figure 5). Similarly, consistent asymptotic behavior is reflected in other

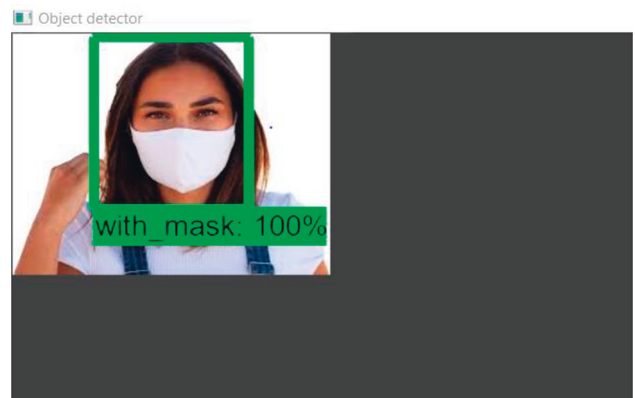


FIGURE 10: Output-I results.



FIGURE 11: Output-II results.

TABLE 2: Comparison of accuracy achieved through different approaches.

Sr. no.	Methodology	Accuracy (%)
1	Naive Bayes	91.3
2	SVM	89.6
3	Decision tree	85.1
4	Kernel approximation method	83.5
5	Proposed approach	97

training graphs, i.e., localization loss, RPN loss, clone loss, and total loss in Figure 6–9, respectively.

4.4. Python Shell. Open the Anaconda command prompt and type “idle” (with virtual environment selected) followed by ENTER to run any scripts. This will start IDLE, allowing us to open and run some of the scripts. Image Object Detection with Tensor-flow Classifier will open when we open the Python Shell. After that, the run module was selected.

4.5. Output Results. In this research, faster RCNN proved to be more efficient and precise in providing 97% accurate results and showed that the processing time for the whole process is less than the other traditional techniques. This study proposed to decide which person is wearing the mask correctly using DNN. When we train our model after 200,000 epochs, the total loss is 0.0503, which tends to decrease. However, the mean average precision does not increase as the loss decreases. When the number of epochs is

200,000, the mean average accuracy is 0.0502. Sample results are shown in Figures 10 and 11.

5. Conclusion

In this article, we introduced a reliable DNN-based system for identifying people wearing masks. Faster RCNN was employed to train the data in this method, resulting in high accuracy. This model is trained on a GPU to obtain a low computational cost. To achieve our goals, we used a multiphase detection model: First, to label the face mask, and second to detect the edge and compute edge projection for the chosen face region within the face mask. The current findings revealed that faster RCNN was efficient and precise, giving 97% accuracy. The overall loss after 200,000 epochs is 0.0503, with a trend to decrease. While the loss is decreasing, we are getting more accurate results. As a result, the faster RCNN technique effectively identifies whether a person is wearing face masks or not, and the training period was decreased with better accuracy. In the future, Deep Neural Network (DNN) might be used first to train the data and then compress the dimensions of the input to run it on low-powered devices, resulting in a lower computational cost. The results achieved from the proposed approach reflect significant accuracy as compared to the other commonly used approaches, i.e., Table 2.

Data Availability

No such private data were used to support the findings of the study. Only publicly available data has been used.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] T. Greenhalgh, M. B. Schmid, T. Czypionka, D. Bassler, and L. Gruer, "Face masks for the public during the covid-19 crisis," *BMJ*, vol. 369, p. 1435, 2020.
- [2] S. Feng, C. Shen, N. Xia, W. Song, M. Fan, and B. J. Cowling, "Rational use of face masks in the COVID-19 pandemic," *The Lancet Respiratory Medicine*, vol. 8, no. 5, pp. 434–436, 2020.
- [3] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [4] A. Kumar, Z. J. Zhang, and H. Lyu, "Object detection in real time based on improved single shot multi-box detector algorithm," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, 204 pages, 2020.
- [5] L. Jiao, F. Zhang, F. Liu et al., "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, Article ID 128837, 2019.
- [6] Z. Li and Wu, "Efficient object detection framework and hardware architecture for remote sensing images," *Remote Sensing*, vol. 11, no. 20, p. 2376, 2019.
- [7] H. Schneiderman and T. Kanade, "A statistical method for 3D object detection applied to faces and cars," vol. 1, pp. 746–751, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, vol. 1, pp. 746–751, IEEE, Hilton Head, SC, USA, June 2002.
- [8] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proceedings of the International Conference on Image Processing*, vol. 1, September 2003.
- [9] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing features: efficient boosting procedures for multiclass object detection," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, June 2004.
- [10] K. Hotta, "Robust face detection under partial occlusion," *Systems and Computers in Japan*, vol. 38, no. 13, pp. 39–48, 2007.
- [11] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.
- [12] Y.-Y. Lin and T.-L. Liu, "Robust face detection with multiclass boosting," vol. 1, pp. 680–687, in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 680–687, IEEE, San Diego, CA, USA, June 2005.
- [13] L. Goldmann, U. J. Monich, and T. Sikora, "Components and their topology for robust face detection in the presence of partial occlusions," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 559–569, 2007.
- [14] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.
- [15] V. B. Subburaman and S. Marcel, *Fast bounding box estimation based face detection*, In ECCV, Workshop on Face Detection: Where We Are, and What Next? Lausanne, Switzerland, 2010.
- [16] G. Mesnil, Y. Dauphin, G. Xavier et al., "Unsupervised and transfer learning challenge: a deep learning approach," in *Proceedings of the ICML Workshop on Unsupervised and Transfer Learning*, vol. 27, pp. 97–110, Washington, WA, USA, 2012.
- [17] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2879–2886, IEEE, Providence, RI, USA, June 2012.
- [18] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," in *Proceedings of the IEEE International Joint Conference on Biometrics*, pp. 1–8, Sheraton Sand Key Resort Clearwater, FL, USA, 2014.
- [19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, Oregon, OR, USA, 2013.
- [20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [21] S. S. Farfade, M. J. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval - ICMR*, vol. 15, China, 2015.
- [22] B. F. Klare, B. Klein, E. Taborsky et al., "Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1931–1939, Boston, MA, USA, June 2015.
- [23] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271, Hawaii, HA, USA, 2016.
- [24] Z. Dong, J. Wei, X. Chen, and P. Zheng, "Face detection in security monitoring based on artificial intelligence video retrieval technology," *IEEE Access*, vol. 8, Article ID 63421, undefined 2020.
- [25] X. Sun, P. Wu, and S. C. H. Hoi, "Face detection using deep learning: an improved faster RCNN approach," *Neuro-computing*, vol. 299, pp. 42–50, 2018.
- [26] M. S. Ejaz, M. R. Islam, M. Sifatullah, and A. Sarker, "Implementation of principal component analysis on masked and non-masked face recognition," in *Proceedings of the 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pp. 1–5, IEEE, Dhaka, Bangladesh, May 2019.
- [27] Z. Wang, "Masked face recognition dataset and application," 2020, <https://arxiv.org/abs/2003.09093>.
- [28] X. Fan and M. Jiang, "RetinaFaceMask: a single stage face mask detector for assisting control of the COVID-19 pandemic," 2020, <https://arxiv.org/abs/2005.03950>.
- [29] T. V. Janahiraman and M. S. M. Subuhan, "Traffic Light Detection Using Tensorflow Object Detection Framework," in *Proceedings of the 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, pp. 108–113, IEEE, Shah Alam, Malaysia, October 2019.
- [30] S. Yin, H. Li, and L. Teng, "Airport detection based on improved faster RCNN in large scale remote sensing images," *Sensing and Imaging*, vol. 21, no. 1, p. 49, 2020.
- [31] J. Zhang, G. Ye, Z. Tu et al., "A spatial attentive and temporal dilated (SATD) GCN for skeleton-based action recognition," *CAAI Transactions on Intelligence Technology*, vol. 7, no. 1, pp. 46–55, 2022.

- [32] K. Jafarbigloo, H. Danyali, Sanaz, and H. Danyali, "Nuclear atypia grading in breast cancer histopathological images based on CNN feature extraction and LSTM classification," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 4, pp. 426–439, 2021.
- [33] S. Fatima, N. Aiman Aslam, I. Tariq, and N. Ali, "Home security and automation based on internet of things: a comprehensive review," *IOP Conference Series: Materials Science and Engineering*, vol. 899, no. 1, Article ID 012011, 2020.
- [34] Q. Zou, K. Xiong, Q. Fang, and B. Jiang, "Deep imitation reinforcement learning for self driving by vision," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 4, pp. 493–503, 2021.
- [35] M. A. Aslam, M. Naveed Salik, F. Chughtai, N. Ali, S. Hanif Dar, and T. Khalil, "Image classification based on mid-level feature fusion," in *Proceedings of the 2019 15th International Conference on Emerging Technologies (ICET)*, pp. 1–6, IEEE, Peshawar, Pakistan, December 2019.
- [36] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, IEEE, Santiago, Chile, December 2015.
- [37] R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra, and J. M. Z. Maningo, "Object detection using convolutional neural networks," in *Proceedings of the TENCON 2018 - 2018 IEEE Region 10 Conference*, pp. 2023–2027, IEEE, Jeju, Republic of the Korea, October 2018.
- [38] J. Krpec and M. Nemeč, "Face detection cuda accelerating," in *Proceedings of the ACHI 2012, The Fifth International Conference on Advances in Computer-Human Interactions*, pp. 155–160, Citeseer, New Jersey, NJ, USA, 2012.
- [39] Y. Ito, R. Matsumiya, and T. Endo, "ooc_cuDNN: accommodating convolutional neural networks over GPU memory capacity," in *Proceedings of the 2017 IEEE International Conference on Big Data*, pp. 183–192, IEEE, Boston, MA, USA, December 2017.
- [40] A. Kadiyala and A. Kumar, "Applications of Python to evaluate environmental data science problems," *Environmental Progress & Sustainable Energy*, vol. 36, no. 6, pp. 1580–1586, 2017.

Research Article

Hybrid Approach for Shelf Monitoring and Planogram Compliance (Hyb-SMPC) in Retails Using Deep Learning and Computer Vision

Mehwish Saqlain ¹, Saddaf Rubab ^{1,2}, Malik M. Khan ¹, Nouman Ali ³,
and Shahzeb Ali ⁴

¹National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan

²Department of Computer Engineering, College of Computing and Informatics, University of Sharjah, Sharjah 27272, UAE

³Department of Software Engineering, Mirpur University of Science and Technology (MUST), Mirpur 10250, (AJK), Pakistan

⁴COMSATS University, Islamabad 44000, Pakistan

Correspondence should be addressed to Saddaf Rubab; saddaf@mcs.edu.pk

Received 4 February 2022; Accepted 10 May 2022; Published 15 June 2022

Academic Editor: Abdul Qadeer Khan

Copyright © 2022 Mehwish Saqlain et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In retail management, the continuous monitoring of shelves to keep track of the availability of the products and following proper layout are the two important factors that boost the sales and improve customer's level of satisfaction. The studies conducted earlier were either performing shelf monitoring or verifying planogram compliance. As both the activities are important, to tackle this problem, we presented a deep learning and computer vision-based hybrid approach called Hyb-SMPC that deals with both activities. The Hyb-SMPC approach consists of two modules: The first module detects fine-grained retail products using one-stage deep learning detector. For the detection part, the comparison of three deep learning-based detectors, You Only Look Once (YOLO V4), YOLO V5, and You Only Learn One Representation (YOLOR), is provided and the one giving the best result will be selected. The selected detector will perform detection of different categories of SKUs and racks. The second module performs planogram compliance; for this purpose, the company-provided layout is first converted to JavaScript Object Notation (JSON) and then the matching is performed with the postprocessed retail images. The compliance reports will be generated at the end for indicating the level of compliance. The approach is tested in both quantitative and qualitative manners. The quantitative analysis demonstrates that the proposed approach achieved an accuracy up to 99% on the provided dataset of retail, whereas the qualitative evaluation indicates increase in sales and customers' satisfaction level.

1. Introduction

Retailing encompasses the selling of goods and services. It is an integral part of the modern society and also acts as a driving force by contributing significantly to the GDP and aims at encouraging sustained growth [1]. The improvement of living standards of society leads to evolution of retails at an accelerated rate. As Artificial Intelligence (AI) is revolutionizing every sphere of life, enterprises are also focusing on using AI to reshape the ecology of retail industry [2]. Retail management and improving customer's experience

are a challenging task due to multitude of tasks required to be performed in concurrent manner, for example, inventory management, shelves organization, and customer's support.

The predefined arrangement of products within shelves is called *planogram*: it demonstrates the layout of placing each product within shelves and indicates how many *facings* should be present, that is, how many stock-keeping units (SKUs) of the same product should be visible in the front row of the shelf [3]. The effective organization of SKUs on the shelves attracts more customers and helps them to choose and pick the products in an efficient manner [4]. To achieve this objective,

corporations invest in tools and studies to create planograms that are part of their optimal store policy. After proper planning, a layout of placing stock-keeping units (SKUs) is decided by the headquarter, and that particular layout is communicated to the retailers. Retailers are then offered certain discounts or monetary benefits for following the planogram provided to them. As the organizations are investing time and money, they also need to verify the compliance of their planogram by the retailers and stores. At present, the verification of planogram compliance is the responsibility of the store personnel and is routinely performed [5].

The auditing of the shelves, that is, keeping track of SKUs availability, their number, and positioning at different locations, is necessary for optimized management of the retails. The expansion of retail industry with the passage of time also makes shelf monitoring a tough job. Traditionally, the auditing of the shelves is performed manually by store representatives in retail environment. The manual approach is very time-consuming, requires extensive human labor, and is subject to human errors. All these aspects contribute to making inventory management a difficult task.

The important factors that boost sales and improve satisfaction level of customers are (a) the availability of products and (b) the arrangement of products on supermarket shelves [6]. One of the major problems in retail environment is out-of-stock products; a study conducted in [7] showed that, in case of no availability of the required products, 31% of the customers prefer to move to other stores, 26% switch to another brand, and 15% delay their purchase to some other time, whereas 9% buy nothing. This illustrates that on-time availability of the products is a crucial factor affecting the sales environment.

Research also indicates that following 100% optimal planogram can amplify the sales to 7.8% and boost the profit [4]. It also helps merchandisers to make more effective decisions about inventory management. The management of proper counters of available products and producing alerts in case of misplaced SKUs and decreasing the levels of products will encourage the organizations to take appropriate steps and decrease the stocking issues.

For the optimized retail management, planogram compliance and shelf monitoring should be performed in an automated way. To automate this process, object detection in the images of shelves can solve problem of monitoring different categories and subcategories of SKUs, completing missing SKUs, and matching planogram continuously. Fine-grained object recognition in retail industry is a challenging task due to below-mentioned reasons.

Racks are not properly organized and variation in product poses causes problems; products are placed in different order; they are often placed in a horizontal manner. This cluttered condition causes complexity of scene as depicted in Figure 1(a). Different resolution of image capturing device produces different quality images, making product detection difficult. In different strategies of capturing images and variation in the image parameters, the length of different products is mapped to different resolution of pixel (Figure 1(b)). The jitter and camera shake while capturing images cause blurry images which make it difficult

to recognize the products as the details are not clearly visible to be detected (Figure 1(c)).

The catchy, glary, and glossy packages of products and uneven illumination, shadows, and lightening conditions cause reflection as illustrated in Figures 1(d) and 1(e). The images captured from the oblique angles and not from the frontal view cause distortion (Figure 1(f)). Different shapes, sizes, colors, and minute visual difference in product packages require fine-grained classification (Figures 1(g) and 1(h)).

Due to all these problems, fine-grained product recognition becomes difficult. The analysis of existing studies indicated that the studies conducted in the past were either performing shelf monitoring or checking planogram compliance. As both activities are critical in optimized management of retails, a hybrid approach is required to perform both activities in an efficient manner. To deal with this challenge, a hybrid approach for shelf monitoring and planogram compliance which is called Hyb-SMPC is proposed in this work. For the detection part, three deep learning models (YOLO V4, YOLO V5, and YOLOR) will be compared and the one having accurate results will be used. The following are the contributions of this research work:

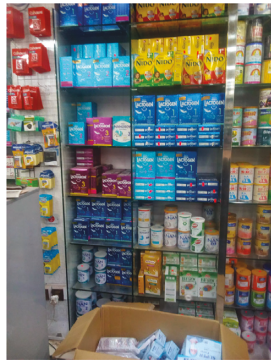
- (i) Fine-grained SKUs detection in the retail by using deep learning.
- (ii) Shelf monitoring, that is, keeping track of the instances of the SKUs present on the shelves.
- (iii) Verifying the compliance of the planogram using techniques of computer vision.

The organization of the rest of this paper is as follows. Section 2 presents related work. Section 3 provides the details of the proposed approach. Results are provided in Section 4 and, finally, the conclusion and the future work are provided in Section 5.

2. Related Work

2.1. State of the Art in Computer Vision. In our daily life retails are playing a significant role. The number of products in the retail increases every day with the increasing number of new products coming into the market; in this situation, the traditional way of retail management is very difficult as product detection and inventory management require extensive human labor.

Deep learning and computer vision are used in various applications such as image classification, object detection in industrial production [8], medical image analysis, and action recognition [9, 10]. To automate this manual process, various computer vision-based solutions have been proposed in the literature. The first attempt of product recognition in grocery stores was done in [11], where the authors applied three different object recognition algorithms based on local invariant features. The proposed methodology did not perform well considering precision and efficiency but one of the important contributions of that work was the provision of a dataset called GroZi-120 consisting of 120 different grocery products. The dataset is publicly available for further research [11]. Feature-based product detection methods



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

FIGURE 1: Improper product placements on the racks. (a) Unorganized products. (b) Difference in dimension. (c) Blurry image. (d) Uneven illumination. (e) Lightening reflection. (f). Nonfrontal image. (g). Minute difference of products. (h) Change in product size.

were extensively studied in the past. These features are key-point-based, gradient-based, color-based, pattern-based, and deep-learning-based [12].

2.1.1. Key-Point-Based Features. Key-point-based feature detection method detects the key points from the images, and this is the technique frequently employed in the retail scenario. Scale-Invariant Feature Transform (SIFT) [13] and Speeded-Up Robust Feature (SURF) [14] are the well-known feature extraction algorithms. SIFT features are invariant to scale, rotation, and illumination of image. By using SIFT in [15], features of input images were compared with stored features of the objects in the database and recognition was performed; however, in [16], the on-shelf availability and misplaced products were detected by using SURF descriptors. In the first phase, counting is performed; by looking at the duplicate properties provided by SURF descriptor, each item of the product is counted. In the second phase, rectangular bounding boxes are fitted around each product and the products are identified with SURF and color properties.

Reference [17] employed logo-detection-based algorithm for the recognition of products on shelves. The algorithm includes two steps. Products were detected and classified on the basis of their brands by matching SIFT key points and the finer classification was performed by using color information and the exact product label was recognized. Visual monitoring of shelves was performed in [4]; the system provided in the study analyzed the images and verified compliance of the planogram. For the detection of the objects, three different approaches were used; the vote map approach used SURF descriptors and outperformed the other approaches.

2.1.2. Gradient-Based Features. These features focus on the structure and shape of the object. Histogram of Oriented Gradients (HOG) and Sobel and Prewitt operators are generally used for gradient computation. These operators incorporate geometric shapes, edges, and corners of the products detected from the images [12]. Gradient-based features like color and shape were captured in [18] for the purpose of product detection in retails. Sobel operator was used in [19] and the authors presented an automated planogram compliance check in retails by providing a framework based on visual analysis. The framework consists of three modules which performed row extraction and occupancy computation and identified completely and partially missing case. Through exploiting color and texture properties, the counting of the products was performed, and their placement was checked. Nevertheless, the study did not perform product detection. In [20], a heuristic approach to count the instances of the same product and detect the missing item on the shelf without using classifiers is specified. The proposed algorithm includes morphological operations, template matching, and histogram comparison. The experimental results demonstrate that the satisfactory results are achieved with the algorithm only when the user manually selects the most substantial label of the product from the shelf image.

2.1.3. Pattern-Based Features. In the detection of retail products using recurring patterns, Haar and Haar-like features are extensively used pattern-based features [12]. A fine-grained recognition of grocery products by integrating VGG-16 with recurring features and attention maps was proposed in [21]. Recurring features detect the candidate region and give coarse labels to the products; afterwards attention maps help the classifier to concentrate on the fine details in the candidate region of interest (ROI). The mean average precision (mAP) of 0.75 is achieved with the proposed method. The authors in [11] analyzed the performance of boosted Haar-like features, SIFT, and color histogram matching algorithms and found that SIFT outperformed the others. An automatic method for checking the planogram compliance is suggested in [22] without the requirement of template images through the detection of recurring features. Graph matching technique was used to match the provided layout with the extracted layout.

All of the above-mentioned studies conducted for object detection and planogram compliance used traditional computer vision techniques. These traditional techniques require manually designed features; these hand-crafted features do not always reflect sufficient information. Each feature characterization requires working with a plethora of parameters [23]; with increasing number of classes, feature extraction becomes inconvenient and it becomes the responsibility of CV engineers to select the features which identify different classes of objects in the best manner. To deal with this difficulty, deep-learning-based methods were introduced, which are based on the concept of end-to-end learning [23]. Recent research is focused on the use of mid-level features and deep learning models to build robust decision support systems such as smart vehicles and IoT applications [24–28].

2.2. State of the Art in Deep Learning. Over the time, deep learning has emerged efficiently by showing improved performance. Deep learning has the ability to learn the features automatically from the images [23]. Another merit of deep learning is the deeper layers, which can extract precise features, whereas simple neural networks are not able to do that [2]. The most frequently used technique in deep learning is convolutional neural network- (CNN-) based object detection. You Only Look Once (YOLO) [29], Single-Shot Detector (SSD) [30], and Region-based Convolutional Neural Networks (R-CNN) [31] are the modern variants of CNN which are very efficient. CNN outperformed traditional methods based on hand-crafted features such as SIFT [13] and SURF [14] as they are unable to extract deep information from images.

Many researchers contributed to using CNN for detection of products in retails. Reference [32] used convolution neural network to resolve the issue of in-store product recognition and achieved an accuracy of 78.9%. The challenge of large-scale fine-grained structure classification was handled in [33] by exploiting contextual information along with deep network.

To investigate the number of products present on the shelf, [34] took the help of surveillance cameras to record the

videos of the shelves to take account of the number of the present products. The study tracks the changed regions by background subtraction method; afterwards moving objects are removed and CNN based on CaffeNet is used for the classification of changed regions. The success rate of 89.6% was achieved with this study [34]. The extension of the work is provided in [35] which used images from the surveillance camera for monitoring the availability of products. The Hungarian method distinguishes the foreground from the successive image. The classification of detected changed region is performed by two deep networks, that is, CIFAR-10 and CaffeNet. This methodology also helps in the determination of shelves which are accessed commonly [35]. A fast detection and recognition method based on fine-grained categories of products is anticipated in [36] when very limited training data is available for training. The results indicate that 52.16% mAP was achieved for recognition of each product.

Reference [37] provided a template-free, zero-short product detection system which avoided templates and detected the products by segmenting the shelves horizontally into layers and vertically into products. The classifications of horizontal layers are performed by GoogLeNet, whereas vertical division is performed by another trained GoogLeNet. The results indicate better performance compared to the existing methods; however, the empty regions between the products influence the method negatively, making it less robust.

A deep learning approach was suggested by [38] for planogram compliance in retail stores. The images are collected through the robot NAVii and also from the Internet and then split into three different training sets for training three different CNN models. The CNN model trained on both Internet and store images gives better accuracy than other models and can generalize in a much better way because of exposure to the great variation of products.

Deep-learning-based object detection methods are divided into two categories: two-stage detectors and one-stage detectors. Region-based Convolutional Neural Networks (R-CNN) [39], Faster R-CNN [31], and Mask R-CNN [40] come under the category of two-stage detectors. One-stage detectors deal with object detection as a simple problem of regression. RetinaNet [41], YOLO [29], and SSD [42] are well known one-stage detectors.

Reference [43] provided a semisupervised deep-learning-based image classification approach for shelf auditing. The study merged the two ideas of “semisupervised” and “on-shelf availability (SOSA).” Semisupervised learning took advantage of both labeled and unlabeled data. Deep learning architecture YOLO V4 is used for on-shelf auditing (OSA); it makes comparison of three different approaches of deep learning (RetinaNet, YOLO V3, and YOLO V4) for monitoring OSA and the best results were achieved with YOLO V4; however, the study did not perform planogram compliance.

There are very few studies regarding checking of planogram compliance in retails. In [44], deep-learning-based hybrid approach based on image classification and object

detection is provided to solve the problem of planogram compliance in retails. For assessment of quality of the images, Blind/Referenceless Image Spatial QUality Evaluator BRISQUE [45] technique was integrated into the framework. Eight different types of templates were taken into account to train the model. The products were classified into two classes, that is, “Exact 7 by 4” and “No Exact 7 by 4.” VGG-16 was used for classification and Tiny YOLO V2 [46] was used for object detection. The overall accuracy achieved by this hybrid approach reaches 95% [44].

YOLO V5 and YOLOR are recently released versions, so no work has yet been done in the retail industry using these models. The proposed study is the first one to provide comparison of these models in shelf monitoring and planogram compliance. The summary of the studies conducted in the past is given in Table 1 which gives the clear idea that all the studies conducted in the past were performing one of two tasks: shelf monitoring and planogram compliance. Hence, it was discovered that there is a dire need for a hybrid approach that can perform both shelf monitoring and planogram compliance in retails. The novelty of the proposed approach is to perform both tasks.

3. Proposed Approach: Hybrid Approach for Shelf Monitoring and Planogram Compliance (Hyb-SMPC)

The proposed technique is a hybrid approach which combines both concepts, “shelf monitoring” and “planogram compliance,” for the first time to facilitate retail management. The process involves object detection at fine-grained level followed by verification of planogram compliance using shelf images. For this purpose, the study used three one-stage deep learning detectors, that is, YOLO V4 [47], YOLO V5 [48], and YOLOR [49], to detect and classify the products. The proposed study is the first one to provide comparison of these three detectors. The approach is broadly divided into two modules as illustrated in Figure 2.

The first module is product detection module. It performs detection and localization of racks as well as SKUs in parallel manner. The process is followed by the next module called planogram compliance module, which verifies the specific placement policy of SKUs formulated by the company.

The general process of training the models is represented in Figure 3. The study used image-based dataset which is provided by the industry partner and is collected from different retails, stores, and supermarkets. As the dataset is image-based, preprocessing involves images resizing, denoising, and image labeling. Image labeling is a process of providing annotations to the images. Labels are provided on the basis of type of products. Different retails contain variety of racks, so racks are also labeled in the preprocessing stage. After preprocessing stage, the next phase is the splitting of data into two subsets, that is, training and testing, and then training of three different detectors is performed using labeled data of training subset. For training 200 images per product, SKUs are used.

TABLE 1: Comparison of proposed approach with the previous approaches.

Reference no.	Year	Object detection		Planogram compliance	Methods
		Traditional method	Deep learning Two-stage One-stage		
[4]	2015			✓	SURF
[16]	2015	✓			SURF
[18]	2015		✓		SVM
[19]	2015			✓	SURF + color histogram
[22]	2016			✓	Recurring pattern detection
[38]	2016			✓	CNN
[36]	2017		✓		VGG-F
[17]	2018	✓			SIFT
[21]	2018		✓		VGG-16 with recurring features and attention maps
[37]	2018		✓		GoogLeNet
[35]	2019		✓		CIFAR-10, CaffeNet
[43]	2020			✓	RetinaNet, YOLO V3, YOLO V4
[44]	2020			✓	VGG-16, Tiny YOLO V2
Proposed method			✓	✓	YOLO V4, YOLO V5, YOLOR

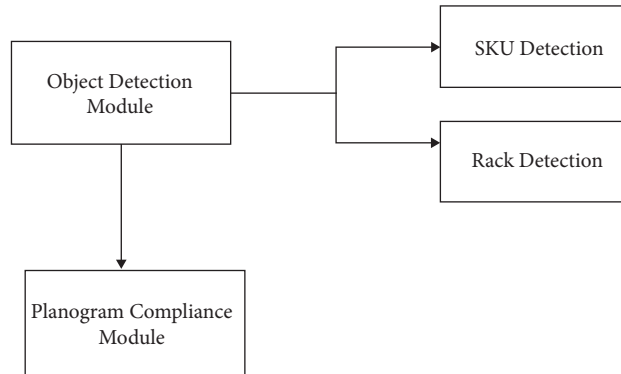


FIGURE 2: Modular view of Hyb-SMPC.

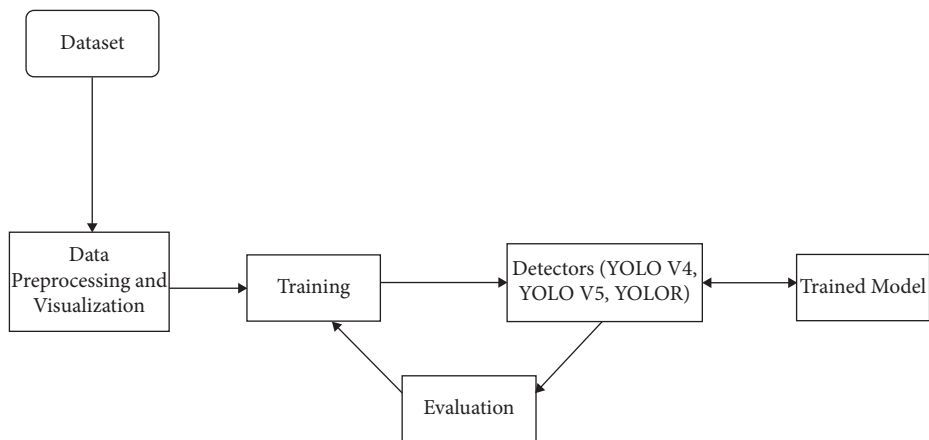


FIGURE 3: Model training process.

Three one-stage detectors are trained by providing labeled data (YOLO V4, YOLO V5, and YOLOR). The SKUs of the dataset are categorized under twelve categories which are mentioned in Table 2. The testing of the detectors is performed with the test set and performance is evaluated using different accuracy metrics.

After training and tuning of detectors are performed, the detector producing the most accurate results among the three will then be deployed on the server to perform detection. The detailed flow of the proposed hybrid approach is represented in Figure 4. The first step is the provision of labeled images from the real retail environment to the three detectors for training. The next step is the selection of the best model depending upon the results of different evaluation metrics. Afterwards unlabeled images would be provided to the best trained detector; the selected detector will then perform detections.

After detection, the detector provides the processed images as indicated in Figures 5(a) and 5(b) which contain the list of detected SKUs, along with their counters and IDs to illustrate the number of instances of the SKUs present on the shelves. Figures 5(c) and 5(d) show the racks detected by the detector. Later SKUs and racks are sorted with respect to x , y coordinates and postprocessed retail images are generated. The sorting of SKUs and racks with respect to coordinates is very important as it will act as an input for the second module and provide help in planogram checking. The postprocessed retail images illustrated in Figures 6(a) and 6(b) contain sorted SKUs in the sorted racks.

After the detection of SKUs and racks, the second module called the planogram compliance module will start its working. For planogram matching, the study explored two methods:

- (1) Planogram matching using color detection
- (2) Planogram matching by generating JSON

3.1. Planogram Matching Using Color Detection. In planogram matching using color detection for verification of layout, the postprocessed retail images received from the first module are converted into planogram images by using Python libraries called OpenCV and PIL image. Afterwards, the generated planogram and the company-provided planogram are matched.

For making planogram image, all the racks, shelves, and SKUs will be represented as 2D blocks. A rectangular block which depicts the whole shelf is first generated (yellow block in Figure 7(b)) by taking into account the shape of a retail shelf. The racks are then generated one by one (green box in Figure 7(b)) by using the information contained in the postprocessed retail images. Each rack contains different two-dimensional (2D) boxes filled with colors representing different categories of SKUs. Different colors are assigned to different categories of SKUs as depicted in Figures 7(a) and 7(b). The postprocessed retail image obtained from the first module is represented in Figure 7(a), whereas Figure 7(b) represents the planogram image generated from it. The company-generated layout is given in Figure 7(c) as a

template. We are using a color dataset that contains Red, Green, Blue (RGB) values with their corresponding names. The CSV file for the color dataset has been taken from [50]. The dataset file includes 865 color names along with their RGB and hex values. Now we assign each RGB color value to the classes belonging to each category. Hence, each color represents a specific SKU in the planogram.

Matching is performed on the basis of same image size and the same number of racks as the company-provided template; for example, the planogram of category chiller with four racks will only be matched with the planogram image of chiller with four racks generated by our planogram module. As we have all the coordinates of racks and SKUs provided by detectors, we pick the cropped image part of each SKU from both planogram images (company-provided and module-generated) to verify its color. If the RGB values of both cropped images get matched, then we store TRUE as a string in an object. Similarly, if they cannot be matched, then we store FALSE.

Afterwards, by counting the number of total TRUEs and FALSEs, we will generate the report of planogram compliance. The threshold values are decided as 10%–90%; a value lying between these limits indicates that the planogram is followed partially, whereas a value above 90% indicates planogram to be followed fully and a value below 10% indicates that planogram is not followed at all.

3.2. Planogram Matching by Generating JSON. In this method, the company-provided template which is shown in Figure 8(b) is given as an input and the module will generate JSON from it using Python functions. These functions will extract racks and SKUs. The generated JSON will be matched with the information extracted from postprocessed retail image (Figure 8(a)) which is also saved in the form of JSON to find whether the planogram is followed fully or partially or is not followed at all.

The matching of company-generated layout with postprocessed retail images of shelves occurs at real time so this process must be efficient. Hence, for this purpose, we used JSON for matching. The matching occurs rack by rack and SKU by SKU starting from the rack one. When the string of both JSONs gets matched, we store TRUE, and if they cannot be matched, we store FALSE as a string in another object. Threshold values are decided as 10%–90%; a value lying between these limits indicates that the planogram is followed partially, whereas a value above 90% indicates planogram to be followed fully and a value below 10% indicates that planogram is not followed at all. The report is generated at the end and is sent to the company as well as to the retailers. Figure 8(a) indicates that the planogram is followed 100%, whereas Figure 8(c) indicates an image which is not following planogram at all.

The comparison of the two methods described above is performed and the average processing time taken for matching planogram for different categories of the products is calculated. Table 3 contains the details. On the basis of the average processing time, it is evident that the planogram matching by generating JSON is more efficient compared to

TABLE 2: Main categories of dataset.

No.	Main categories
1	Juices
2	Chiller
3	Dairy liquid
4	Dairy powder
5	Coffee
6	Milk modifier
7	PTW (powder tea whitener)
8	Infant nutrition
9	BFC (breakfast cereals)
10	Nutrition
11	Nestrade
12	Sachets

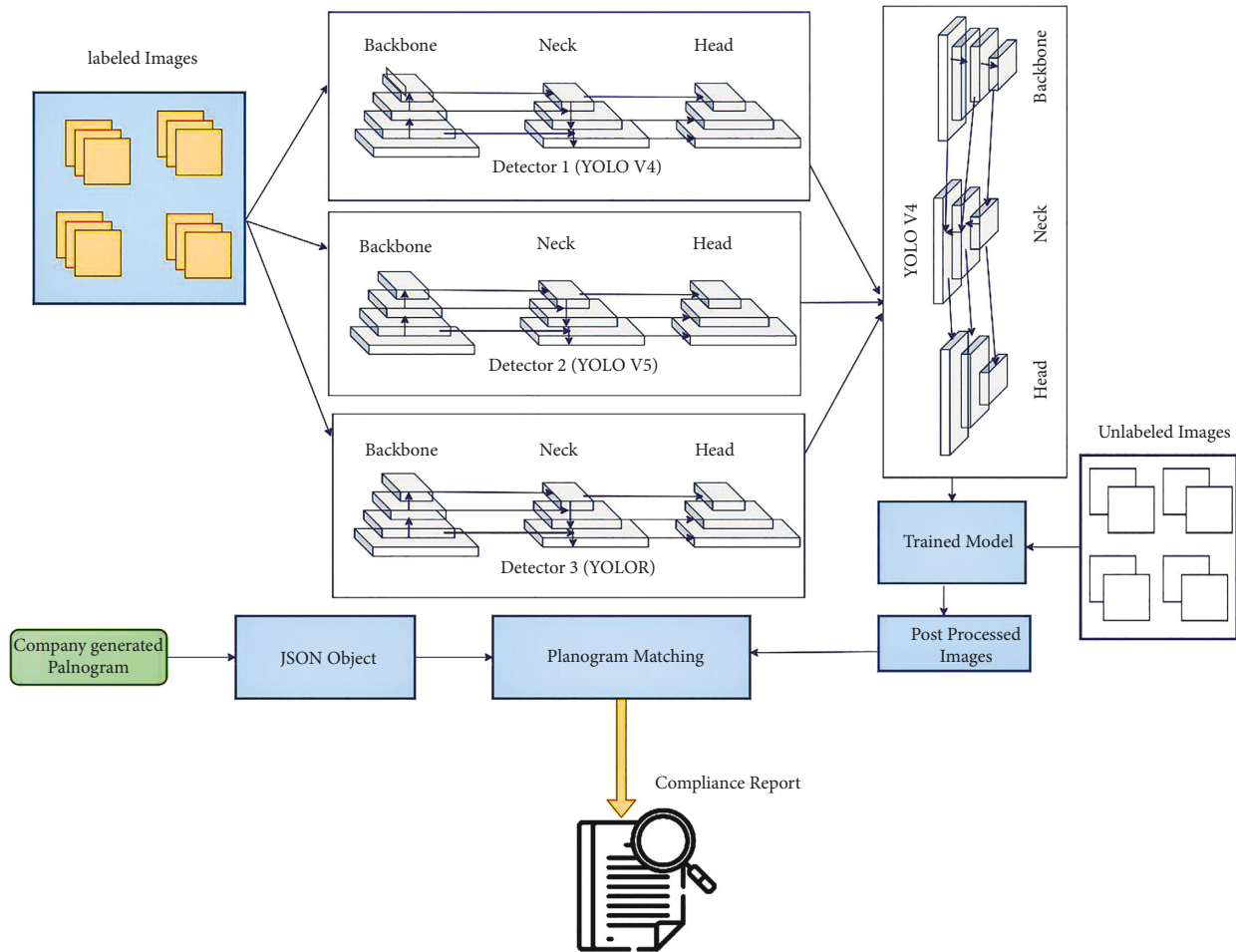


FIGURE 4: Workflow of Hyb-SMPC.

the other method. Therefore, in Hyb-SMPC, the planogram is matched through generating JSON.

3.3. Formal Definition of Hyb-SMPC Approach. Assume that $D = \{(a_1, b_1), (a_2, b_2), (a_3, b_3), \dots, (a_p, b_p)\}$ has p images with labeled SKUs. In each factor (a, b) , $a \in A$, input space, whereas $b \in B = \{l_1, l_2, l_3, \dots, l_q\}$ has q class labels. The proposed approach considers a function $f: A \leftrightarrow B$ for mapping unseen input images (IM) to correct class labels

(b). $M = \{m_1, m_2, \dots, m_j\}$, where m refers to one-stage detectors; in the proposed approach, m_1 refers to YOLO V4, m_2 refers to YOLO V5, and m_3 refers to YOLOR; hence, $j = 3$.

The general flow of the proposed approach is given in Algorithm 1 containing five steps. The first step is division of dataset D into D_{train} and D_{test} by 80 percent and 20 percent and training of detectors is performed. In the second step, all the three detectors will be tested using labeled test dataset D_{test} and the best detector is selected by comparing the obtained results. The third step of the algorithm gives the input images (IM) to



(a)



(b)



(c)



(d)

FIGURE 5: Processed images containing ((a) and (b)) detected SKUs and ((c) and (d)) detected racks.



(a)



(b)

FIGURE 6: Postprocessed retail images: (a) postprocessed image 1; (b) postprocessed image 2.

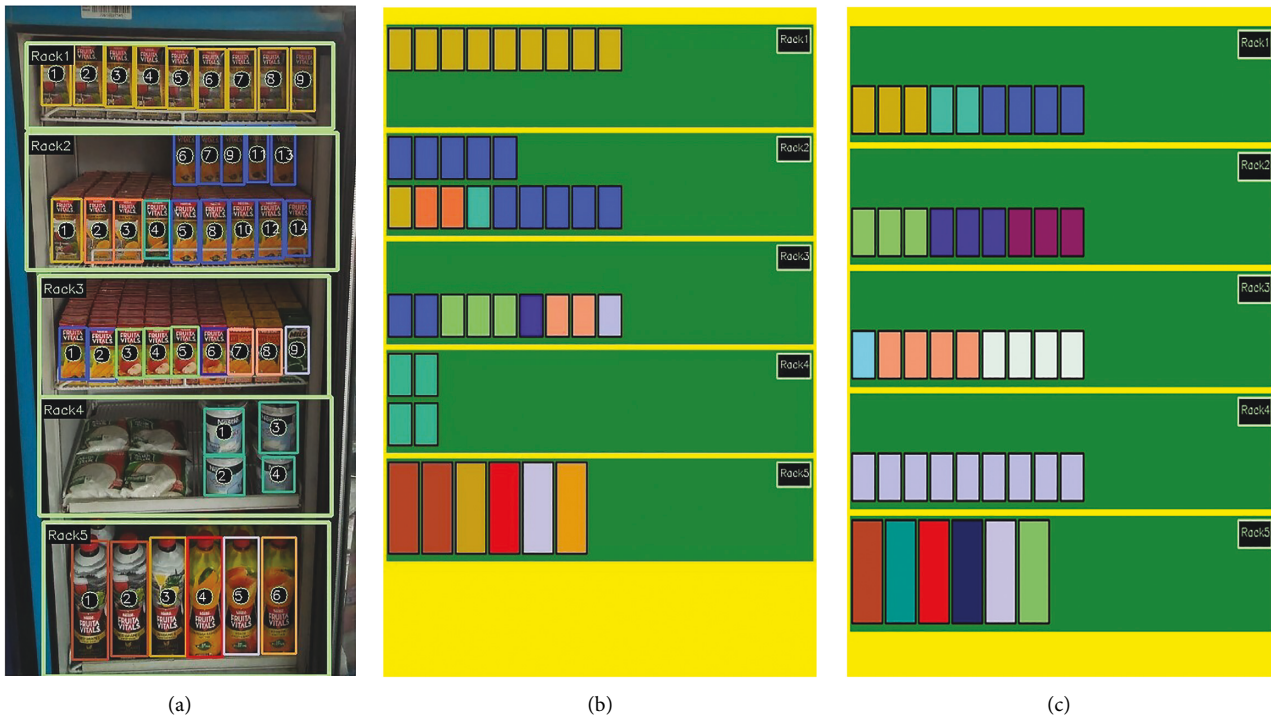


FIGURE 7: Planogram matching by color detection. (a) Postprocessed retail image. (b) Generated planogram. (c) Company-provided planogram.

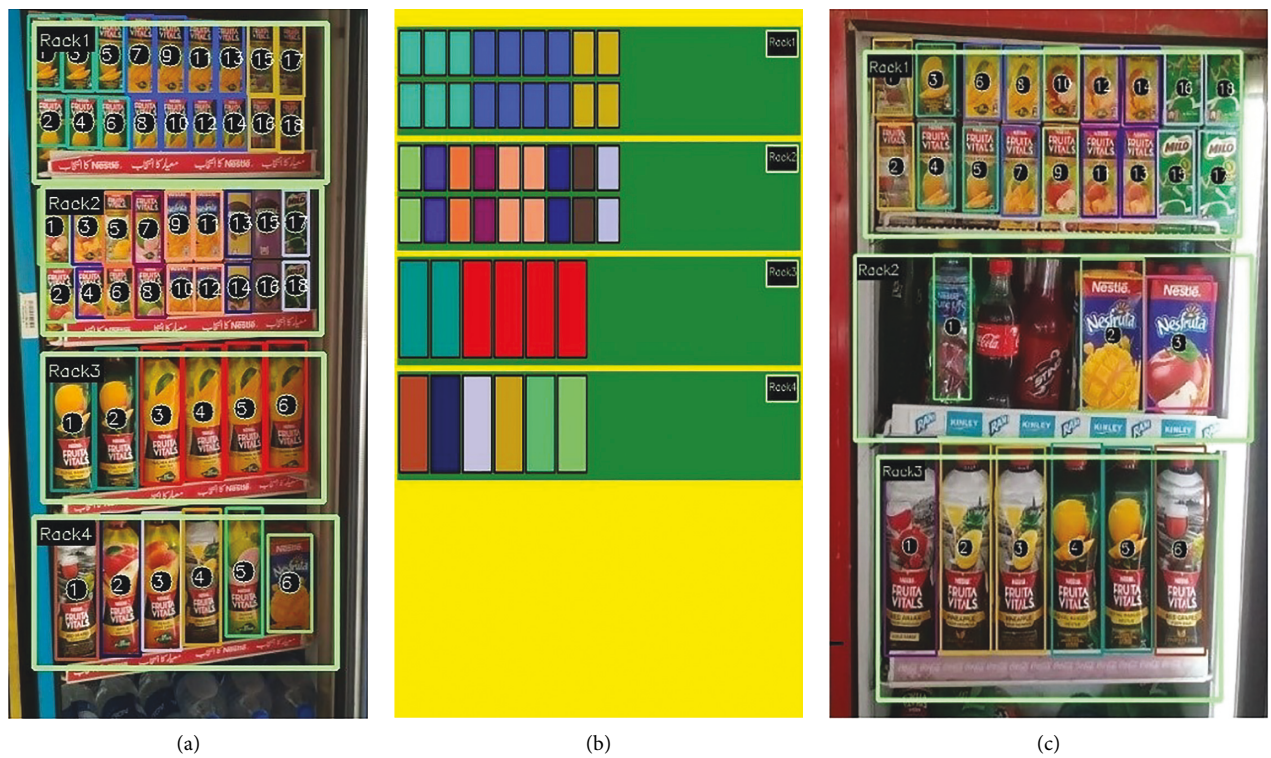


FIGURE 8: Planogram matching by generating JSON. (a) Postprocessed retail image. (b) Company-generated planogram. (c) Unmatched postprocessed retail image.

TABLE 3: Average processing time for planogram matching.

Categories	Planogram matching through color detection (sec)	Planogram matching by generating JSON (sec)
Juices	7	1.5
Chiller	6	1.7
Dairy liquid	7.1	1.2
Dairy powder	7.2	1.3
Coffee	8.2	1.2
Milk modifier	8	1.5
PTW	7.8	1.8
Infant nutrition	6	1.7
BFC	4.9	1.4
Nutrition	5.9	1.5
Nestrade	6.7	1.5
Sachets	7.7	1.9

```

Input:  $D$ : Labeled dataset  $D = \{(a_1, b_1), (a_2, b_2), (a_3, b_3), \dots, (a_p, b_p)\}$  with  $p$  images
 $B = \{l_1, l_2, l_3, \dots, l_q\}$  with  $q$  classes
 $M = \{m_1, m_2, \dots, m_j\}$  with  $j$  models
 $IM$  = input images
Output: Trained models
 $\hat{b}$  = labels of Classes for the SKUs included in input images
Start:
 $D_{\text{train}} = \text{Split}(D, p * 80)$ 
 $D_{\text{test}} = \text{Split}(D, (p - (p * 80)))$ 
//Step 1—Training of models with labeled data
for  $n = 1$  to  $j$ :
  for every epoch:
    for every  $(a_i, b_i)$  in  $D_{\text{train}}$ :
       $m_j = \text{Train}(a_i, b_i)$ 
    end
  end
end for
//Step 2—Testing models
for  $k = 1$  to  $j$ :
  for every  $(a_i, b_i)$  in  $D_{\text{test}}$ :
    Prediction =  $m_k(a_i)$ 
  end
end for
//Step 3—Detecting SKUs in input image  $\hat{b} = TM(a_i)$ 
Output: Processed images ( $PI$ )
//Step 4—Sorting SKUs and Racks
 $PPI = \text{Sorting}(PI)$ 
//Step 5—Generating Planogram from JSON object and comparing post processed image with Planogram layout
 $JO = \text{contour}(Pg)$ 
foreach  $a_i$  in  $D$  Compare ( $PPI, JO$ )
End Algorithm

```

ALGORITHM 1: Algorithm for Hyb-SMPC

find the predicted class labels \hat{b} for different SKUs through trained detectors M and produces processed images (PI). In the fourth step, SKUs and racks are sorted with respect to x, y coordinates and postprocessed retail images (PPI) are obtained. In the fifth step, JSON (JO) is generated from company-provided planogram template; this step will also match post-processed retail images (PPI) with JO for checking compliance.

4. Experimentation and Results

Evaluation is the vital part of any system and the performance of the models is generally evaluated through experimentation. Different accuracy metrics were used to gauge the efficiency of the proposed approach. The details are provided below.

4.1. Evaluation Metrics. This study evaluates the approach both quantitatively and qualitatively. For evaluating our approach quantitatively, the metrics of precision, average precision (AP), mean average precision (mAP), recall, and the value of $F1$ -score are used to estimate the accuracy of the models [51].

True Positive (TP): Correctly identified the correct SKU.

True Negative (TN): Correctly identified that it is not the correct SKU.

False Positive (FP): Also called false alarm, identifies the wrong SKU.

False Negative (FN): The SKU is not identified when actually it should be identified.

Precision specifies correct detections over total number of detections.

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive (FP)}}. \quad (1)$$

Recall indicates the number of totally corrected SKUs from the list of SKUs visible in the image:

$$\text{Recall} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative (FN)}}. \quad (2)$$

$F1$ -score merges both precision and recall:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (3)$$

Average precision is calculated by taking average of all the values of precision:

$$\text{AP} = \frac{1}{N} \sum_{i=0}^N \text{Precision}. \quad (4)$$

Mean average precision is the average of all the APs:

$$\text{mAP} = \frac{1}{|T|} \sum_{i \in T} AP_i. \quad (5)$$

Here, N indicates the number of instances present in the test set and $|T|$ is total number of all AP s computed for each class.

Compliance accuracy of retail image with respect to company-provided planogram layout is calculated by the following equation, where $P_{matched}$ is the number of matched SKUs and P_{total} indicates the total number of SKUs:

$$\text{Compliance Accuracy} = 1 - \frac{(P_{matched} - P_{total})}{P_{total}}. \quad (6)$$

4.2. Dataset Description. There is always a need for a huge amount of data for deep learning models. For evaluating effectiveness of the proposed approach, the dataset used in this study is provided by the industry partner which contains products of different categories and subcategories used for fine-grained recognition; the dataset contains 30,000 images,

all collected manually from real diverse environment, that is, retails, departmental stores, marts and supermarkets, and shops with natural lightings through different mobile cameras.

The average resolution of the images was 1024×1024 with jpeg format. Images were captured from the distance of 1.5 to 5 feet from the front of the shelves. Each image contains multiple products. The testing set was also collected from real-life scenario through handheld devices. The dataset has a hierarchical structure containing a total of 12 main categories which cover diverse appearance, for example, boxes, bottles, poach, and chiller, and contains 106 different fine-grained SKUs. Figure 9 shows the number of SKUs in each subcategory.

There is very minor difference in the packages of fine-grained categories. Rich annotations are provided to each product including the category, count, sizes, and flavors. 3000 images were labeled by using a labeling tool called LabelImg as presented in Figures 10(a) and 10(b). As each image contains multiple SKUs, almost 50 or more, high accuracy can be achieved during training. The percentages of training set and testing set were 80% and 20%, respectively. Training set and testing set contain 2400 and 600 images, respectively. The images of racks in the dataset were collected from 100 different types of racks which approximately contain six different levels. The annotation tool we used gave .txt file for each image. The text file contains class and location information in the form of class number and x, y, w, h coordinates.

4.3. Quantitative Evaluation. For evaluating Hyb-SMPC, the Amazon Web Service (AWS) instance called Elastic Compute Cloud (EC2) is used. In this study, for training process, Graphic Processing Unit (GPU) used is NVIDIA Tesla V100. At first, the training of the detectors was performed one by one on the GPU. In the proposed study, the Darknet-based framework is used for YOLO V4, whereas YOLO V5 and YOLOR are based on PyTorch-based framework.

Transfer learning is the concept of reusing the knowledge acquired from one specific task in another related newer task. This makes the learning process fast and enhances the performance of the deep learning models. Various models have been trained on challenging datasets which are then used for tackling related problems. In this work, the pre-trained model used was “yolov4.conv.137” for YOLO V4. During training process, the training dataset was divided into small units called batches to perform learning of models. In this work, we used batch size of 64 and number of epochs is 72000. Input size of images was 512×512 , with learning rate of 0.00261.

The training progress plot of the best category is illustrated in Figure 11. This plot helped us in monitoring the training process which is showing the “training accuracy.” The details of average precision achieved by three different detectors trained for different categories are given in Table 4.

The highest average precision of 99% was achieved for the categories of coffee, milk modifier, and powder tea whitener. Furthermore, comparison of mAP , recall, and $F1$ -

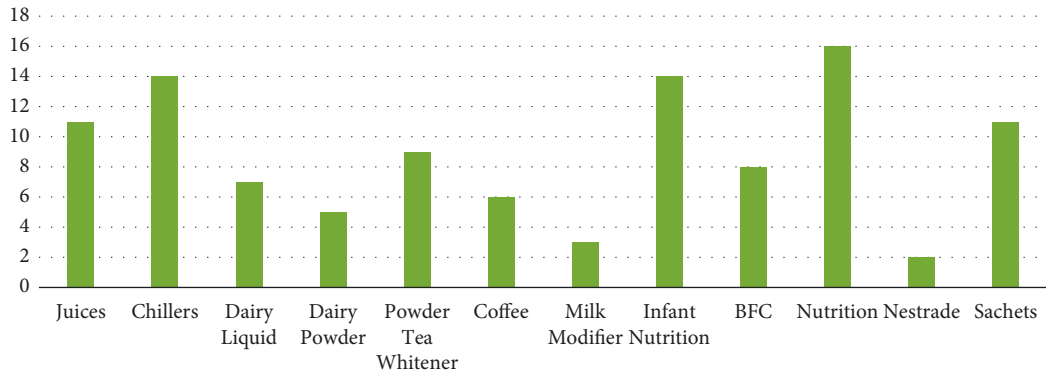
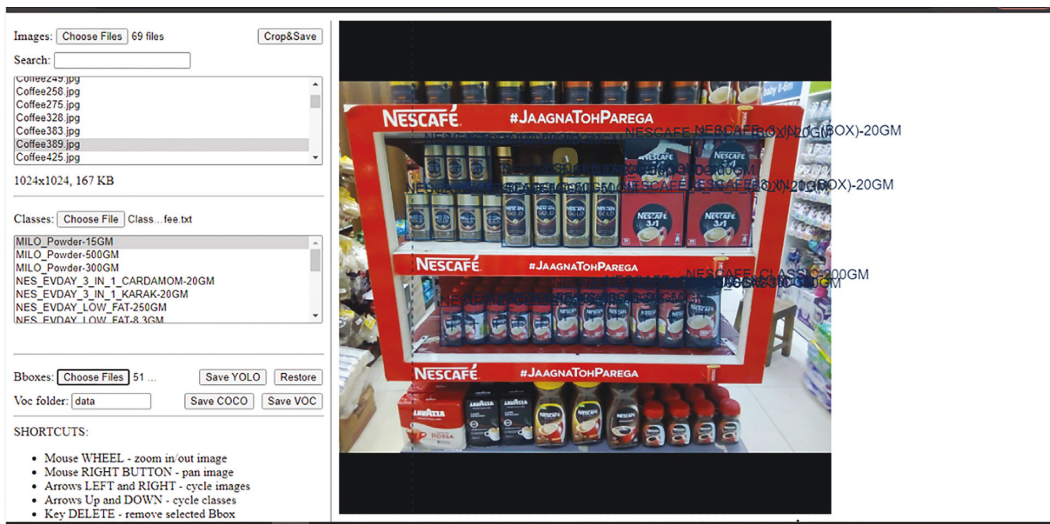
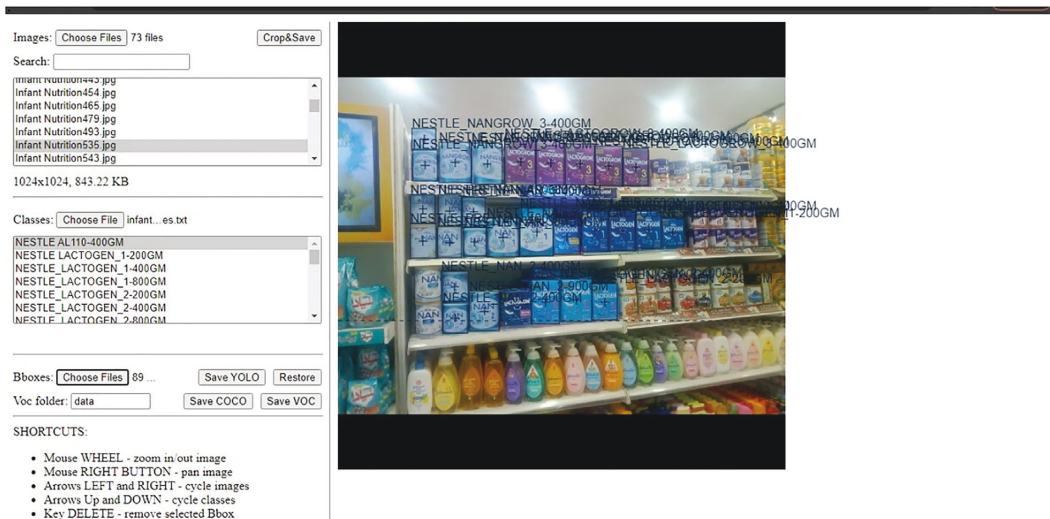


FIGURE 9: Number of SKUs in each category.



(a)



(b)

FIGURE 10: Labeled images. (a) Labeled image for coffee. (b) Labeled image for infant nutrition.

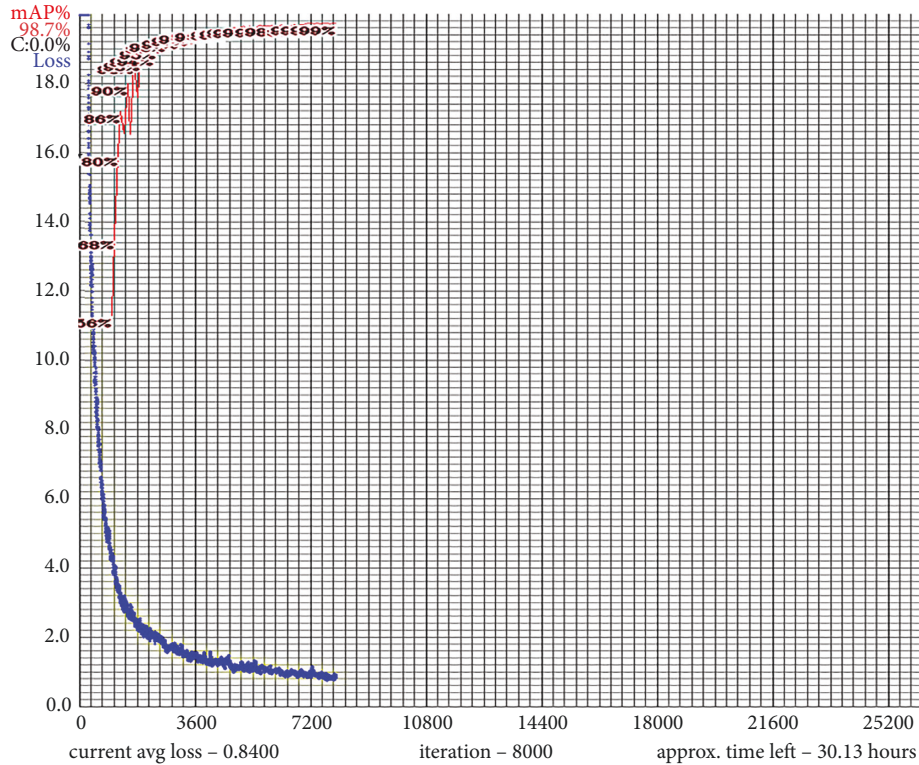


FIGURE 11: Training progress plot for categories of coffee, milk modifier, and PTW.

TABLE 4: Comparison of average precision achieved by three different detectors.

Main categories	YOLO V4	YOLO V5	YOLOR
	AP		
Juices	0.97	0.81	0.77
Chiller	0.97	0.82	0.79
Dairy liquid	0.965	0.826	0.89
Dairy powder	0.966	0.866	0.87
Coffee	0.987	0.85	0.85
Milk modifier	0.982	0.85	0.84
PTW	0.984	0.88	0.87
Infant nutrition	0.95	0.84	0.80
BFC	0.92	0.88	0.73
Nutrition	0.92	0.82	0.79
Nestrade	0.95	0.81	0.76
Sachets	0.95	0.81	0.78

score of three different detectors is provided in Table 5 and graphically demonstrated in Figure 12. The details of the planogram compliance accuracy achieved by Hyb-SMPC for different categories of the SKUs are provided in Table 6.

To evaluate the significance of Hyb-SMPC with the conventional methods, the test cases based on the size of the products are made. The effectiveness of the Hyb-SMPC is demonstrated in Table 7, which provides the comparison of the proposed approach with the conventional methods of [5, 22]. The results indicate that the Hyb-SMPC outperformed the conventional methods.

4.4. Qualitative Evaluation. The study also presents the qualitative evaluation of the proposed approach; for this

purpose, the user's feedback is collected and analyzed. The users are divided into two groups; both groups provided their feedback by completing a survey which is incorporated in the annexure (included as separate file). We report on group 1 (retailer's group) as it is the most significant. The findings from both groups are presented below.

4.4.1. Retailer's Feedback. This group is comprised of professional retailers working in the domain of retailing, and the following are the summarized results:

- (1) All the members were pleased to see new automated system.
- (2) All the members felt content using the new system, checking reporting mechanism, and reviewing it.

TABLE 5: Comparison of *mAP*, *recall*, and *F1-score* of three different detectors.

Metrics	YOLO V4	YOLO V5	YOLOR
mAP	0.96	0.833	0.82
F1-score	0.95	0.822	0.801
Recall	0.898	0.827	0.81

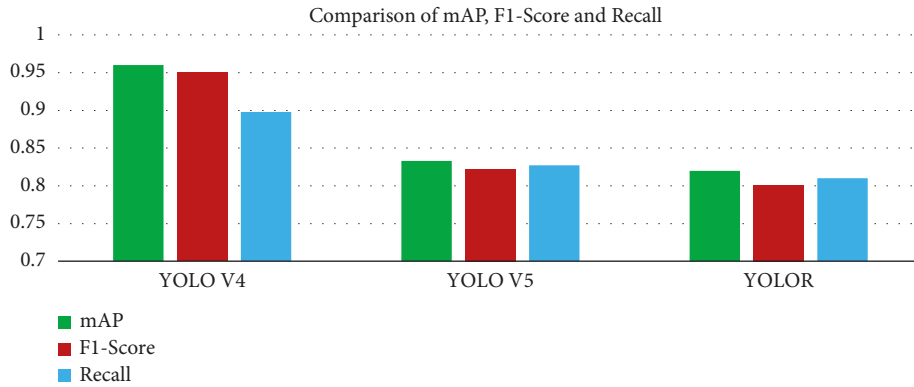


FIGURE 12: Success rate of three detectors.

TABLE 6: Compliance accuracy of Hyb-SMPC for different categories.

Categories	Compliance accuracy (%)
Juices	99.8
Chiller	99.6
Dairy liquid	98.1
Dairy powder	98.3
Coffee	99.7
Milk modifier	99.6
Powder tea whitener	99.65
Infant nutrition	97.5
BFC	96.5
Nutrition	96
Nestrade	97.43
Sachets	97.2

TABLE 7: Comparison of *compliance accuracy* of Hyb-SMPC with conventional method.

Product size	Recurring patterns (%)	Hyb-SMPC (%)
Small	95.32	99
Medium	90.61	97
Large	85.24	98

- (3) Almost all members were enthusiastic about incorporating the new system to enhance their work efficiency.

Regarding the survey, statements S1 to S7 express positive statements for the approach. The responses obtained regarding all these statements were 4 or 5 (4: agree, 5: strongly agree) except for S7 which mostly got the response of 4 (agree). To summarize, S1 obtained 100% response, S2 got 98%, S3 achieved 96%, S4 and S5 got 95%, and S6 achieved 97%, whereas S7 got 96%. Figure 13(a) represents level of user satisfaction

regarding the aspects of system. A Wilcoxon signed-rank test was performed to determine whether there was a difference in the retailer’s satisfaction level by using our approach compared to the previous manual technique. There was a statistically significant difference between the groups at 0.05 level; the *p* value equals 0.0156250; the test statistic *Z* equals -2.417559 , which is not in the 95% region of acceptance: $[-1.9600: 1.9600]$. $W = 0.0$, is not in the 95% region of acceptance: $[3.0000: 24.0000]$. The observed standardized effect size, Z/\sqrt{n} , is large (0.86). That indicates that the retailers are quite satisfied with our approach.

Statements S8 to S14 represent negative statements regarding the presented approach. All the participants gave the score of 1 or 2 (1: strongly disagree, 2: disagree) except for S11 where 10.5% responded with “agree,” thus collectively demonstrating higher level of user satisfaction represented in Figure 13(b).

4.4.2. Customer’s Feedback. This group consists of customers (both males and females) visiting the retail stores. The results obtained from the questionnaire provided to them indicate that the participants gave a score of 4 or 5 to the positive statements. S1 and S3 obtained 100% response, whereas S2 obtained 89%. The participants gave the score of 1 or 2 to the negative statements; S4 to S6 indicate negative statements. Only 2% of participants were undecided about S4. Figure 14 shows the feedback of customers.

The results indicated that properly organized products increase the satisfaction level of the customers and let the customers visit the stores more often, which contributes to increasing the sales of the stores. Hence, the proposed approach can enhance the sales of stores to a significant level.

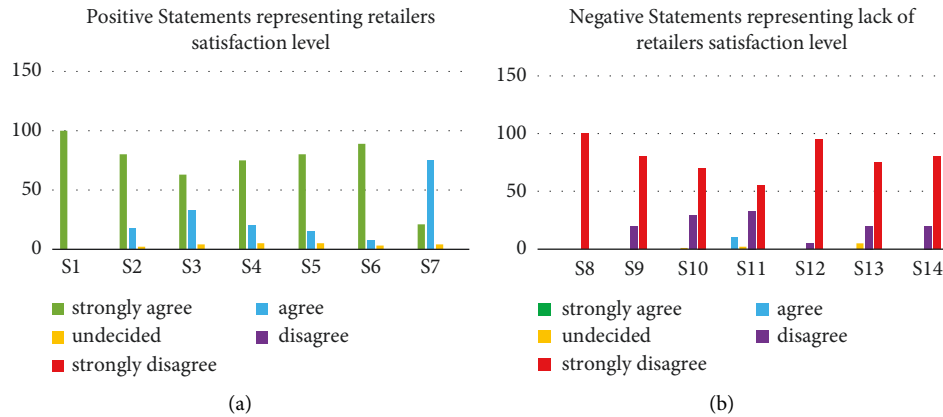


FIGURE 13: Retailer satisfaction level. (a) Positive statements. (b) Negative statements.

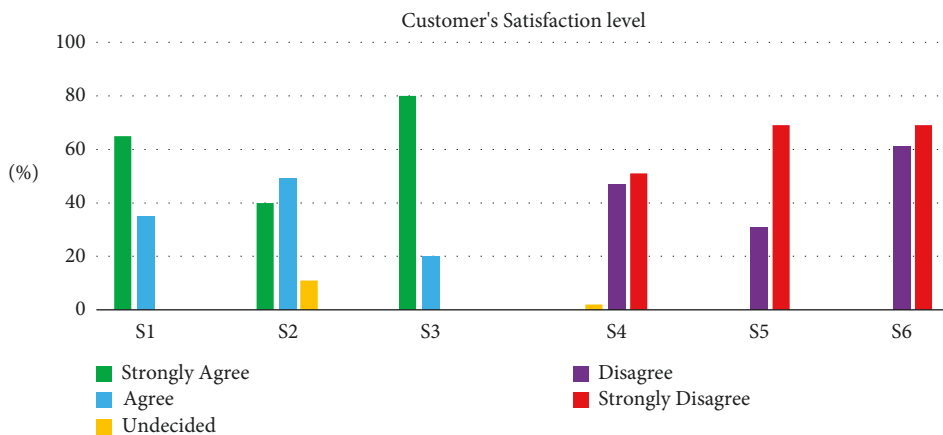


FIGURE 14: Customer's satisfaction level.

5. Conclusion

Effectively monitoring retail shelves and satisfying planogram are the two main factors that can boost sales of retail sector. The earlier studies conducted in this domain were either performing shelf monitoring or checking planogram compliance. As both activities are important, the proposed study presented a hybrid approach that deals with both activities. The study presented an approach to detect fine-grained retail products using deep learning and also verify the compliance of planogram. For the detection part, three one-stage detectors, YOLO V4, YOLO V5, and YOLOR, were trained on the dataset consisting of 30,000 retail images having 106 different SKUs belonging to 12 main categories. The use of one-stage detector makes the detection part fast. The best trained model performed efficiently on real retail environment and achieved accuracy up to 99%. The proposed method also checked planogram compliance by matching the provided planogram with the postprocessed images of retail and generated report indicating that the planogram is followed fully or partially or is not followed at all. There can be several extensions of the presented work. Some of the future considerations are described as follows:

- (i) Augmenting Internet of things (IoT) to automate the manual process of capturing images by the personnel, instead the cameras mounted at different locations of store will capture the images and upload them on the servers for further processing.
- (ii) Using strong quality assessment techniques to monitor the quality of captured images. In case of blurry, noisy, and distorted images, the system must not accept such images and ask the image-capturing entities to capture the images again. This technique will help improve accuracy.
- (iii) Another extension of this work is to formulate a way to work with unlabeled data as manually labeling the SKUs in the images is a time-consuming and laborious task.

Data Availability

Data will be provided if required.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Supplementary Materials

The authors have supplied the description of dataset as supplementary materials. The dataset used is provided by the industry partner which contains products of 12 different categories (e.g., boxes, bottles, poach, chiller, etc.) and subcategories of 106 different fine-grained SKUs. The dataset contains 30,000 images manually collected from real diverse environment, that is, retail, departmental stores, marts and supermarkets, and shops with natural lightings through different mobile cameras. The average resolution of the images was 1024×1024 with jpeg format. Images were captured from the distance of 1.5 to 5 feet from the front of the shelves. (*Supplementary Materials*)

References

- [1] B. Knezevic, S. Renko, N. Knego, and N. Knego, "Changes in retail industry in the Eu," *Business, Management and Education*, European Online Library, vol. 9, no. 1, pp. 34–49, 2011.
- [2] Y. Wei, S. Tran, S. Xu, B. Kang, and M. Springer, "Deep learning for retail product recognition: challenges and techniques," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8875910, 23 pages, 2020.
- [3] A. Tonioni and L. Di Stefano, "Product recognition in store shelves as a sub-graph isomorphism problem," in *Proceedings of the International Conference on Image Analysis and Processing*, pp. 682–693, LNCS, Catania, Italy, September 2017.
- [4] M. Marder, S. Harary, A. Ribak, Y. Tzur, S. Alpert, and A. Tzadok, "Using image analytics to monitor retail store shelves," *IBM Journal of Research and Development*, vol. 59, no. 2/3, pp. 1–3, 2015.
- [5] S. Liu and H. Tian, "Planogram compliance checking using recurring patterns," in *Proceedings of the 2015 IEEE International Symposium on Multimedia (ISM)*, Miami, FL, USA, December 2015.
- [6] T. Elbers, *The Effects of In-Store Layout- and Shelf Designs on Consumer Behaviour*, 2016.
- [7] D. Corsten and T. Gruen, "Desperately seeking shelf availability: an examination of the extent, the causes, and the efforts to address retail out-of-stocks," *International Journal of Retail & Distribution Management*, vol. 31, no. 12, pp. 605–617, 2003.
- [8] X. Zhang and G. Wang, "Stud pose detection based on photometric stereo and lightweight YOLOv4," *Journal of Artificial Intelligence and Technology*, vol. 2, no. 1, pp. 32–37, 2021.
- [9] A. Shabbir, A. Rasheed, A. Rasheed et al., "Detection of glaucoma using retinal fundus images: a comprehensive review," *Mathematical Biosciences and Engineering*, vol. 18, no. 3, pp. 2033–2076, 2021.
- [10] S. Karimi Jafarbigloo and H. Danyali, "Nuclear atypia grading in breast cancer histopathological images based on CNN feature extraction and LSTM classification," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 4, pp. 426–439, 2021.
- [11] M. Merler, C. Galleguillos, and S. Belongie, "Recognizing groceries in situ using in vitro training data," in *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, June 2007.
- [12] B. Santra and D. P. Mukherjee, "A comprehensive survey on computer vision based approaches for automatic identification of products in retail store," *Image and Vision Computing*, vol. 86, pp. 45–63, 2019.
- [13] K. Mikolajczyk and K. Mikolajczyk, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.
- [14] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: speeded up robust features," in *Proceedings of the Computer Vision - ECCV 2006*, pp. 404–417, Graz, Austria, May 2006.
- [15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [16] R. Moorthy, S. Behera, S. Verma, S. Bhargave, and P. Ramanathan, "Applying Image Processing for Detecting On-Shelf," in *Proceedings of the Third International Symposium on Women in Computing and Informatics*, pp. 451–457, Kochi, India, August 2015.
- [17] T. Mittal, B. Laasya, and J. Dinesh Babu, "A logo-based approach for recognising multiple products on a shelf," in *Proceedings of the SAI Intelligent Systems Conference (IntelISys) 2016*, vol. 16, pp. 15–22, London, UK, September 2018.
- [18] G. Varol and R. S. Kuzu, "Toward retail product recognition on grocery shelves," in *Proceedings of the Sixth International Conference on Graphic and Image Processing (ICGIP 2014)*, vol. 9443, pp. 1–7, Beijing, China, October 2014.
- [19] A. Saran, E. Hassan, and A. K. Maurya, "Robust visual analysis for planogram compliance problem," in *Proceedings of the 2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pp. 576–579, Tokyo, Japan, May 2015.
- [20] E. Frontoni, M. Contigiani, G. Ribighini, and I. Dii, "A Heuristic Approach to Evaluate Occurrences of Products for the Planogram Maintenance," in *Proceedings of the 2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, Senigallia, Italy, September 2014.
- [21] W. Geng, F. Han, J. Lin et al., "Fine-grained grocery product recognition by one-shot learning," in *Proceedings of the 26th ACM international conference on Multimedia*, vol. 2, pp. 1706–1714, Seoul, Republic of Korea, October 2018.
- [22] S. Liu, W. Li, S. Davis, C. Ritz, and H. Tian, "Planogram Compliance Checking Based on Detection of Recurring Patterns," *Computer Vision and Pattern Recognition*, vol. 3, pp. 1–8, 2016.
- [23] N. O. Mahony, S. Campbell, A. Carvalho et al., "Deep Learning vs. Traditional Computer Vision," *Cv*, 2019.
- [24] M. A. Aslam, M. N. Salik, F. Chughtai, N. Ali, S. H. Dar, and T. Khalil, "Image classification based on mid-level feature fusion," in *Proceedings of the 2019 15th International Conference on Emerging Technologies (ICET)*, Peshawar, Pakistan, December 2019.
- [25] "Detection and prediction of traffic accidents using deep learning techniques," *Angewandte Chemie International Edition*, vol. 6, no. 11, pp. 951–952, 2022.
- [26] S. Fatima, N. Aiman Aslam, I. Tariq, and N. Ali, "Home security and automation based on internet of things: a comprehensive review," in *Proceedings of the IOP Conference Series: Materials Science and Engineering*, vol. 899, no. 1, Article ID 12011, Chennai, India, September 2020.
- [27] Q. Zou, K. Xiong, Q. Fang, and B. Jiang, "Deep imitation reinforcement learning for self-driving by vision," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 4, pp. 493–503, 2021.
- [28] N. A. Othman and I. Aydin, "A new IoT combined body detection of people by using computer vision for security application," in *Proceedings of the 2017 9th International*

- Conference on Computational Intelligence and Communication Networks (CICN)*, vol. 2018, pp. 108–112, Girne, Northern Cyprus, September 2017.
- [29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2016, pp. 779–788, Las Vegas, NV, USA, June 2016.
- [30] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: scalable and efficient object detection,” in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Article ID 10778, Seattle, WA, USA, June 2020.
- [31] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [32] P. Jund, N. Abdo, A. Eitel, and W. Burgard, *The Freiburg Groceries Dataset*, <http://arxiv.org/abs/1611.05799>, 2016.
- [33] E. Goldman and J. Goldberger, “CRF with deep class embedding for large scale classification,” *Computer Vision and Image Understanding*, vol. 191, pp. 1–11, 2019.
- [34] K. Higa and K. Iwamoto, “Robust estimation of product amount on store shelves from a surveillance camera for improving on-shelf availability,” in *Proceedings of the 2018 IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 1–6, Krakow, Poland, October 2018.
- [35] K. Higa and K. Iwamoto, “Robust shelf monitoring using supervised learning for improving on-shelf availability in retail stores,” *Sensors*, vol. 19, no. 12, pp. 2722–12, 2019.
- [36] L. Karlinsky, J. Shtok, Y. Tzur, and A. Tzadok, “Fine-grained recognition of thousands of object categories with single-example training,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 965–974, Honolulu, HI, USA, July 2017.
- [37] H. Sun, J. Zhang, and T. Akashi, “TemplateFree: product detection on retail store shelves,” *IEEE Transactions on Electrical and Electronic Engineering*, vol. 15, no. 2, pp. 242–251, 2020.
- [38] T. Chong, I. Bustan, and M. Wee, “Deep learning approach to planogram compliance in retail stores,” *Semant. Sch.*, pp. 1–6, 2016.
- [39] R. Girshick, J. Donahue, T. Darrell, J. Malik, U. C. Berkeley, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, p. 5000, Columbus, OH, USA, June 2014.
- [40] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020.
- [41] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.
- [42] Y. Konishi, Y. Hanzawa, M. Kawade, and M. Hashimoto, “Fast 6D pose estimation from a monocular image using hierarchical pose trees,” in *Proceedings of the Computer Vision - ECCV 2016*, vol. 1, pp. 398–413, Amsterdam, The Netherlands, October 2016.
- [43] R. Yilmazer and D. Birant, “Shelf auditing based on image classification using semi-supervised deep learning to increase on-shelf availability in grocery stores,” *Sensors*, vol. 21, no. 2, pp. 327–426, 2021.
- [44] P. Wajire and E. Pune, *Image classification for retail*, 2020.
- [45] A. Mittal, A. K. Moorthy, and A. C. Bovik, “No-reference image quality assessment in the spatial domain,” *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.
- [46] H.-W. Zhang, L.-J. Zhang, P.-F. Li, and D. Gu, “Yarn-dyed fabric defect detection with YOLOV2 based on deep convolution neural networks,” in *Proceedings of the 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, vol. 17, pp. 170–174, Enshi, China, May 2018.
- [47] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” 2020, <http://arxiv.org/abs/2004.10934>.
- [48] J. Solawetz, “YOLOv5 New Version - Improvements And Evaluation,” 2020, <https://blog.roboflow.com/yolov5-improvements-and-evaluation/>.
- [49] C. Wang, I. Yeh, and H. M. Liao, “You Only Learn One Representation: Unified Network for Multiple Tasks,” pp. 1–11, 2021, <https://arxiv.org/abs/2105.04206>.
- [50] Codebrainz, “Color-names,” 2021, <https://github.com/codebrainz/color-names/blob/master/output/colors.csv>.
- [51] H. Y. Ha, “Integrating Deep Learning with Correlation-Based Multimedia Semantic Concept Detection,” 2015.

Research Article

COMSATS Face: A Dataset of Face Images with Pose Variations, Its Design, and Aspects

Mahmood Ul Haq,¹ Muhammad Athar Javed Sethi ,¹ Rehmat Ullah ,¹ Aamir Shazhad,² Laiq Hasan,¹ and Ghulam Mohammad Karami ³

¹Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Peshawar, Pakistan

²Department of Electrical and Computer Engineering, COMSATS University Islamabad, Abbottabad Campus, Abbottabad, Pakistan

³SMEC International Pvt Limited, Kabul 1007, Afghanistan

Correspondence should be addressed to Ghulam Mohammad Karami; ghulam.karami@smec.com

Received 1 March 2022; Accepted 15 April 2022; Published 23 May 2022

Academic Editor: Nouman Ali

Copyright © 2022 Mahmood Ul Haq et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the three-dimensional formation and flexibility, a human face may appear different in numerous events. Researchers are developing robust and efficient algorithms for face detection, face recognition, and face expression analysis, causing several difficulties due to face poses, illumination, face expression, head orientation, occlusion, hairstyle, etc. To determine the effectiveness of the algorithms, it needs to be tested using a specific benchmark of face images/databases. Face pose is an important factor that severely reduces the recognition ability. In this paper, two contributions are made: (i) a dataset of face images with multiple poses is introduced. The dataset includes 850 images of 50 individuals under 17 different poses (0°, 5°, 10°, 15°, 20°, 25°, 30°, 35°, 55°, -5°, -10°, -15°, -20°, -25°, -30°, -35°, -55°). These images were captured closed to real-world conditions in the time span of five months in COMSATS University, Abbottabad Campus. Face images included in this dataset can reveal the efficiency and robustness of future face detection and face recognition algorithms. (ii) A comparative analysis of three face recognition algorithms such as PAL, PCA, and LDA is presented based on the proposed face database.

1. Introduction

Face recognition (FR) is an essential biometric technique that compares the face features of two different images to determine the similarity between these images. FR is a rapidly growing and the most popular research area. Face appearance depends on various factors such as illumination variations, face pose variations, and occlusion [1]. To examine the robustness of the face recognition algorithm to these multiple aspects, a database of significant size and diversity is required. Two main classes of face recognition are (a) face detection and (b) face verification [2]. Face recognition is legally and commercially used in applications, with high collectability and acceptability [3]. By using new imaging sensors, a new range of possibilities is open to boost the performance of face recognition systems [4].

Much has been published about FR in previous literature [5–8]. These conventional methods are successful, but their robustness is being challenged by different factors such as bad lighting conditions and the low resolution of face images [8]. Currently, there are existing techniques that show considerable accuracy if the images face being processed is of sufficient resolution. Below we briefly illustrate the latest face recognition developments.

Researchers in [9] developed a Laplacian face approach (LFA). In this approach, through the process of optimal linear approximation of eigenfaces, Laplacian faces are obtained. Based on the published results, the LFA based face recognition approach attains much lower error rates. Simonyan et al. [10] use Fisher vectors on densely sampled Scaled Invariant Feature Transform (SIFT) features to classify faces. The overall accuracy of the proposed algorithm

for the standard Labelled Faces in the Wild (LFW) dataset is 87.7%. Wavelets based facial recognition systems have been proposed in [11]. The wavelets transform provides insensitivity towards illumination changes and pose variations. The proposed algorithm shows acceptable results on FERET facial database. The overall false rejection ratio is 0.12%. However, the author has not provided information about the occlusion in the paper. In [12], the authors used PCA based face recognition to recognize criminal faces through CCTV cameras installed in public and private areas. Their algorithm achieved an accuracy of 80%. But the proposed algorithm was not tested by real-time criminal face images.

In [13], authors have proposed MFDD and RMFRD datasets for training and testing deep learning-based face recognition algorithms. MFDD dataset was designed for the detection of face wearing while RMFRD dataset was collected for validation or fine-tuning dataset in training and testing datasets in real situations. Authors in [14] presented a new face recognition algorithm MagFace to learn unified features. The proposed algorithm achieved 95.97% accuracy as compared to other algorithms. Recently researchers in [15–20] have presented different features extraction schemes that can be investigated for developing a robust FR algorithm. The main contribution of this paper is as follows:

- (i) This paper briefs the creation of the COMSATS face database developed in COMSATS University Abbottabad Campus. This database contains 850 face images of 50 individuals with seventeen different poses.
- (ii) The image acquisition process has been repeated with a very comparable setup in the two labs. For each subject, three separate sessions were performed with time of four months.
- (iii) The proposed dataset has been tested on three baseline algorithms such as PCA, PAL, and LDA by changing the face poses and image resolution.
- (iv) We cover a large range of face pose and image resolution in simulation to test the three baseline face recognition algorithms. We investigated the images of having resolution of 144×256 , 140×140 , 70×70 , 40×40 , 20×20 , 10×10 , and 5×5 pixels.
- (v) The comparison has been presented under a very challenging situation, when there is only one test image.

The rest of the paper is organized as follows. Section 2 provides a general idea of available face databases. Section 3 describes the acquisition setup of database. Section 4 reports the three baseline face recognition algorithms tested on the COMSATS face database. Finally, Section 5 lists the results of FR algorithms and Section 6 presents some final remarks. For each section, Table 1 shows the nomenclature used in this paper.

2. Available Face Databases

A huge number of databases are available in the face recognition community, and the face recognition algorithms perform differently on different datasets. Researchers' teams gathered these databases, which varied in scope, purpose, and size. Here, we briefly review the key features of these

TABLE 1: Nomenclature.

Notation	Description
FR	Face recognition
AdaBoost	Adaptive boosting
PCA	Principal component analysis
LDA	Linear discriminant analysis
PAL	Principal component analysis with adaptive boosting of linear discriminant analysis
I cropped	Cropped face image
LBP	Local binary pattern
I_m^s	Mean image of S subject

available face recognition databases such as number of subjects and images, condition, image resolution, and type. But due to the inaccessibility of information, these databases are not discussed with the same level of detail. AT&T database contains 400 images with 40 distinct subjects collected by Cambridge University. Each subject has 40 different images. Images were taken with different facial expressions (closed/open eyes, smiling/not smiling), varying lighting conditions, and facial details (with glasses, without glasses).

Face recognition data contain 395 subjects including males and females having 20 images per subject. Most subjects of this database are 18–20 years old, with some older subjects. Some subjects have beards and glasses. The images format of this database is a colour JPEG image of 24-bit.

Facial recognition technology (FERET) [21] was started in 1993. A total of 14051 face images of 1209 people have been included in this database covering a large range of variations in facial expressions, illuminations, viewpoints, and acquisition time. AR database [22] was collected by Alex Martinez and Robert Benavente. This database contains 4,000 colour face images of 126 subjects including 70 men and 56 women. The dataset included images with frontal view, illumination, facial expressions, and occlusions like glasses and scarves. JAFFE database is also called the Japanese female face database containing 213 images of 10 Japanese models with seven facial expressions (neutral and basic facial expression). Indian face database [23] was collected in IIT Kanpur Campus during February 2002 in JPEG format. This dataset contains 40 subjects including males and females with eleven images of each subject. The size of the images is 640×480 pixels, an 8-bit image. These images contain faces looking upwards left, looking down, and looking upwards right. Available expressions of this dataset are neutral, smiling, laughter, and sadness. Georgia Tech Face database was collected by the Georgia Institute of Technology in the time span of five months. This database contains 50 subjects and 15 colour images per subject in JPEG format with different scales and locations. Various images were captured in two sessions to consider the variations in expression, illumination conditions, and appearance. PUT face database was created by CIE biometrics containing 10000 images of 100 subjects. These images were captured in a controlled environment. This database includes additional data such as rectangles containing eyes, mouth, nose, landmarks positions, and face and is accessible for research work. CMU PIE database [24] contains 68

TABLE 2: Available datasets along with their features.

Database	Images	Features
AT&T 29 LFW	400 images with 40 people	Light and expression variations with glasses
AR face 21	4000 images with 126 people	Expression, occlusion, illumination, and frontal pose
Face recognition data	395 subjects and 20 images/ subject	18–20 years old with some older subjects. Some subjects have beards and glasses
FERET 23	Containing 14126 images with 199 subjects	Pose, expression and time variations, colour images
Yale face 15	15 subjects with 165 images	Eyeglasses, expressions, and lightening
Yale face B 10	10 subjects with 5760 images	Illumination and pose variation
The Extended M2VTS database, University of Surrey, UK	Four recordings of 295 subjects	Speaking headshot, rotating headshot, high-quality colour images
JAFFE database	213 images of 10 female models	Seven facial expressions
PIE database, CMU 68	41368 images with 68 subjects	Illumination expression and pose
CMU Multi-PIE	750000 facial images of 337 subjects	Nineteen poses and different viewpoints
LFW database	More than 13000 images with 1680 subjects	Pose, illumination, expression background variation, and occlusion

subjects with 41,368 images having 43 illumination conditions and 13 poses, with four different expressions. This database is also called the CMU Pose, Illumination, and Expression database. This database was collected from November 2000 up to December 2000. CMU Multi-PIE [25] database includes a vast collection of images captured with different pose angles. CMU Multi-PIE database was collected in five months having more than 750000 facial images of 337 subjects taken at several viewpoints displaying a range of expressions and poses.

LFW [26] dataset has images with different poses, expressions, illumination variations, and occlusion. LFW database contains more than 13000 images with 1680 subjects. Yale face database [27] contains 15 subjects with 165 images. This database includes 11 images per person with different facial expressions, lighting conditions and occlusion such as glasses. Yale face B database contains 5760 images of 10 persons, 576 with 9 poses, and 64 illumination conditions per subject. The Basel Face model is collected by the University of Basel and is available on their website. The Morphable model has registered 100 male and 100 female 3D scans faces. The chokePoint dataset is a video dataset of 48 videos including 64,204 face images. This dataset includes person reidentification, image set matching, clustering, 3D face reconstruction, face tracking, background subtraction, and estimation. In Table 2, we present a review of face recognition dataset which can help the development and validation of new FR algorithms.

3. COMSATS Face Dataset

3.1. Equipment. Different instruments are used to collect dataset like total station (Trimble M3 DR5), theodolite (DT-5), staff rod, stand, permanent marker, background sheet, and a digital camera. The angles were measured using theodolite and total station. The staff rod of 5 meters is used to find the elevation of angles. Theodolite is used to measure the angles of the vertical axis and horizontal axis. Theodolite and the total station were used because of their fine accuracy. The images were captured by a professional photographer

with cannon EOS6D in the lightening of fluorescent lamps as shown in Figure 1(a). The optic was a Canon 85 mm, f1.8 with an aperture of f5.6 and shutter speed of 1/60th.

3.2. Observation. An organized indoor atmosphere was set up with fluorescent lamps and natural light. The participant was asked to sit at the predefined point in front of the camera at 0.5 to 0.8 meters and follow the predefined structure as shown in Figure 1(b). A white sheet was placed behind the background to produce uniformity. The camera operator observed the participant face angle for the desirable results before taking the images.

3.3. Image Acquisition. Fifty volunteers participated in the collection of the dataset, and all belonged to the same gender (Male) with different ages, weight, colour, and cast. Their ages limits were from 18 to 35 years. Most of them were students of COMSATS University Islamabad Abbottabad (Campus) with few alumni. The database collection work was performed in the survey lab of civil engineering department of COMSATS University Islamabad Abbottabad (Campus). The dataset was completed in the duration of five months. These images were captured in two separate sessions at a lab explicitly prepared for purpose of the dataset. Samples were sited in front of a white sheet. Two of the image processing experts were selected to provide the mental state term and to set the face of an actor according to the corresponding face angle. To prepare himself for the interpretation of the related face angle, the participator was given time as needed. When the participator provided a thumb gesture to the photographer, the picture was taken in the desired view angle. Importantly, for a guarantee and natural interpretation of a given face angle, the participators were restricted not to tilt the head. The participant then immediately turned to the next face angle as advised by the instructors from the camera, and a second picture was taken. The camera operator collected the dataset images at the end of the experiment. This database consists of 850 images of 50 subjects under 17 different poses (0°, 5°, 10°, 15°, 20°, 25°, 30°,

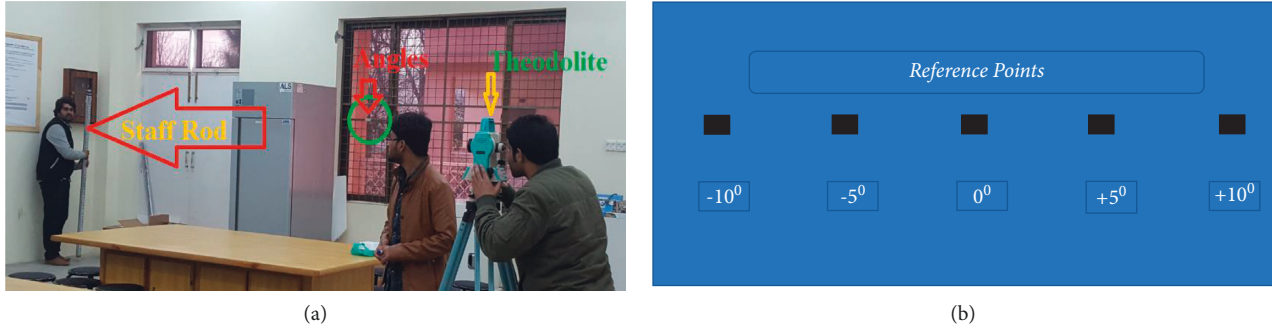


FIGURE 1: (a) Angle measurement process and predefined structure, (b) measured angles.

TABLE 3: Dataset specifications table.

Subject area	Electronics and computer engineering
Specific subject area	Computer vision, image processing, face recognition, electronics
Type of data	Images
How dataset was acquired	The dataset was acquired by using the following instruments 1- Camera (Cannon 20.6 mega pixel) 2- Theodolite (DT-5) 3- Total station (Trimble M3 DR5) 4- Fluorescent lamp (for lightening)
Data format	JPG
Experimental factors	Images with 17 different poses (-55 to $+55$) of a total of 50 individuals were captured with the help of a photographer, lab assistant, and lab engineer
Experimental features	The database consists of 850 images of fifty subjects with seventeen different poses ($0^0, 5^0, 10^0, 15^0, 20^0, 25^0, 30^0, 35^0, 55^0, -5^0, -10^0, -15^0, -20^0, -25^0, -30^0, -35^0, -55^0$)
Data source location	COMSATS University Islamabad, Abbottabad (Campus), Pakistan
Data availability	Data is available on http://cuiatd.edu.pk/COMSATSFacePoseVarProj.html

$35^0, 55^0, -5^0, -10^0, -15^0, -20^0, -25^0, -30^0, -35^0, -55^0$) with each subject having different age, weight, height, and facial colour.

A consent form has been signed by every individual, which ensures that their face images will be used for research purposes. Specifications of the dataset are presented in Table 3.

3.4. Image Specification. The database contains 850 jpg image files with a resolution of 2988×5312 pixels (colour images) with the built-in flash of the camera. Each image was then preprocessed, and their resolution has been changed to 144×256 . The size of each preprocessed image is less than 1 MB. Properties of images, i.e., dimensions and pixels before and after preprocessing, are presented in Figure 2. In the database preprocessing step, all the images of each individual were renamed by their face angles. These images were resized by MATLAB using nearest neighbor interpolation algorithm and the dimensions of images were changed to get relevant results. These images were cropped manually to get the specific (important) portion of an image. Raw images can be obtained upon request from the authors. Researchers can use these images for face detection, face recognition, age estimation, facial expression recognition, and face pose recognition.

3.5. Dataset Structure. The database consists of 850 images of fifty subjects under seventeen different poses ($0^0, 5^0, 10^0, 15^0, 20^0, 25^0, 30^0, 35^0, 55^0, -5^0, -10^0, -15^0, -20^0, -25^0, -30^0, -35^0, -55^0$). The images of individuals are presented in Figure 3. These images were captured close to real-world conditions for a duration of five months. Figure 4 shows 17 different poses of each individual. Face images involved in this dataset can reveal the effectiveness and robustness of different face detection and recognition algorithms. These images were cropped in preprocessing step to get the specific (face) portion of an image. However, for research purposes, raw images can be obtained upon request from the authors.

3.6. Data Records

- This dataset will be used for the evaluation of the performance of different algorithms proposed for security and attendance purposes.
- This data will be a source for different algorithms like LDA [28], Local Binary Pattern [29], eigenfaces [30], and Deep Learning and will be a challenge for recently published face recognition algorithms [31–33].
- It includes the poses in the range of -55 to $+55$ of all the subjects. These poses are $0^0, 5^0, 10^0, 15^0, 20^0, 25^0, 30^0, 35^0, 55^0, -5^0, -10^0, -15^0, -20^0, -25^0, -30^0, -35^0, -55^0$. This dataset includes fifty subjects having different ages of people; the age range is from 18 to 25 years.

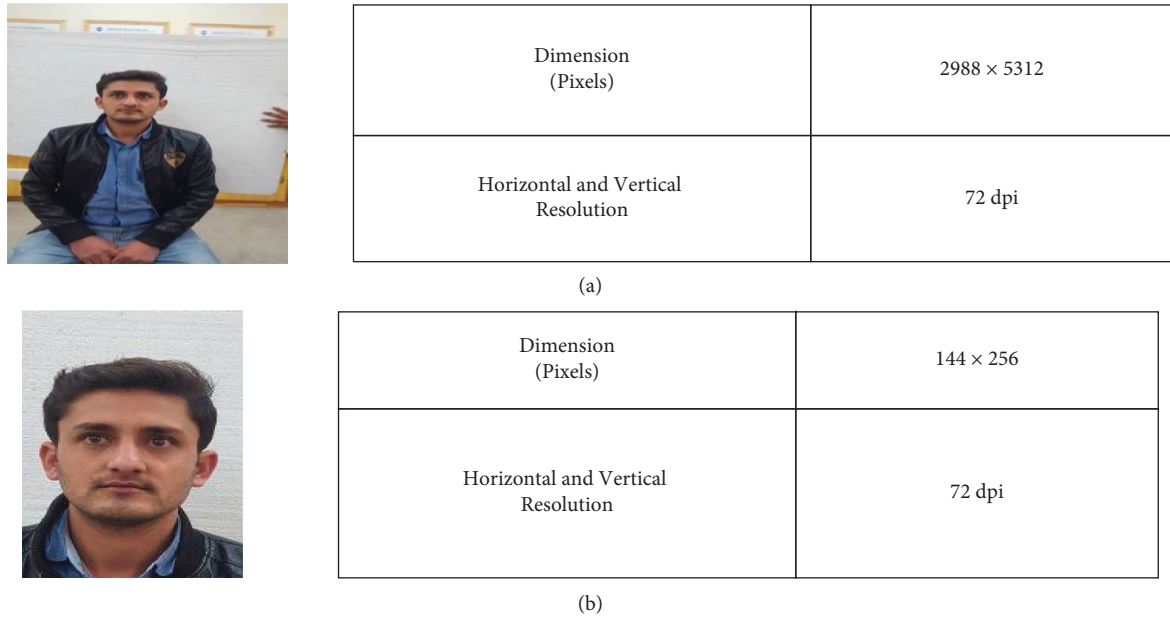


FIGURE 2: Properties of a single image: (a) original image, (b) image after preprocessing.



FIGURE 3: Example images of subjects.

4. Face Recognition Algorithms

This section describes the comparison of face recognition algorithms based on the abovementioned database. The study was performed using different sizes of images. Images

of fifty subjects (three images per person) were chosen for training (gallery), whereas these algorithms were tested on seventeen images per subject of different sizes such as 144×256 , 140×140 , 70×70 , 40×40 , 20×20 , 10×10 , and 5×5 pixels.



FIGURE 4: Seventeen different poses of individual subject.

4.1. PCA Based (Eigenfaces) Face Recognition Algorithm. Principal Component Analysis is a statistical procedure in which transformation is used as set of observed possible correlated variables into linearly noncorrelated variables which are called the principal of components. In the face recognition system, PCA plays a vital role as it is a very efficient method for face recognition. As in PCA all images of the training set are represented as a combination of weighted eigenfaces and calculate covariance matrices. By the covariance matrix of a training set of images, eigenvectors are obtained. Weights of eigenvectors are found by the set of eigenfaces that are most relevant. Recognition of faces is done by projecting a test image on the subspace of eigenfaces. The distance between the test image and training images is calculated using

$$d_{b_i b_j} = \sum_{a=1}^n (x_{k_i} - x_{k_j})^2, \quad (1)$$

where b_i and b_j represent two matrices for training and test samples, respectively, and $(x_{k_i} - x_{k_j})^2$ is the Euclidean distance (ED) between two image components X_{k_i} and X_{k_j} . Test image must have minimum Euclidean distance with a recognized image that exists in the training images. There are three possible scenarios in PCA based face recognition algorithm when the test image is tested with the face database as described below.

Scenarios:

- (i) If the test face image is far away from face space, it is not a face image.
- (ii) If the test face image is near face space and far away from face class, then the image is not recognized by the algorithm.
- (iii) If the test image is close to both face class and face space, then the face image is correctly recognized in the face database. For implementation and detail of the PCA based FR algorithm, readers are referred to [34].

4.2. Linear Discriminant Analysis (LDA) and Fisher's Face. The LDA is proposed as an enhancement to Principal Component Analysis (PCA). LDA constructs a discriminant subspace that reduces the scatter between the same class images and maximizes the scatter between images of different classes. Let $c = [X_1, X_2, \dots, X_c]$ be the face classes in the database and let each face class X_i has face images x_j , where $j = 1, 2, \dots, k$. Within class, variance can be calculated using with in-class scatter matrix.

$$S_w = \sum_{i=1}^c \sum_{j=1}^k (x_k - \mu_i)(x_k - \mu_i)^T, \quad (2)$$

where, for all classes (c), x_k denotes the j^{th} sample, while μ_i represents mean of i^{th} class and can be calculated by

$$\mu_i = \frac{\sum_{j=1}^k x_j}{k}. \quad (3)$$

Similarly, the between-class scatter matrix (S_b) can be defined as

$$S_b = \sum_{i=1}^c N_i (\mu_i - \mu) (\mu_i - \mu)^T, \quad (4)$$

where μ represents the mean of all classes and can be calculated as

$$\mu = \frac{\sum_{i=1}^c \mu_i}{c}. \quad (5)$$

After computing S_b and S_w , find the product of S_w^{-1} and S_b and compute the eigenvectors of the product ($S_w^{-1}S_b$). To reduce the scatter matrix dimensionality, use the same approach as eigenfaces (PCA). The last step is to project each face image to face space

$$S_i = U_T(x_j - \mu). \quad (6)$$

For a detailed study, readers are referred to [35].

4.3. PAL Face Recognition Algorithm. In the PAL FR algorithm, initially 68 specific points on training and testing faces are detected after face detection using a machine learning algorithm. In next step, all these faces are cropped according to these 68 landmarks. The mean and standard deviation of each face image are calculated and updated according to the relation given in (7) to reduce the error due to lighting variations

$$I_n = \frac{(I_{\text{cropped}} - \bar{X}) \times \sigma_{def}}{\sigma_i + \bar{X}_{def}}, \quad (7)$$

where \bar{X} represents the mean and σ_i represents the standard deviation of each input image while \bar{X}_{def} and σ_{def} are predefined mean and standard deviation suggested for all input images to reduce light variations. In this technique mean image of each class is taken to reduce time complexity, memory requirements, and errors due to pose variations. Mean image can be calculated as

$$I_m^s = \frac{\sum_{j=1}^J I_{nj}^s}{J}, \quad (8)$$

where I_{nj}^s is the j th training image (normalized) of subject 's' and J represents a total number of training images of 's' subject.

Furthermore, these images are fed to AdaBoost combined with LDA for recognition. A scoring value of the test image with each class is attained using the final classifier and the maximum scoring value achieved with the class will be considered as recognized image with desired class. For detailed study, readers are referred to [36]. The pseudocode of the proposed algorithm is presented in Table 4.

The proposed face database is tested on three baseline techniques such as PAL, PCA, and LDA. Tables 5 and 6 show the overall accuracy of the above-mentioned algorithms on the proposed face database.

5. Simulation Results

The experiments were performed using a Super-Server 7047 GR machine having 92 GB of RAM with MATLAB

2019 as a simulation tool. To test the above-mentioned FR algorithms, numerous tests were carried out on the proposed database which has several face images with two different conditions, such as face poses and image resolutions.

For each algorithm, three frontal images (0° , 5° , -5°) were chosen for training as shown in Figure 5(a) and seventeen different test images with a variation in pose (0° , 5° , 10° , 15° , 20° , 25° , 30° , 35° , 55° , -5° , -10° , -15° , -20° , -25° , -30° , -35° , -55°) as shown in Figure 5(b)

5.1. Face Image Resolution Analysis. In this study, face images of 144×256 , 140×140 , 70×70 , 40×40 , 20×20 , 10×10 , and 5×5 pixels were reinvestigated. Table 5 details the results of the three FR algorithms. From Table 5, essential explanations are as follows.

- (i) For face images having resolution of 70×70 pixels and above, PAL yields the highest recognition rate of 86.66% followed by PCA having an accuracy of 78.4% and LDA having an accuracy of 66.2%
- (ii) For face resolution of 5×5 , the accuracy of PCA and LDA has been decreased to 32.7% and 26.2% while the recognition rate of the PAL algorithm is 71.3%
- (iii) PAL algorithm is most effective across all aforementioned ranges of image resolutions

5.2. Face Pose Analysis. Some features of an individual's face are occluded due to variations in the facial pose. A good FR algorithm should be robust to pose variations and should be able to recognize a face with different viewing angles. In this study seventeen face poses of 50 subjects are investigated.

As shown in Table 6, the PAL method, the PCA, and LDA based face recognition algorithms yield 100% recognition accuracy for frontal face images of 144×256 and 70×70 pixels.

- (i) The PAL method comprehensively outperforms other face recognition algorithms from frontal to $\pm 55^\circ$ of pose variation.
- (ii) We observed the LDA based face recognition algorithm is less effective under low resolution by achieving the maximum accuracy of 47% for frontal facial images. For $\pm 55^\circ$ of face pose, the LDA barely yields any recognition results.

5.3. Computational Complexity. Figure 6 presents the execution times of algorithms for different image resolution face images.

From Figure 6, important observations are as follows.

- (i) For each face image resolution category, PAL algorithm consumes over 9 seconds and is most computationally complex as compared to PCA and LDA.
- (ii) For image resolution of 40×40 pixels and below, the compared algorithms consume less than 4 seconds. The LDA is unable to recognize face image resolution of 10×10 pixels and below.

TABLE 4: Proposed PAL approach.

Input: A set of input images $A = \{a_{i=1}^j\}_{i=1}^I$ with $I = \{1, 2, \dots, I\}$ classes and J images of each class.

Do for $i = 1, \dots, I$

- (2) convert RGB images to grey,
- (3) estimate and crop face (I_{cropped}).
- (4) update mean and standard deviation of each image, $I_n = (I_{\text{cropped}} - \bar{X}) \times \sigma_{def}/\sigma_i + \bar{X}_{def}$
- (5) calculate mean image of each class, $tr_i = \sum_{j=1}^J a_j^i/J$.

Final training images of each class, $Tr = \{tr_1, \dots, tr_I\}$.

Initialize mislabelled distribution over m , $D_1(i) = 1/m = 1/N (-1)$

Do for $t = 1, \dots, T$:

- (1) if $t = 1$, choose i samples per class for the learner.
- (2) train LDA feature extractor.
- (3) build a g classifier h_t .
- (4) calculate pseudo loss, e_t
- (5) calculate $\beta_t = e_t/(1 - e_t)$
- (6) if $\beta_t = 0$, abort the loop
- (7) update the distribution

Final g classifier of training image, $hf(z) = \arg \max(\sum(\log 1/\beta_t)h_t(z, y))$.

Generate a matching score.

Output: Maximum matching score (M_{score}), $I_{\text{recog}} = \arg \max(M_{\text{score}})$.

TABLE 5: Recognition accuracy, precision, and recall for different image resolutions.

Image resolution (in pixel)	Algorithm	Recognition accuracy (%)	Precision	Recall
144 × 256	PCA algorithm	78.4	0.7844	0.8451
	LDA algorithm	62.2	0.6622	0.7462
	PAL algorithm	86.66	0.866	0.9069
140 × 140	PCA algorithm	78.4	0.7844	0.8451
	LDA algorithm	62.2	0.6622	0.7462
	PAL algorithm	86.66	0.866	0.9069
70 × 70	PCA algorithm	78.4	0.7844	0.8451
	LDA algorithm	62.2	0.6622	0.7462
	PAL algorithm	86.66	0.866	0.9069
40 × 40	PCA algorithm	73.22	0.7322	0.7881
	LDA algorithm	58.3	0.5833	0.6290
	PAL algorithm	86.66	0.8666	0.9112
20 × 20	PCA algorithm	53.46	0.5346	0.5721
	LDA algorithm	41.49	0.4149	0.4523
	PAL algorithm	81.43	0.8143	0.8491
10 × 10	PCA algorithm	45.05	0.4505	0.4732
	LDA algorithm	37.34	0.3734	0.3972
	PAL algorithm	77.75	0.7775	0.8133
5 × 5	PCA algorithm	32.7	0.3277	0.423
	LDA algorithm	26.2	0.2623	0.3477
	PAL algorithm	71.3	0.7133	0.7889

TABLE 6: Comparison of classification accuracy algorithms for pose variations.

Image resolution	FR algorithms	Recognition accuracy%								
		±55°	±35°	±30°	±25°	±20°	±15°	±10°	±5°	0°
144 × 256 pixel	PCA [34]	30	49	47	53	53	64	100	100	100
	LDA [35]	43	61	66	71	77	88	100	100	100
	PAL technique [36]	64	74	79	84	87	92	100	100	100
70 × 70 pixel	PCA [34]	30	49	47	53	53	64	100	100	100
	LDA [35]	43	61	66	71	77	88	100	100	100
	PAL technique [36]	64	74	79	84	87	92	100	100	100
5 × 5 pixel	PCA [34]	6	17	17	24	30	36	39	55	71
	LDA [35]	0	8	14	13	25	33	38	51	54
	PAL technique [36]	32	48	57	62	71	72	100	100	100

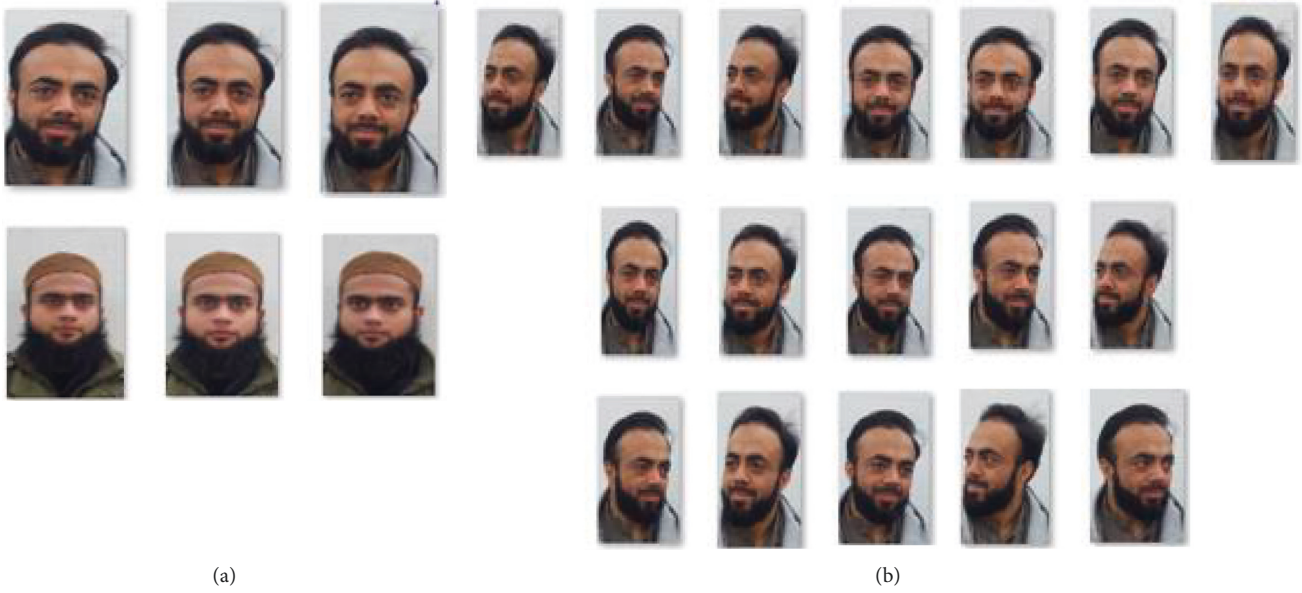


FIGURE 5: (a) Training images, (b) testing images.

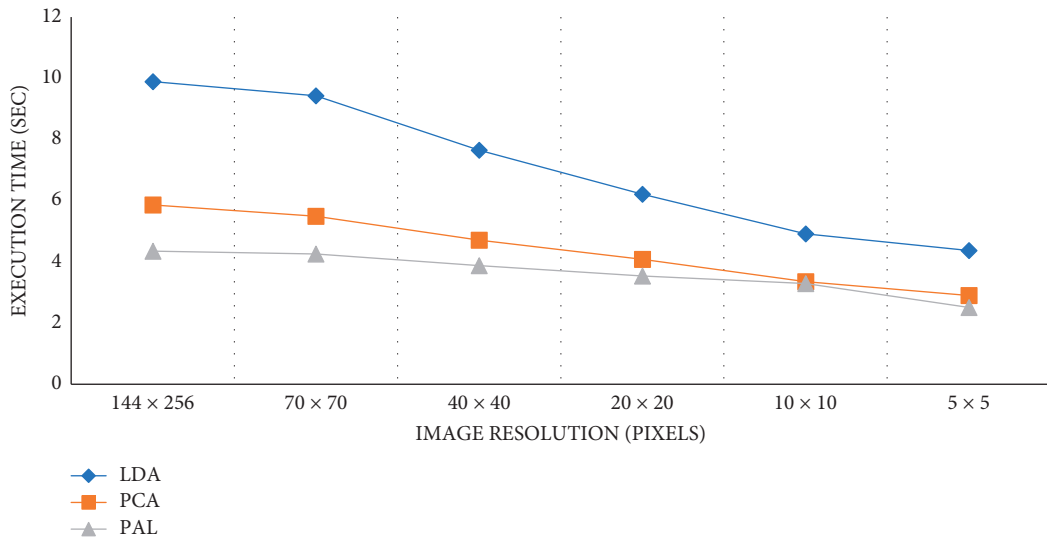


FIGURE 6: Computational complexity of the compared algorithms.

(iii) For the different face image resolutions, on average 3.23 seconds is consumed by PAL while executing even on a high-performance Super-Server machine. From the above analysis, it can be concluded that the compared algorithms are near real time.

5.4. Discussion. To develop a robust FR algorithm that can mimic the human vision system, continuous efforts are in progress. Table 7 further highlights the importance of the FR algorithms.

- (i) For extremely low-resolution frontal images, such as 20×20 pixels, PAL and PCA algorithms can be used.
- (ii) For low-resolution nonfrontal images, such as crime scenes, only PAL should be used.

TABLE 7: Selection of the FR algorithms based on performance.

Description	Algorithm
Low-resolution frontal images	PAL [36] and PCA [34]
Low-resolution images with face poses	PAL [36]
Time efficient with average accuracy	PCA [34]
Time efficient	PCA [34] and LDA [35]

(iii) For less computational complexity, face poses, and average accuracy, readers are suggested to use the PCA algorithm.

6. Conclusion and Future Work

This paper presents a dataset of face images with multiple poses (COMSATS face database). These images were

captured close to real-world conditions in the time span of five months in COMSATS University, Abbottabad (Campus). Face images included in this dataset can reveal the efficiency and robustness of future face detection and recognition algorithms. This database can be used for other research areas such as gender classification, age estimation, emotion recognition, face pose recognition, age estimation, and face modelling.

In the next step, a comparison of three well-known face recognition algorithms based on the proposed dataset is presented which are (i) PCA based face recognition (eigenfaces), (ii) LDA based face recognition, and (iii) PAL face recognition algorithm. Simulation results on the proposed database show that PAL face recognition algorithm can be reliably used for low resolution up to 5×5 -pixel images and from frontal (0°) ranges to $\pm 55^\circ$ of face pose variation near real time.

In our future work, we intend to develop a new face recognition algorithm that can recognize low-resolution face images up to 5×5 -pixel images and pose variation of $\pm 90^\circ$

Data Availability

The data are available with the first author and will be provided on request for research purposes.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.


References

- [1] Z. Mahmood, N. Muhammad, N. Bibi, and T. Ali, "A review on state-of-the-art face recognition approaches," *Fractals*, vol. 25, no. 2, Article ID 1750025, 1–19 pages, 2017.
- [2] F. Munawar, K. Uzair, S. Aamir, U. H. Mahmood, and M. Zahid, "An empirical study of image resolution and pose on automatic face recognition," in *Proceedings of the 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, Islamabad, Pakistan, January 2019.
- [3] M. Jacquet and C. Champod, "Automated face recognition in forensic science: review and perspectives," *Forensic Science International*, vol. 307, Article ID 110124, 2020.
- [4] D. T. Nguyen, D. P. Tuyen, T. D. Pham, N. R. Baek, and K. R. Park, "Combining deep and handcrafted image features for presentation attack detection in face recognition systems using visible-light camera sensors," *Sensors*, vol. 18, no. 3, 2018.
- [5] J. Wang, C. Lu, M. Wang, P. Li, S. Yan, and X. Hu, "Robust face recognition via adaptive sparse representation," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2368–2378, 2014.
- [6] Z. Mahmood, T. Ali, S. Khattak, and U. K. Samee, "A comparative study of baseline algorithms of face recognition," in *Proceedings of the 12th International Conference on Frontiers of Information Technology*, pp. 263–268, IEEE, Islamabad, Pakistan, December 2014.
- [7] M.-H. Yang, "Kernel eigenfaces vs. Kernel fisherfaces: face recognition using kernel methods," in *Proceedings of the Fifth IEEE International Conference on Automatic Face Gesture Recognition*, vol. 2, Washington, DC, USA, May 2002.
- [8] H. Qiu, D. Gong, Z. Li, W. Liu, and D. Tao, "End2End occluded face recognition by masking corrupted features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [9] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.
- [10] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher vector faces in the wild," *BMVC*, vol. 2, no. 3, p. 4, 2013.
- [11] S. Kakarwal and R. Deshmukh, "Wavelet transform based feature extraction for face recognition," *International Journal of Bioinformatics Research and Applications*, vol. 1, no. 1, Article ID 9740767, 2010.
- [12] N. A. Abdullah, M. J. Saidi, N. H. A. Rahman, C. C. Wen, and I. R. A. Hamid, "October. Face recognition for criminal identification: an implementation of principal component analysis for face recognition," in *AIP Conference Proceedings*, vol. 1891, no. 1, AIP Publishing LLC, Article ID 020002, 2017.
- [13] B. Huang, Z. Wang, G. Wang et al., "Masked face recognition datasets and validation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1487–1491, Montreal, BC, Canada, October 2021.
- [14] Q. Meng, S. Zhao, Z. Huang, and F. Zhou, "Magface: a universal representation for face recognition and quality assessment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Article ID 14225, 2021.
- [15] J. Zhang, G. Ye, Z. Tu et al., "A spatial attentive and temporal dilated (SATD) GCN for skeleton-based action recognition," *CAA I Transactions on Intelligence Technology*, vol. 7, 2020.
- [16] Q. Zou, X. Kang, Q. Fang, and B. Jiang, "Deep imitation reinforcement learning for self-driving by vision," *CAA I Transactions on Intelligence Technology*, vol. 6, no. 4, pp. 493–503, 2021.
- [17] L.-H. Juang, M.-N. Wu, and C.-H. Lin, "Affective computing study of attention recognition for the 3D guide system," *CAA I Transactions on Intelligence Technology*, vol. 5, no. 4, pp. 260–267, 2020.
- [18] J. Zhou, L. Liu, W. Wei, and J. Fan, "Network representation learning: from preprocessing, feature extraction to node embedding," *ACM Computing Surveys*, vol. 55, no. 2, pp. 1–35, 2022.
- [19] H. A. Zainab, A. M. Moamin, H. A. Karrar et al., "Comprehensive review of machine learning (ML) in image defogging: taxonomy of concepts, scenes, feature extraction, and classification techniques," *IET Image Processing*, vol. 16, no. 2, pp. 289–310, 2022.
- [20] R. K. Tripathi and A. Singh Jalal, "A robust approach based on local feature extraction for age invariant face recognition," *Multimedia Tools and Applications*, pp. 1–18, 2022.
- [21] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
- [22] A. Martinez and R. Benavente, "The AR face database," CVC Technical Report, No. 24, 1998.
- [23] V. Jain, "The Indian face database," 2002, <http://vis-www.cs.umass.edu/%7E%20vidit/IndianFaceDatabase/>.
- [24] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, 2003.

- [25] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.
- [26] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," vol. 1, no. 2, Technical Report 07-49, Amherst, Massachusetts, MA, USA, 2007.
- [27] A. Georghiades, P. N. Belhumeur, and D. J. Kriegman, *Yale Face Database*, vol. 2, 1997, <http://cvc.yale.edu/projects/yalefaces/yalefa>.
- [28] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using LDA-based algorithms," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 195–200, 2003.
- [29] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.
- [30] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [31] J. Wu and Z. H. Zhou, "Face recognition with one training image per person," *Pattern Recognition Letters*, vol. 23, no. 14, pp. 1711–1719, 2002.
- [32] S. Tan, X. Sun, W. Chan, L. Qu, and L. Shao, "Robust face recognition with kernelized locality-sensitive group sparsity representation," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4661–4668, 2017.
- [33] Y. Su, "Robust video face recognition under pose variation," *Neural Processing Letters*, vol. 47, no. 1, pp. 277–291, 2018.
- [34] J. Yang, D. Zhang, A. F. Frangi, and J. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.
- [35] M. Yang, "Kernel eigenfacesvs kernel Fisherfaces: face recognition using kernel methods," in *Proceedings of the 5th International Conference on Automatic Face and Gesture Recognition (AFGR)*, pp. 215–220, Washington, DC, USA, May 2002.
- [36] M. Haq, A. Shahzad, Z. Mahmood, A. Shah, N. Muhammad, and T. Akram, "Boosting the face recognition performance of ensemble based LDA for pose, non-uniform illuminations, and low-resolution images," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 6, pp. 3144–3164, 2019.
- [37] R. Ullah, H. Hayat, A. A. Siddiqui, U. A. Siddiqui, J. Khan, and F. Ullah, "A Real-Time Framework for Human Face Detection and Recognition in CCTV Images," *Mathematical Problems in Engineering*, vol. 2022, 2022.

Research Article

Image Deblurring Algorithm Based on the Gaussian-Scale Mixture Expert Field Model

Jing Zhang ¹ and Tao Zhang²

¹Basic Science Department, Jilin University of Architecture and Technology, Changchun 130114, Jilin, China

²Quality Manufacturing, Volkswagen (Anhui) Automotive Co., Ltd., Hefei 230091, Anhui, China

Correspondence should be addressed to Jing Zhang; zhangjingdlut@126.com

Received 8 October 2021; Revised 26 February 2022; Accepted 31 March 2022; Published 17 May 2022

Academic Editor: Nouman Ali

Copyright © 2022 Jing Zhang and Tao Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the proliferation of portable digital products, image quality degradation has received a lot of attention. As the most common phenomenon in image degradation, the issue of image deblurring is the focus of much attention. Blind motion blur removal is the main target of this paper. The heavy-tailed distribution is the most dominant statistical feature of natural images. However, most image deblurring methods use a gradient prior with fixed parameters to recover a clear image, which leads to loss of details in the recovered clear image and does not consider the higher order prior of the natural image. Therefore, this paper proposes a new regularized image recovery model based on the Gaussian-scale mixture expert field (GSM-FoE) model. First, the GSM-FoE model learns filters and corresponding parameters with higher order prior information of images by training images in a natural image library; second, these learning results are used to guide the image recovery process. The GSM-FoE model and gradient-fidelity based image recovery model is proposed, which can be used with an iterative re-weighted least squares (IRLS) method. Experiments demonstrate that the suggested recovery approach is simple to use and successful at reducing blur and noise, as well as suppressing ringing effects while preserving image information. Moreover, the image restoration method performs well for large blurring kernels. The results fully reflect the effectiveness and robustness of the proposed method for complex noise scenarios. The quality of the generated images is significantly better than that of several classical methods.

1. Introduction

Images are the primary form in which humans acquire, express, and communicate visual information. Motion blur is commonly used to portray the relative motion between the target object and the camera using a blur kernel. The goal of deblurring is to recover a clear-edged image from the observed blurred image for subsequent use in intelligent applications. Therefore, the problem of motion blur image restoration is of great theoretical and practical importance.

Algorithms on image deblurring have also evolved in the image field. Qin et al. proposed to remove motion blur based on the feature information (transparency information) of the image content itself [1]. Liu et al. proposed a deep learning approach to estimate the probability distribution of motion blur blocks using the CNN and Markov random field

model and then use the information based on the image blocks to solve the global inconsistent motion blur problem, but the blind deblurring problem of a single image increases the difficulty [2]. Abdelrahim estimated the blur kernel by extracting the salient edges of the blurred image, but there is a large amount of noise and ringing in the recovered image [3]. Sun et al. proposed a constraint based on the Laplace prior that can better preserve the edge and detail information of the recovered image [4]. Sun et al. introduces a new model to guide the image restoration process, namely, the use of continuous segmented function stitching to approximate the gradient distribution [5]. To better characterize the sparse nature of image gradients, Kja et al. proposed a constraint based on a super-Laplacian prior that makes the recovered image more consistent with natural scene properties, but the method cannot adaptively adjust

the strength of the penalty for different regions in the image. [6]. For the image noncoherent motion blur problem, Wang proposed a deep learning method of convolutional neural networks that can estimate and remove noncoherent motion blur more effectively [7]. Yang et al. used the TV blind convolution method, applying a sparse gradient prior as a constraint to solve the blind convolution problem [8]. Based on the edge information of the image, Xu et al. proposed a blind deconvolution method for MAP estimation model of the image [9]. In order to solve the problem of noncoherent motion blur, Jin et al. proposed the method of estimating blur kernel using a learning convolutional neural network (CNN) [10]. In recent years, some scholars have broken away from the original research idea and proposed the concept of learning-based image restoration to replace the smoothing constraint term based on regularization methods [11–15]. The basic idea of this class of methods is to obtain a priori knowledge of natural images through learning algorithms. For image prior terms, Huang et al. proposes a Gaussian scale mixture learning method combined with the Bayesian minimum mean squared error estimation to train the model [16]. In low-level computer vision, Nazarinzhad et al. proposed to learn prior information of natural images with the higher-order Markov random fields (MRF) [17].

Digital image processing techniques are increasingly used in high-end fields, and the study of image deblurring is the key factor to promote its development. Considering the fact that common algorithms are still prone to multi-peaks, this paper proposed the GSM-FoE model, which represents the spatial structure information of images, to mine the higher-order prior knowledge of natural images, and learns eight 3×3 filters that contain the higher-order prior knowledge of natural images. In the image restoration process, the gradient information of the image is also introduced into the image prior term in this paper. Experimental results and comprehensive comparison analysis demonstrate its superiority.

Concretely, our contributions are four-fold as follows:

- (i) This paper argues that although the gradient distributions of natural images all obey a heavy-tailed distribution, it is not appropriate to take a function to approximate this distribution directly, which would increase the error in the image recovery step. This will increase the error in the image recovery step.
- (ii) This paper argues that it is not sufficient to consider only the first-order a priori information of natural images. Based on a profound learning of the GSM-FoE model, this paper uses the GSM-FoE model to learn higher-order prior knowledge of natural images, and the results of these learned filters acting on the images reflect their intrinsic feature information.
- (iii) The deblurring algorithm, which combines the learning results with the gradient fidelity term, is used to maintain the image details and edge information well and suppress the ringing effect.
- (iv) For the image restoration model in this paper, we give an effective solution that works well for large images or large blur kernels.

2. GSM-FoE Model Offline Natural Image a Priori Learning Method

2.1. FoE Method. In a regularization-based approach, the Markov field (MRF) model can be used to model the spatial structure information of an image using potential functions to form a priori constraints. However, it is limited in that it can only use a simple neighborhood structure (each pixel is only related to its four nearest neighboring pixels), whereas the FoE (Field of Expert) model, which represents the spatial structure information of an image, can better address this limitation and can learn higher-order prior knowledge from the training image library. Figure 1 shows the flow process of the FoE model.

An a priori model based on image spatial information is introduced into the objective function of image deblurring. In other words, the FoE model is incorporated into the image restoration model. A neighborhood system is defined for a $m \times m$ (m is generally odd) square region such that it connects all nodes within the region. There are $km \times m$ systems of neighborhoods that may overlap with each other throughout the image x . Each neighborhood center pixel k ($k = 1, \dots, K$) then has an extremely large group $x_{(k)}$. The potential function of the group $x_{(k)}$ is denoted by $f(x_k)$, $f(x_k) = \prod_{i=1}^N \phi(J_i^T x_{(k)}; \Theta)$. Under the FoE model, the probability density function of the image x is as follows:

$$P(x) = \frac{1}{Z} \prod_{k=1}^K f(x_{(k)}) \quad (1)$$

$$= \frac{1}{Z} \prod_{k=1}^K \prod_{i=1}^N \phi(J_i^T x_{(k)}; \Theta).$$

Among them, N denotes the number of expert functions, ϕ_i is the expert function to be defined, J_i is the filter to be learned, Θ denotes the set of parameters to be learned, and the filter Z is the normalized parameter. The number of parameters in the model depends only on the size of the group and the number of filters, and there is no requirement for the size of the learned image x . In practice, the model is often transformed into the form of a Gibbs distribution for convenience.

$$P(x) = \frac{1}{Z} \exp(-E_{FoE}(x)), \quad (2)$$

$$E_{FoE}(x) = - \sum_k \sum_{i=1}^N \log \phi(J_i^T x; \Theta).$$

2.2. Selection of Canonical Terms and Expert Functions under the GSM-FoE Model. The literature [18] mentions, respectively, the use of TV regularization and l_p parametric regularization methods to fit the heavy-tailed distribution of image gradients. In addition to TV regularization and l_p parametric regularization methods, the concept of learning-

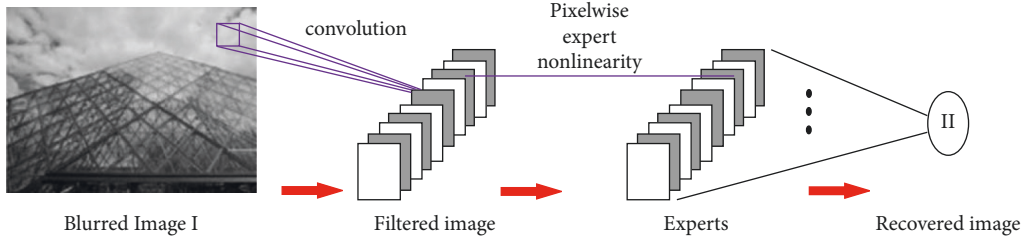


FIGURE 1: Flow process of the FoE model.

based image restoration has been proposed by some scholars in recent years. In addition to first-order derivative information, the intrinsic features of an image include higher-order prior information, and learning methods that train the prior can learn these higher-order priors from a given image database and are therefore more accurate than methods that approximate the prior.

If the image x in the training image library is transformed into the image I in the image restoration process, then the regular term describing the higher order prior of the image $p(I)$ under the FoE model can be expressed as follows:

$$P(I) = \frac{1}{Z} \prod_{k=1}^K \prod_{i=1}^N \phi(J_i^T I_k; \Theta). \quad (3)$$

Among them, N denotes the number of expert functions, ϕ_i is the expert function to be defined, J_i is the filter to be learned, Θ denotes the set of parameters to be learned, and the filter Z is the normalized parameter.

Under the MAP model, the selected expert function needs to satisfy the condition that it is guaranteed to be logarithmically continuous and differentiable. There are three classical expert functions: one is based on the student-t distribution, the second is Charbonnier's light and heavy-tailed expert function, and the third is the Gaussian scale mixture expert function. In spite of losing detail, the Student-t expert function's logarithmic distribution is more consistent with the heavy-tailed distribution than the Charbonnier expert function. The Gaussian scale mixture expert function [19] not only retains more detail information in the image but can also eliminate noise better, so the GSM expert function is used in this paper.

Although the gradient distributions of natural images all obey a neutral distribution, it is not reasonable to take a function to approximate this distribution directly, which would increase the error in the image recovery step. Therefore, in this paper, the expert function represented by the Gaussian scale mixtures with zero mean is chosen.

$$\phi(J_i^T I; \Theta) \propto \sum_{l=1}^L \frac{\pi_l}{\sigma_l} e^{-(J_i^T I)^2 / 2\sigma_l^2}. \quad (4)$$

Here, π_l and σ_l are the parameters of Θ . Filter J_i with higher order information about the image and the parameters Θ can be learned from the Berkeley image library using the EM (Maximum Expectation) algorithm [20] with the following training procedure:

Algorithm 1 describes the basis rotation algorithm.

Figure 2 shows the logarithmic distribution of the GSM expert function. The logarithmic distribution shows that the GSM expert function has thicker tails on both sides and a small spike in the middle, so the log-weighted tail distribution is more approximate. The advantage of the Gaussian scale hybrid expert function is that it includes a variety of represented expert functions, including Student-t experts and Charbonnier experts, and the scales and parameters of the GSM experts are adjustable, so the GSM experts are more flexible and diverse. It is not sufficient to consider only the first-order priori information of natural images. Based on a deep learning of the GSM-FoE model, we learn the higher-order prior knowledge of the natural images. 15 scales are selected in the training process, and eight 3×3 filters are learned. The result of these filters acting on the image reflects its intrinsic feature information. From the distribution of the weights in Figure 2, it can be seen that the yellow, red, and orange curves are more consistent with the heavy-tailed distribution at the $-3, -4$ scale, which means that the weight distribution at the $-3, -4$ scale fits the higher-order prior of the image better and can better reflect the intrinsic characteristics of the image itself. Based on the above analysis, the GSM expert function is selected in this paper for the image restoration process.

3. Image Deblurring Algorithm Based on the GSM-FoE Model

The quality of image recovery can be improved by using a priori constraints on image spatial structure information in regularized image recovery methods. The advantages of the FOE model in representing spatial structure information have attracted increasing attention [21–23].

By taking into account the higher-order prior of natural images, this paper introduces an a priori model based on image spatial information in the objective function of declaring and applies the learning results under the model of GSM FoE to guide image restoration. In the image restoration process, considering that the image gradient term can better suppress noise, this paper incorporates the gradient fidelity term into the image deblurring model as well, improves the traditional regularization term, proposes a regularization method based on GSM FoE and gradient fidelity, applies it to the single image blind deblurring problem, and gives an effective algorithm based on IRLS.

Input: Rotation filter J_i^T (base) as column composition matrix J , Berkeley training image bank, small image blocks I_k
 (1) Step E: $q_j(k) \propto \pi_j/\sigma_j e^{-1/2\sigma_j^2(J_i^T I_k)^2}$
 (2) Step MM: R is the orthogonal matrix to be learned, satisfying $W = RJ$, where r is the column vector of the orthogonal matrix R
 $r = \text{eig min } J^T (\sum_{k,j} q_j(k)/\sigma_j^2 I_{(k)} I_{(k)}^T) J$
 $J = Jr$
 Output: filter J

ALGORITHM 1: Basis Rotation Algorithm.

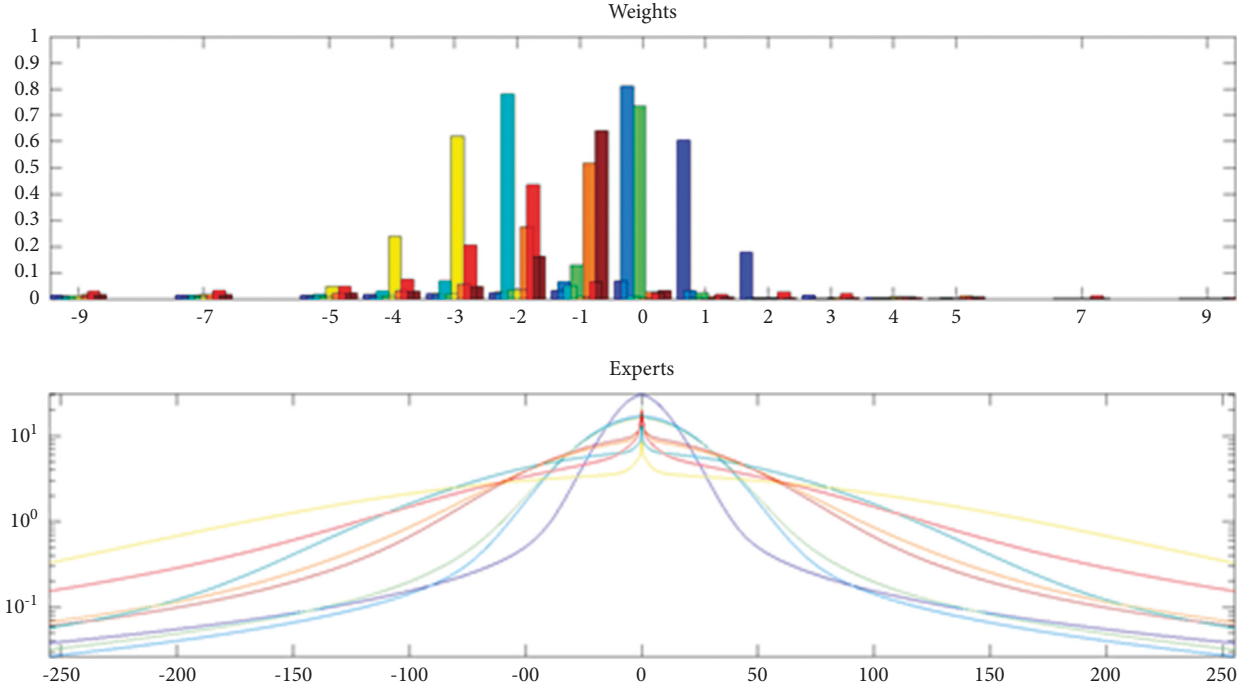


FIGURE 2: Logarithmic distribution of GSM-FoE functions.

3.1. Regularization Methods Based on the GSM-FoE Model.
 The energy function describing the higher order prior of the image under the GSM FoE model can be expressed from the reasoning in the literature [24] as follows:

$$E(I) = \min_q \sum_{i,l} q_l \left(\frac{1}{2\sigma_l^2} (J_i^T I)^2 - \ln \frac{\pi_l}{\sigma_l} + \ln q_l \right). \quad (5)$$

Of these, $q_l \propto \pi_l/\sigma_l e^{-1/2\sigma_l^2(J_i^T I_k)^2}$, $\sum_l q_l = 1$, l represent scales, 15 scales were selected during the course of the training. J_i denotes the filter learned by training, and $\{\sigma_l, \pi_l\}$ is the parameter learned.

From the distribution about the weights of the GSM experts at different scales, it is known that the weights at the -3 , -4 scale are more in line with the heavy-tailed distribution, so the -3 , -4 scale is chosen in this section, which can better fit the higher-order prior of the image. This soft fit can reduce the error in fitting the energy function and improve the accuracy of the image prior when the restriction

is changed from $(J_i^T I)^2$ to $J_i^T I$. Based on the above analysis, the energy function used in this paper is as follows:

$$E(I) = \sum_i \frac{1}{2} \left(\frac{1}{2\sigma_{(-3)}^2} (J_i^T I) + \frac{1}{2\sigma_{(-4)}^2} (J_i^T I) - \ln 4 \frac{\pi_{(-3)} \times \pi_{(-4)}}{\sigma_{(-3)} \times \sigma_{(-4)}} \right). \quad (6)$$

A conventional image restoration model based on regularization takes the following concrete form:

$$\min_I \|k * I - B\|_2^2 + \gamma \|\nabla I\|^2. \quad (7)$$

The second of these is noise suppression through image gradient information. More clear images with intrinsic priori information are obtained using an image prior term approach based on an expert field model. Within the framework of the MAP model, this paper incorporates the higher-order prior knowledge of natural images learned from the GSM-FOE model into the image prior term, while

suppressing noise by introducing image gradients information. The optimization model is as follows:

$$\begin{aligned} \min_I & \|I \otimes k - B\|_2^2 + \lambda_I \sum_i \frac{1}{2} \left(\frac{1}{2\sigma_{(-3)}^2} (J_i^T I) + \frac{1}{2\sigma_{(-4)}^2} (J_i^T I) - \ln 4 \frac{\pi_{(-3)} \times \pi_{(-4)}}{\sigma_{(-3)} \times \sigma_{(-4)}} \right), \\ \min_I & \|I - \hat{I}\|_2^2 + \gamma \|\nabla I\|^2. \end{aligned} \quad (8)$$

To solve model (9), we use an algorithm based on the iterative reweighted least squares (IRLS) method and the conjugate gradient method [20] for the iterative solution. The weights updated for each iteration are as follows:

$$w_v^{(i)} = \frac{1}{2(\sigma_{(-3)}^2 + \sigma_{(-4)}^2)(J_i^T I_{v-1})}, \quad (9)$$

where v denotes the position of the pixel in the image. The advantage of the new improved model is that this soft fit reduces the error of the energy function and thus improves the accuracy of the image prior.

Before giving the algorithms in this paper, we illustrate some notation as follows:

- (1) The symbol C_ϕ is used to denote a Toeplitz matrix, i.e., the matrix indistinct is first pulled into a row vector form, C_ϕ denotes the matrix formed by the elements of this row cycling backwards each time and then arranging them in rows.
- (2) The convolution $B = k \otimes I$ of the image is expressed in a matrix form: $B = C_k I$. To ensure that the matrix is multipliable, the rest of the values of the elements of each row in C_k , except for the elements of the fuzzy kernel k , are complemented by 0.
- (3) J denotes the learned filter.

Algorithm 2 describes the estimation process for clear images:

3.2. Process of Blind Image Deblurring. The recovery of blurred images in this paper is divided into three parts: offline GSM higher-order prior learning training, blur kernel estimation, and clear image recovery, as shown in flowchart Figure 3.

Assuming that motion blur is spatially globally consistent, the recovery model in this paper estimates the blur kernel k and the clear image I by iterating the following two equations alternatively.

$$\begin{aligned} \min_I & \|I \otimes k - B\|_2^2 + \gamma \|\nabla I\|^2 + \lambda_I E(I), \\ \min_k & \|I \otimes k - B\|_2^2 + \lambda_k \|k\|_2^2 + \mu C(k). \end{aligned} \quad (10)$$

Among them, K is the unknown fuzzy kernel and \otimes is the 2D convolution operator. Under the assumption of Gaussian noise, the fidelity term $E(B|I, k)$ is generally

denoted as $\|I \otimes k - B\|_2^2$. $E(I)$ is an image prior term based on the GSM-FoE model, which guides the recovery of clear images by mining the higher-order prior information of natural images. $\|\nabla I\|^2$ is based on image gradient information to suppress noise. $\lambda_k \|k\|_2^2$ is the fuzzy kernel prior term. The weights λ_k and λ_I are the parameters of the kernel prior and the image prior, respectively, are the parameters of the image gradient and are the parameters of the discrete metric $C(k)$.

The fuzzy image is first converted into a gray-scale image; in the estimation stage of the fuzzy kernel, given the input iterative image I , this paper uses a constraint-based l_0 approach to extract the significant structure; in order to retain more structural information, the strong edges are restored with impact filtering; finally, the fuzzy kernel is estimated.

In the image recovery phase, the offline GSM-FoE model is used to train the Berkeley image database so as to learn higher-order prior knowledge of natural images, which is used to guide the recovery of clear images; the GSM-FoE prior model and the gradient-fidelity regularization term are used as model constraints, and an optimization algorithm is proposed.

The detailed parts of blurred images tend to cause inaccuracy in the estimation of the blur kernel. The double-sided filter allows more low frequency information to be retained. The specific algorithm is as follows:

$$F(I(x)) = \frac{1}{Z_x} \sum_{y \in T} f(|x - y|) g(|I(x) - I(y)|) I(y). \quad (11)$$

Among them, x and y are the coordinates of the pixel points in the image, W is the image pixel space, Z_x is the normalized term, I is the image to be filtered, $F(I(x))$ is the image after bilateral filtering, and both f and g denote domain filters.

Considering the problem of noise in the image, this paper performs a bilateral filtering algorithm on I_b of Algorithm 2 to obtain the image $I_{b'}$. Calculate the image detail layer as $I_d = I_b - I_{b'}$. Finally, a clear image $I'_s = I + I_d$ with more detailed information is obtained.

In recent years, many papers have considered sparsity constraints in the kernel estimation process, but they tend to lead to non-convexity problems. To avoid this problem and to take into account the continuity of the kernel, the kernel estimation model [24] used in this paper is as follows:

Input: Fuzzy kernels k , Filters J , Fuzzy image B , steps iter
 Initialization: value $w_i^{(0)} = 1$;
 (1) Calculate $A = C_k^T C_k + \lambda \sum_N C_{I_i}^T C_{I_i}$, $b = C_k^T B$;
 For $v = -1$: iter
 (2) Calculate $\bar{A} = \sum_N^{i=1} A^T w_v^{(i)} A$, $\bar{b} = \sum_N^{i=1} A^T w_v^{(i)} b$;
 (3) Solving systems of equations using the conjugate gradient method $\bar{A}I = \bar{b}$, result is I_v
 (4) Value $u_v = AI_v - b$,
 If $u_v > \text{ther}$ then $w_{v+1}^{(i)} = 1/2(\sigma_{(-3)}^2 + \sigma_{(-4)}^2)J_i^T I_{v-1}$
 Else exit
 End
 Output: Clear image I^*

ALGORITHM 2: Algorithms IRLS.

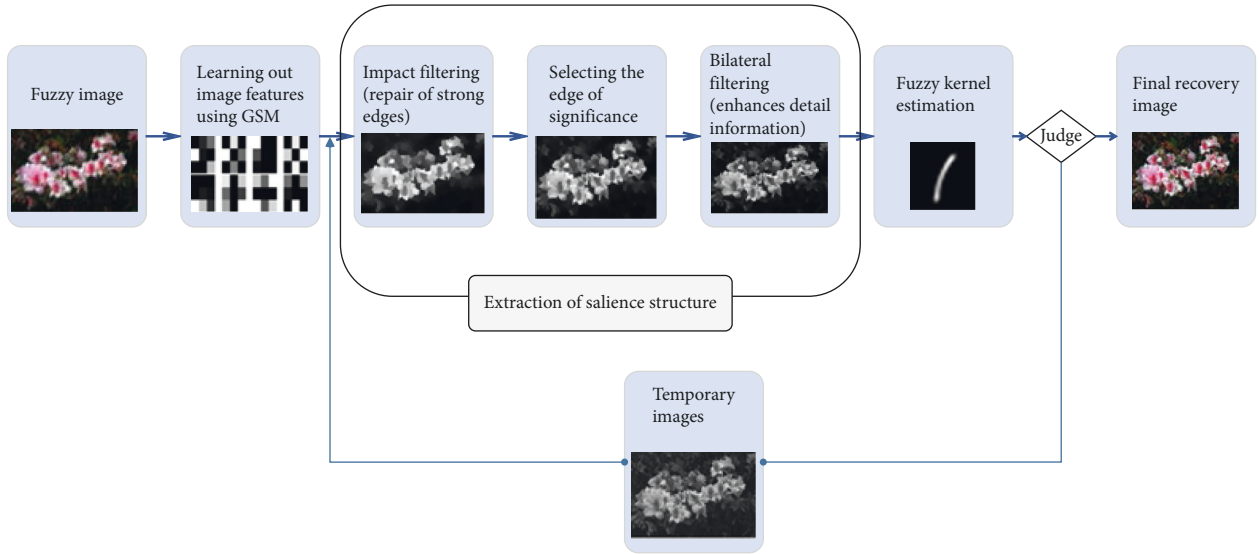


FIGURE 3: Flowchart of the proposed blind deblur algorithm.

$$\begin{aligned} \min_k \|\nabla I \otimes k - \nabla B\|_2 + \lambda_k \|k\|_2^2 + \mu C(k), \\ \text{s.t. } k(x, y) \geq 0 \text{ and } \sum_{\{(x,y)\}} k(x, y) = 1. \end{aligned} \quad (12)$$

Among them, $C(k) = \#\{(x, y) \mid |\partial_x k(x, y)| + |\partial_y k(x, y)| \neq 0\}$. Due to the use of discrete metrics $C(k)$, the solution is to alternate iterations of the following two processes.

$$\begin{aligned} \min_k \|\nabla I \otimes k - \nabla B\|_2 + \lambda_k \|k\|_2^2, \\ \min_k \|\hat{k} - k\|_2 + \mu C(\hat{k}). \end{aligned} \quad (13)$$

λ_k and μ denote adjustable parameters to balance the strength of the fidelity and kernel similarity terms. The first process is a convex function on the fuzzy kernel k . The second process is the L_0 gradient minimization problem.

Once the fuzzy kernel has been estimated, the image can be recovered using the fuzzy image and the estimated fuzzy kernel. In order to make full use of the learned higher order prior knowledge, while avoiding noise in the image recovery process, a GSM FoE prior model and gradient fidelity regular terms are used and the recovery model is as follows:

$$\min_I \|k * I - B\|_2^2 + \gamma \|\nabla I\|^2 + \lambda_I \sum_i \frac{1}{2} \left(\frac{1}{2\sigma_{(-3)}^2} (J_i^T I) + \frac{1}{2\sigma_{(-4)}^2} (J_i^T I) - \ln 4 \frac{\pi_{(-3)} \times \pi_{(-4)}}{\sigma_{(-3)} \times \sigma_{(-4)}} \right). \quad (14)$$

$\{J_l, \pi_l, \sigma_l, l = -3, -4\}$ indicates the filter that has been learned and the parameters.

Value $a = \sigma_{-3}^2 + \sigma_{-4}^2/2\sigma_{-3}^2\sigma_{-4}^2, b = \ln 4\pi_{-3} \times \pi_{-4}/\sigma_{-3} \times \sigma_{-4}$. The relative energies ratio of the regularization methods based on the GSM-FoE model is $\sum_{i,p \in I} a(J_i^T I) - b/\sum_{i,p \in I} a(J_i^T B) - b$. Experiments show that the property that a regular term based on GSM-FoE model and gradient fidelity has an image energy greater than that of a clear image, making the model converge more towards the clear image.

4. Numerical Experiments and Analysis

In this paper, a coarse to fine multiscale approach is used to recover a clear image. The total number of layers is determined by the size of the fuzzy kernel k . The size of k determines the number of iteration steps per layer to be 25. First, the color image is converted into a gray-scale image, on which the saliency structure is extracted; second, the estimation of the blur kernel and the recovery of the image are performed; then, the recovered image is upsampled and used as the initial input for the next iteration. In the final image restoration stage, the three channels of the color image are processed separately.

The parameters of the experiments are set as follows: the parameters in the model $\beta = 0.7, \lambda_k = 0.001$, and $\mu = 0.01$. The parameters under the GSM-FoE model λ_l and γ are adjustable parameters, and the experiments generally take $\lambda_l = 0.001, \gamma = 0.01$, the number of expert functions $N = 8$, and the window size 3×3 . The EM learning algorithm is applied to learn eight filters and their corresponding parameters from the training database set.

The simulation platform is MATLAB R2018a, the computer configuration is: 64 bit windows 10 system, Pentium dual-core 2.8 GHz, running memory is 2 GB. In our experiments, our learning data came from the Berkeley Segmentation Benchmark image library. There are 500 benchmark images in this database, including many common natural scenes in everyday life, such as animals, people, landscapes, buildings, etc. It is the most commonly used image database for image segmentation, edge detection, and other related fields. The eight 3×3 filters used in the following experiments are the result of learning this database as a whole, which is undoubtedly very time consuming. Given the efficiency of the experiments, the user can also manually select images in the image library that have some correlation with the blurred image B for training.

4.1. Experimental Comparison of Image Prior Terms with and without the Gradient Fidelity Term. Tikhonov first proposed to use this constraint on the squared norm $\|\nabla I\|_2^2$ of the gradient as a regular term to solve the discomfort problem of images to better remove noise. In this paper, we combine the gradient fidelity term with the GSM-FoE model, proposing a novel image regularity term. The experimental results show that the addition of a gradient fidelity term significantly suppresses noise in the recovered image and reduces the generation of the ladder effect. Since the gradient fidelity

term is a convex problem that can be solved by the fast Fourier transform, it does not destroy the existence of the optimal solution of the deblurred model.

Figure 4 shows an experimental comparison of the image prior term with and without the gradient fidelity term. Figure (a) shows a clear image, figure (b) shows a blurred image, figure (c) shows the image restoration results with only the GSM FoE model introduced in the image prior term, and figure (d) represents the introduction of the GSM-FoE model and the gradient fidelity term in the image prior term. In terms of the recovery results, some information such as edges and textures are partially missing in figure (c). Figure (d) not only eliminates the noise but also suppresses the ringing effect by adding the image gradient fidelity term, and the recovered image has a clearer and more natural visual effect at the edges and textures.

4.2. Comparison of Quantitative Experiments. Using accuracy and time-consuming experiments as indicators, literature [5], literature [18], and literature [20] were used as control groups for the experiments and their experimental results were compared with the experimental results of this research method. The Bregman reweighted alternating minimization (BRAM) was applied to image deblurring [5]. Nonblind image deblurring was proposed via deep learning (DL) in a complex field [18]. The image restoration method used in this paper is the GSM-FoE model. The gradient-based conditional generative adversarial network (CGAN) was used to image deblurring [20]. The experimental data samples were obtained from the Berkeley image database, and the effect of different methods on the recovery of degraded images was verified through synthetic datasets.

4.2.1. Recovery Time Comparison. 500 blurred images under synthetic data were selected for blind recovery, and the algorithm in this paper was compared with other three algorithms for recovery time, and the experimental results are shown in Figure 5.

According to the analysis of the data in Figure 5, the recovery times for the methods of literature [18], literature [20] and this paper are relatively short when the number of images is small. Between 190 and 230 images, there is a relatively large qualitative change in the recovery time of the methods of literature [5], literature [18], and literature [20] as the number of images increases, followed by a relatively stable oscillation. Overall, as the number of images increases, the recovery time of the method in this paper is relatively stable and the time required for recovery is short and efficient.

4.2.2. Comparison of the Degree of Recovery. To further validate the performance of this method, PSNR (peak signal-to-noise ratio), MSE (mean square error), and the degree of recovery were used as test metrics, and the methods of literature [5], literature [18], and literature [20] were used as control groups to blindly recover blurred images,

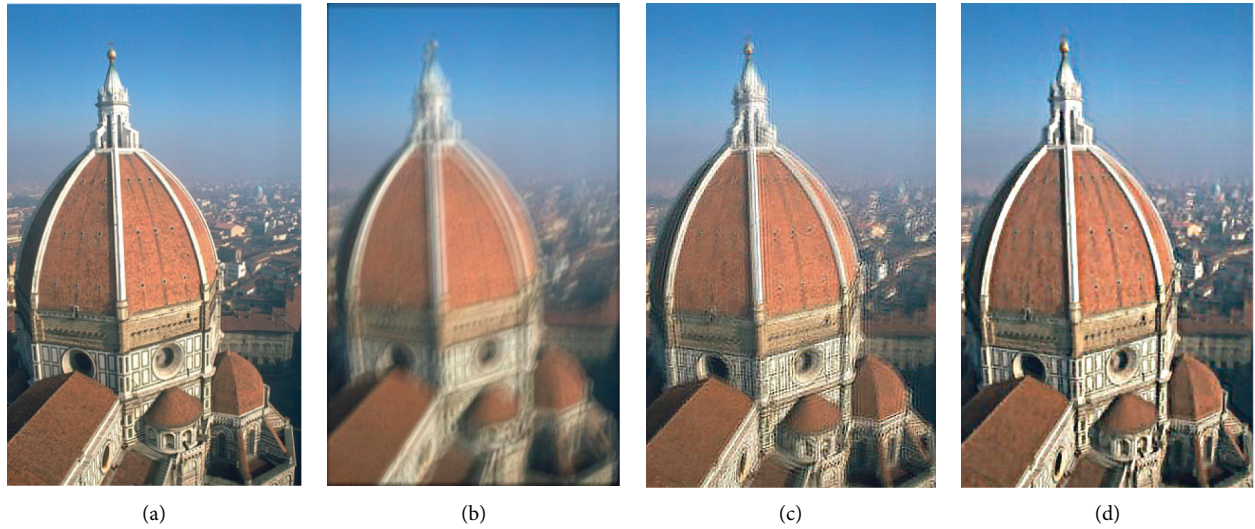


FIGURE 4: Results with and without the gradient fidelity term in the GSM-FoE model. (a) Clear image. (b) Blurred image. (c) No gradient fidelity term. (d) With gradient fidelity term.

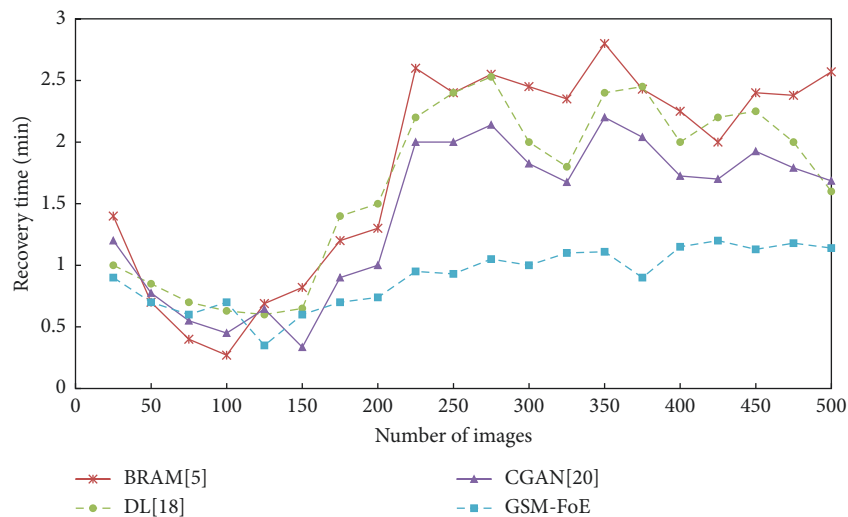


FIGURE 5: Comparative results of recovery times for different methods.

respectively, to compare the recovery effects of the five algorithms, and the metric data are shown in Table 1.

The experiments show that the peak signal-to-noise ratio value of the method in this paper is 93.25 db, which is at least 7 db higher than that of other methods in the literature. The mean square error value is 15.18%, which is much lower than that of other methods. Figure 6 shows the comparison of the recovery degree of different methods. As shown in Figure 6, as the number of images increases, the recovery degree of the literature [18] method shows a sudden low change between the number of 10 and 50 images, followed by a smooth decrease. Overall, there is a steady oscillation in the degree of recovery in literature [5] and literature [20] as the number of images increases, while the degree of recovery of the method in this paper then increases and is significantly higher than that of the other three literature methods, indicating that the

recovery results in this paper are clearer and better maintain the details of the images.

4.3. Recovery Effect. The blind recovery experiments of the method are shown in Figure 7, where the fuzzy kernel is unknown. On the left is the fuzzy image, and on the right is the recovery result. In addition to the first-order derivative information, the intrinsic features of the image also include higher-order prior information, and the training prior learning method can learn these higher-order priors from a given image database, so the regular term method in this paper is more accurate than the approximation prior method. In terms of the overall visual effect, this paper effectively suppresses the ringing effect while removing blur, and maintains the edges of the image well.

TABLE 1: Evaluation indicator values for each method.

Evaluation indicators	BRAM [5]	DL [18]	CGAN [20]	GSM-FoE
PSNR/DB	76.59	83.45	85.64	93.25
MSE/%	24.6	38.17	34.94	15.18

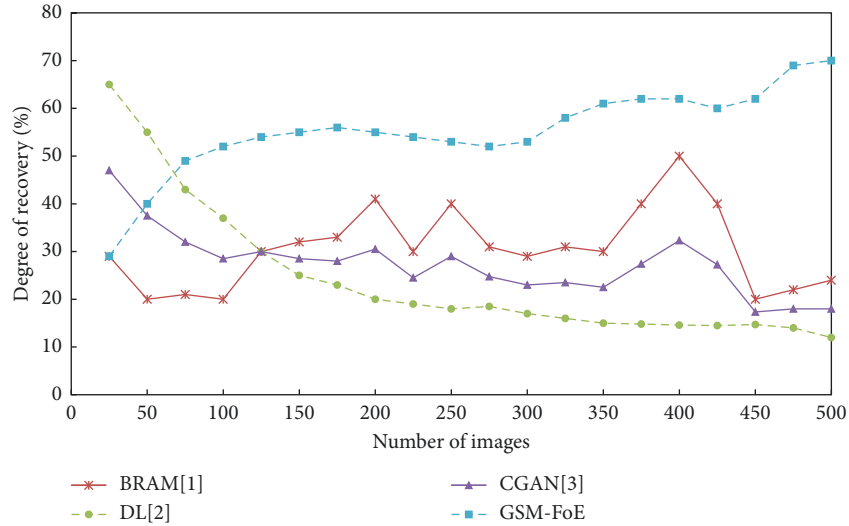


FIGURE 6: Comparative results of the degree of recovery by different methods.



FIGURE 7: Experimental results of the method in this paper.

5. Conclusion

Image deblurring techniques have a wide range of applications in daily life, industrial production, and other fields and have received widespread attention in research areas such as image processing and computer vision. The method of learning with the GSM-FoE model can better fit the higher-order prior of natural images and accurately portray the global prior knowledge of natural images. The graph of the logarithmic distribution of the expert function shows that the Gaussian mixed-scale expert function fits the heavy-tailed distribution better than other expert functions. In this paper, using the GSM-FoE model, 500 images from the Berkeley database were trained to learn eight filters and the corresponding parameters.

In this paper, these learning results under the line of GSM-FoE model are used to guide the image restoration process in the objective function of image deblurring. In the image restoration process, the gradient fidelity term is also incorporated into the image deblurring model in this paper, considering that the image gradient term can better suppress noise, so the traditional regular term is improved. An effective algorithm based on IRLS is proposed for the GSM-FoE model and the image gradient fidelity based regular term, which adaptively changes the parameter values during the iterative process and the recovered image can better maintain the details. In this paper, an efficient algorithm with alternate iterations of fuzzy kernel and image recovery is used, and experiments show that the algorithm can effectively handle the case of large fuzzy kernels.

In future research work, more consideration needs to be given to the following issues. First, training against the Berkeley image library is time-consuming, so we need to find a relatively fast learning method; second, given the diversity of learning models, further research studies will follow on other learning models and their application in the direction of image deblurring. Therefore, the next step is to investigate a more efficient method of extracting saliency structure, which can incorporate it into the global inconsistent image deblurring problem.

Data Availability

Some or all data, models, or codes generated or used during the study are available in a repository or online in accordance with funder data retention policies.

Conflicts of Interest

The authors declare that there are no conflicts of interest with any financial organizations regarding the material reported in this manuscript.

Acknowledgments





This work was supported by Jilin Province Higher Education Association Project (JGJX2020D516).

References

- [1] C. Qin, P. Ji, C. C. Chang, J. Dong, and X. Sun, "Non-uniform watermark sharing based on optimal iterative BTC for image tampering recovery," *IEEE Multimedia*, vol. 25, 2018.
- [2] Z. Liu, X. Li, P. Luo, C. Loy, and X. Tang, "Deep learning Markov random field for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, 2018.
- [3] E. M. Abdelrahim, "Hierarchical adaptive genetic algorithm based T-S fuzzy controller for non-linear automotive applications," *International Journal of Fuzzy Systems*, vol. 24, pp. 1-15, 2021.
- [4] H. Sun, X. Yang, and H. Gao, "A spatially constrained shifted asymmetric Laplace mixture model for the grayscale image segmentation," *Neurocomputing*, vol. 331, no. FEB.28, pp. 50-57, 2019.
- [5] T. Sun, L. Qiao, and D. Li, "Bregman reweighted alternating minimization and its application to image deblurring," *Information Sciences*, vol. 503, pp. 401-416, 2019.
- [6] J. Kja, S. Ying, L. Qixin, L. Jun, W. Xiaofei, and Z. Wensheng, "Image restoration using overlapping group sparsity on hyper-Laplacian prior of image gradient," *Neurocomputing*, vol. 420, pp. 57-69, 2021.
- [7] Y. T. Wang, X. L. Zhao, T. X. Jiang, L. J. Deng, Y. Chang, and T. Z. Huang, "Rain streaks removal for single image via kernel-guided convolutional neural network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 99, pp. 1-13, 2020.
- [8] Y. Yang and J. Jia, "An image reconstruction algorithm for electrical impedance tomography using adaptive group sparsity constraint," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 9, pp. 1-11, 2017.
- [9] Z. Xu, H. Chen, and Z. Li, "Fast blind deconvolution using a deeper sparse patch-wise maximum gradient prior," *Signal Processing: Image Communication*, vol. 90, no. 3, Article ID 116050, 2021.
- [10] K. H. Jin, M. T. Mccann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, vol. 26, no. 99, pp. 4509-4522, 2016.
- [11] T. Fan and J. Xu, "Image classification of crop diseases and pests based on deep learning and fuzzy system," *International Journal of Data Warehousing and Mining*, vol. 16, no. 2, pp. 34-47, 2020.
- [12] B. Zhang, Z. Zhu, and C. Xu, "A primal-dual multiplier method for total variation image restoration," *Applied Numerical Mathematics*, vol. 145, no. Nov, pp. 145-158, 2019.
- [13] P. Yifan, S. Qilin, D. Xiong, W. Gordon, H. Wolfgang, and H. Felix, "Learned large field-of-view imaging with thin-plate optics," *ACM Transactions on Graphics*, vol. 38, no. 6, pp. 1-14, 2019.
- [14] K. Singh, D. K. Vishwakarma, and G. S. Walia, "Blind image deblurring via gradient orientation-based clustered coupled sparse dictionaries," *Pattern Analysis & Applications*, vol. 22, no. 2, pp. 549-558, 2019.
- [15] J. Liu and S. Osher, "Block matching local SVD operator based sparsity and TV regularization for image denoising[J]," *Journal of Scientific Computing*, vol. 78, no. 1, pp. 1-18, 2019.
- [16] D. Hazarika, V. K. Nath, and M. Bhuyan, "SAR image d based on a mixture of Gaussian distributions with local parameters and m edge detection in lapped Transform domain," *Sensing and Imaging*, vol. 17, no. 1, p. 15, 2016.
- [17] J. Nazarinezhad and M. Dehghani, "A contextual-based segmentation of compact PolSAR images using Markov Random Field (MRF) model," *International Journal of Remote Sensing*, vol. 40, no. 3, pp. 985-1010, 2019.
- [18] Y. Quan, P. Lin, Y. Xu, and Y. H. Nan, "Nonblind image deblurring via deep learning in complex field," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 70, no. 99, pp. 1-14, 2021.
- [19] T. Tirer and R. Giryes, "Image restoration by iterative d and backward projections," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1220-1234, 2019.
- [20] H. Zhao, D. Wu, H. Su, and S. J. Zheng, "Gradient-based conditional generative adversarial network for non-uniform blind deblurring via DenseResNet," *Journal of Visual Communication and Image Representation*, vol. 74, Article ID 102921, 2021.
- [21] S. Bourouis, H. Sallay, and N. Bouguila, "A competitive generalized gamma mixture model for medical image diagnosis[J]," *IEEE Access*, vol. 30, no. 99, 2021.
- [22] C. G. Wilson, T. F. Shipley, and A. K. Davatzes, "Evidence of vulnerability to decision bias in expert field scientists," *Applied Cognitive Psychology*, vol. 34, no. 5, pp. 1217-1223, 2020.
- [23] K. Morris-Binelli, S. Muller, F. V. Rens, G. H. Allen, and M. R. Simon, "Individual differences in performance and learning of visual anticipation in expert field hockey goalkeepers," *Psychology of Sport and Exercise*, vol. 52, 2020.
- [24] Z. Zhang, S. Liu, J. Peng, and M. G. J. Yao, "Simultaneous spatial, spectral, and 3D compressive imaging via efficient Fourier single-pixel measurements," *Optica*, vol. 5, no. 3, p. 315, 2018.

Research Article

An LSTM with Differential Structure and Its Application in Action Recognition

Weifeng Chen ^{1,2}, Fei Zheng ^{2,3}, Shanping Gao ¹ and Kai Hu ²

¹Quanzhou University of Information Engineering, Quanzhou, Fujian, China

²School of Automation, Nanjing University of Information Science & Technology, Nanjing, China

³China Telecom Ningbo Branch, Zhejiang, Ningbo, China

Correspondence should be addressed to Weifeng Chen; cwf6426@nuist.edu.cn

Received 22 January 2022; Accepted 6 April 2022; Published 10 May 2022

Academic Editor: Saadat Hanif Dar

Copyright © 2022 Weifeng Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Because of the broad application of human action recognition technology, action recognition has always been a hot spot in computer vision research. The Long Short-Term Memory (LSTM) network is a classic action recognition algorithm, and many effective hybrid algorithms have been proposed based on basic LSTM infrastructure. Although some progress has been made in accuracy, most of those hybrid algorithms have to have more and more complex structures and deeper network levels. After analyzing the structure of the classic LSTM from the perspective of control theory, we determined that the classic LSTM could strengthen the differential characteristics of human action recognition technology to reflect the change of speed. Thus, an improved LSTM structure with an input differential characteristic module is proposed. Furthermore, in this article, we considered the influence of first-order and second-order differential on the extraction of movement pose information, that is, the influence of movement speed and acceleration on action recognition. We designed four different LSTM units with first-order and second-order differential. Moreover, the experiments were performed for the four units on three common datasets repeatedly. We found that the LSTM network with the input differential feature module proposed in this article can effectively improve action recognition accuracy and stability without deepening the complexity of the network and can be used as a new basic LSTM network architecture.

1. Introduction

With the widespread use of virtual reality technology [1], human-computer interaction, intelligent transportation [2, 3], and other fields [4] in real life, action recognition research has been rapidly developing, and action recognition occupies a pivotal position in computer vision. The goal of this research was to detect the action in video or image sequences, judge action categories, or predict further actions. At present, action recognition research methods can be divided into two categories: one is based on manual feature extraction [5–9], and the other is based on deep neural network learning features.

The method based on manual feature extraction takes the traditional machine learning method to extract features from the video, then encode the features, normalize the coding vector, train the model, and finally predict and

classify the actions. Its advantage lies in extracting features according to needs, strong pertinence, and simple implementation; however, the datasets present lighting, similar actions (jogging and running), dynamic background, and other noises in action recognition [10]. These noises make the manual extraction features challenging to classify in subsequent classification tasks; therefore, the research work on action recognition based on manual feature extraction methods is currently limited—the most representative one is iDT (improved Dense Trajectories). The iDT algorithm is the most stable in this type of algorithm, but its computation speed is slow, and real-time requirements cannot be satisfied due to a large amount of calculation required.

Most of the existing network frameworks of action recognition algorithms based on deep learning [11] are developed from the convolutional neural network [12–15]. Because action recognition objects are video sequences,

they increase time-series information compared with a single image. Action recognition algorithms based on deep learning are generally used to learn the features of a time series. Long short-term memory (LSTM) is a classic action recognition algorithm used in deep networks. It is a kind of time recurrent neural network, specially designed to solve the long-term dependence problem of a general recurrent neural network (RNN). Because LSTM can process time-series information, the LSTM network is often applied in action recognition, and many effective hybrid algorithms are derived. Yue-Hei Ng et al. [16] proposed the two-stream convolutional network model combined with LSTM, reducing computational cost and learning the global video features. The two-stream convolutional network uses a convolutional neural network (CNN: AlexNet or GoogLeNet) on ImageNet to extract the image features and optical flow features of the video frames and then inputs the extracted image features and optical flow features to the LSTM network for processing to get the final result. Although the effect achieved by this network is general, it provides a new idea for the research of action recognition. Even if there is a large amount of noise in the optical flow images, the network combined with LSTM can be helpful for classification. Du et al. [17] proposed an end-to-end recurrent pose-attention network (RPAN). The RPAN combines the attention mechanism with the LSTM network to represent more detailed actions. Long et al. [18] proposed an RNN framework with multimodal keyless attention fusion. The network divides visual features (including RGB image features and optical flow features) and acoustic features into equal-length segments and inputs them to LSTM. The network's advantage is that it reduces computation cost and improves computation speed. The LSTM is applied to extract different features in this network. Song et al. [19] used skeleton information to train the LSTM and divided the network into two sub-networks: a temporal attention subnetwork and a spatial attention subnetwork. Tang et al. [20] proposed a novel coherence constrained graph (GCC) LSTM with spatio-temporal context coherence (STCC) and GCC to effectively recognize group activity, by modeling the relevant motions of individuals while suppressing the irrelevant motions. Shu et al. [21] proposed a novel hierarchical long short-term concurrent memory (H-LSTCM) to model the long-term inter-related dynamics among a group of persons for recognizing human interactions. Shu et al. [22] also proposed a novel skeleton-joint co-attention recurrent neural network (SC-RNN) to capture the spatial coherence among joints, and the temporal evolution among skeletons simultaneously on a skeleton-joint co-attention feature map in spatiotemporal space. Networks of action recognition based on deep learning are mainly based on three types: two-stream convolution network, 3D convolution network, and the LSTM network [23, 24]. With the further development of computer vision, the study of action recognition is limited to the above three networks. The attention mechanism and the NTU RGB+D skeleton dataset have also been researching hotspots in action recognition in recent years. Simultaneously, most of the existing action

recognition algorithms based on deep learning are based on the classic LSTM model, which has derived many effective hybrid models.

From the development of action recognition in recent years, we can see that LSTM network is widely used in the research of action recognition. The action recognition algorithm based on the LSTM network depends on the more and more complex network framework, and the improvement of accuracy depends on the depth of the network and the number of parameters. The overcomplex hybrid networks have high requirements for machine hardware and do not improve the attention to the action fine features. At present, action recognition is mostly applied in the human-computer interaction, such as the conversion of action between a real person and a simulated digital person in somatosensory games, which pays great attention to the fineness of the action. So action recognition should pay more attention to the action posture and the extraction of action features. In order to better deal with the problems existing in the video datasets of action recognition, such as complex background, illumination transformation, and action similarity (such as walking and running), and to improve the recognition accuracy without deepening the complexity of the algorithm framework, an action recognition algorithm based on improved LSTM network is studied.

Observing the development process of action recognition research in recent years, we believe that the research work of action recognition based on the LSTM network tends to more complex mixture models, but the research results on the information of LSTM itself appear less. However, in many practical applications, the research still cares about the details of the action itself. Moreover, an overly complicated network will make recognition speed slow. In further studying the classic LSTM, we believe that if we consider the LSTM structure from control theory, the LSTM has a proportion (P) and integral (I). If we refer to the standard PID control, we can see that the classic LSTM lacks a differential (D) link. The first-order differential represents the speed of motion from the robot control, and the second-order differential represents acceleration. We can further consider adding multiple first-order or second-order enhanced input differential modules to implement different basic network models.

The contributions of this article are as follows:

- (1) Improves the classical LSTM ability to capture action's speed. The idea of an input differential LSTM unit is proposed. The concept of control differential in PID is introduced into the deep learning network. It can increase the impact of time series on action recognition and consider the different speeds and accelerations of the human body. The first-order differential corresponds to the movement speed, and the second-order differential corresponds to the action acceleration. Therefore, we intend to add the differential input module in a classical LSTM structure, to enhance the capture of speed and acceleration information in motion and improve the recognition accuracy.

(2) On the basis of improving the classical LSTM architecture, this article applies it to LRCN to improve the performance of LRCN motion recognition. Based on the input gate, forget gate, and output gate of the original LSTM unit, the input of action differential (including the first-order differential and the second-order differential) is added. Furthermore, the basic LSTM algorithm with four kinds of enhanced input differential modules is designed. By testing three classic datasets, the accuracy is improved compared with the original LSTM unit, and the stability is not decreased compared with the original LSTM unit, but the training speed is weak. The enhanced input differential LSTM unit can replace the original LSTM unit and flexibly be applied in various network frameworks to realize different application scenarios. The enhanced input differential LSTM unit has a good development prospect.

This article is divided into five sections. Section 1 introduces the development process of action recognition research; Section 2 introduces related knowledge and methodology; Section 3 introduces the four related models proposed in this article; Section 4 describes the experiments performed on the three kinds of datasets to test the performance of the 4 LSTM units proposed in this article; Section 5 summarizes the work of this article.

2. Related Knowledge and Methodology

2.1. Related Knowledge

2.1.1. Recurrent Convolutional Neural Network. The recurrent neural network [25] establishes a weight connection among the input layer's neurons, hidden layer, and output layer in the neural network. The output of the network module's hidden layer at each moment depends on the information of the previous moment. The recurrent network module of RNN can learn the current moment's information and save the information of the previous time series. However, for long-time-series information, RNN is prone to the problem of gradient disappearance. Therefore, the LSTM network is proposed to solve this problem.

The LSTM network replaces the hidden layer node in the original RNN model with a memory unit [26]. The key lies in the cell state to store historical information. There are three gate structures [27] to update or delete information in the cell state through the Sigmoid function and point-by-point product operation. Figure 1 shows an LSTM unit's internal structure; from left to right are the forget gate, the input gate, and the output gate. The LSTM network can process sequence information through the cumulative linear form to avoid gradient disappearance [28] and learn long-period information. Thus, the LSTM network can be used to learn long-time sequence information.

The equation of the forget gate is

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

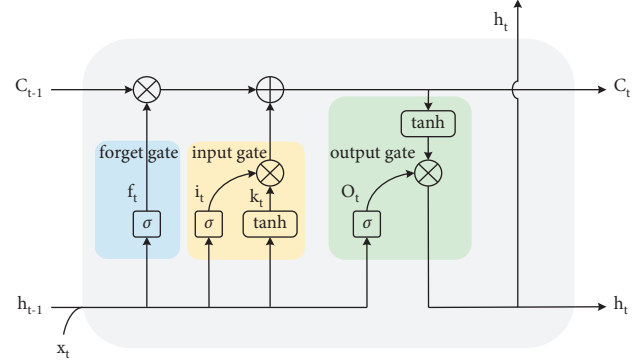


FIGURE 1: This is the internal structure of the long short-term memory (LSTM) network unit (basic LSTM).

where f_t is the output value of the forget gate, h_{t-1} is the output value of the last moment, x_t is the input value of the current moment, and w_f and b_f are the weight matrix and bias vector in the Sigmoid function of the forget gate, respectively. $[h_{t-1}, x_t]$ is the connection matrix of h_{t-1} and x_t .

The equations of the input gate are

$$i_t = \sigma(w_i * [h_{t-1}, x_t] + b_i), \quad (2)$$

$$k_t = \tanh(w_k * [h_{t-1}, x_t] + b_k), \quad (3)$$

where i_t and k_t are the input gate's output values, and w_i and b_i are the weight matrix and bias vector in Sigmoid function of the input gate, respectively; w_k and b_k are the weight matrix and bias vector in the tanh function of the input gate, respectively.

The equations of the output gate are

$$O_t = \sigma(w_o * [h_{t-1}, x_t] + b_o), \quad (4)$$

$$h_t = O_t * \tanh(C_t), \quad (5)$$

where O_t is the output value of the output gate, w_o and b_o are the weight matrix and bias vector in the Sigmoid function of the output gate, respectively, and h_t is the output value of the current moment.

The updated cell state is

$$C_t = f_t * C_{t-1} + i_t * k_t, \quad (6)$$

where C_t is the cell state of the current moment and C_{t-1} is the cell state of the last moment.

2.1.2. PID Control. PID control is the abbreviation of proportional, integral, and differential control, which has the advantages of a simple algorithm, good robustness, and high reliability. In the control system, the PID controller constitutes the control error according to the given value and the actual output value and performs proportion, integral, and differential computations operation on the error. The three computation results are linearly combined to obtain the total control value and then control the controlled object. PID control is a linear control algorithm based on the estimation of error "past," "present," and "future" information [29]. The principle of the conventional PID control system is shown in Figure 2.

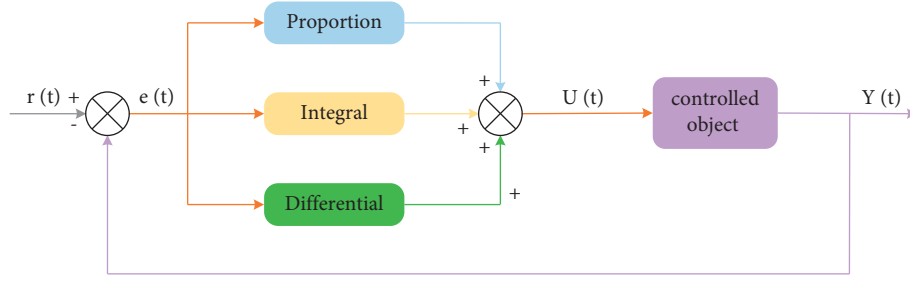


FIGURE 2: This is the schematic diagram of the PID control system.

Among them, $r(t)$ is the system input, $U(t)$ is the controller output, $Y(t)$ is the system output, $e(t)$ is the system error, and $e(t) = r(t) - Y(t)$.

The formula of the controller output is

$$U(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}, \quad (7)$$

where K_p is the proportional coefficient, T_i is the integral-time constant, T_d is the differential-time constant, K_i is the integral coefficient, $K_i = K_p/T_i$, and K_d is the differential coefficient, $K_d = K_p * T_d$.

Figure 2 shows that PID has three correction links: proportion, integral, and differential. The proportion link proportionally reflects the error signal $e(t)$ of the control system. Once the error occurs, the proportional controller will perform at the fastest speed to reduce error and control the “now” error. Because the adjustment function of proportional control is based on the error, it reflects PID control’s rapidity. The integral link can remember the error. For the “past” error of the system, it is mainly to eliminate the steady-state error. The strength of the integral effect is mainly determined by the integral-time constant T_i . The larger the T_i is, the weaker the integral action. The integral action reflects the accuracy of PID control. The differential link can reflect the trend of the error signal (rate of change). Given the “future” error, the dynamic characteristics of the closed-loop system can be improved through advanced action, which reflects the stability of PID control.

We extract the idea of differential control in PID control. The first-order differential can increase the information capture of the LSTM unit on the action speed. The second-order differential can increase the network’s information capture of the action acceleration. The improved input differential LSTM unit can improve the network’s stability while improving the accuracy of the network’s action recognition.

2.2. Methodology. By analyzing the classic LSTM model, we believe that the recurrent memory network retains the last video frame h_{t-1} and inputs video frame x_t , using different weights w_f and w_i to express the relationship between the frame’s information. This is the relationship between the information of the LSTM frame and the current video frame.

When w_f and w_i are positive, it is an integral (I) relationship between the information. When w_f and w_i are negative, it is a differential (D) relationship between the information. Simultaneously, the weight added to the current video frame’s information also becomes a proportion (P) relationship. Considering that w_f and w_i are positive, when programming, the classic LSTM only contains the proportion (P) and integral (I). We believe that from the PID control, we can try to add a differential (I) relationship to the classic LSTM. From deep learning, it is also a feature enhancement idea.

From the perspective of robot kinematics, the action information features include motion limb status, posture, speed, and acceleration. Take the manipulator arm of a robot as an example, the arm’s movement includes the translation of the center of mass and rotation around the center of mass. When the Newton–Euler equation analyzes the manipulator’s arm, the dynamic equation is as follows:

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta), \quad (8)$$

where $M(\theta)$ is the $n \times n$ mass matrix of the operating arm, $V(\theta, \dot{\theta})$ is the centrifugal force of $n * 1$, and the Gothic force vector, which depends on the position and speed, $G(\theta)$, is the gravity vector of $n \times 1$. $M(\theta)$ and $G(\theta)$ are complex functions of the position of all the joints θ of the manipulator’s arm. $\dot{\theta}$ is the first order of angle change and represents speed. $\ddot{\theta}$ is the second-order change of angle change and represents acceleration. In control theory, the control of a robot requires first-order and second-order differential.

At present, the action recognition networks based on deep learning focus on the action posture information. So, increasing the information extraction of the action limbs’ speed and acceleration can increase the network’s final performance. The action speed and acceleration are the first-order and second-order differential of the posture, reflecting the posture changes trend. In this article, we introduce the differential in PID control combined with the classic LSTM

unit to realize the extraction of multiple information such as the posture, speed, and acceleration of the action.

Although the current action recognition research based on LSTM focuses on the influence of the time series on action recognition, a basic LSTM unit considers only two time series in a short period of time: the current moment and the last moment; only a part of the previous time series is retained because of the forget gate of LSTM. However, for a complete action, the action is continuous. An action cannot be completed in just two short-time sequences. A simple action (such as bowing) requires at least 3-4 time series to complete. Besides, the actions in the dataset are more complex and require more time series to complete. So, it is more effective to retain more time-series information for action recognition.

From the above ideas, this article combines the original LSTM basic unit with the differential input module to build the improved input first-order and second-order differential LSTM units. In this article, we structure a basic network framework and a multilayer LSTM to show the improved input differential network's performance. We hope that the improved input differential LSTM unit can improve the network's recognition performance in the end through the experimental results. Furthermore, we hope that the LSTM unit, combined with PID control differentiation, can capture more abundant action information. The proposed LSTM units can be flexibly applied in different networks to realize different applications.

3. LSTM Network Based on Input Differentiation

Although this article's network framework is relatively simple, it can better reflect the effect of improving the input differential LSTM unit in terms of accuracy and stability.

3.1. Improved LSTM Unit with First-Order Input Differential.

Figure 3 shows that the improved first-order input differential LSTM unit adds a new input module to the original LSTM. In the mathematical model, the first-order differential of the x_t part in the differential module is $dx(t)/dt$. In the design of this article, since t is a fixed value and is a small value, the first-order differential of the input $dx(t)/dt$ is approximately as $x_t - x_{t-1}$, that is, $dx(t)/dt \approx x_t - x_{t-1}$; in this way, the differential can be realized, and the calculation is convenient. From the observation of basic image processing, $x_t - x_{t-1}$, the optical flow method in image processing provides the information on image change.

The state equations of the forget gate, input gate, and output gate of the LSTM unit with improved input first-order differential are shown in equations (1)–(5).

The state equations of the first-order input differential are

$$\begin{aligned} d_t &= \sigma(w_d * [h_{t-1}, x_t - x_{t-1}] + b_d), \\ e_t &= \tanh(w_e * [h_{t-1}, x_t - x_{t-1}] + b_e), \end{aligned} \quad (9)$$

where d_t is the output value of the first-order differential in Sigmoid function, and e_t is the output value of the first-order

differential in tanh function, $x_t - x_{t-1}$ is the first-order input differential, w_d and b_d are the weight matrix and bias vector in Sigmoid function of the first-order input differential, respectively, and w_e and b_e are the weight matrix and bias vector in the tanh function of the first-order input differential, respectively.

The updated cell state is

$$C_t = f_t * C_{t-1} + i_t * k_t + d_t * e_t. \quad (10)$$

3.2. Improved LSTM Unit with Second-Order Input Differential.

Figure 4 shows that the improved second-order input differential LSTM unit adds a second-order differential input module to the original LSTM unit. The improved input second-order differential LSTM unit is applied to the network model so that the network can extract the dual information of action features and action acceleration.

The state equations of the forget gate, input gate, and output gate of the LSTM unit with improved second-order input differential are shown in equations (1)–(5).

The state equations of the second-order input differential are:

$$\begin{aligned} d_t &= \sigma(w_d * [h_{t-1}, x_t - x_{t-2}] + b_d), \\ e_t &= \tanh(w_e * [h_{t-1}, x_t - x_{t-2}] + b_e), \end{aligned} \quad (11)$$

where d_t is the output value of the second-order differential in Sigmoid function, e_t is the output value of the second-order differential in tanh function, $x_t - x_{t-2}$ is the second-order input differential, w_d and b_d are the weight matrix and bias vector in Sigmoid function of the second-order input differential, respectively, and w_e and b_e are the weight matrix and bias vector in the tanh function of the second-order input differential, respectively.

The updated cell state is

$$C_t = f_t * C_{t-1} + i_t * k_t + d_t * e_t. \quad (12)$$

3.3. Improved LSTM Unit with Double First-Order Input Differentials.

Figure 5 shows that the improved double first-order input differentials LSTM unit adds two first-order differential input modules to the original LSTM unit. The improved input double first-order differential LSTM unit is applied to the network model to extract more action characteristics and speed information.

The state equations of the forget gate, input gate, and output gate of the LSTM unit with improved double first-order input differentials are shown in equations (1)–(5).

The state equations of the double first-order input differentials are

$$\begin{aligned} d_t &= \sigma(w_d * [h_{t-1}, x_t - x_{t-1}] + b_d), \\ e_t &= \tanh(w_e * [h_{t-1}, x_t - x_{t-1}] + b_e), \\ p_t &= \sigma(w_p * [h_{t-1}, x_t - x_{t-1}] + b_p), \\ q_t &= \tanh(w_q * [h_{t-1}, x_t - x_{t-1}] + b_q), \end{aligned} \quad (13)$$

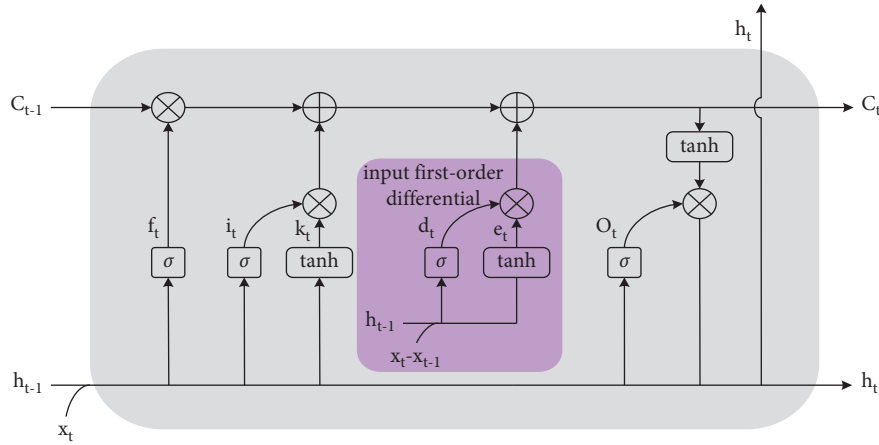


FIGURE 3: This is the improved LSTM unit with the first-order input differential (1st D Lstm).

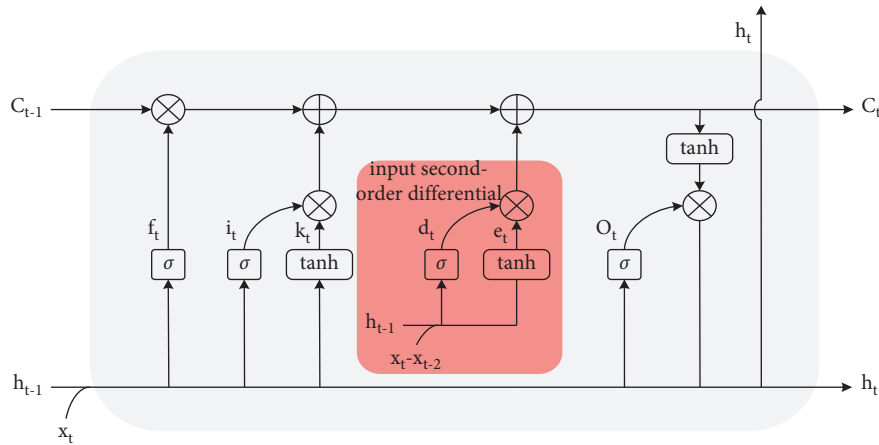


FIGURE 4: This is the improved LSTM unit with second-order input differential (2nd D Lstm).

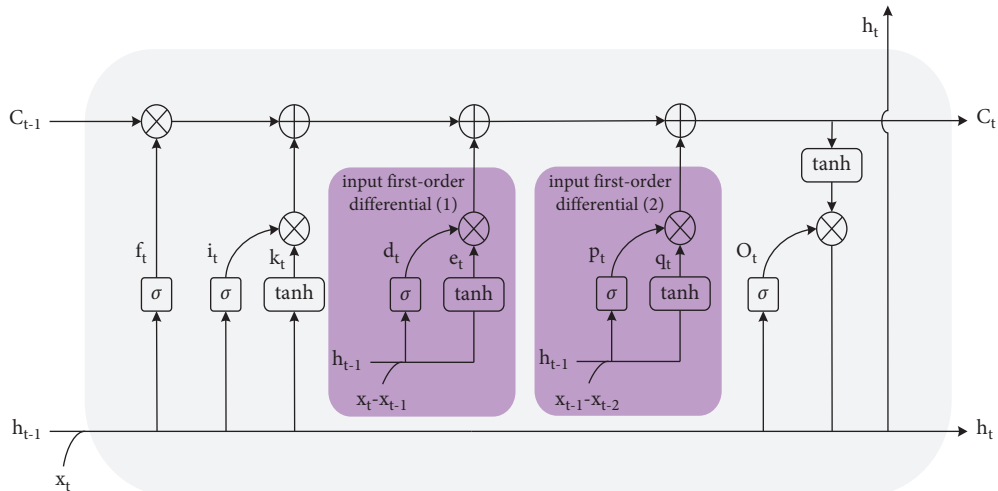


FIGURE 5: This is the improved LSTM unit with double first-order input differentials (double 1st D Lstm).

where d_t and p_t are the output values of the first-order differential in Sigmoid function, and e_t and q_t are the output values of the first-order differential in tanh function, $x_t - x_{t-1}$ is the first-order input differential, w_d and w_p are

the weight matrices in Sigmoid function of the double first-order input differentials, b_d and b_p are bias vectors in Sigmoid function of the double first-order input differentials, w_e and w_q are the weight matrices in the tanh function of the

double first-order input differentials, and b_e and b_q are bias vectors in the tanh function of the double first-order input differentials.

The updated cell state is

$$C_t = f_t * C_{t-1} + i_t * k_t + d_t * e_t + p_t * q_t. \quad (14)$$

3.4. Improved LSTM Unit with First-Second-Order Input Differentials. As shown in Figure 6, the improved first-second-order input differential LSTM unit adds a first-order differential module and a second-order differential module to the original LSTM unit. The improved first-second-order input differential LSTM unit is applied to the network model, enabling the network to extract multiple information of motion characteristics and motion speed and acceleration.

The state equations of the forget gate, input gate, and output gate of the LSTM unit with improved first-second-order input differentials are shown in equations (1)–(5), and the state equations of first-second-order input differentials are shown in equations (15)–(18).

The state equations of the first-second-order input differentials are

$$d_t = \sigma(w_d * [h_{t-1}, x_t - x_{t-1}] + b_d), \quad (15)$$

$$e_t = \tanh(w_e * [h_{t-1}, x_t - x_{t-1}] + b_e), \quad (16)$$

$$p_t = \sigma(w_p * [h_{t-1}, x_t - x_{t-2}] + b_p), \quad (17)$$

$$q_t = \tanh(w_q * [h_{t-1}, x_t - x_{t-2}] + b_q), \quad (18)$$

where d_t is the output value of the first-order differential in Sigmoid function, e_t is the output value of the first-order differential in tanh function, $x_t - x_{t-1}$ is the first-order input differential, p_t is the output value of the second-order differential in Sigmoid function, q_t is the output value of the second-order differential in tanh function, $x_t - x_{t-2}$ is the second-order input differential, w_d and w_p are the weight matrices in Sigmoid function of the first-second-order input differentials, b_d and b_p are the bias vectors in Sigmoid function of the first-second-order input differentials, w_e and w_q are the weight matrices in the tanh function of the first-second-order input differentials, and b_e and b_q are bias vectors in the tanh function of the first-second-order input differentials.

The updated cell state is

$$C_t = f_t * C_{t-1} + i_t * k_t + d_t * e_t + p_t * q_t. \quad (19)$$

4. Experiment

4.1. Datasets. Research teams, both overseas and domestic, usually use human action datasets in algorithm training to detect the algorithm's accuracy and robustness. The dataset has at least the following two important functions:

- (1) The researchers need not care about the process of collection and pretreatment.

- (2) Ability to detect and compare different performances of different algorithms under the same standard.

The KTH dataset [30] was released in 2004. The KTH dataset includes six kinds of actions (walking slowly, jogging, running, boxing, waving, and clapping) performed by 25 people in 4 different scenes. The dataset has 2391 video samples and includes scale transformation, clothing transformation, and lighting transformation. However, the shooting camera is fixed and the background is relatively single.

The Weizmann dataset [31] was released in 2005 and includes nine people completing ten kinds of actions (bending, stretching, high jump, jumping, running, standing, hopping, walking, waving1, and waving2). In addition to category tags, the dataset contains silhouettes of people in the foreground and background sequences to facilitate background extraction. However, the dataset has a fixed perspective and simple backgrounds.

The above two datasets were released early. The citation rate of the datasets in the traditional methods of action recognition is high, which significantly promotes action recognition for the future. However, with the rapid development of action recognition, there are shortcomings: the background is simple, the perspective is fixed, and each video has only one person. The above two datasets already cannot satisfy real action recognition requirements, so now they are rarely used.

The Hollywood2 dataset [32] was released in 2009. The video data in the dataset were collected from Hollywood movies. There are 3669 video clips in total, including 12 action categories (answering the phone, eating, driving, etc.) extracted from 69 movies and 10 scenes (outdoor, shopping mall, kitchen, etc.). The dataset is close to real situations.

The University of Central Florida released the UCF-101 dataset [33] in 2012. The dataset samples include various action samples collected from TV stations and video samples saved from the video website YouTube. There are 13,320 videos, including five types of actions (human-object interaction, human-human interaction, limb movements, body movement, and playing musical instruments), and 101 class-specific small actions. The dataset has many samples and rich action categories and can train the algorithm well, so it is widely used.

Brown University released the HMDB-51 dataset [34] in 2011. The video samples come from the video clips of the movie and video website YouTube. There are 51 types of sample actions and 6849 videos in total. Each type of sample action in the dataset contains at least 101 videos.

The UCF-101 dataset and the HMDB-51 dataset have many action data types and a wide range of actions and are more classic in action recognition. The scenes in the Hollywood2 dataset are more complex and closer to real life. To comprehensively reflect the improved LSTM unit's performance proposed in this article, we adopted three databases, including UCF-101, 1 HMDB-51, and Hollywood2, for training and testing. Furthermore, the improved LSTM units were tested and improved performance in the above three databases.

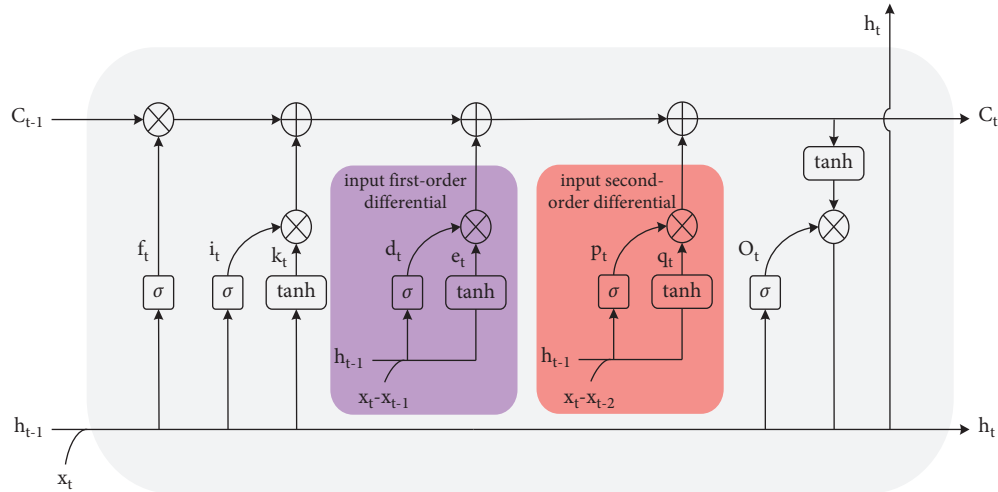


FIGURE 6: This is the improved LSTM unit with first-second-order input differentials (1st + 2nd D Lstm).

4.2. Experimental Method. To more intuitively and effectively test the effect of the improved input differential LSTM unit proposed in this article on action recognition, the experiments adopted a relatively simple network framework model: long-term recurrent convolutional networks (LRCNs) [35]. In future research, the four different input differential LSTM units proposed in this article can be directly used instead of the original basic LSTM units or directly replaced into a more complex network model containing LSTM units to achieve better application performance.

The LRCN directly connects the LSTM model to the convolutional neural network and simultaneously learns temporal and spatial features. The LRCN network framework is shown in Figure 7. The model converts the video data in the dataset into frame images and then uses the pretrained CNN network to extract frame images' features. Moreover, the LRCN inputs the extracted features to the improved input differential LSTM network to extract the time sequence information and finally classifies the results by SoftMax.

This article uses the method in Donahue's work [33]. The method uses the convolutional network to extract the spatial features and the LSTM network to extract the temporal features. However, the method in this article is slightly different from the original text. In the CNN feature extraction step, we adopted InceptionV3 to extract more accurate frame image features. The InceptionV3 requires little computation close but has high performance. In the step of extracting time sequence information by LSTM network, the number of network layers is customized according to computer performance and recognition accuracy requirements. Moreover, the different orders of the input differential LSTM units proposed in this article were adopted in the LRCN.

The LSTM units (discussed in Sections 2.1 and 3.1–3.4) were applied to the network model framework in Figure 7. The improved LSTM units were evaluated from the three indexes of accuracy, loss, and standard deviation. To better reflect the improved LSTM units' performance, the

experiments were carried out on three datasets of HMDB-51, UCF-101, and Hollywood2, respectively. The experiments use only a single variable of the LSTM unit. The input data model, training parameters, and other parameters were consistent. The batch_size is 32, the hidden layers' parameter is 1024, the full connection layers' parameter is 512, and the loss function is the classic cross-entropy function. Moreover, the optimizer is Adadelta optimizer, the learning rate is 0.001, and the decay rate is 0.95. In LRCN, we adopted 5 levels, and each level has 1024 LSTMs to build its structure. All works are end-to-end training and end-to-end models.

The recording was as follows: recording the original LSTM unit as basic Lstm; recording the first-order input differential LSTM unit as 1st D Lstm (the model in Section 3.1); recording the second-order input differential LSTM unit as 2nd D Lstm (the model in Section 3.2); recording the double first-order input differentials LSTM as double 1st D Lstm (the model in Section 3.3); and recording the first-second-order input differentials LSTM unit as 1st + 2nd D Lstm (the model in Section 3.4).

The assessment method used the direct hold-out method. To avoid the influence of the other bias introduced by the data division process on the result and increase the final evaluation result's fidelity, the training set and the testing set were divided equally at each type of action in every dataset in the experiment. The training set accounts for 70% of the total dataset, and the testing set accounts for 30% of the total dataset. Simultaneously, to make the results more stable and reliable, this article uses multiple hold-out to take the average value as the experiment's final evaluation results. Each LSTM unit uses the hold-out method described above to divide the dataset and then conduct the experiment. After an experiment concluded, the dataset was redivided, and the experiment was performed again and repeated. The experiments were performed using three datasets of five different LSTM units; each was repeated three times. At last, the average accuracy of three experimental results is the result of the LSTM unit.

Our experiment's hardware configuration was an Intel I7-9700K CPU, two Nvidia GeForce GTX2080Ti graphics

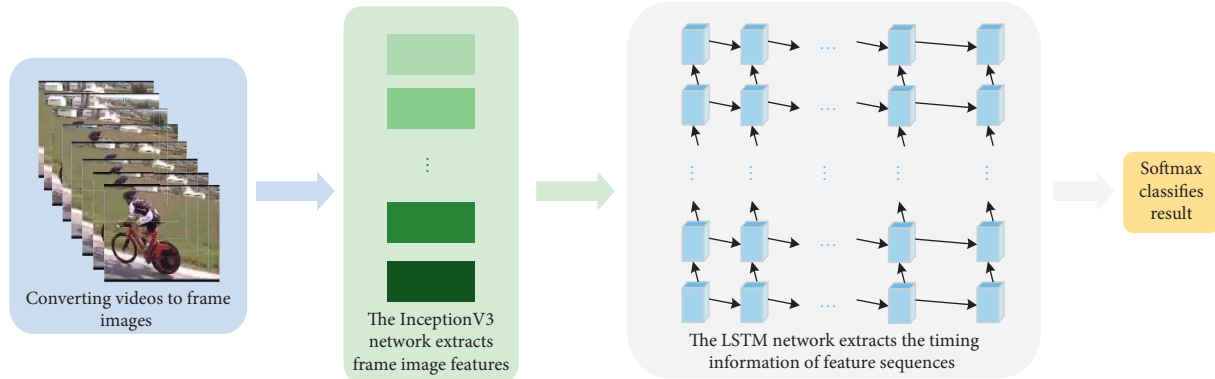


FIGURE 7: This is the LSTM network framework based on improved input differentiation.

cards, 4 * 16 G total 64 GB memory. The software environment was configured as Ubuntu 16.04, CUDA 8.0, Cudnn 6.0 for CUDA 8.0, TensorFlow 1.4, and python 3.5.

4.3. Experimental Results and Analysis

4.3.1. Performance Experiments of Five Models in This Article on the HMDB-51 Dataset. Figures 8(a) and 8(b) show the comparison graphs of accuracy and loss of action recognition applied to five different LSMT units on the HMDB-51 dataset.

Figure 8(a) shows that on the HMDB-51 dataset, the accuracy of basic Lstm is 39.99%, the accuracy of 1st D Lstm is 41.34%, the accuracy of 2nd D Lstm is 41.44%, the accuracy of double 1st D Lstm is 42.27%, and the accuracy of 1st + 2nd D Lstm is 43.30%.

The above experimental data show that in the HMDB-51 dataset, the four different LSTM units proposed in this article have different degrees of improvement in the accuracy of action recognition than the original LSTM unit. However, the train-step is delayed to some extent when the accuracy reaches a stable level. Although the basic LSTM unit's loss is low overall, sometimes there is a step phenomenon in the loss. The loss of the 1st D LSTM unit is slightly higher than that of the basic LSMT unit. Compared with the loss of the above two LSTM units, the loss of double 1st D LSTM unit, 2nd D LSTM unit, and 1st + 2nd D LSTM unit is higher. In general, in the HMDB-51 dataset, the LSTM algorithm with enhanced input differential features is improved in accuracy compared with the classic LSTM algorithm without input differential features.

4.3.2. Performance Experiments of Five Models in This Article on the UCF-101 Dataset. Figures 9(a) and 9(b) show the comparison graphs of accuracy and loss of action recognition applied to five different LSMT units on the UCF-101 dataset.

Figure 9(a) shows that, on the UCF-101 dataset, the accuracy of basic Lstm is 71.15%, the accuracy of 1st D Lstm is 79.88%, the accuracy of 2nd D Lstm is 73.42%, the accuracy of double 1st D Lstm is 71.99%, and the accuracy of 1st + 2nd D Lstm is 72.67%.

From the above experimental data, it can be concluded that on the UCF-101 dataset, the 1st LSTM unit has the highest accuracy, and the overall loss is low, but there are still higher steps. Besides, with the superposition of training steps, there is a fluctuation in the loss. The other three LSTM units' networks' accuracy also improved compared to the original LSTM unit. In general, in the UCF-101 dataset, the LSTM algorithm with enhanced input differential features is improved in accuracy compared to the classical LSTM algorithm without input differential features.

4.3.3. Performance Experiments of Five Models in This Article on the Hollywood2 Dataset. Figures 10(a) and 10(b) show the comparison graphs of accuracy and loss of action recognition applied to five different LSMT units on the Hollywood2 dataset.

Figure 10(a) shows that on the Hollywood2 dataset, the accuracy of basic Lstm is 46.49%, the accuracy of 1st D Lstm is 47.85%, the accuracy of 2nd D Lstm is 47.89%, the accuracy of double 1st D Lstm is 46.54%, and the accuracy of 1st + 2nd D Lstm is 47%. The videos in the Hollywood2 dataset are relatively long, and the scenes are complicated. There are some interference actions except for tags in a video. Maybe for the above reason, 1st D Lstm performed better on the Hollywood2 dataset.

In general, in the Hollywood2 dataset, the LSTM algorithm with enhanced input differential features improved accuracy compared with the classical LSTM algorithm without input differential features. Table 1 shows the ablation experimental data (including mean accuracy and deviation) on 3 sets of datasets, and the bolded data are the highest.

Other researchers [36] used the LRCN model to perform action recognition, obtaining an accuracy of 38.8% and 68.3% on the HMDB-51 and UCF-101 datasets, respectively. The results are similar to the experimental results in this article; thus, this article's experimental data are credible. Through experiments, we found that different classes of LSTM differential units have different accuracy on different datasets. Compared with the original LSTM unit, the accuracy of four differential LSTM units had specific improvements. For the dataset with only a single action in videos, the first-order input differential LSTM unit might

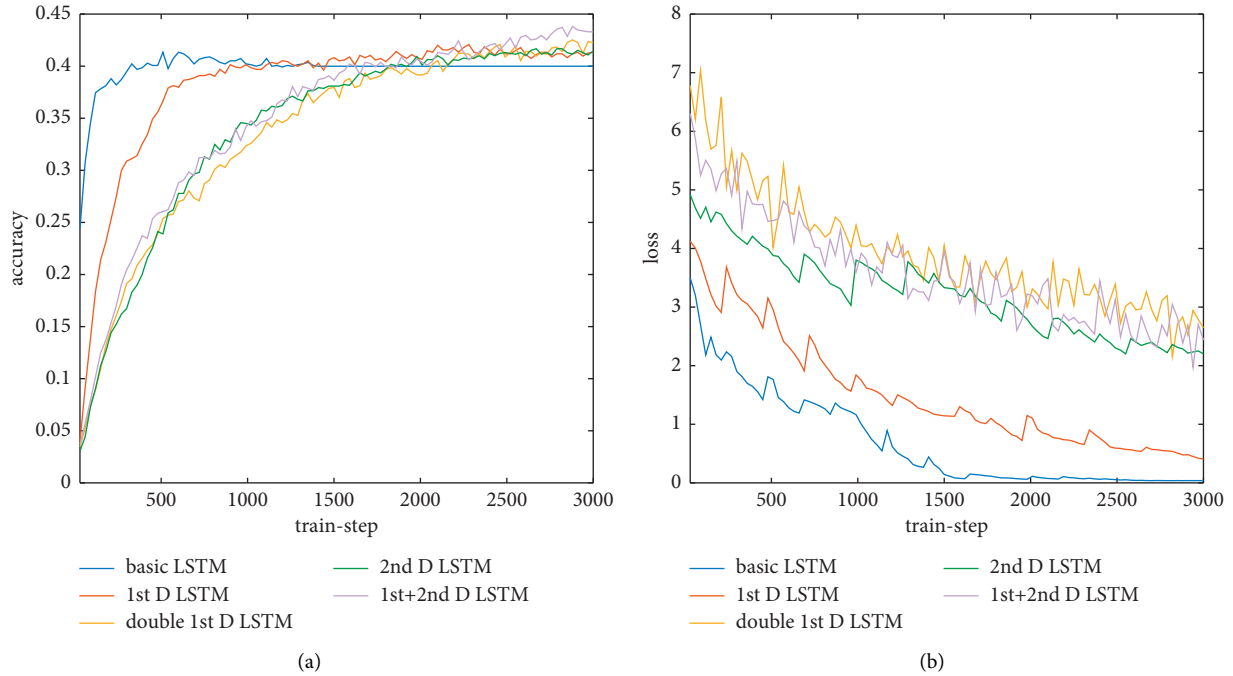


FIGURE 8: The comparison graph of accuracy and loss applied to five different LSMT units on the HMDB-51 dataset. (a) Accuracy and (b) Loss.

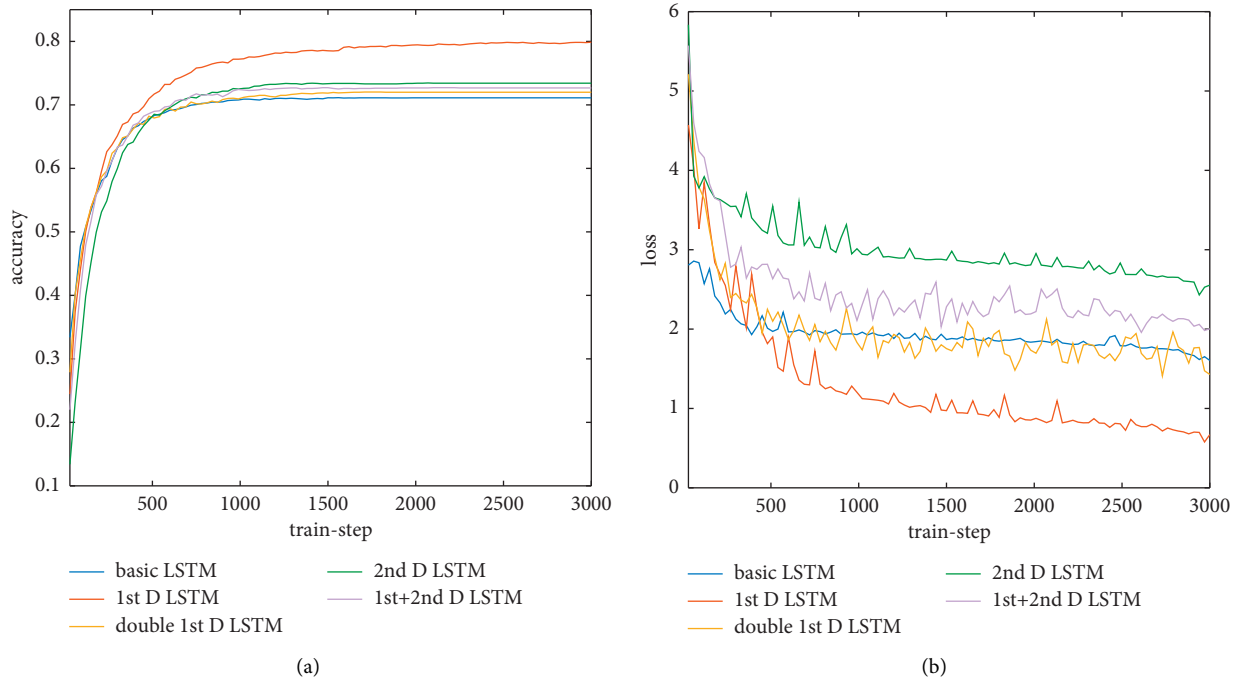


FIGURE 9: The comparison graph of accuracy and loss applied to five different LSMT units on the UCF-101 dataset. (a) Accuracy and (b) Loss.

work better. For the dataset with complex scenes and many other action interferences, the second-order input differential LSTM unit might work better. The four input differential LSTM units proposed in this article need to be studied further regarding loss functions. Using different optimizers or redefining loss functions may be the approach required to achieve an optimal model.

At the same time, we noticed that, in Figures 8(b), 9(b), 10(b), all D LSTMs' loss function fluctuates; we thought that, according to control theory, the differential elements easily introduce high-frequency measurement noise, which made all D LSTMs' loss more volatile than classical LSTM and 2nd D LSTMs' loss more volatile than 1st D LSTMs' loss, which are the shortcomings of all D LSTMs.

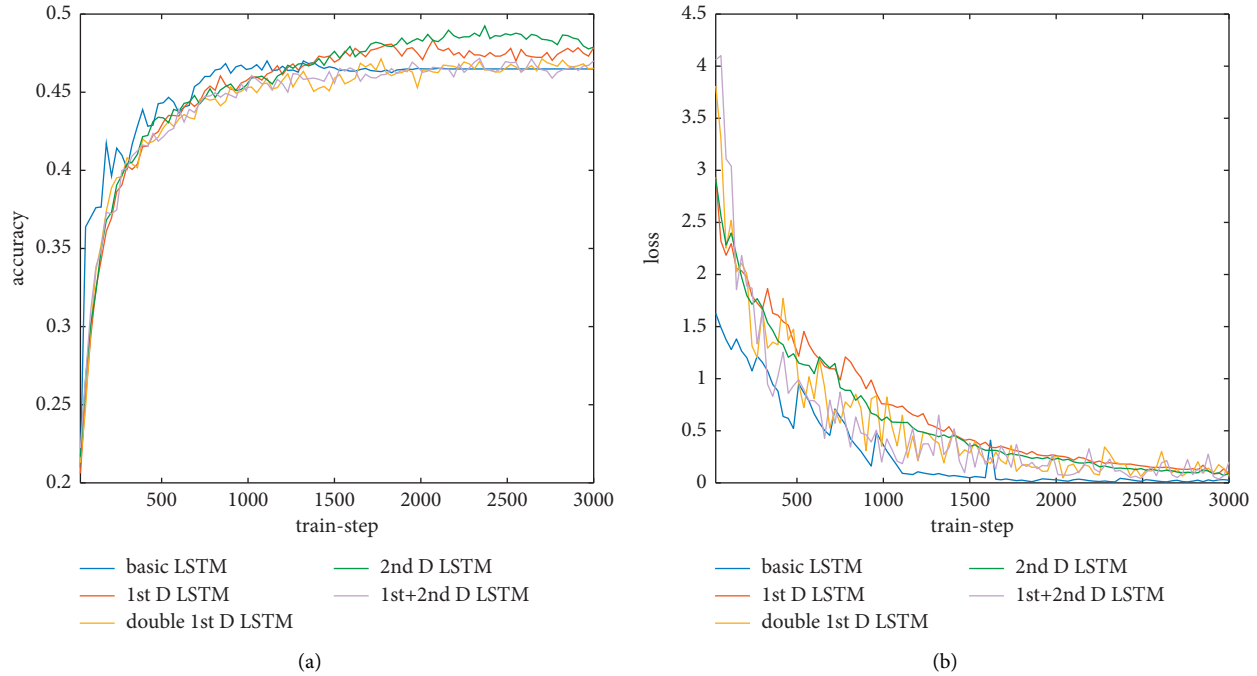


FIGURE 10: The comparison graph of accuracy and loss applied to five different LSMT units on the Hollywood2 dataset. (a) Accuracy and (b) Loss.

TABLE 1: The accuracy and standard deviation of different LSTM units on the dataset.

	HMDB-51	UCF-101	Hollywood2
Basic D Lstm	39.99% ± 1.31%	71.15% ± 0.37%	46.49% ± 0.68%
1st D Lstm	41.34% ± 0.83%	79.88% ± 0.38%	47.85% ± 0.99%
2nd D Lstm	41.44% ± 0.91%	73.42% ± 0.43%	47.89% ± 1.23%
Double 1st D Lstm	42.24% ± 1.05%	71.99% ± 0.61%	46.54% ± 0.74%
1st + 2nd D Lstm	43.30% ± 1.46%	72.67% ± 0.87%	47.00% ± 1.05%

4.4. Accuracy Comparison of Deep Learning Action Recognition Algorithms. In order to further verify the four input differential LSTM units proposed in Section 3, the improved differential LSTM units are compared with other deep learning algorithms. And experiments are carried out on UCF-101 and HMDB-51 datasets commonly used in deep learning. Table 2 is the result of a comparison experiment, and the bolded data are the highest. From Table 2, we can see that LRCN combined with 1st D LSTM is the best, and LRCN combined with 1st + 2nd D LSTM is the best.

In this section, the accuracy of two-stream convolutional network, LRCN network with attention mechanism, and LRCN network with BiLSTM is compared with the accuracy of the improved differential LSTM unit. Through experiments, it is found that the accuracy of the 1st D LSTM unit is the highest on the UCF-101 dataset and that of the 1st + 2nd D LSTM unit on the HMDB-51 dataset is the highest. The 1st D LSTM unit can better deal with features with short completion time and large category gap. The video in the UCF-101 dataset only has label actions, and there are great differences among 101 types of actions. The HMDB-51 dataset has more irrelevant actions than UCF-101 dataset. The 1st + 2nd D LSTM unit can handle both long- and short-time sequence features at the same time, so it can deal with noise actions better.

TABLE 2: The accuracy comparison of various deep learning algorithms on UCF-101 and HMDB-51 datasets.

	UCF-101 (%)	HMDB-51 (%)
Two-stream convolutional network [37]	73.00	40.50
Basic LSTM	71.15	39.99
1st D LSTM	79.88	41.34
2nd D LSTM	73.42	41.44
LRCN Double 1st D LSTM	71.99	42.24
1st + 2nd D LSTM	72.67	43.30
LSTM + attention	72.40	41.50
BiLSTM [31]	70.00	39.81

4.5. The Stability Experiments of Five LSTM Models. In our research, the networks of five different LSTM units were tested on three datasets, repeated three times for each dataset. The average accuracy and standard deviation of the final stable results were calculated. The standard deviation can reflect the accuracy dispersion degree of the LSTM unit on the corresponding dataset. Figure 11 and Table 1 show that each LSTM unit’s standard deviations applied to different datasets are small; therefore, in terms of stability, the four different input differential LSTM units proposed in this

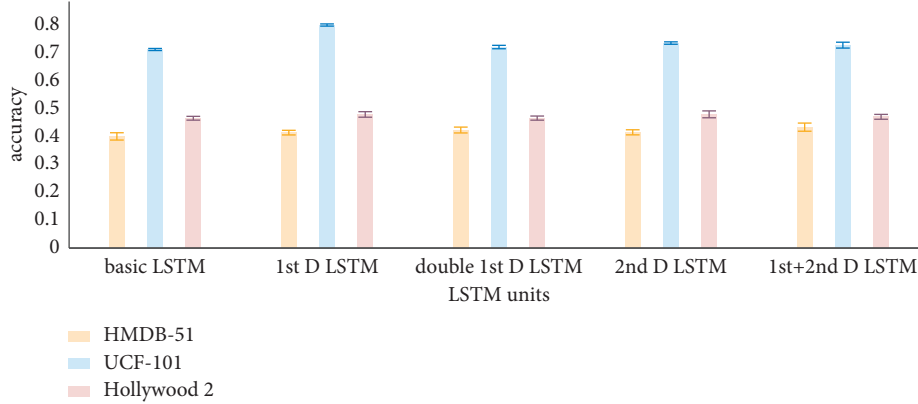


FIGURE 11: The comparison chart of accuracy and standard deviation of five different LSMT units.

TABLE 3: This is the frames per second (FPS) of different LSTM units trained on different datasets.

	HMDB-51	UCF-101	Hollywood2
Basic Lstm	61	64	32
1st D Lstm	36	36	25
2nd D Lstm	35	36	25
Double 1st D Lstm	18	20	11
1st + 2nd D Lstm	18	18	11

article have good stability in various datasets. Compared to the classic LSTM without input differential features link, the stability is not much different.

4.6. Algorithms Execution Time Experiments. The frames per second (FPS) evaluation index is a definition in the image field. Image detection and recognition generally refer to the number of images that can be processed in one second. In this experiment, FPS refers to the number of video frames that can be processed in one second.

Table 3 shows that in the process of training data, on the whole, the original LSTM unit processes more image frames in one second. As the amount of data processed by the network doubles, the input differential LSTM units used in the HMDB-51 and UCF-101 datasets show slower data processing. However, in the Hollywood2 dataset, it is equivalent to the original LSTM unit. The speed of processing data in different LSTM units may be affected by the video content's complexity. In terms of training time, compared with the classical LSTM algorithm without input differential features, the four methods proposed in this article are all inferior.

When the trained model parameters were used for recognition on different datasets, the original LSTM unit's recognition speed and the four LSTM units proposed in this article on a limited number of datasets are similar, which is roughly around 180 frames per second. However, overall, the original LSTM unit's recognition speed is 4 to 7 frames per second faster than the proposed four LSTM units proposed in this article.

5. Conclusion and Prospect

Human action recognition has more application requirements today and has received significant attention from researchers in related fields [38]. In this study, we combined the differentiation idea in PID control with the LSTM unit in the deep learning network and proposed four kinds of LSTM units with input differentiation, which increases the influence of information difference in time series on action recognition. Compared with the complex hybrid models, the differential LSTM unit can maintain the simplicity of the network structure and improve the recognition accuracy, so that it can be better applied to the real use scene. Due to the different habits and speeds of different characters in the dataset, the input differential LSTM units proposed in this article can pay attention to body movement speed to increase the characteristic information of actions in the time series. The experiments prove that the four different LSTM units proposed in this article have different degrees of improvement in action recognition accuracy compared with the original LSTM units. According to the video's length and the video's actions, different differential units have different performances in each dataset. Compared with other action recognition algorithms based on deep learning, the input differential LSTM unit has advantages in recognition accuracy, and it can be used in application scenarios such as attitude estimation and image caption generation.

In summary, since most of the human actions in current action recognition datasets involve short-length videos of human actions, the accuracy of the LSTM with the first-order input differential is higher in the action recognition

network. We used a simple network structure and network parameters to reflect the input differential LSTM unit's action recognition performance. Although the input differential LSTM unit intuitively reflects good accuracy, the loss function's processing is not detailed enough and may be optimized in future applications.

In general, the first-order/second-order input differential LSTM unit proposed in this article achieved good results in action recognition. Compared with the original LSTM unit, it has improved accuracy while maintaining stability, although its training speed is weak. The proposed unit can replace the original LSTM unit, can be flexibly applied in various network frameworks to realize different application scenarios, and has a good development prospect.

Data Availability

The code used to support the findings of this study are available from the corresponding author upon request. The data are from the open dataset of HMDB-51 (UCF-101" title="https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/), UCF-101 (https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/), UCF-101 (http://www.crcv.ucf.edu/data/UCF101.php), and Hollywood2 (http://www.di.ens.fr/~laptev/actions/hollywood2/).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors drafted the manuscript, and read and approved the final manuscript.

References

- [1] E. A. Suma, D. M. Krum, B. Lange, S. Koenig, S. Rizzo, and A. Bolas, "Adapting user interfaces for gestural interaction with the flexible action and articulated skeleton toolkit," *Computers & Graphics*, vol. 37, no. 3, pp. 193–201, 2013.
- [2] C. Chen, B. Liu, S. Wan, P. Qiao, and P. Pei, "An edge Traffic flow detection Scheme based on deep learning in an intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1840–1852, 2021.
- [3] X. Gejing and L. Yang, "Research on the impact of Internet evolution on accounting information system based on data Mining," *Journal of Physics: Conference Series*, vol. 1345, no. 5, Article ID 052055, 2019.
- [4] C. Chen, Q. Hui, W. Xie, S. Wan, S. Zhou, and Y. Pei, "Convolutional Neural Networks for forecasting flood process in Internet-of-Things enabled smart city," *Computer Networks*, vol. 186, Article ID 107744, 2021.
- [5] X. Yang and Y. Tian, "Action recognition using super Sparse coding vector with spatio-temporal Awareness," in *Proceedings of the European Conference on Computer Vision*, pp. 727–741, Switzerland, September 2014.
- [6] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with Stacked Fisher vectors," in *Proceedings of the European Conference on Computer Vision*, pp. 581–595, Springer, Cham, September 2014.
- [7] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Computer Vision and Image Understanding*, vol. 150, pp. 109–125, 2016.
- [8] R. Arandjelovic and A. Zisserman, "All about VLAD," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1578–1585, IEEE, Portland, OR, USA, June 2013.
- [9] I. C. Duta, B. Ionescu, K. Aizawa, and N. Sebe, "Spatio-temporal VLAD encoding for human action recognition in videos," in *Proceedings of the International Conference on Multimedia Modeling*, pp. 365–378, Reykjavik, Iceland, January 2017.
- [10] H. Zhu, C. Zhu, and Z. Xu, "Research advances on human activity recognition datasets," *Acta Automatica Sinica*, vol. 44, pp. 978–1004, 2018.
- [11] Q. Wang and K. Chen, "Multi-label zero-shot human action recognition via joint latent ranking embedding," *Neural Networks*, vol. 122, pp. 1–23, 2020.
- [12] M. Xia, W. Song, X. Sun, J. Liu, T. Ye, and Y. Xu, "Weighted Densely connected convolutional networks for Reinforcement learning," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 04, Article ID 2052001, 2020.
- [13] M. Xia, W. A. Liu, K. Wang, X. Zhang, and Y. Xu, "Non-intrusive load disaggregation based on deep dilated residual network," *Electric Power Systems Research*, vol. 170, pp. 277–285, 2019.
- [14] M. Xia, J. Qian, X. Zhang, J. Liu, and Y. Xu, "River Segmentation based on Separable attention residual network," *Journal of Applied Remote Sensing*, vol. 14, no. 03, p. 1, 2019.
- [15] M. Xia, X. Zhang, W. a. Liu, L. Weng, and Y. Xu, "Multi-stage feature Constraints learning for Age estimation," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2417–2428, 2020.
- [16] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: deep networks for video classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4694–4702, IEEE, Boston, MA, USA, June 2015.
- [17] W. Du, Y. Wang, and Y. Qiao, "Rpan: an end-to-end recurrent pose-attention network for action recognition in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3725–3734, IEEE, Venice, Italy, October 2017.
- [18] X. Long, C. Gan, G. De Melo et al., "Multimodal keyless attention fusion for video classification," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, USA, February 2018.
- [19] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proceedings of the Thirty-first AAAI conference on artificial intelligence*, San Francisco California USA, February 2017.
- [20] J. Tang, X. Shu, R. Yan, and L. Zhang, "Coherence Constrained graph LSTM for group Activity recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 636–647, 2022.
- [21] X. Shu, J. Tang, G.-J. Qi, W. Liu, and J. Yang, "Hierarchical long short-term Concurrent memory for human interaction recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 1110–1118, 2021.
- [22] X. Shu, L. Zhang, G.-J. Qi, W. Liu, and J. Tang, "Spatio-temporal Co-attention recurrent neural networks for human-

- skeleton motion prediction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3300–3315, 2022.
- [23] K. Hu, Y. Ding, J. Jin, L. Weng, and M. Xia, “Skeleton motion recognition based on multi-scale deep spatio-temporal features,” *Applied Sciences*, vol. 12, no. 3, Article ID 1028, 2022.
- [24] K. Hu, F. Zheng, L. Weng, Y. Ding, and J. Jin, “Action recognition algorithm of spatio-temporal differential LSTM based on feature enhancement,” *Applied Sciences*, vol. 11, no. 17, p. 7876, 2021.
- [25] J. Donahue, L. Anne Hendricks, S. Guadarrama et al., “Long-term recurrent convolutional networks for visual recognition and Description,” in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2625–2634, Boston, MA, USA, June 2015.
- [26] M. X. Jiang, C. Deng, Z. G. Pan, L. F. Wang, and X. Sun, “Multiobject Tracking in videos based on LSTM and deep Reinforcement learning,” *Complexity*, vol. 2018, Article ID 4695890, 12 pages, 2018.
- [27] M. Xia, W. a. Liu, K. Wang, W. Song, C. Chen, and Y. Li, “Non-intrusive load disaggregation based on composite deep long short-term memory network,” *Expert Systems with Applications*, vol. 160, Article ID 113669, 2020.
- [28] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, “A CNN-LSTM-Based model to Forecast Stock Prices,” *Complexity*, vol. 2020, Article ID 6622927, 10 pages, 2020.
- [29] H. Wang, y. Luo, W. An, Q. Sun, J. Xu, and L. Zhang, “PID controller-based stochastic optimization acceleration for deep neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5079–5091, 2020.
- [30] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local SVM approach,” in *Proceedings of the 17th International Conference on Pattern Recognition*, pp. 32–36, Cambridge, UK, August 2004.
- [31] B. Moshé, G. Lena, and S. Eli, “Actions as Space-time Shapes,” in *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV’05)*, pp. 1395–1402, Beijing, China, October 2005.
- [32] M. Marszalek, I. Laptev, and C. Schmid, “Actions in context,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2929–2936, Miami, USA, June 2009.
- [33] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: a dataset of 101 human actions classes from videos in the wild,” 2012, <https://arxiv.org/abs/1212.0402>.
- [34] H. Kuehne, H. Jhuang, and E. Garrot, “HMDB: a large video database for human motion recognition,” in *Proceedings of the International Conference on Computer Vision*, pp. 2556–2563, Barcelona, Spain, November 2011.
- [35] J. Donahue, L. A. Hendricks, M. Rohrbach et al., “Long-term recurrent convolutional networks for visual recognition and Description,” in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 677–691, IEEE, 2017.
- [36] Q. He, “Video content recognition technology research based on deep learning,” Master Thesis, University of Electronic Science and Technology of China, Xian, China, 2017.
- [37] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” 2014, <https://arxiv.org/abs/1406.2199>.
- [38] X. Yu, Z. Zhang, L. Wu et al., “Deep ensemble learning for human action recognition in still images,” *Complexity*, vol. 2020, Article ID 9428612, 23 pages, 2020.

Research Article

Efficient Localization of Multitype Barcodes in High-Resolution Images

Jinwang Yi  and Yuanbiao Xiao 

Xiamen University of Technology, Xiamen 361024, China

Correspondence should be addressed to Yuanbiao Xiao; 1922031040@s.xmut.edu.cn

Received 26 November 2021; Revised 21 February 2022; Accepted 7 March 2022; Published 30 March 2022

Academic Editor: Nouman Ali

Copyright © 2022 Jinwang Yi and Yuanbiao Xiao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Barcode positioning technology is one of the important components of barcode technology. However, most of the current algorithms are only applicable to a single type of barcode positioning or limited to low-resolution images due to a large amount of calculation, and thereby it is still a challenge to locate multitype barcodes in high-resolution images. In response to the above problems, this paper proposes a reliable multitype barcode localization method for multibarcode localization in high-resolution images where one-dimensional (1D) barcodes, two-dimensional (2D) barcodes, or multitype barcodes are present simultaneously. The method consists of three main steps: first, extracting multiple types of barcode features through a joint edge detection algorithm; next, marking target barcode regions with a bidirectional contour labeling method; and finally, extracting barcode regions by an improved affine transformation. The experimental results show that, in terms of localization accuracy, the proposed method has a better accuracy of 97.83% than existing algorithms in low-resolution images and can locate multitype barcodes in high-resolution images with an accuracy rate of 98.04%. Besides, in terms of time cost, the proposed method effectively reduces the time cost by 50% and improves the barcode localization efficiency.

1. Introduction

Barcode positioning technology is widely used in many industries and fields, such as warehousing, library management, health care, industrial production, and express logistics [1, 2]. However, its efficiency is insufficient for multiple types of barcode localization in high-resolution images because most of the existing methods are only applicable to a single type of barcode localization and the localization process uses complex algorithms, which lead to high time costs.

Produced by Gallo and Manduchi, a simple and fast algorithm is used to calculate the vertical and horizontal gradients of each pixel and perform global binarization for the segmentation of the 1D barcode region [3], which, however, is unsuitable for locating slanted barcodes. Yun and Kim [4] proposed a new 1D barcode localization method that detects barcode features based on the similar structure of entropy and edge orientation in 1D barcodes,

thus realizing the localization of arbitrarily tilted 1D barcodes. Literature [5] exploited the stacked edge features inside 2D barcodes to develop a histogram-based 2D barcode detection method, but it is vulnerable to the interference of dense text. Rincon et al. [6] locate quick response (QR) codes in complex contexts by detecting the structural features of three rectangles in QR codes, effectively addressing the impact of dense text. The above algorithms have high accuracy for single type barcode positioning, but none of them consider the localization of multiple types of barcodes in one image. D. T. Lin and C. L. Lin [7] proposed a multitype barcode localization framework, which utilizes an adaptive thresholding method to extract black and white linear features inside barcodes to achieve multitype barcode localization. However, this method only performs multitype barcode positioning tests in low-resolution images with a resolution of 300×400 pixels. Katona and Nyúl [8] performed edge feature detection of barcodes by improving the Sobel and Feldman algorithm [9], which enables the

framework to perform edge detection of multiple types of barcodes in barcode images with a resolution of 720×480 pixels, compared to the previous method, but its algorithm complexity is high. These algorithms are limited to feature detection of a single type of barcode, or their algorithms have high complexity and face the disadvantages of high time cost and computational inefficiency in high-resolution images.

In the image processing-based barcode localization, barcode region marking is also of great interest. D. T. Lin and C. L. Lin [7] developed a two-channel connected domain analysis algorithm, which can improve the efficiency of barcode region labeling to replace the traditional method. Another important contribution of this algorithm is the ability to label multiple types of barcode regions. Chang et al. [10] studied a contour-detection-based marking method that has a wide range of applications by using the contour-detection technique to track and label the external and internal contours of each target location. By improving this algorithm, Chen et al. [11] proposed a two-stage method for marking the external contours of barcode regions, namely, orientation-based region contour tracking and contour-based region marking. This algorithm can not only mark multiple types of barcode regions but also do it more efficiently because global marking is not required. Liu et al. [12] successfully applied the above algorithm to multiple 2D barcode regions, and the experimental results also showed that the contour-detection-based barcode region marking method is better than the connected domain analysis method, but its marking time is affected by the resolution size of a barcode image.

Effective extraction of barcode areas is a prerequisite for obtaining barcode information. Lin and Fuh [13] used the position detection patterns of QR codes to extract QR codes that could recognize different types of QR code images. In the literature [14], a QR code extraction algorithm based on the Hough transform [15] is proposed. The algorithm achieves QR code extraction by forming a bitmap with edge detection and Hough space and preserving the intersection of vertical line segments. Chen [16] provided an algorithm to effectively extract the tilted 1D barcode region. It first obtains the 1D barcode edge information in the image by Canny edge detection [17], then detects the straight lines inside the barcode using the Hough transform, obtains the tilt angle of the 1D barcode region, and finally rotates the extracted barcode to a horizontal state. The above barcode region extraction algorithm can effectively extract tilted single-type barcodes, but it does not consider the extraction of multitype barcodes, and the process requires high time cost due to complex preprocessing, especially in high-resolution images.

The development of deep learning and machine learning has inspired many new barcode positioning algorithms. Zamberletti et al. [18] introduced a machine learning-based barcode detection method in which the algorithm effectively recognizes 1D barcodes. Katona et al. [19] used a vector machine to learn the characteristics of QR codes and use them to extract QR barcodes. Unfortunately, none of the above algorithms take into account multitype barcodes. Ren et al. [20] provided a fast region-based convolutional neural

network (CNN) model that can quickly detect and classify multiple targets. Based on this model, Tian et al. [21] developed a barcode detector called BAN, which can quickly detect multiple types of barcodes in barcode images. Unfortunately, it has only been tested in images with resolution of 1024×768 pixels. Jia et al. [22] developed a multitype barcode recognition model based on an improved CNN and trained it using images with resolution of 1920×1080 pixels. Additionally, another focus of this model is that the accurate position and distorted barcode shape can be determined and corrected. The performance of these localization algorithms is tied to the quality and quantity of training data and the training time of the model.

Based on the above research, the problems identified are high computational cost, restricted to low-resolution images, limited to barcode types, etc. In order to address these problems, a reliable localization method for multitype barcodes in high-resolution images is proposed. The main contributions of these works can be summarized as follows:

- (1) Effectively reduces time costs and further improves barcode positioning accuracy.
- (2) The proposed algorithm can perform barcode localization in high-resolution images and has excellent performance.
- (3) Efficiently extracts arbitrarily tilted barcodes and provides good horizontal barcodes for decoders.
- (4) The proposed method can locate 1D barcodes, 2D barcodes, and multitype barcodes.

2. Methods

Since the majority of current barcode localization algorithms are confined to a single kind of barcode and the high computational complexity is limited to low-resolution pictures, there is still room for improvement in their localization performance and efficiency. Aiming at the problems mentioned above, this paper proposes a barcode positioning method for high-resolution images. The method includes three main modules: feature extraction of the barcode edge, labeling of the barcode region, and extraction of the barcode region. The barcode positioning method is shown in Figure 1.

2.1. Feature Extraction of the Barcode Edge. Effective extraction of edge features in the barcode region is a prerequisite for barcode localization. Nevertheless, existing techniques are only appropriate for extracting a single type of barcode feature or are limited to low-resolution images due to the high complexity of the algorithm. Therefore, this work proposes a simple and quick algorithm, namely joint edge detection.

2.1.1. Grayscale Conversion. The input image is a high-resolution image with three red, green, and blue (RGB) channels. It not only fails to reflect the features of the barcode region but also leads to data redundancy. To reduce data redundancy and improve calculation efficiency, images

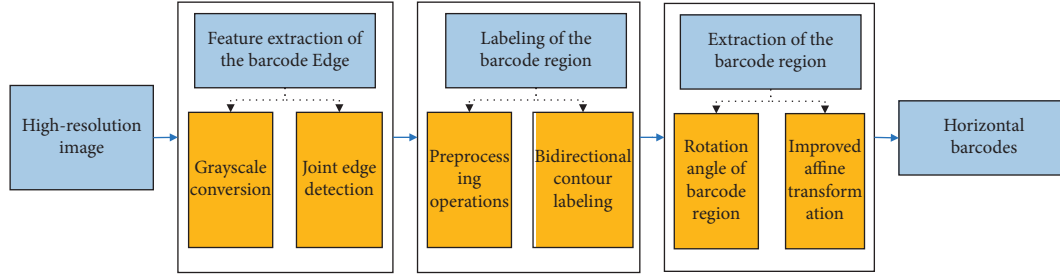


FIGURE 1: The barcode positioning method.

are first converted to single-channel grayscale images by the following equation:

$$G(x, y) = 0.299 \times r(x, y) + 0.587 \times g(x, y) + 0.114 \times b(x, y), \quad (1)$$

where r , g , and b represent the pixel value of red, green and blue channels of the image at the (x, y) position, respectively. Through this operation, the high-resolution image is converted into a grayscale image, which effectively keeps the edge feature information of the barcode and reduces data redundancy.

2.1.2. Joint Edge Detection. The joint edge detection algorithm proposed in this paper mainly detects the edge features of barcodes in all directions through horizontal, vertical, and double diagonal directions. Detection directions are shown in Figure 2.

The suggested method is divided into three stages, as follows:

In the beginning, the horizontal and vertical gradients $I_x(n)$ and $I_y(n)$ for each pixel in the greyscale image G are computed in the horizontal and vertical directions. Then, they are combined in a nonlinear manner, and their absolute values are calculated. Finally, a horizontal-vertical edge gradient map $I_e(n)$ is generated by equation (2). This approach fully detects the barcode with horizontal (vertical) edge features and improves the robustness of the barcode horizontal (vertical) edge feature extraction direction.

$$I_e(n) = ||I_x(n)| - |I_y(n)||. \quad (2)$$

Then, the barcodes with horizontal (vertical) edge features are extracted in the gradient map $I_e(n)$, and all the extracted barcode regions D_i are removed by equation (3) in the grayscale image G , forming a new grayscale image G_{new} , where i is the number of extracted barcode regions.

$$G_{\text{new}}(x, y) = G(x, y) - D_i(x, y). \quad (3)$$

Finally, 45-degree and 135-degree gradients $I_{45}(n)$ and $I_{135}(n)$ for each pixel n in the new grayscale image G_{new} are computed in double diagonal directions. Then, they are combined in a nonlinear manner, and their absolute value is calculated. Finally, a double diagonal edge gradient map $I_d(n)$ is generated by equation (4). Through this step, the edge features of inclined barcodes and 2D barcodes can be

effectively detected, and the robustness of the extraction direction of the edge features of inclined barcodes and 2D barcodes can also be improved.

$$I_d(n) = ||I_{45}(n)| - |I_{135}(n)||. \quad (4)$$

The edge feature detection results of the high-resolution image through the joint edge detection algorithm are shown in Figure 3, where it can be seen from Figures 3(b) and 3(c) that the edge features of horizontal (vertical) barcodes, inclined barcodes, and 2D barcodes are effectively detected by this method. The higher the energy, the better the effect of barcode edge feature detection.

2.2. Labelling of the Barcode Region. The labeling of the barcode region is an important process for accurate barcode localization. The traditional connected-component marking approach requires that all pixels in the target region be marked. The efficiency is low, and the cost of marking time is heavily influenced by image resolution. Therefore, a new connected-component method has been developed to improve efficiency.

2.2.1. Preprocessing Operations. In order to reduce the influence of small background clutter and weak target areas in gradient maps $I_e(n)$ and $I_d(n)$, a block filter of size 35×35 over gradient maps $I_e(n)$ and $I_d(n)$ is used to reduce noise and improve the weak barcode region. The size of the filter was chosen based on the range of the resolution of the input images by our method. It is worth noting that block filtering can be implemented efficiently so that only few operations per pixel are required. In addition, the thresholds are determined by taking the adaptive thresholding approach of Otsu [23]. The binary images $B_e(n)$ and $B_d(n)$ are obtained by equation (5) after the thresholds T_e and T_d are obtained, and morphological techniques are utilized to fill in the gaps in the barcode feature region.

$$B_e(n) = \begin{cases} 1, & \text{if } I_e(n) \geq T_e, \\ 0, & \text{otherwise,} \end{cases} \text{ or } B_d(n) = \begin{cases} 1, & \text{if } I_d(n) \geq T_d, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

In this formula, $I_e(n)$ is the horizontal-vertical gradient map, $I_d(n)$ is the diagonal gradient map. The preprocessing results of $I_e(n)$ and $I_d(n)$ are shown in Figures 4(b) and 4(c), respectively.

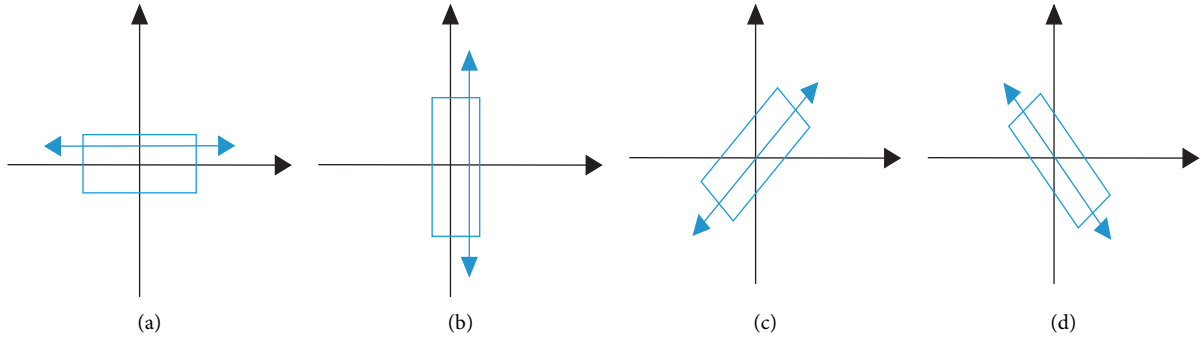


FIGURE 2: Detection directions: (a) horizontal, (b) vertical, (c) 45-degree, and (d) 135-degree.

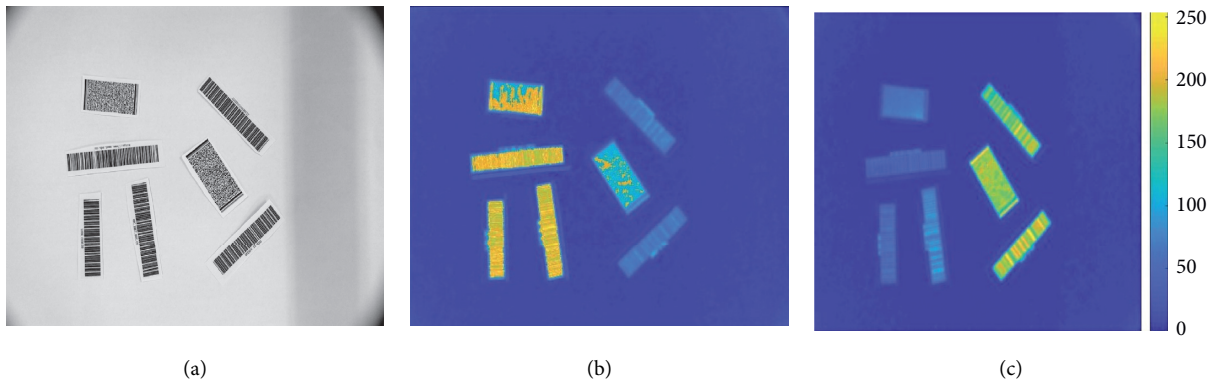


FIGURE 3: Edge feature detection results: (a) high-resolution image, (b) the corresponding energy for $I_e(n)$, and (c) the corresponding energy for $I_d(n)$.

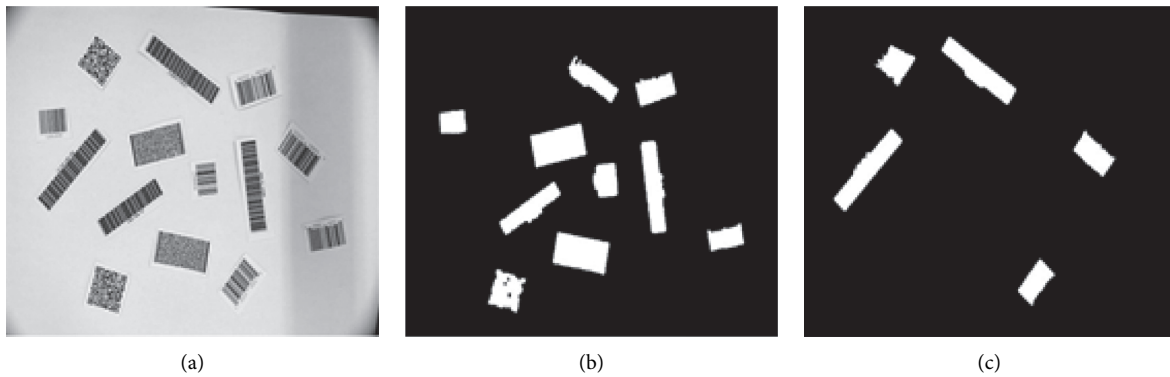


FIGURE 4: Preprocessing results: (a) high-resolution image, (b) preprocessing result of the horizontal-vertical gradient map, and (c) preprocessing result of the diagonal gradient map.

2.2.2. Bidirectional Contour Labelling. The traditional approach is abandoned in this section, and a bidirectional labeling method based on contour information is proposed. This method tracks and labels the outer contour of each barcode region clockwise and counterclockwise, as realized by parallel computing [24] on the CPU. The bidirectional contour labeling method consists of five steps used iteratively to examine all the pixels in the image.

In Step 1, the current pixel P_0 is the starting point of the outer contour of the new barcode region if it is a foreground

point and the prior pixel P_{-1} is a background point, as illustrated in Figure 5(a). After the contour beginning point is found, the beginning and middle points P_S and P_C can be set as $P_S = P_{-1}$ and $P_C = P_0$, respectively, for initialized contour tracking.

In Step 2, the next contour point P_N is tracked clockwise and marked, with P_S as the starting point of clockwise tracking and P_C as the center point of eight-neighborhoods, as shown in Figure 5(b).

In Step 3, the next contour point P_N is tracked counterclockwise and noted, with P_S as the starting point of

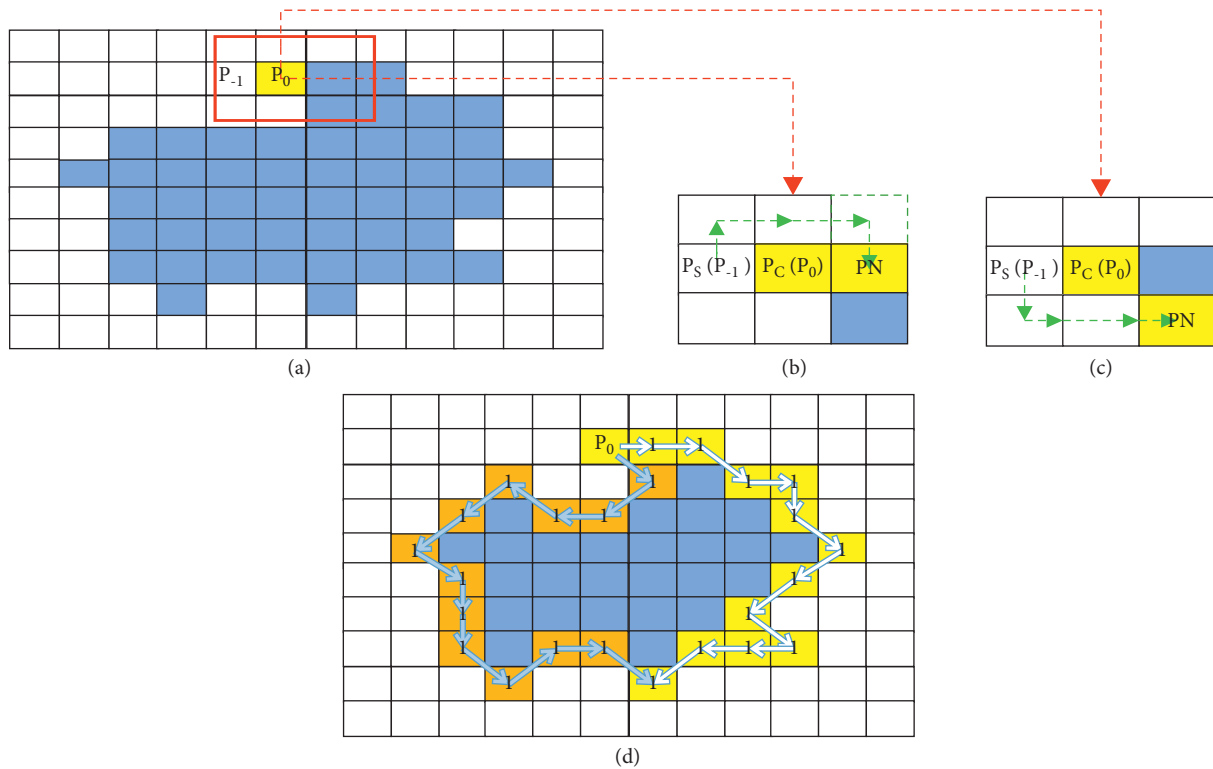


FIGURE 5: Bidirectional contour labelling process: (a) target region, (b) clockwise contour tracking, (c) counterclockwise contour tracking, and (d) marking results of the target region.

counterclockwise tracking and P_C as the center point of eight-neighborhoods, as presented in Figure 5(c).

In Step 4, the starting and center points P_S and P_C can be set as $P_S = P_C$ and $P_C = P_N$, respectively, for clockwise tracking (counterclockwise tracking) when a new contour point P_N is tracked clockwise (counterclockwise) and marked.

In Step 5, the parallel computing technique is adopted to achieve simultaneous contour tracking and marking both clockwise and counterclockwise. The method proceeds to determine the starting point of the next contour if clockwise and counterclockwise tracking to the pixel point P_N has been marked, indicating that the contour tracking and labeling of the current barcode region have been finished. Figure 5(d) shows the outcome of the bidirectional contour marking approach in the target region.

2.3. Extraction of the Barcode Region. In order to improve decoding efficiency, the marked barcode areas need to be extracted. Since the existing barcode extraction technology is restricted to a particular type of barcode extraction, a quick and simple extraction algorithm is proposed in this section. The procedure consists of two steps: calculating the rotation angle of the inclined barcode and rotating the inclined barcode region to a horizontal state.

2.3.1. Rotation Angle of Barcode Region. Based on the rectangular shape of the barcode region, a maximum outer rectangle is fitted to the marked barcode region, and the fitted barcode region is extracted from the binary image

$B(n)$. The rotation operation with the extracted barcode region instead of the entire image can effectively reduce the data volume and improve the computation speed.

The minimum outer rectangle is fitted to the extracted barcode region, and the tilt angle θ of the barcode region is calculated based on the fitted minimum outer rectangle. The rotation angle for rotating the tilted barcode area to a horizontal state is calculated by the following two steps:

In Step 1, based on the width W and the height H of the minimum outer rectangle, the width-to-height ratio R of the minimum outer rectangle is calculated by the following equation:

$$R = \frac{W}{H}. \quad (6)$$

In Step 2, the rotation angle β is determined by the relationship between the ratio R and the threshold values R_1 and R_2 , which are empirical values herein, as shown in the following equation:

$$\beta = \begin{cases} \theta, & R \geq R_1, \\ 90^\circ + \theta, & 0R \leq R_2, \quad \theta \in [-90^\circ, 0]. \\ \theta, & R_2 < R < R_1. \end{cases} \quad (7)$$

In this formula, R_1 and R_2 are set to 1.2 and 0.9, respectively, based on experience, and the tilt angle θ is the angle of the first side of the minimum outer rectangle rotating clockwise and touching the X -axis of the coordinate system. The above side is the width W of the smallest outer rectangle, and the adjacent side is the height H , as shown in Figure 6. This method fully guarantees that the long side of

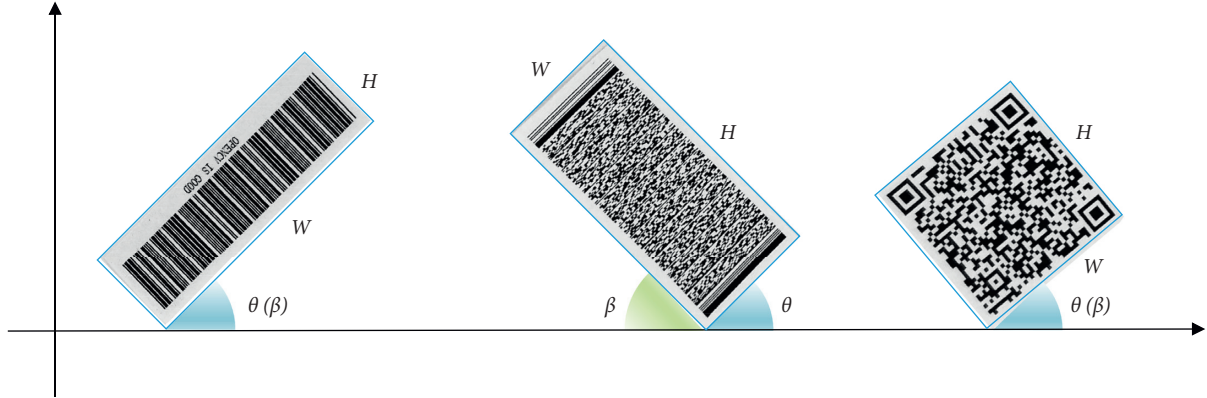


FIGURE 6: Schematic diagram of the width (W), height (H), tilt angle θ , and rotation angle β of the smallest outer rectangle.

the barcode is horizontal and the short side of the barcode is vertical after rotating β with any tilt barcode.

2.3.2. Improved Affine Transformation. The tilted barcode area is rotated to a horizontal state around the center point of the barcode area by the affine transformation, where the rotation matrix A and translation vector B required for the affine transformation are determined by the rotation angle β and the center of rotation. The rotation relationship is as follows:

$$\begin{aligned} \begin{bmatrix} x' \\ y' \end{bmatrix} &= A \begin{bmatrix} x \\ y \end{bmatrix} + B, \\ B &= \begin{bmatrix} (1 - \cos \beta) \times C_x - \sin \beta \times C_y \\ \sin \beta \times C_x + (1 - \cos \beta) \times C_y \end{bmatrix}. \end{aligned} \quad (8)$$

In this formula, $A = \begin{bmatrix} \cos \beta & \sin \beta \\ -\sin \beta & \cos \beta \end{bmatrix}$, C_x and C_y are the horizontal and vertical coordinates of the center point of the barcode area, (x, y) are the horizontal and vertical coordinates of the pixel point to be rotated, and (x', y') are the horizontal and vertical coordinates of the pixel point after rotation.

However, as the size and the center of the barcode area are changed after rotation, the traditional affine transformation for rotation will result in missing or incomplete edges of the barcode area. To address this shortcoming, the algorithm is improved in this paper to recalculate the barcode area size and the new translation vector after the barcode area rotation by the following equations:

$$\begin{cases} W' = W|\cos \beta| + H|\sin \beta|, \\ H' = W|\sin \beta| + H|\cos \beta|, \end{cases} \quad (9)$$

$$B = \begin{bmatrix} (1 - \cos \beta) \times C_x - \sin \beta \times C_y \\ \sin \beta \times C_x + (1 - \cos \beta) \times C_y \end{bmatrix} + \begin{bmatrix} \frac{W' - W}{2} \\ \frac{H' - H}{2} \end{bmatrix}, \quad (10)$$

where W and H are the width and the height of the barcode area before rotation and W' and H' are the width and the

height of the barcode area after rotation. The results of barcode region extraction are shown in Figure 7, where the red and green boxes in Figure 7(a) are the results of fitting the maximum outer rectangle and the minimum outer rectangle, respectively. The results show that the proposed algorithm can extract arbitrarily skewed multitype barcode regions in high-resolution images and rotate them to a horizontal state, providing good barcode images for decoders.

3. Results and Discussion

3.1. Experimental Environment and Datasets. In this paper, the proposed algorithm is simulated using the OpenCV framework and C++, and the barcode images are captured using a high-resolution camera. The barcode images are divided into three data sets as follows:

- (A) Dataset A contains 900 low-resolution images (600 images of a single type of barcode and 300 images of multiple types of barcodes), where the barcode image resolution is 1024×768 pixels, and each image has an average of 12 barcodes.
- (B) Dataset B contains 1000 high-resolution images (600 images of a single type of barcode and 400 images of multiple types of barcodes), where the barcode image resolution is 4208×3120 pixels, and each image has an average of 14 barcodes.
- (C) Dataset C contains 400 barcode images of multiple types, where the resolution and number of barcodes are shown in Table 1.

To verify the performance of the proposed algorithm, this paper compares the accuracy and time cost of barcode localization with those of the traditional localization algorithms.

3.2. Accuracy of Positioning. The accuracy of barcode positioning includes single type barcodes and multiple type barcodes in both low-resolution images and high-resolution images. The details are as follows:

First, the localization accuracy of the proposed algorithm for barcodes in low-resolution images is tested in Dataset A. Tables 2–4 show the accuracy of the proposed algorithm

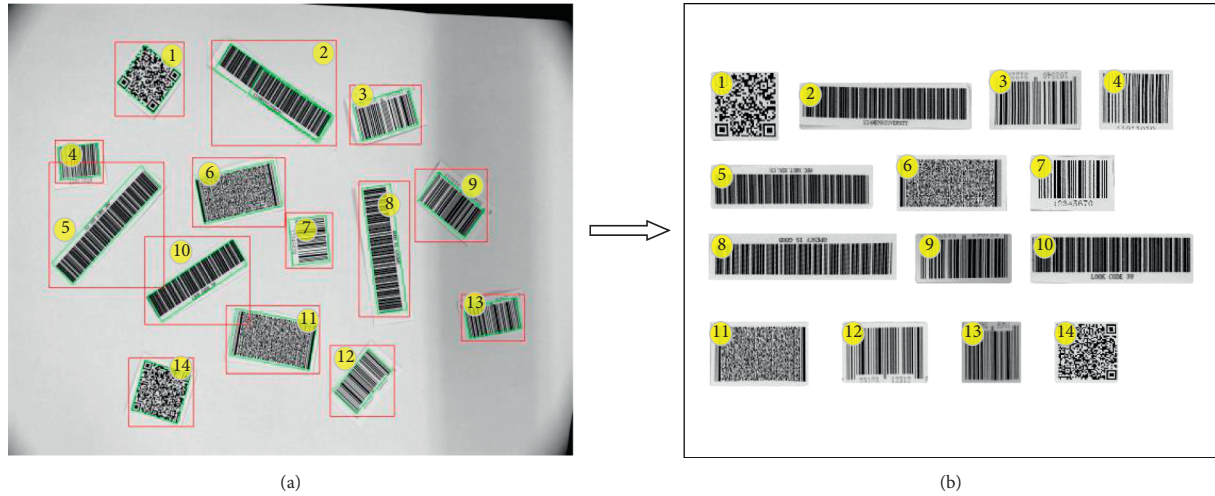


FIGURE 7: Extraction results: (a) fitting results of barcode area and (b) horizontal status barcode area.

TABLE 1: Time cost for locating multiple types of barcodes in low-resolution images.

Methods	Resolution	Number of barcodes	Time cost (ms)	Proposed (ms)
Chen et al. [11]	300 × 400	3–5	230	98
D. T. Lin and C. L. Lin [7]	640 × 480	3–5	256	165
Katona and Nyúl [8]	720 × 480	3–5	310	176
Katona et al. [19]	1024 × 768	3–5	550	233

TABLE 2: Localization accuracy of 1D barcodes in low-resolution images.

Barcode type	Chen et al. [11]	Katona and Nyúl [8] (%)	D. T. Lin and C. L. Lin [7] (%)	Proposed (%)
Code 128	91.07	84.8	94.64	96.30
Code 39	96.43	90.5	100	97.41
UPC-A	90.7	95.4	95.35	99.24
EAN 13	94.55	99.02	98.18	99.36
EAN 18	90	95.9	92.43	98.05
Average	92.55	93.16	96.12	98.07

TABLE 3: Localization accuracy of 2D barcodes in low-resolution images.

Barcode type	Ren et al. [20]	Tian et al. [21] (%)	D. T. Lin and C. L. Lin [7] (%)	Proposed (%)
QR code	92.8	93.3	96.52	98.28
PDF 417	93	96	91.04	97.25
Data matrix	93.2	94.4	97.67	98.32
Average	93	94.57	95.08	97.95

TABLE 4: Localization accuracy of multitype barcodes in low-resolution images.

Methods	Katona and Nyúl [8]	Chen et al. [11] (%)	Tian et al. [21] (%)	D. T. Lin and C. L. Lin [7]	Proposed (%)
Accuracy	92.97	94.49	94.66	96.52	97.82

for 1D, 2D, and multitype barcode localization and compare the experimental results of such a method with those of D. T. Lin and C. L. Lin, Katona and Nyúl, Chen, Ren et al. and Tian et al. [7, 8, 11, 20, 21], respectively. The experimental results show that the proposed algorithm has an average accuracy of

98.07% in 1D barcode localization, 97.95% in 2D barcode localization, and 97.82% in multitype barcode localization. From the data in the tables, it can be seen that the proposed algorithm can effectively locate both single type and multitype barcodes in low-resolution barcode images and that its

TABLE 5: Localization accuracy of 1D barcodes in high-resolution images.

Barcode type	Number of barcodes	Number of positioning	Locate rate (%)
Code 128	884	872	98.64
Code 39	735	725	98.64
UPC-A	916	910	99.34
EAN 13	940	931	98.04
EAN 18	1230	1223	99.42
Total	4705	4661	99.06

TABLE 6: Localization accuracy of 2D barcodes in high-resolution images.

Barcode type	Number of barcodes	Number of positioning	Locate rate (%)
QR code	873	871	99.77
PDF 417	690	686	99.42
Dara matrix	772	771	99.87
Total	2335	2328	99.70

TABLE 7: Localization accuracy of multitype barcodes in high-resolution images.

Barcode type	Number of barcodes	Number of positioning	Locate rate (%)
Multi-type	3056	2996	98.04

localization performance is better than that of the existing barcode localization algorithms.

Second, the localization accuracy of the proposed algorithm for barcodes in high-resolution images is tested in Dataset B. Tables 5–7 show the accuracy of the proposed algorithm for 1D, 2D, and multitype barcode localization. From the data in the table, it can be seen that the proposed algorithm can effectively locate both single type and multiple type barcodes in high-resolution barcode images with high accuracy. Specifically, the average positioning accuracy of 1D, 2D, and multitype barcodes was 99.06%, 99.70%, and 98.04%, respectively.

3.3. Time Cost of Positioning. Effectively reducing the time cost of barcode positioning is another focus of the proposed algorithm. The time cost of barcode positioning includes multiple types of barcodes in low-resolution images and high-resolution images, respectively. The details are as follows:

Firstly, the time cost of the proposed algorithm for multiple types of barcode localization in low-resolution images is tested in Dataset C, where the barcode image resolution and the number of barcodes correspond to those provided by Lin, Katona, and Chen. As shown in Table 1, the time costs of the proposed algorithm are 98 ms, 165 ms, 176 ms, and 233 ms, respectively, which are lower than the time costs of the conventional localization algorithms, i.e., 230 ms, 256 ms, 310 ms, and 550 ms, respectively. It can be seen from the data in the table that the proposed algorithm can effectively reduce the time cost of multitype barcode positioning in low-resolution images by 36%–57% compared to the traditional method.

Secondly, the multitype barcode images in Dataset B are converted to five different resolutions, which are 526×390 pixels, 1052×780 pixels, 2104×1560 pixels, 3208×2120 pixels, and 4208×3120 pixels, respectively. Meanwhile, the time cost

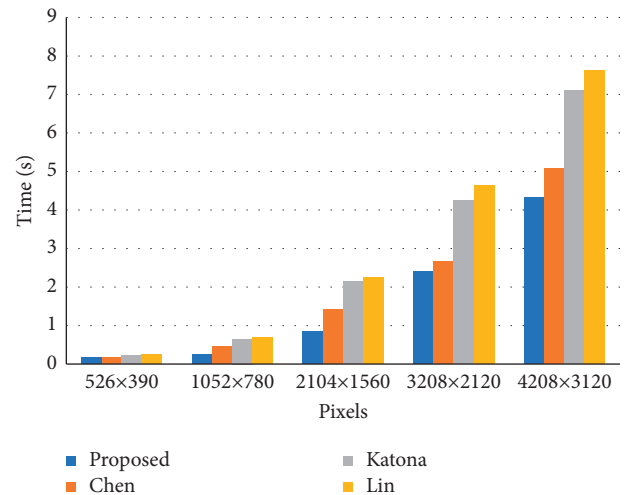


FIGURE 8: Time cost of barcode positioning at different resolutions.

of the proposed algorithm to locate the multitype barcodes in the high-resolution images is being tested. The test results are shown in Figure 8. It is clear that the time cost of barcode positioning is greatly influenced by the image resolution and the number of barcodes to be positioned, but the increase in the time cost of the proposed algorithm is lower than that of existing methods. On the other hand, the time cost of existing algorithms is higher than that of the proposed algorithm at different resolutions. The experimental results show that the proposed algorithm can effectively reduce the time cost of barcode positioning and improve the efficiency of barcode positioning, especially in high-resolution images.

Through the above experimental analysis, the accuracy of the barcode localization algorithm proposed in this paper in locating 1D barcodes, 2D barcodes, and multitype barcodes in

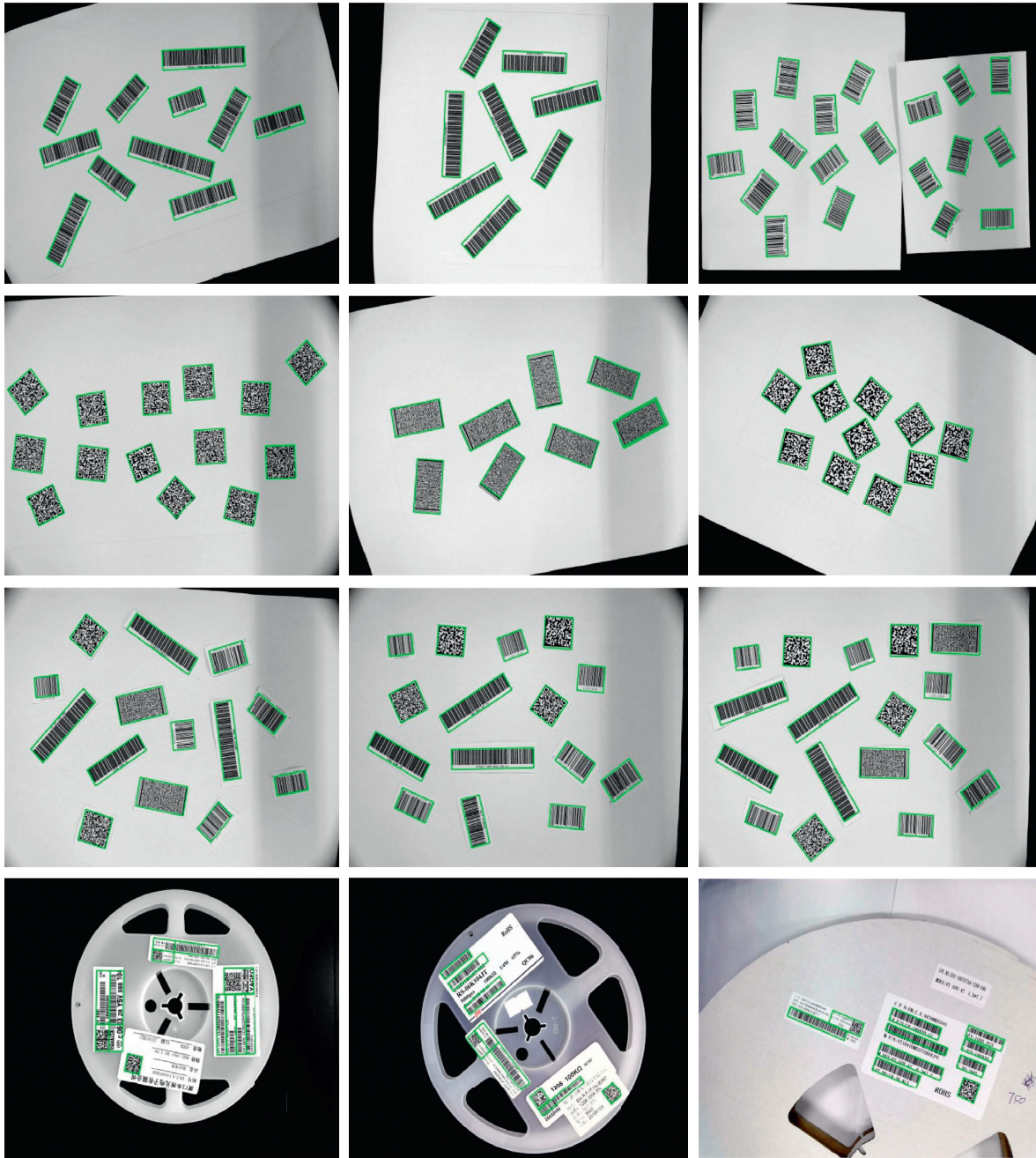


FIGURE 9: Barcode positioning effect in high-resolution images.

low-resolution images is higher than that of existing algorithms. Secondly, it can effectively locate single-type barcodes and multitype barcodes in high-resolution images, and some of the localization effects are shown in Figure 9. Finally, the time cost of the proposed algorithm is lower than that of traditional algorithms for multitype barcode localization in both low-resolution images and high-resolution images, which can effectively reduce time cost and improve localization efficiency.

4. Conclusions

In this study, a new reliable localization method is proposed to extract arbitrary tilting real barcodes from one image,

especially when 1D, 2D, or multitype barcodes are present in high-resolution images. The proposed method mainly comprises three steps: first, detection of barcode edge features; second, marking of barcode regions; and finally, extraction of barcode regions. In the proposed method, the edge features of arbitrary barcodes are quickly extracted by a joint edge detection algorithm, and potential barcode regions are efficiently marked by using a bidirectional contour marking method. Finally, the problem of tilted barcode extraction is solved by an improved affine transformation. The experimental results show that the proposed method can effectively locate multitype barcodes with higher accuracy in both low-resolution and high-resolution images. On the

other hand, the proposed method can effectively reduce the time cost of barcode localization and further improve its efficiency, especially for multitype barcodes in high-resolution images.

Data Availability

The datasets used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments



This work was supported by the National Natural Science Foundation of China (no. 61701422).

References

- [1] S. M. Youssef and R. M. Salem, "Automated barcode recognition for smart identification and inspection automation," *Expert Systems with Applications*, vol. 33, no. 4, pp. 968–977, 2007.
- [2] D. H. Chen and H. Liu, *Barcode Technology and Applications*, pp. 1–10, Chemical Industry Press, Beijing, China, 2015.
- [3] O. Gallo and R. Manduchi, "Reading 1D barcodes with mobile phones using deformable templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1834–1843, 2011.
- [4] I. Y. Yun and J. Kim, "Vision-based 1D barcode localization method for scale and rotation invariant," in *Proceedings of the of the 2017 IEEE Region 10 Conference (TENCON)*, pp. 2204–2208, Penang, Malaysia, November 2017.
- [5] K. C. A. FabijbDska, "Detection of QR-Codes in digital images based on histogram similarity," *Image Processing and Communications*, vol. 20, no. 2, pp. 41–48, 2015.
- [6] O. L. Rincon, O. Starostenko, A. A. Vicenteand, and J. C. Galan-Hernandez, "Binary large object-based approach for QR code detection in uncontrolled environments," *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 4613628, 15 pages, 2017.
- [7] D. T. Lin and C. L. Lin, "Automatic location for multi-symbology and multiple 1D and 2D barcodes," *Image Processing and Communications*, vol. 21, no. 6, pp. 663–668, 2013.
- [8] M. Katona and L. G. Nyúl, "Efficient 1D and 2D Barcode Detection Using Mathematical morphology," in *Proceedings of the Mathematical Morphology And its Applications To Signal And Image Processing*, pp. 464–475, Uppsala, Sweden, May 2013.
- [9] R. O. Duda and P. E. Hart, *Classification and Scene Analysis*, pp. 271–272, Wiley, New York, NY, USA, 1973.
- [10] F. Chang, C.-J. Chen, and C.-J. Lu, "A linear-time component-labeling algorithm using contour tracing technique," *Computer Vision and Image Understanding*, vol. 93, no. 2, pp. 206–220, 2004.
- [11] Y. Chen, Z. X. Yang, and Z. F. Bai, "Simultaneous Real-Time Segmentation of Diversified Barcode Symbols in Complex Background," in *Proceedings of the First International Conference On Intelligent Networks And Intelligent Systems*, pp. 527–530, Wuhan, China, November 2008.
- [12] Q. L. Liu, X. C. Li, M. Zou, and Z. Jun, "The Multi-QR Codes Extraction Method in Illegible Image Based on Contour Tracing," in *Proceedings of the 2011 IEEE International Conference on Anti-Counterfeiting, Security and Identification*, pp. 51–56, Xiamen, China, June 2011.
- [13] J. A. Lin and C. S. Fuh, "2D barcode image decoding," *Mathematical Problems in Engineering*, vol. 2013, Article ID 848276, 10 pages, 2016.
- [14] M. Dubská, A. Herout, and J. Havel, "Real-time precise detection of regular grids and matrix codes," *Journal of Real-Time Image Processing*, vol. 11, no. 1, pp. 193–200, 2016.
- [15] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.
- [16] J. M. Chen, "One-dimension Barcode Localization Algorithm with Complex Background," M.S. thesis, Zhejiang University, Hangzhou, China, 2015.
- [17] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
- [18] A. Zamberletti, I. Gallo, and S. Albertini, "Robust Angle Invariant 1D Barcode Detection," in *Proceedings of the 2013 Second IAPR Asian Conference on Pattern Recognition (AVPR)*, pp. 160–164, Naha, Japan, November 2013.
- [19] M. Katona, P. Bodnár, and L. G. Nyúl, "Distance transform and template matching based methods for localization of barcodes and QR codes," *Computer Science and Information Systems*, vol. 17, no. 1, pp. 161–179, 2019.
- [20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [21] Y. Tian, Z. H. Che, G. T. Zhai, and Z. Gao, "BAN, A barcode accurate detection network," in *Proceedings of the 2018 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1–5, Taichung, Taiwan, China, December 2018.
- [22] J. Jia, G. Zhai, J. Zhang et al., "EMBDN: an efficient multiclass barcode detection network for complicated environments," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9919–9933, 2019.
- [23] N. Ostu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [24] D. Darriba, G. L. Taboada, R. Doallo, and D. Posada, "JModelTest 2: more models, new heuristics and parallel computing," *Nature Methods*, vol. 9, no. 8, p. 772, 2012.

Research Article

Video Style Transfer based on Convolutional Neural Networks

Sun Dong ^{1,2}, Youdong Ding,² Yun Qian ³, and Mengfan Li¹

¹Pan Tianshou College of Architecture, Art and Design, Ningbo University, Ningbo 315211, China

²Shanghai Engineering Research Center of Motion Picture Special Effects, Shanghai 200072, China

³School of Fine Arts, Anhui Normal University, Wuhu 241002, China

Correspondence should be addressed to Yun Qian; qianyun@ahnu.edu.cn

Received 25 November 2021; Revised 16 February 2022; Accepted 28 February 2022; Published 25 March 2022

Academic Editor: Savvas A. Chatzichristofis

Copyright © 2022 Sun Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Video style transfer using convolutional neural networks (CNN), a method from the deep learning (DL) field, is described. The CNN model, the style transfer algorithm, and the video transfer process are presented first; then, the feasibility and validity of the proposed CNN-based video transfer method are estimated in a video style transfer experiment on *The Eyes of Van Gogh*. The experimental results show that the proposed approach not only yields video style transfer but also effectively eliminates flickering and other secondary problems in video style transfer.

1. Introduction

In the deep learning field, image style transfer is an important research topic [1]. Some traditional methods for style transfer include texture synthesis, support vector machines, histogram matching, and automatic sample collection [2–4]. Although special effects can be produced, image distortion as well as other prominent problems can also occur, such as the loss of detail, bending and deformation of straight lines, and color change over a large range. In addition, special algorithms are usually needed to further correct mistakes, resulting in low-style transfer efficiency and poor image quality. Recently, convolutional neural network DL models have been successfully applied to image style transfer problems, reigniting interest in this research field [5–8]. In the present study, a style transfer algorithm was developed and tested on *The Eyes of Van Gogh*, an American biography and feature film directed by Alexander Barnett, with the main roles played by Dane Agostini and John Alexander, which narrates the secret story of Van Gogh in St-Remy, Bedlam for 12 months; this film shows the legend of the talented artist who created, loved, and changed

the world through hallucinations, nightmares, and painful memories.

The purpose of this paper is that we tried adopting CNN based style transfer method for the video style transfer experiment. The foregoing CNN-based style transfer method is seldom used in the video field. The proposed style transfer algorithm uses techniques from the DL field and is based on a CNN; the feasibility and validity of the proposed CNN-based video style transfer algorithm are estimated in an experiment on the transfer of the painting style of Van Gogh's *The Starry Night* to the film special effects.

2. Methods

2.1. CNN. A CNN is a DL method developed recently that has attracted considerable attention. In general, a CNN is a multilayered network [9–11]; a typical CNN is shown schematically in Figure 1. A CNN consists of a series of convolution (C) and subsampling (S) layers. Each layer is composed of multiple 2D planes, with each serving as a feature map; the network also includes some fully connected (FC) hidden layers. There is only one input layer in a CNN.

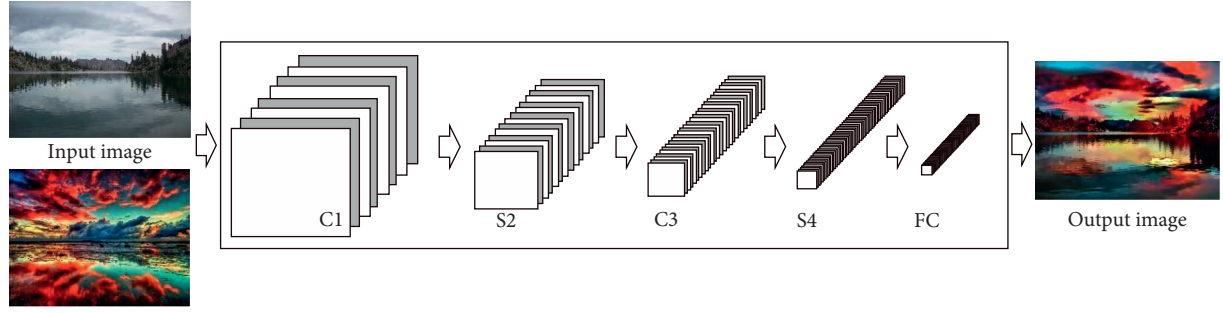


FIGURE 1: CNN structure.

This input layer receives two-dimensional objects directly, and the process of feature extraction into samples is performed by the convolution and sampling layers. Multiple fully connected hidden layers are used mostly used to accomplish specific tasks [12].

2.2. Style Transfer Algorithm. The methodology of Gatys et al. [13, 14] is reviewed. On this basis, the feature extraction and storage of style images and content images (single frames of video) are proposed. A style image \vec{a} is transmitted through the network (expressed as A^l), and the styles included in all layers are computed and stored ($A^l \in R^{N_i \times N_i}$). A content image \vec{p} is transmitted through the network (expressed as P^l) and stored in layer l ($P^l \in R^{N_i \times D_l}$). Therein, N_i represents the number of filters in that layer; and D_l is the spatial dimension of the feature map, namely the product of width and height. Then, a random white noise image \vec{x} is transmitted through the network; both the content feature F^l and the style feature G^l are computed. $F^l[\cdot] \in R^{N_i \times D_l}$ and F_{ij}^l are the activations of the i -th filter at j in layer l ; $G^l[\cdot] \in R^{N_i \times N_i}$ and G_{ij}^l are the vectorization results of layer l in i and j feature maps; the feature correlation is obtained as $G_{ij}^l = \sum_{k=1}^{D_l} F_{ik}^l F_{jk}^l$.

For each layer of the style image, the mean quadratic deviation of elements between G^l and A^l is computed, and the style loss $\mathcal{L}_{\text{style}}$ is computed using equation (1).

$$\mathcal{L}_{\text{style}}(\vec{a}, \vec{x}) = \sum_{l \in L_{\text{style}}} \frac{1}{N_l^2 D_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2. \quad (1)$$

The mean quadratic deviation between F^l and P^l is computed, and the content image loss $\mathcal{L}_{\text{content}}$ is computed from equation (2).

$$\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) = \sum_{l \in L_{\text{content}}} \frac{1}{N_l D_l} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2. \quad (2)$$

The total loss $\mathcal{L}_{\text{singleframe}}$ is a linear combination of the style and content loss functions; it could propagate computation reversibly with errors, pertaining to the derivatives of pixel values; a gradient is used to iterate and upgrade the image \vec{x} until it matches the style feature of the style image \vec{a} and the content feature of the content image \vec{p} simultaneously; weight factors, including both α and β , determine

the importance of the two components, content and style, which is calculated from equation (3).

$$\mathcal{L}_{\text{singleframe}}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{\text{style}}(\vec{a}, \vec{x}). \quad (3)$$

2.3. Elimination of Flickering in Video Style Transfer. At present, most restoration methods require modeling to account for flickering in the image sequence; the flicker parameters of the model are first estimated, and then color correction and restoration are performed. However, the existing methods cannot treat flicker problems arising from video style transfer. A color transfer algorithm proposed by Reinhard et al. [15] is put forward in this paper, based on which the steps of video frame color correction and sequence restoration are simplified; the steps are specific to flickering after video style transfer, and the interframe color transfer is used directly. Thus, the data processing load and computational complexity are reduced, yet the restoration efficiency is increased. Video color transfer is an algorithm that changes the frame color [16]. A synthetic frame with the form of the original frame and reference frame color can be obtained by defining a reference frame that provides the original frame for both the structure and color layout, which is especially suitable for continuous video processing. The specific algorithm is as follows:

- (1) Both the original frame and the reference frame of the video are converted to the l , α , and β color space from the RGB color space, and the correlation between the two frames is removed.
- (2) The mean and the standard deviations of the original frame and the reference frame in each channel of the l , α , and β color space are computed, respectively. The mean values of the three channels of the original frame are m_S^l , m_S^α , and m_S^β ; the standard deviations of the original frame are σ_S^l , σ_S^α , and σ_S^β ; the mean values of the reference frame are m_R^l , m_R^α , and m_R^β ; and the standard deviations of the reference frame are σ_R^l , σ_R^α , and σ_R^β .
- (3) In accordance with equation (4), the overall color information of the mean value of all pixel values weakening the original frame is subtracted from all pixel values for each channel.

$$\begin{cases} l'_S = l_S - m_S^l, \\ \alpha'_S = \alpha_S - m_S^\alpha, \\ \beta'_S = \beta_S - m_S^\beta. \end{cases} \quad (4)$$

Here, l_S , α_S , and β_S are all pixel values for the three channels of the original frame, and l'_S , α'_S , and β'_S are the pixel values for the three channels of the original frame after weakening.

- (4) The standard deviation ratio of the original frame and reference frame is taken as the coefficient of channel value offset; the detailed information of the reference frame is mapped to the original frame in accordance with the following equation:

$$\begin{cases} l' = \frac{\sigma_R^l}{\sigma_S^l} * l'_S, \\ \alpha' = \frac{\sigma_R^\alpha}{\sigma_S^\alpha} * \alpha'_S, \\ \beta' = \frac{\sigma_R^\beta}{\sigma_S^\beta} * \beta'_S. \end{cases} \quad (5)$$

Here, l' , α' , and β' are all pixel values for the synthetic frame in the three channels of the l , α , and β color space.

- (5) The overall information about the reference frame is added to the synthetic frame; that is, this information is added to the mean value for each channel of the reference frame as shown in the following equation, and the final synthetic frame is thereby obtained.

$$\begin{cases} l = l' + m_R^l, \\ \alpha = \alpha' + m_R^\alpha, \\ \beta = \beta' + m_R^\beta. \end{cases} \quad (6)$$

Here, l , α , and β are all pixel values for the three channels finally obtained by the synthetic frame.

- (6) After color transfer, the synthetic frame from the l , α , and β color space is converted to the RGB color space.

2.4. Video Style Transfer Process

- (1) The video is converted into a single frame. Pre-processing should be performed on the transferred video; single-frame processing is performed for continuous videos, and the preprocessed video is saved as a JPG file. MATLAB software is applied for processing, and classified conservation is conducted based on the shot sequence.
- (2) *Video style transfer*. The style transfer algorithm is used to perform video frame style transfer using a CNN. The same group of shot circles is usually

selected in the transfer process to conduct the video style transfer experiment.

- (3) *Video color transfer*. Secondary flicker problems frequently occur in video style transfer. The essence of flicker is that adjacent frames vary significantly in brightness or hue, and the visual perception of flickering also appears when the video is played continuously. The flicker problem in video transfer is treated using the color transfer algorithm.
- (4) *Single-frame synthetic video*. After continuous video transfer and treatment of flickering, MATLAB is used for the single-frame synthesis of AVI-formatted videos to evaluate the results.

3. Experimental Work

3.1. Model Parameters. Model selection and parameter optimization are key steps in video style transfer, and a proper model and parameters can significantly enhance the transfer of high-quality artistic videos. First, four artificial intelligence models, namely CaffeNet, GoogLeNet, VGG16, and VGG19, were selected, all of which have their own unique advantages [17–20]. CaffeNet is a classical DL model; its advantages include network expansion and the ability to solve fitting problems. It is also the simplest network among the four models; since these models have been proposed, several deeper network structures have been proposed. GoogLeNet utilizes the concept of an inception module, aiming at strengthening the function of basic feature extraction modules. It considerably enhances the feature extraction ability of a single layer, but does not significantly increase the computed amount. Although VGG-Net has inherited some network frameworks from LeNet and AlexNet, the former is not identical to the latter ones; VGG-Net uses more layers, usually from 16 to 19. VGG-Net mainly increases the network structure depth, while reducing parameter configuration. The model suitable for the style transfer of the video *The Eyes of Van Gogh* must be analyzed in advance. Second, the style/content conversion rates (10^{-1} , 10^{-2} , 10^{-3} , 10^{-4} , and 10^{-5}) are key parameters for transfer, and a preexperiment is also required for parameter selection. Therefore, the video clip of *The Eyes of Van Gogh* was selected for the style transfer experimental analysis.

Figure 2 shows the experimental results obtained when the style/content conversion rates of the four models were set to 10^{-1} , 10^{-2} , and 10^{-3} ; the video frame style transfer was not sufficient because the images in the original video remained, owing to the excessively low conversion rate. When the conversion rate was 10^{-5} , the style transfer was excessive and some important information, such as the form and structure of the original video frame, was lost. The CaffeNet-based style transfer revealed several serious errors, such as distortion, making it not suitable for this video transfer; although GoogLeNet performed slightly better than CaffeNet, there were still many errors; the performances of VGG16 and VGG19 were better, and these methods achieved optimal results, especially for the conversion rate of 10^{-4} . The experimental results for the VGG16/19 model were further compared, and the results of the VGG19-based transfer were considered to have higher fidelity,



FIGURE 2: Comparison of models and parameters.

more plentiful layers, and a high transfer efficiency (marked by a red frame in Figure 2). Figure 3 shows the computational times for the different models and parameters.

As a result, the VGG19 model and the style/content conversion rate of 10^{-4} were selected for the style transfer in the video transfer experiment on *The Eyes of Van Gogh*.

3.2. Video Style Transfer Experiment. Based on the CNN [21–24], the style transfer algorithm based on the Caffe platform was used to input the video frame to be transferred. The VGG-19 network trained in advance was used for computing the loss; conv4_2 in this network represents the content; conv1_1, conv2_1, conv3_1, conv4_1, and conv5_1 represent the style; the loss of the parameter weight used was not more than 0.02%; and the number of iterations was 512. The hardware device used was a high-performance workstation (HPZ840 workstation, parameter configuration: Intel Xeon E5 Eight-core, two GPUs, Nvidia TITAN Xp. memory: 64 GB). The following experimental steps were performed: first, we imported 24 frames (800×480 per frame in size) at a time for continuous style transfer. The model was VGG19, and the

conversion rate was 10^{-4} ; second, Van Gogh’s representative work, *The Starry Night*, was selected as the source style image; its short yet thick brushwork and fiery color are filled with personality characteristics and artistic charm; finally, CUDA was used for parallel computing and the outputting of the JPG video frame (the maximal length was 1024).

Figure 4 shows the results of the style transfer experiment for a continuous video. Figures 4(a) and 4(c) were selected from the original frames in the experimental video *The Eyes of Van Gogh*; we chose two groups of shots, with low indoor brightness and high outdoor brightness, respectively, for the video style transfer experiment. This group of video shots was relatively fixed, and the characters had no large displacements. Figures 4(b) and 4(d) show the realization of the video style transfer using equations (1)–(3) that were presented in Section 2 of this paper. The experimental results show that, on the one hand, the style transfer retains the form structure of the original video frame and other information, and when mixed with the brushwork texture and color elements of *The Starry Night*, their combination produces a unique visual effect; on the other hand, the video frame details obtained in the style transfer are excellent with rich colors, without any

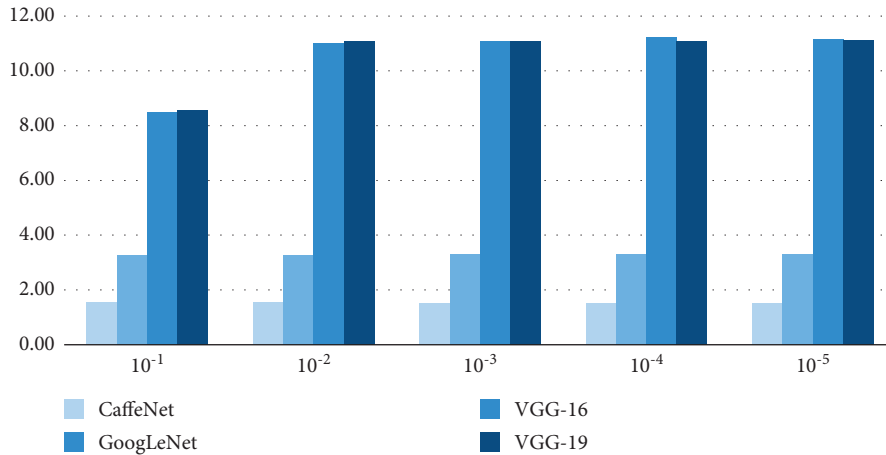


FIGURE 3: Comparison of time spent.



FIGURE 4: Result of the video style transfer experiment (a–d).

style transfer errors such as pseudoscopic images or fuzziness, and without interframe flickering.

Considering the real and effective experimental results, we selected a set of continuous frames in which the characters had large deviations from the video for the style transfer experiment, to estimate the reliability and validity of the style transfer algorithm for video style transfer applications. The experimental steps and model parameters were the same as those mentioned previously.

Figure 5 shows the results of this video style transfer. Although the form structure information, mixture with

textures and colors of the source style image, and other elements of the target video frame were reserved to produce visual effects, we also noted some mistakes in the details of the video style transfer. A prominent problem was interframe flickering, as shown in the part marked with a red frame in Figure 5(b). Therein, some parts of several single frames exhibited hue and brightness deviations, which caused flickering as secondary damage during continuous play. Thus, we used the color transfer algorithm to further process and eliminate flickering, thereby attaining optimal video transfer.

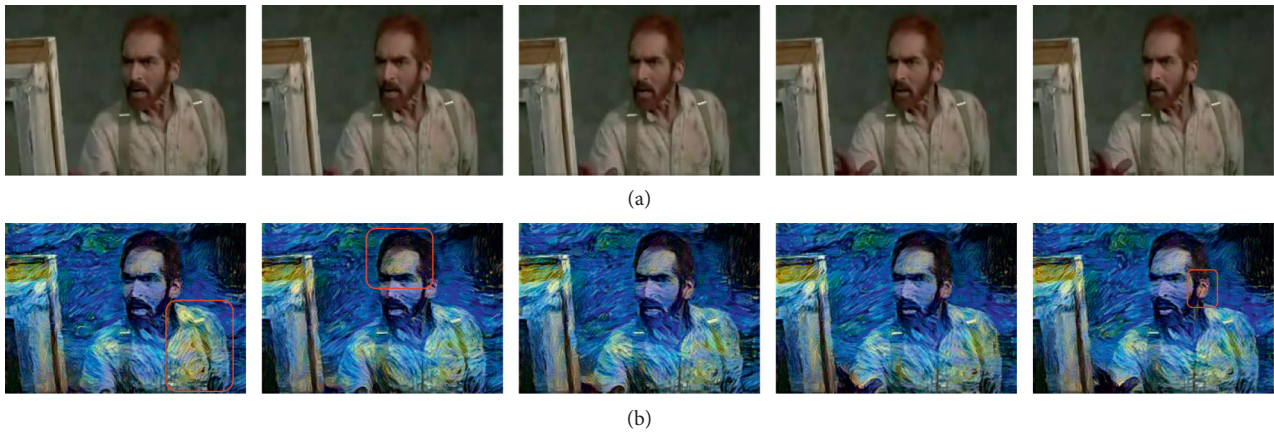


FIGURE 5: Results of the video style transfer experiment (a, b).

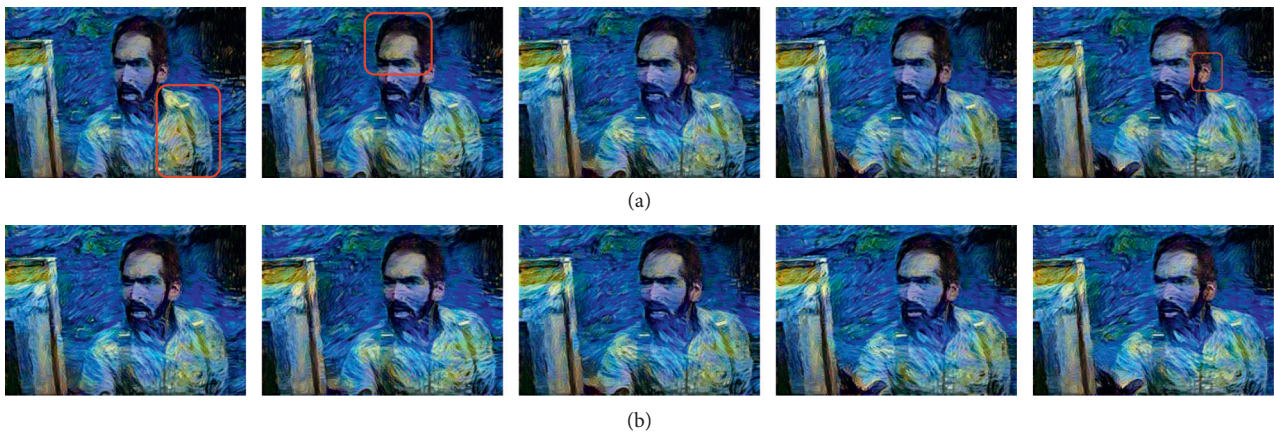


FIGURE 6: Results of the video flickering elimination experiment: (a) before and (b) after.

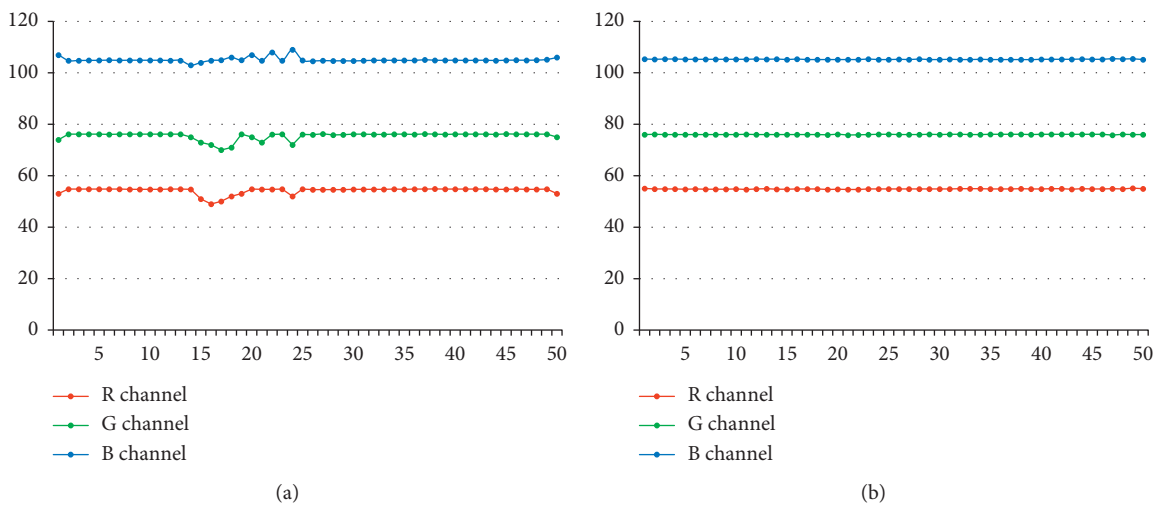


FIGURE 7: Comparison of the mean statistics of videos: (a) before and (b) after.

3.3. *Elimination of Flickering.* The shooting process was performed for various videos imported in accordance with equations (4)–(6) that were presented in Section 2 of this

paper. A proper frame in the same shot was selected as a reference frame (here, we chose the middle frame as the reference frame), and the color feature of the reference frame



FIGURE 8: Results of the video style transfer experiment.

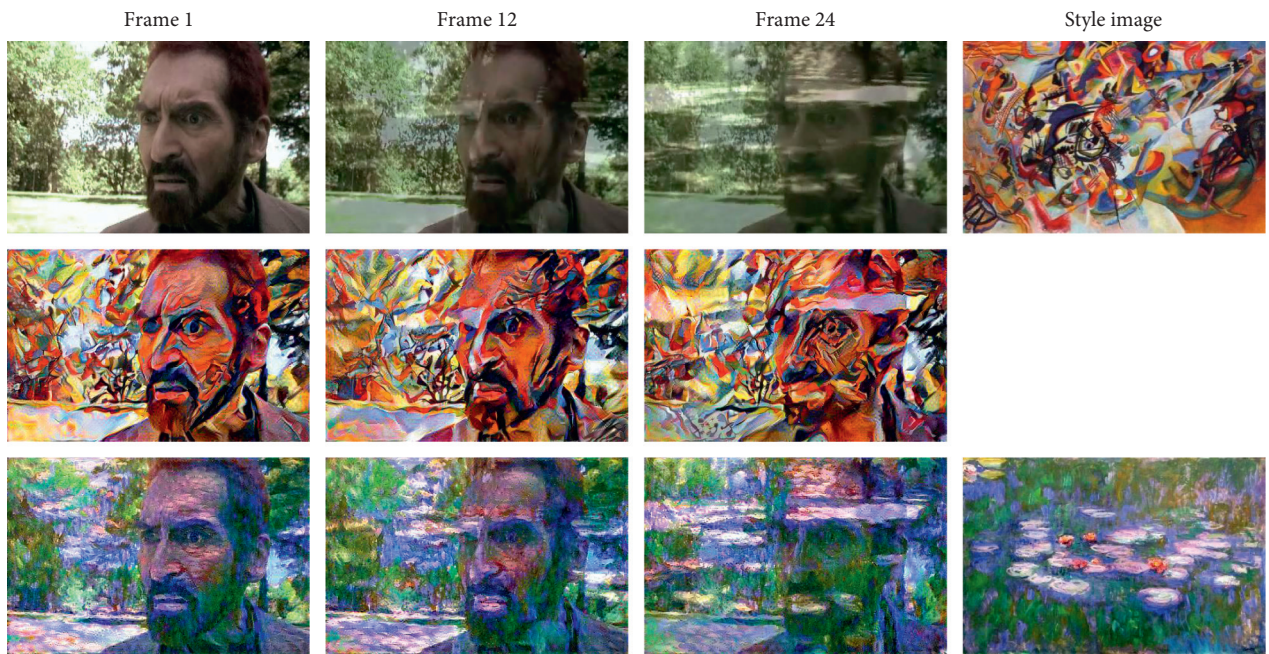


FIGURE 9: Result of different video style transfer experiment.

was transferred to each frame in the same group of shots successively; after the video processing of the same group, the above steps were repeated until all of the frames imported into the video were processed. Figure 6(a) shows the video frames after the elimination of flickering using the color transfer algorithm; we observe that compared with the areas marked with red frames in Figure 6(a) (the character’s chest, forehead, and other body parts), the style transfer errors were effectively eliminated.

Figure 7 shows the mean statistics of videos before and after the elimination of flickering; Figure 7(a) shows the statistics before the flickering elimination process, and

Figure 7(b) depicts the statistics after the flickering elimination process.

Figure 8 shows a scene from *The Eyes of Van Gogh* where two people walk through the scene. Without the color transfer algorithm, the stylized videos demonstrate flickering between adjacent frames after the people pass by. Figure 9 shows another scene from *The Eyes of Van Gogh* with fast camera motion. The color transfer algorithm eliminated the interframe flickering. The experimental results show that the color transfer algorithm can effectively eliminate secondary flickering arising from video style transfer, and the resulting video is full of colors and exhibits a uniform hue.

4. Conclusion

The CNN-based style transfer algorithm quickly and effectively generates diverse and stylized videos, as well as unique visual effects. The experiment proved that the video style transfer method proposed herein is feasible and effective. In terms of parameter optimization of the video style transfer model, we found that the style transfer results are strongly determined by the style/content conversion rate and model selection. The experiment also showed that for the film, *The Eyes of Van Gogh*, the optimal model was VGG19 and the optimal conversion rate was 10^{-4} . It should be noted that model parameters should have been selected in combination with different videos; a sample analysis experiment should be conducted in advance to obtain the best results. In addition, as flickering and other secondary problems often occur in video style transfers, the video after style transfer requires further processing using the color transfer algorithm to obtain high-quality experimental results.

In future work, we hope to explore the use of the proposed CNN-based style transfer algorithm for other video transformation tasks, such as the production of stable and visually appealing stylized videos even in the presence of fast motion and strong occlusion. Owing to the subjectivity of video quality evaluation, we also plan to establish a subjective evaluation index system for better evaluation of style transfer video quality. Video style transfer is a common problem like loss of details, bending and deformation, or color change over a large range, which is to cause secondary video damage like a flicker. Subjective evaluation is the most commonly used method in video quality evaluation. However, subjective evaluation is a time-consuming task. For this, we plan to employ a forced-choice evaluation on Amazon Mechanical Turk (AMT) with 200 different users to evaluate our experimental results. This is a part of further research. In addition, we plan to extend the dataset to include more videos, which would make our approach more generalizable.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

All authors declare that there are no conflicts of interest with this study.

Acknowledgments

This research was supported by the National Natural Science Foundation of China [Grant no.61402278, 61303093], the Teaching and Research Project of Ningbo University [Grant no. JYXMXZD2022019], the Social Science Foundation of Anhui Province [Grant no. AHSKY2018D74], and the Outstanding Young Talents Foundation by the Ministry of Education of Anhui Province [Grant no. gxyq2018002].

References

- [1] M. Elad and P. Milanfar, "Style transfer via texture synthesis," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2338–2351, 2017.
- [2] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proceedings of the European Conference on Computer Vision*, pp. 694–711, Amsterdam, The Netherlands, October 2016.
- [3] J. Liao, Y. Yao, L. Yuan, H. Gang, and B. K. Sing, "Visual attribute transfer through deep image analogy," *ACM Transactions on Graphics*, vol. 36, 2017.
- [4] T. Dutta and H. P. Gupta, "Leveraging smart devices for automatic mood-transferring in real-time oil painting," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 1581–1588, 2017.
- [5] H. S. Faridul, T. Pouli, C. Chamaret et al., "Colour mapping: a review of recent methods, extensions and applications," *Computer Graphics Forum*, vol. 35, 2016.
- [6] W. H. Bangyal, R. Qasim, Z. Ahmad et al., "Detection of fake news text classification on COVID-19 using deep learning approaches," *Computational and Mathematical Methods in Medicine*, vol. 2021, Article ID 5514220, 14 pages, 2021.
- [7] W. H. Bangyal, A. Hameed, J. Ahmad, and R. Etengu, "New modified controlled bat algorithm for numerical optimization problem," *Computers, Materials & Continua*, vol. 70, pp. 2241–2259, 2021.
- [8] W. H. Bangyal, K. Nisar, A. A. B. Ibrahim, M. R. Haque, J. J. P. C. Rodrigues, and D. B. Rawat, "Comparative analysis of low discrepancy sequence-based initialization approaches using population-based algorithms for solving the global optimization problems," *Applied Sciences*, vol. 11, no. 16, Article ID 7591, 2021.
- [9] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.
- [10] Y. Nikulin and R. Novak, "Exploring the neural algorithm of artistic style," 2016, <https://arxiv.org/abs/1602.07188>.
- [11] S. Hicsonmez, N. Samet, F. Sener, and P. Duygulu, "DRAW: deep networks for recognizing styles of artists who illustrate children's books," in *Proceedings of the 2017 ACM on international conference on multimedia*, Yokohama Japan, June 2017.
- [12] Y. Jing, Y. Yang, Z. Feng, Y. Yizhou, and S. Mingli, "Neural style transfer: a review," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, 2017.
- [13] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, June 2016.
- [14] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos," in *Proceedings of the Computer Vision and Pattern Recognition*, Hannover, Germany, September 2016.
- [15] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, 2001.
- [16] X. Huang, Y. D. Ding, and B. Wu, "Implementation and design of old film global flicker restoration system," *Video Engineering*, vol. 40, no. 12, pp. 125–129, 2016.
- [17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Vision and Pattern Recognition*, 2014.
- [18] M. S. Ryoo and L. Matthies, "Video-based convolutional neural networks for activity recognition from robot-centric

- videos,” in *Proceedings of the SPIE Defense Security. International Society for Optics and Photonics*, pp. 98370R–98376R, Bellingham, WA, USA, September 2016.
- [19] A. Zhai, D. Kislyuk, Y. Jing et al., “Visual discovery at pinterest,” in *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 515–524, Perth, Australia, April 2017.
- [20] F. Okura, K. Vanhoey, A. Bousseau, A. A. Efros, and G. Drettakis, “Unifying color and texture transfer for predictive appearance manipulation,” *Computer Graphics Forum*, vol. 34, no. 4, pp. 53–63, 2015.
- [21] D. Zheng, “A novel method for fabric color transfer,” *Color Research & Application*, vol. 40, no. 3, pp. 304–310, 2015.
- [22] O. Frigo, N. Sabater, J. Delon, and H. Pierre, “Video style transfer by consistent adaptive patch sampling,” *The Visual Computer*, vol. 35, 2019.
- [23] Y. Liu, M. Cohen, M. Uyttendaele, and S. Rusinkiewicz, “AutoStyle: automatic style transfer from image collections to users’ images,” *Computer Graphics Forum*, vol. 33, no. 4, pp. 21–31, 2014.
- [24] A. Gupta, J. Johnson, A. Alahi, and L. Fei Fei, “Characterizing and improving stability in neural style transfer,” in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.

Research Article

A Real-Time Framework for Human Face Detection and Recognition in CCTV Images

Rehmat Ullah ¹, **Hassan Hayat**,² **Afsah Abid Siddiqui**,² **Uzma Abid Siddiqui**,² **Jebran Khan** ³, **Farman Ullah**,² **Shoaib Hassan**,² **Laiq Hasan**,¹ **Waleed Albattah** ⁴, **Muhammad Islam**,⁵ and **Ghulam Mohammad Karami** ⁶

¹Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Peshawar, Pakistan

²Department of Electrical and Computer Engineering, COMSATS University Islamabad, Attock Campus, Attock, Pakistan

³Department of Artificial Intelligence, AJOU University, Suwon, Republic of Korea

⁴Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia

⁵Department of Electrical Engineering, College of Engineering and Information Technology, Onaizah Colleges, Al-Qassim, Saudi Arabia

⁶SMEC International Pvt. Limited, Kabul 1007, Afghanistan

Correspondence should be addressed to Ghulam Mohammad Karami; ghulam.karami@smec.com

Received 2 October 2021; Accepted 15 December 2021; Published 3 March 2022

Academic Editor: Nouman Ali

Copyright © 2022 Rehmat Ullah et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper aims to develop a machine learning and deep learning-based real-time framework for detecting and recognizing human faces in closed-circuit television (CCTV) images. The traditional CCTV system needs a human for 24/7 monitoring, which is costly and insufficient. The automatic recognition system of faces in CCTV images with minimum human intervention and reduced cost can help many organizations, such as law enforcement, identifying the suspects, missing people, and people entering a restricted territory. However, image-based recognition has many issues, such as scaling, rotation, cluttered backgrounds, and variation in light intensity. This paper aims to develop a CCTV image-based human face recognition system using different techniques for feature extraction and face recognition. The proposed system includes image acquisition from CCTV, image preprocessing, face detection, localization, extraction from the acquired images, and recognition. We use two feature extraction algorithms, principal component analysis (PCA) and convolutional neural network (CNN). We use and compare the performance of the algorithms K-nearest neighbor (KNN), decision tree, random forest, and CNN. The recognition is done by applying these techniques to the dataset with more than 40K acquired real-time images at different settings such as light level, rotation, and scaling for simulation and performance evaluation. Finally, we recognized faces with a minimum computing time and an accuracy of more than 90%.

1. Introduction

Today's organizations face significant security challenges; they need several specially trained personnel to achieve the required security. However, humans make mistakes that affect safety. Closed-circuit television (CCTV) is currently used for various purposes in everyday life. The development of video surveillance has transformed simple passive monitoring into an integrated intelligent control system.

Face detection and its new applications for secure access control, financial transactions, etc. Biometric systems (faces, palms, and fingerprints) have recently gained new importance. With advances in microelectronics and vision systems, biometrics has become economically viable. Facial recognition is an essential part of biometrics. In biometrics, human fundamentals are mapped to current data. The facial features are hauled out and implemented using an efficient algorithm, and some variations are made to improve the

existing algorithm model. Face recognition from the computer can be applied to a variety of applied applications, including crime ID, security systems, and authentication. A facial recognition system typically involves steps of face detection where the face of the input image is detected, and then the image process cleans the face image for easy recognition.

In this modern age, face recognition has become a necessity as the individual's identification increases daily with globalization. Since the last two decades, face recognition has received much attention because of its various applications, invaluable image analysis, and understanding domains. Face recognition is also becoming important in other fields like image processing, animation [1], security [2], human-computer interface [3], and medicine [4]. Face recognition is natural, noninvasive, and easy to use. The face recognition system has a wide choice of applications in public safety, entertainment, attendance management, and financial payment. While today's facial recognition systems work well in relatively controlled environments, they suffer from significant problems when used in existing surveillance systems due to image resolution, background clutter, lighting variations, and face and expression posture.

Face recognition systems consist of three steps, such as preprocessing of the image, feature extraction, and classification technique for recognition [5]. Features extracted from the face, such as the mouth, nose, eyebrows, etc., are geometric features. The detected and processed face is compared to a database of known faces to determine who the person is. The surveillance system needs people to monitor it. Human monitoring involves reliability issues, scalability issues, and the inability to identify everyone.

Facial occlusions, such as beards and accessories (glasses, hats, and masks), involve evaluating facial recognition systems, making the subject diverse and challenging to function in a nonsimulated environment. Another essential factor to consider is the different terminologies of the same distinct: macro and microterminologies find their place on someone's face because of changes in an emotional state, and because of the many expressions of this type, effective recognition becomes difficult. A perfect face recognition system should be able to tolerate changes in lighting, expressions, poses, and occlusions and can scale for many users who need to capture the fewest images simultaneously.

The overall contributions of the research paper can be summarized as follows:

- (i) A machine learning-based framework for detecting and recognizing faces in CCTV images with various clutter backgrounds and occlusion
- (ii) A dataset of 40K images with different environmental conditions, clutter backgrounds, and occlusion
- (iii) Performance comparison of classical machine learning and deep learning algorithms for faces recognition in CCTV images

The rest of the paper is organized as follows: Section 2 briefly introduces the related works. Section 3 explains the

methodology, and the results are discussed in Section 4. Finally, we conclude the paper in Section 5.

2. Related Work

In this section, we briefly introduce the related works about face detection and recognition using classical approaches and deep learning.

2.1. Face Detection Algorithms

2.1.1. Geometric Methods for Face Detection. In the early stages of computer vision, researchers explored many algorithms that extracted the image characteristics and utilized geometric requirements to comprehend the provisions of all features. This was partly due to very limited computational resources. The reduction of information from the extraction of functionality has made computer vision possible in the first computers [6, 7].

2.1.2. Template-Based Face Detection [8]. Most of the face detection algorithms are model-based, they encode facial images directly on the basis of pixel intensity. Probabilistic models are mostly used for the characterization of these images of facial images also by neural networks or by some other mechanisms. The parameters of these models are automatically adjusted by sample images or manually.

2.1.3. Simple Templates. If you are using a skin-based method and another skin color is found in the image (like arms and hands), these algorithms show false results. Many researchers tried to overcome this by using simple models to integrate results from the color matching of skin. These models have varied from some ovals related to the image of the edge of the entrance to the correlation models for the regions of skin color and skin color (like lips, hands, or eyes). However, these techniques can enhance the robustness of detectors by color, but with also the enhancement of speed.

2.2. Face Recognition algorithms. Face recognition is a technique that has now attained consideration in machine learning and artificial intelligence. It plays an essential role in many social security applications. There are many studies and practices now under research that can solve the problem of face recognition. Vivek and Guddeti [9] proposed combining cat swarm optimization (CSO), particle swarm optimization (PSO), and genetic algorithm (GA). This hybrid technique has inspired many others to work similarly. Ali et al. combined SVM, higher-order spectral (HOS), and random transformation (RT) [10].

2.2.1. Iterative Closest Point-Based Alignment. The objective of the alignment approach [11, 12] is based on the closest iterative point to determine the translation and the rotation parameters in an iterative way to convert the point cloud. Clouds' mean square error becomes minimal while both point clouds are aligned. So, distance among point clouds is

reduced to a minimum by translating and rotating one of the point clouds with respect to others, also determine by identifying the distance with every point in the initial point clouds every second, also calculating the average of all distances. An important disadvantage of the alignment approach based on the closest iterative point is that it needs an initial alignment of the convergence course. This approach is computationally very expensive, so that's another disadvantage.

2.2.2. Simulated Annealing-Based Alignment. It is an algorithm based on a stochastic process used for local research [13]. The difference between hill-climbing and simulated annealing is that it can compute an even worse solution than the current one in the iteration process. As simulated annealing is not bounded by local minima, it is more likely that you will find a solution. Six parameters are required for simulated annealing (in which three for every translation also the rotation referencing to a 3D coordinate system) which is used to define transformation matrix which is used for an alignment between two 3D faces. This approach aligns images of the face in three phases: (1) alignment in initial level, (2) alignment in an approximate level, and (3) alignment in the last level [14]. Initially, the center of the two-sided mass is being aligned. By using this approach, it serves to minimize an approximation measure which uses the consensus of multiple estimators M (MSAC) together with the mean square error corresponding point of two faces that will compare. Then, an accurate alignment is obtained with the mean of a search algorithm that is based upon simulated annealing, which uses the measurement of the interpenetration of surfaces (SIM) as an estimation criterion. The disadvantage of alignment based on simulated annealing is its more calculation time which is comparable to the alignment based on the nearest iterative point.

2.2.3. Average-Based Face Model. This alignment is based on the medium-based face model [15]. First of all, the reference points are on the face automatically or manually. Subsequently, the average of pivotal coordinates calculated, followed by procrustes examination and transformed milestone [16], are again mediated to obtain a face model. While in this method, the image of the probe face aligns with the average model using an alignment on the nearest iterative point. A notable weakness of the alignment based on the medium face model is the low precision index [17] and part of the spatial material lost during the creation of the medium face model.

The first step in face recognition is preprocessing. Images taken from a camera or in real-time video surveillance setups may suffer from various degradations during the process of capture, transformation, conversion, or compression [18]. For instance, blurry, noisy, and low-resolution images affect the face recognition process. Such issues may lead to significant challenges in the face recognition scheme and decrease its performance. Therefore, pre-processing is an essential step in any face recognition system. Many color normalization, statistical, and convolutional methods are

used as preprocessing tools [19]. Another big problem in face recognition through surveillance cameras is that too many images of a person are collected and applying a face recognition algorithm to each of them proves costly in terms of processing and energy consumption. Vignesh et al. [20] presented a technique for image quality assessment (IQA) using CNN to take the person's best image. Tudavekar et al. [21] proposed video inpainting to fill the missing regions in a video by dual-tree complex wavelet transformation.

PCA is the most widely used technique in signal and image processing. They are also known as eigenfaces, the orthogonal vectors that help in face recognition. Drume and Jalal proposed a two-level classification technique that uses principal component analysis (PCA) in level one and boosts its results by support vector machine (SVM) at level two [22]. Kanade employed image processing techniques to extract 16 facial parameters with the ratio of distance, angle, and area and used the method of Euclidean distance to achieve a performance of 75% [23]. On this basis, a method called eigenface for face recognition was proposed for the first time [24]. This method leads to the formation of an algorithm called principal component analysis (PCA). From then on, PCA gathered a lot of attention and became the most effective approach for face recognition. Many improvements have been made in the PCA algorithm to get its best results [25–30].

Rala used PCA and Kernel-PCA for feature extraction and face recognition, respectively. They explore the non-linear kernel function for the improvement of PCA [31]. Abdullah et al. optimized the PCA time complexity without affecting the performance of the algorithm [32]. Another approach includes hexagonal feature detection, which works on the principle of edge detection [33]. A part-based method in [34] utilizes PCA, NMF, ICA, LDA, etc., under partial occlusion. Another effective algorithm called AFMC shows the results to be more accurate with the reduced computational cost and proposes eliminating the SSS problem [35]. Viola–Jones algorithm was also presented with the smoothed invalid regions and excluded near-ear regions [36].

Deep hidden ID entity feature (DeepID), a face representation based on CNN, is suggested in [37]. Unlike DeepFace, which learns features from a single large CNN, DeepID learns features from an ensemble of tiny CNNs that are utilized for network fusion. Similarly, a face recognition pipeline, WebFace, is proposed in [38], which uses CNN to learn the face representation. The convolutional neural network (CNN) [39] has been one of the most prominent approaches in computer vision over the last decade, with applications including image classification [40], object identification [41], and face recognition [38]. Different methods, such as PCA-based eigenfaces [42] and LDA-based Fisherfaces [43] employ the nearest neighbor (NN) classifier and its variants [44]. In a face recognition system, supervised classifiers such as support vector machines (SVM) [45] and neural networks [46] are also proposed. Huang et al. [47, 48] developed a novel learning technique for single hidden layer feedforward networks (SLFNs) called the extreme learning machine (ELM), that can be utilized in regression and

classification applications [42, 49–51]. Yang et al. [52] proposed a re-enforcement-based deep learning algorithm for multirobot path planning. Table 1 depicts the summary of the literature review.

3. Proposed Framework for Face Detection and Recognition in CCTV Images

The proposed method consists of four significant steps: (i) image acquisition, (ii) image enhancement, (iii) face detection, and (iv) face recognition, as shown in Figure 1. We performed different machine learning techniques for recognition purposes that include random forest, decision tree, K-nearest neighbor (KNN), and convolutional neural network (CNN).

3.1. Image Acquisition. In this phase, we acquire an image. Images need to be restored from the source (usually a hardware source) camera, making it the first step in the workflow sequence because processing is not possible. Our CCTV constantly reads images, which is our preprocessed input.

3.1.1. Camera Interfacing. An Internet protocol (IP) camera, Hikvision DS-2CD2T85FWD-15/18, is used for image acquisition. It is an 8-megapixel camera and captures 15 frames per second video with a resolution of 1248 * 720. Firstly, the camera will capture the image, which will be saved and accessed using some software tool, such as MATLAB. Table 2 shows the CCTV camera specification used for image acquisition.

The face database includes the faces of those whom it will recognize. Because facial recognition involves classification algorithms, each image in the dataset is labeled. Images of each person's faces have their own unique labels. We have more than 41,320 images of 90 people. Thus, the label of these classes (persons) is from 1 to 90. It means that each label has multiple images. Given below is the dataset description.

So, label 1 has 775 images approximately and same as others displayed in the figure (classes on the x -axis and number of images on the y -axis). Figure 2 shows the sample images in the dataset.

3.2. Preprocessing. After the image acquisition, preprocessing of the image prepares it for further handling. Preprocessing includes two main steps: gray scale conversion and edge detection techniques.

3.2.1. Grayscale Conversion. From the camera, we acquire the RGB image (R for red, G for green, and B for blue). An RGB pixel has 1 pixel of red combined with pixels of blue and green. The RGB image made computation expansive as 1 pixel is of 8 bits, so in RGB, it would become 24 bits. In a grayscale image, each pixel is a scalar, so it will be an 8-bit image. So, the equation that converts RGB to grayscale is

$$\text{Grayscale} = 0.3 * R + 0.59 * G + 0.11 * B. \quad (1)$$

Here R, G, and B represent red, green, and blue pixels, respectively.

3.2.2. Canny Edge Detection. The Canny filter detects edges in pictures by detecting abrupt changes in color in photos. We are using this to enhance the edges of the images. The more the advantages are improved, the more accuracy we can achieve in recognizing facial expressions. The filter consists of Gaussian and Sobel filters. Firstly, a Gaussian filter with a predefined value of σ is applied to grayscale images to smooth edge finding.

$$G = \frac{1}{(2\pi\sigma^2)} e^{-(x^2+y^2)/2\sigma^2}. \quad (2)$$

In the second step, the Sobel filter is applied for finding the edges in the images. The filter used for finding the horizontal edges is

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}. \quad (3)$$

For horizontal edges, the filter is

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \quad (4)$$

The horizontal and vertical edges are calculated in order to find all the edges in the filter.

$$A = x = \sqrt{G_x^2 + G_y^2}. \quad (5)$$

The third and last step of the canny edge detector, the hysteresis threshold, is applied to images containing the edges. The threshold is expressed as

$$H = \frac{1}{1 + e^{-x}}. \quad (6)$$

The maximum and minimum thresholds are selected initially. If the pixel's value is greater than the specified threshold, then one is assigned to the pixel, and if the value of the pixel is less than the threshold, then it is set to 0. Another case is when the value is the same as the threshold; it remains the same. Lastly, the edges are added to the original image to get the final enhanced image. Thus, detection and extraction of facial features become easy and increase the efficiency of the overall system.

3.3. Face Detection. The next step after getting the image from the camera is to detect the face from the images by the Viola–Jones algorithm that distinguishes the face and nonface regions. Then, for further processing, the face region is extracted.

TABLE 1: Literature review.

Ref. no.	Algorithm	Accuracy	Dataset
[53]	Principal component analysis, local binary patterns histograms, K-nearest neighbor, and convolutional neural network	85.6%, 88.9% 81.4%, and 98.3%	400 images for 40 persons
[42]	Local binary pattern	93.3% and 90.8%	30 images over 10 people, 5040 images over 120 people
[43]	Convolutional neural network and support vector machine	97.5%	1400 images for 200 persons
[54]	Virtual geometry group (VGG) face model	92.1%	2.6M images over 2.6K people
[55]	Nearest neighbor	87.3%	14,000 images of over 1000 people
[56]	Recurrent regression neural network	95.6%	4207 images for 337 persons
[57]	Binary quality assessment	95.56%	494 414 images for 10 575 persons
[58]	Eigenfaces, Fisherfaces, and Laplacian faces	79.4%, 94.3%, and 95.4%	41 368 images of 68 persons
[59]	SRC, NN, NS, and SVM	98.4%, 72.7%, 94.4%, and 95.4%	4000 images for 126 persons
[60]	Fisher vector space and deep face	93.1% and 97.3%	2.6M images of 2622 persons

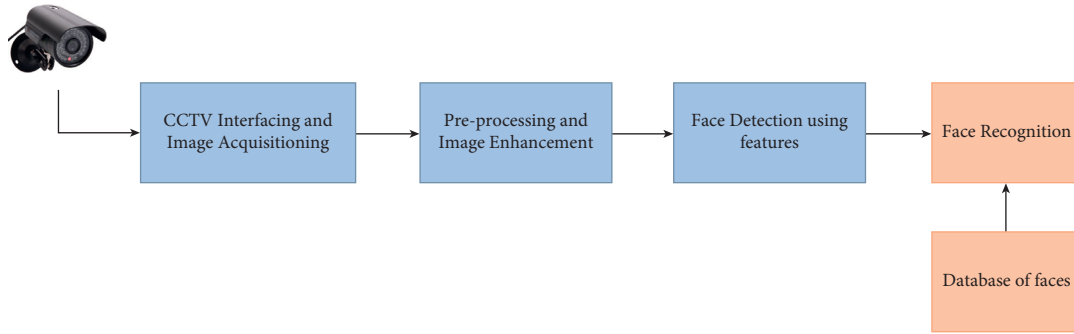


FIGURE 1: Process flow of the proposed system.

TABLE 2: Camera properties.

DS-2CD2T85FWD-15/18
Up to 8 megapixel high resolution
Digital noise reduction
Day and night vision
Max. resolution 3840 × 2160

3.3.1. *Face Detection Using Viola–Jones Algorithm.* Viola–Jones algorithm is the first algorithm that provides competitive object detection rates in real-time. It provides robustness with high detection rates, easy for real-time applications as it can process two frames per second. After applying this, different classification techniques are used to recognize the image. The main steps include the following:

- (1) Haar feature
- (2) Integral image
- (3) Ada boost training
- (4) Cascading classifiers

3.3.2. *ROI Extraction and Resizing.* The face detected by the Viola–Jones technique is extracted and resized as a 40 × 40 image, then used by various feature extraction techniques to find the features.

3.4. *Features Extraction from Detected Face Images.* We have used the principal component analysis (PCA) technique to extract features of the face in order to detect the face in later steps.

3.4.1. *PCA-Based Facial Feature Extraction.* PCA is a technique used to reduce the dimensions of the images in our dataset. It finds the characteristics of images, the difference and variance in pixels in one column from the other [58]. PCA has the following steps as shown in Figure 3:

- (1) Mean of each column

In this step, we have calculated the mean value of each column. The sum of the means of the columns are expressed as

$$\gamma_i = \sum_{i=1}^n \frac{a_{1i} + a_{2i} + a_{3i} + \dots + a_{mi}}{m}. \quad (7)$$

Here, γ_i is the mean of i -th column.

- (2) Covariance matrix

The second step is calculating the covariance of the matrix. The variance of the pixels is calculated as



FIGURE 2: Sample of face images used for recognition.



FIGURE 3: PCA steps for feature extraction.

$$\text{cov}(X_i, X_j) = \frac{1}{n} \sum_{k=1}^m (X_i^k - \gamma_i)(X_j^k - \gamma_j). \quad (8)$$

In the above equation, i is the number of columns in the original image matrix, j is the second column in the image, and k is the number of rows. The following equation shows the result.

$$\begin{bmatrix} \text{cov}(X_1, X_1) & \text{cov}(X_1, X_2) & \dots & \text{cov}(X_1, X_n) \\ \text{cov}(X_2, X_1) & \text{cov}(X_2, X_2) & \dots & \text{cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(X_n, X_1) & \text{cov}(X_n, X_2) & \dots & \text{cov}(X_n, X_n) \end{bmatrix}. \quad (9)$$

(3) Eigenvalues

After the covariance matrix is calculated, the eigenvalues of the covariance matrix can be calculated by.

$$|\text{covariance} - \gamma I_n| = 0. \quad (10)$$

(4) Eigenvectors

Using the eigenvalues calculated in the previous step, we can find the eigenvectors from the following equation:

$$|\text{covariance} - \gamma_i I_i| * X_i = 0. \quad (11)$$

Eigenvalues are the features of an extracted face. These values will be used for recognition.

3.5. Face Recognition Using Machine Learning Algorithms

3.5.1. Random Forest. This is a machine learning approach for solving classification and regression problems. It makes use of ensemble learning, a technique used for solving difficult problems by combining many classifiers. Many

decision trees make up a random forest algorithm. The random forest algorithm's produced "forest" is trained via bagging or bootstrap aggregation. Bagging is a meta-algorithm that enhances accuracy by grouping them together.

3.5.2. Decision Tree. For classification and regression, the decision tree is a nonparametric supervised learning approach. The objective is to learn basic decision rules from data characteristics to construct a model that predicts the value of a target variable. It is a flowchartlike tree structure in which each internal node represents an attribute test, each branch indicates the outcome, and each leaf node (terminal node) carries a class label.

3.5.3. K-Nearest Neighbor. We have used 5, 10, and 15 eigenvectors as our features. The dataset is created with these vectors, and the new face image will pass through all the steps of PCA. Then, we will calculate its distance with the features of other images in the dataset, and the nearest one will be our prediction. We have used the Manhattan distance formula to calculate distance as it is more accurate. The Manhattan distance formula is

$$D(Z, B) = \sum_{x=1}^n |z_x - b_x|. \quad (12)$$

Here, z is for the dataset, and b is for the test image. Then we will check which instance in the dataset has the minimum distance with the test image, which will be our prediction.

3.6. Face Recognition Using Convolutional Neural Network. Convolutional neural networks consist of convolutional layers, pooling layers, and, at the end, a fully connected layer. A CNN has a much different architecture than a simple neural network. It has an input layer, a convolutional layer, a

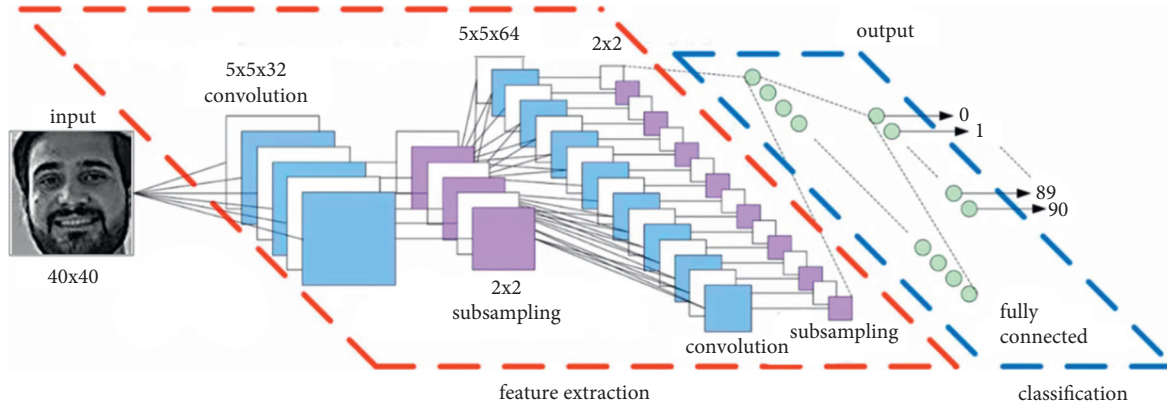


FIGURE 4: The architecture of CNN.

max-pooling layer, and at the end, a fully connected neural network as shown in Figure 4.

We have used Adam optimizer for training in optimizing weights.

3.6.1. Adam Optimizer

$$\begin{aligned}
 v_t &= \beta_1 * v(t-1) - (1 - \beta_1) * g_t, \\
 s_t &= \beta_2 * s(t-1) - (1 - \beta_2) * g_t^2, \\
 \Delta\omega_t &= -\eta \frac{v_t}{\sqrt{s_t + \epsilon}} * g_t, \\
 \omega_{t+1} &= \omega_t + \Delta\omega_t,
 \end{aligned}
 \tag{13}$$

where η : learning rate (0.001), g_t : gradient at time t , v_t : exponential average of the gradient, s_t : exponential average of the square of Gradient, and $\beta_{1,2}$: hyperparameters.

4. Results and Discussion

When we apply PCA, we get eigenvectors; these eigenvectors are our features. We have used different features, such as we have used 5, 10, and 15 eigenvectors.

4.1. K-Nearest Neighbour (KNN) Algorithm Results. Results obtained by simulating different values of k are as shown in Table 3.

Figure 5 with 5 eigenvectors shows the results obtained having a maximum accuracy of 94.7%. When we increase the value of K the accuracy decreased. For $K = 1$, with Manhattan distance, we get approximately 95% accuracy, and with Euclidean distance, we get 89% accuracy.

In Figure 6, PCA features with 10 coefficients are shown. With 10 eigenvectors, we obtained a maximum of 93.7% accuracy with Manhattan distance and with Euclidean distance, we obtained 87.6%. Then the accuracies decreased as the value of K increased. Here we have also noted that Manhattan distance performs better than Euclidean distance. And if the eigenvectors increase, the accuracy also

decreases because the starting eigenvectors show maximum feature importance.

In Figure 7, PCA features with 15 coefficients are shown. Same case here, as the features increase, accuracy decreases. And the same with the value of k .

4.2. Decision Tree Result. For the decision tree, the results obtained for different features are given below, both in tabular form Table 4 and graphical form Figure 8.

4.3. Random Forest Results. The random forest shows the highest accuracy of 93.20% with 5 eigenvectors in Table 5 and Figure 9.

4.4. CNN Results. As in CNN, we must train our dataset. We have trained our data in 5000 steps and obtained 95.7% accuracy with only 30 images for testing and 30 for training.

4.4.1. With 50% Training and Testing Data. We have obtained a maximum of 95.67% accuracy with 50% data of training and testing. We trained it in 4000 steps. In some steps, the training steps, the accuracy increased, and at some points, it decreased, but at the end, we have obtained a maximum accuracy of 95.67%, accuracy as shown in Figure 10.

4.4.2. With 90% Training and 10% Testing Data. Now we have obtained 95% accuracy in this section, maybe because testing data is much less than training. And we have obtained this accuracy in 300 steps, as shown in graph Figure 11.

4.4.3. With 80% Training and 20% Testing Data. Now we have obtained 97.5% accuracy in this section, which may be because testing data is much less than training data. And we have trained data in 5000 steps, as shown in the graph Figure 12.

TABLE 3: Results for KNN.

No. of features	Training data	Numerical methods	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
5	90	Euclidean	89.0115%	89.7889%	79.0841%	76.5861%	75.278%
		Manhattan	94.7623%	90.0457%	88.6113%	86.9326%	86.0086%
	80	Euclidean	87.8664%	80.2338%	77.7842%	75.2137%	73.5403%
		Manhattan	93.7989%	89.0839%	87.6401%	85.9456%	84.8975%
10	90	Euclidean	88.3589%	79.7717%	77.8163%	76.1214%	75.0927%
		Manhattan	93.7989%	89.4494%	88.4072%	87.1582%	77.0927%
	80	Euclidean	86.8185%	77.905%	75.9567%	74.377%	73.4407%
		Manhattan	93.6811%	88.3288%	87.335%	86.0392%	85.3475%
15	90	Euclidean	86.383%	76.6452%	74.7327%	73.293%	72.5651%
		Manhattan	93.9484%	88.2299%	87.3221%	86.1644%	85.618%
	80	Euclidean	84.5773%	74.3589%	72.5164%	71.1934%	70.4926%
		Manhattan	92.8172%	86.6614%	85.9425%	84.7646%	84.1484%

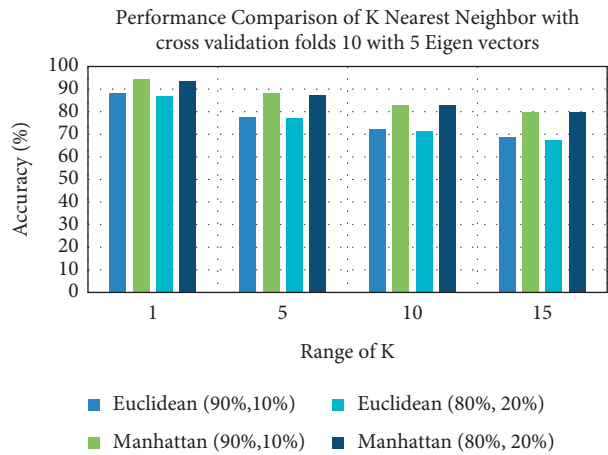


FIGURE 5: Comparison of KNN results for 5 eigenvalues.

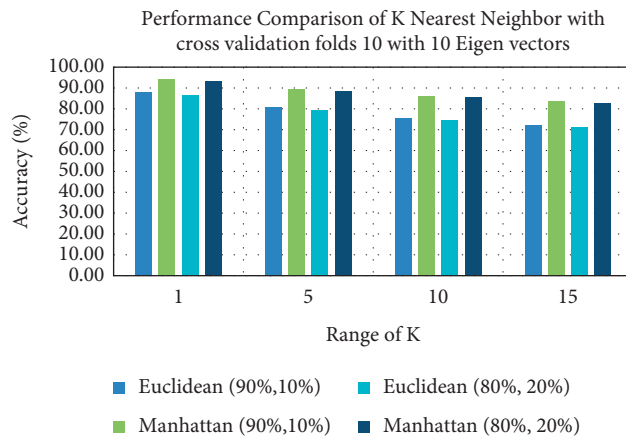


FIGURE 6: Comparison of KNN results for 10 eigenvalues.

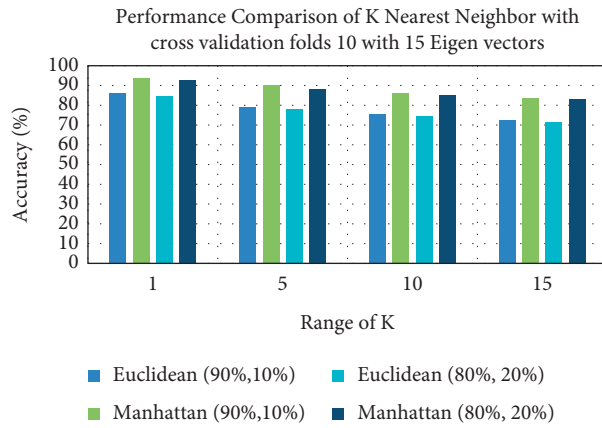


FIGURE 7: Comparison of KNN results for 15 eigenvalues.

TABLE 4: Results for decision tree.

No. of features	Training data	Testing data	Accuracy
5	90%	10%	70.34%
	80%	20%	68.75%
10	90%	10%	68.88%
	80%	20%	68.39%
15	90%	10%	68.64%
	80%	20%	68.28%

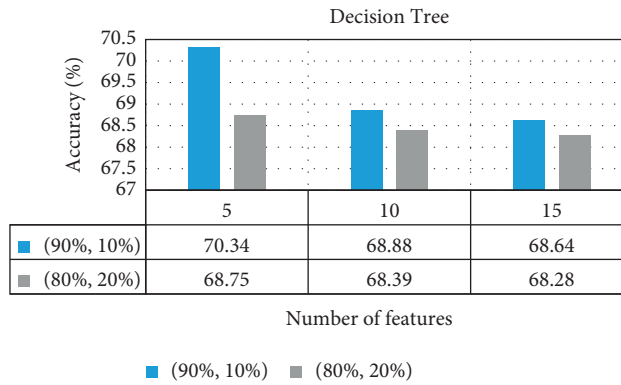


FIGURE 8: Comparison of decision tree results.

TABLE 5: Results for random forest.

No. of features	Training data	Testing data	Accuracy
5	90%	10%	93.20%
	80%	20%	92.65%
10	90%	10%	91.38%
	80%	20%	90.71%
5	90%	10%	89.95%
	80%	20%	88.60%

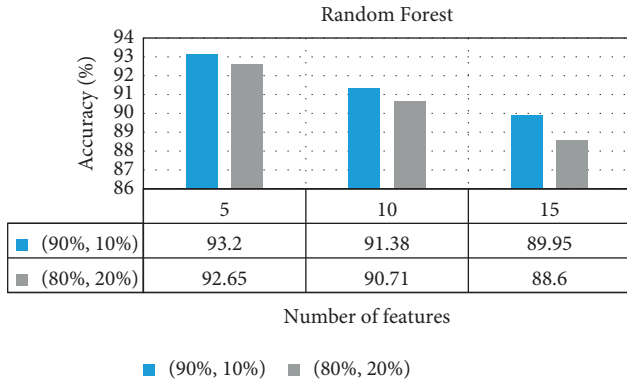


FIGURE 9: Comparison of random forest results.

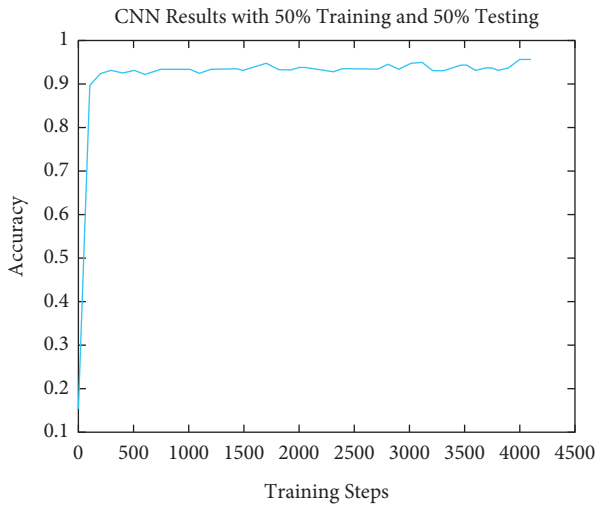


FIGURE 10: Results of 50% training and 50% testing data using CNN.

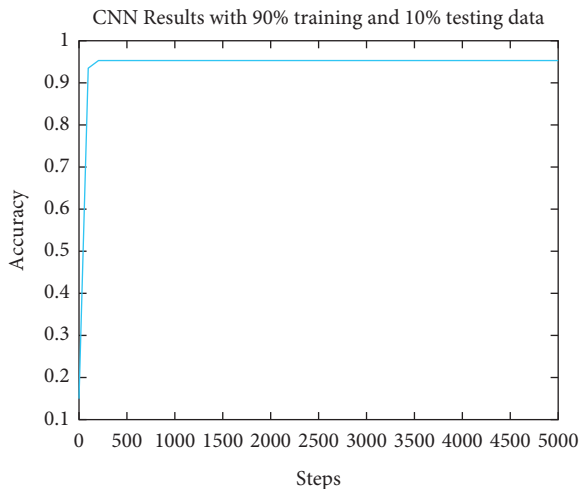


FIGURE 11: Results of 90% training and 10% testing data using CNN.

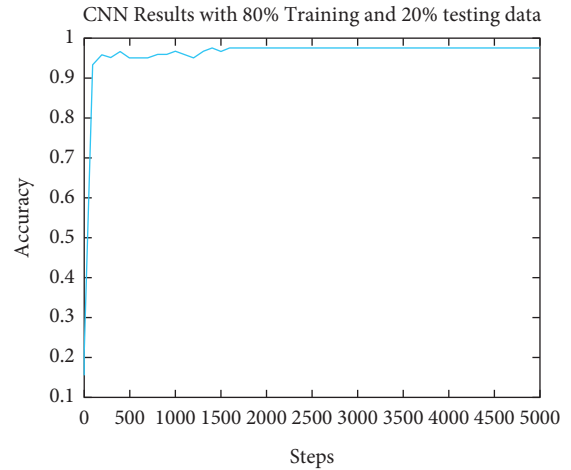


FIGURE 12: Results of 80% training and 20% testing data using CNN.

5. Conclusion

We have developed a framework for automatic face recognition based on CCTV images using different machine learning algorithms in this work. One of the objectives of this work is to collect more than 40,000 face images and compare the performance of algorithms to obtain the highest recognition accuracy. We have implemented different algorithms and have obtained high accuracy for CNN. CNN is much more reliable than PCA with DT, RF, and KNN. KNN is a lazy algorithm, and it checks all the instances in the dataset for prediction while CNN recognizes in very little time from its model. The other reason is that we have used 41,320 images for 90 classes for PCA, and for CNN, we have used ten classes and 30 images per class, and we obtained good accuracy compared to PCA. We collected more than 41,320 images. We will enhance this system by making it a complete security system. We recognize a single face from the image; our next step is to recognize multiple faces in a live-streaming video.

Data Availability

The data are available with the first author and will be provided on request for research purposes.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] A. Deepali, A. Colburn, G. Faigin, L. Shapiro, and B. Mones, "Modeling stylized character expressions via deep learning," in *Proceedings of the Asian Conference on Computer Vision*, pp. 136–153, Springer, Taipei, Taiwan, November 2016.
- [2] S. T. Saste and S. M. Jagdale, "Emotion recognition from speech using MFCC and DWT for security system," vol. 1, pp. 701–704, in *Proceedings of the International conference of*

- Electronics, Communication and Aerospace Technology (ICECA)*, vol. 1, IEEE, Coimbatore, India, April 2017.
- [3] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis et al., "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, 2001.
 - [4] J. Edwards, H. J. Jackson, and P. E. Pattison, "Emotion recognition via facial expression and affective prosody in schizophrenia," *Clinical Psychology Review*, vol. 22, no. 6, pp. 789–832, 2002.
 - [5] S. Umer, B. Chandra Dhara, and B. Chanda, "Face recognition using fusion of feature learning techniques," *Measurement*, vol. 146, 2019.
 - [6] C. Lin and K.-C. Fan, "Human face detection using geometric triangle relationship," vol. 2, pp. 941–944, in *Proceedings of the 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 2, pp. 941–944, IEEE, Barcelona, Spain, September 2000.
 - [7] K. T. Talele and S. Kadam, "Face detection and geometric face normalization," in *Proceedings of the TENCON 2009-2009 IEEE Region 10 Conference*, pp. 1–6, IEEE, Singapore, January 2009.
 - [8] J. Miao, W. Gao, Y. Chen, and J. Lu, "Gravity-center template based human face feature detection," in *Proceedings of the Advances in Multimodal Interfaces - ICMI 2000*, pp. 207–214, Springer, Beijing, China, October 2000.
 - [9] T. V. Vivek and G. R. M. Reddy, "A hybrid bioinspired algorithm for facial emotion recognition using CSO-GA-PSO-SVM," in *Proceedings of the 5th Int. Conf. Commun. Syst. Netw. Technol.*, pp. 472–477, Gwalior, India, April 2015.
 - [10] H. Ali, M. Hariharan, S. Yaacob, and A. H. Adom, "Facial emotion recognition based on higher-order spectra using support vector machines," *Journal of Medical Imaging and Health Informatics*, vol. 5, no. 6, pp. 1272–1277, 2015.
 - [11] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
 - [12] X. Wang, Q. Ruan, Y. Jin, and G. An, "Three-dimensional face recognition under expression variation," *EURASIP Journal on Image and Video Processing*, vol. 6, pp. 281–289, 2014.
 - [13] C. C. Queirolo, L. Silva, O. R. P. Bellon, and M. P. Segundo, "3d face recognition using simulated annealing and the surface interpenetration measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 206–219, 2010.
 - [14] C. C. Queirolo, L. Silva, O. R. P. Bellon, and M. P. Segundo, "3d face recognition using the surface interpenetration measure: a comparative evaluation on the frgc database," in *Proceedings of the International Conference on Pattern Recognition*, pp. 1–5, Tampa, FL, USA, December 2008.
 - [15] N. Alyüz, B. Gökberk, and L. Akarun, "Regional registration for expression resistant 3-d face recognition," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 425–440, 2010.
 - [16] C. Goodall, "Procrustes methods in the statistical analysis of shape," *Journal of the Royal Statistical Society: Series B*, vol. 53, no. 2, pp. 285–321, 1991.
 - [17] B. Gokberk, H. Dutagaci, A. Ulas, L. Akarun, and B. Sankur, "Representation plurality and fusion for 3-d face recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 1, pp. 155–173, 2008.
 - [18] C. Ding and D. Tao, "A comprehensive survey on pose-invariant face recognition," *ACM Transactions on intelligent systems and technology (TIST)*, vol. 7, no. 3, 2016.
 - [19] T. Heseltine, N. Pears, and J. Austin, "Evaluation of image preprocessing techniques for eigenface-based face recognition," in *Proceedings of the 2nd International Conference on Image and Graphics*, vol. 4875, July 2002.
 - [20] S. Vignesh, K. V. S. N. L. Manasa Priya, and S. S. Channappayya, "Face image quality assessment for face selection in surveillance video using convolutional neural networks," in *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, Orlando, FL, USA, December 2015.
 - [21] G. Tudavekar, S. R. Patil, and S. S. Saraf, "Dual-tree complex wavelet transform and super-resolution based video inpainting application to object removal and error concealment," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 4, pp. 314–319, 2020.
 - [22] Y. L. Tian, T. Kanade, and J. F. Cohn, *Handbook of Face Recognition*, Springer, New York, NY, USA, pp. 247–275, 2005.
 - [23] T. Kanade, *Picture Processing System by Computer Complex and Recognition of Human Faces*, PhD. Thesis, Kyoto University, Japan, 1973.
 - [24] X. Xiang, J. Yang, and Q. Chen, "Color face recognition by PCA-like approach," *Neurocomputing*, vol. 152, pp. 231–235, 2015.
 - [25] R. Gottumukkal and V. K. Asari, "An improved face recognition technique based on modular PCA approach," *Pattern Recognition Letters*, vol. 25, no. 4, pp. 429–436, 2004.
 - [26] K. Susheel, V. B. Semwal, and R. C. Tripathi, "Real time face recognition using adaboost improved fast PCA algorithm," 2011, <https://arxiv.org/abs/1108.1353>.
 - [27] C. Li, J. Liu, A. Wang, and K. Li, "Matrix reduction based on generalized PCA method in face recognition," in *Proceedings of the 5th International Conference on Digital Home*, IEEE, Guangzhou, China, November 2014.
 - [28] C. Liu, T. Zhang, D. Ding, and C. Lv, "Design and application of Compound Kernel-PCA algorithm in face recognition," in *Proceedings of the 35th Chinese Control Conference (CCC)*, July 2016.
 - [29] M. Peter, J.-L. Minoi, and H. M. H. Irwandi, *3D Face Recognition Using Kernel-Based PCA Approach*, Springer, Singapore, 2019.
 - [30] L. H. Tran and L. H. Tran, "Tensor sparse PCA and face recognition: a novel approach," 2019, <https://arxiv.org/abs/1904.08496>.
 - [31] M. E. Rala, "Feature extraction using PCA and kernel-PCA for face recognition," in *Proceedings of the 8th International Conference on INFOrmatics and Systems Computational Intelligence and Multimedia Computing Track*, Giza, Egypt, May 2012.
 - [32] M. Abdullah, M. Wazzan, and S. Bo-saeed, "Optimizing face recognition using PCA," 2012, <https://arxiv.org/abs/1206.1515>.
 - [33] M. Sharif, A. Khalid, M. Raza, and S. Mohsin, "Face detection and recognition through hexagonal image processing," *Sind University Research Journal*, vol. 44, no. 2, pp. 541–548, 2012.
 - [34] A. Azeem, M. Sharif, M. Raza, and M. Murtaza, "A survey: face recognition techniques under partial occlusion," *The International Arab Journal of Information Technology*, vol. 11, no. 1, pp. 1–10, 2014.
 - [35] M. Murtaza, M. Sharif, M. Raza, and J. Shah, "Face recognition using adaptive margin Fisher's criterion and linear discriminant analysis," *The International Arab Journal of Information Technology*, vol. 11, no. 2, pp. 1–11, 2014.

- [36] R. Agada and J. Yan, "Edge based mean LBP for valence facial expression detection," in *Proceedings of the 2015 IEEE Int Conf Electr Comput Commun Technol ICECCT*, Coimbatore India, March 2015.
- [37] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1891–1898, IEEE, Columbus, OH, USA, June 2014.
- [38] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," 2014, <https://arxiv.org/abs/1411.7923>.
- [39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [41] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegas, NV, USA, June 2016.
- [42] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [43] W. Zhao, A. Krishnaswamy, and R. Chellappa, "Discriminant analysis of principal components for face recognition," in *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 336–341, Nara, Japan, April 1998.
- [44] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: a literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.
- [45] P. J. Phillips, "Support vector machines applied to face recognition," in *Proceedings of the Advances in Neural Information Processing Systems II*, pp. 803–809, Denver, Colorado, USA, 1999.
- [46] S.-H. Lin, S.-Y. Kung, and L.-J. Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 114–132, 1997.
- [47] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2004)*, vol. 2, pp. 985–990, Budapest, Hungary, July 2004.
- [48] H.-J. Rong, G.-B. Huang, and Y.-S. Ong, "Extreme learning machine for multi-categories classification applications," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2008)*, pp. 1709–1713, IEEE World Congress on Computational Intelligence, Hong Kong, China, June 2008.
- [49] Y. Lan, Y. C. Soh, and G.-B. Huang, "Extreme learning machine-based bacterial protein subcellular localization prediction," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2008)*, pp. 1859–1863, IEEE World Congress on Computational Intelligence, Hong Kong, China, June 2008.
- [50] T. Helmy and Z. Rasheed, "Multi-category bioinformatics dataset classification using extreme learning machine," in *Proceedings of the Eleventh Conference on Congress on Evolutionary Computation (CEC 09)*, pp. 3234–3240, Trondheim, Norway, May 2009.
- [51] C.-W. T. Yeu, M.-H. Lim, G.-B. Huang, A. Agarwal, and Y.-S. Ong, "A new machine learning paradigm for terrain reconstruction," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, pp. 382–386, 2006.
- [52] Y. Yang, J. Li, and L. Peng, "Multi-robot path planning based on a deep reinforcement learning DQN algorithm," *CAA Transactions on Intelligence Technology*, vol. 5, no. 3, pp. 177–183, 2020.
- [53] A. S. Mian, "Representations and matching techniques for 3d free-form object and face recognition," Ph. D. Thesis, , Ph. D. Thesis 2006.
- [54] P. Kamency, M. Benco, T. Mizdos, and R. Radil, "A new method for face recognition using convolutional neural network," *Digital Image Processing And Computer Graphics*, vol. 15, pp. 663–672, 2017.
- [55] T. Bao, C. Ding, M. Zhu, and Y. Wang, *Face Recognition in Real-World Surveillance Videos with Deep Learning Method*, Department of Information and Technology University of Science and Technology of China Hefei, China, 2017.
- [56] A. Gorman, *Cctv Facial Recognition Analysis*, Santa Clara University COEN 150 Project, California, CL, USA, 2011.
- [57] Y. Li, W. Zheng, Z. Cui, and T. Zhang, *Face Recognition Based on Recurrent Regression Neural Network* Research Center for Learning Science, Southeast University, Nanjing, Jiangsu, China, 2017.
- [58] J. Yu, K. Sun, F. Gao, and S. Zhu, "Face biometric quality assessment via light cnn," pp. 25–32, 2018, *Pattern Recognition Letters* 107.
- [59] X. Yan, S. Hu, P. Y. Niyogi, and H. J. Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 328–340, 2005.
- [60] M. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on Neural Networks*, vol. 13, pp. 1450–1464, 2002.

Research Article

A Deep Learning Framework for Leukemia Cancer Detection in Microscopic Blood Samples Using Squeeze and Excitation Learning

Maryam Bukhari ¹, Sadaf Yasmin ¹, Saima Sammad ², and Ahmed A. Abd El-Latif ³

¹Department of Computer Science, COMSATS University Islamabad, Attock Campus, Attock, Pakistan

²Allama Iqbal Open University, Islamabad, Pakistan

³Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shibin Al Kawm 32511, Egypt

Correspondence should be addressed to Maryam Bukhari; maryambukhari09@gmail.com

Received 27 November 2021; Revised 1 January 2022; Accepted 5 January 2022; Published 31 January 2022

Academic Editor: Nouman Ali

Copyright © 2022 Maryam Bukhari et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Leukemia is a fatal category of cancer-related disease that affects individuals of all ages, including children and adults, and is a significant cause of death worldwide. Particularly, it is associated with White Blood Cells (WBC), which is accompanied by a rise in the number of immature lymphocytes and cause damage to the bone marrow and/or blood. Therefore, a rapid and reliable cancer diagnosis is a critical requirement for successful therapy to raise survival rates. Currently, a manual analysis of blood samples obtained through microscopic images is done to diagnose this disease, which is often very slow, time-consuming, and less accurate. Furthermore, in microscopic analysis, the appearance and shape of leukemic cells seem very similar to normal cells which make detection more difficult. In the past decades, deep learning utilizing Convolutional Neural Networks (CNN) has provided state-of-the-art approaches for image classification problems; however, there is still a gap to improve their efficacy, learning procedure, and performance. Therefore, in this research study, we proposed a new variant of deep learning algorithm to diagnose leukemia disease by analyzing the microscopic images of blood samples. The proposed deep learning architecture emphasizes the channel associations on all levels of feature representation by incorporating the squeeze and excitation learning that recursively performs recalibration on channel-wise feature outputs by modeling channel interdependencies explicitly. In addition, the incorporation of the squeeze-and-excitation process enhances the feature discriminability of leukemic and normal cells, and strategically assists in exposing informative features of leukemia cells while suppressing less valuable ones as well as improving feature representational power of deep learning algorithm. We show that piling these learning operations of squeeze and excite together in a deep learning model can improve the performance of the model in diagnosing leukemia from microscopic images based on blood samples of patients. Furthermore, an extensive set of experiments are performed on both cropped cells and full-size microscopic images as well as with data augmentation to address the problem of fewer data and to further boost their performance. The proposed model is tested on two publicly available datasets of blood samples of leukemia patients, namely, ALL_IDB1 and ALL_IDB2. The suggested deep learning model exhibits good results and can be utilized to make a reliable computer-aided diagnosis for leukemia cancer.

1. Introduction

Leukemia is a type of cancer that has a very high mortality rate [1]. It is accompanied by the malicious cloning of abnormal white blood cells (WBC) and is hence referred to as a malignant hematological tumor [2]. Usually, the human

body comprises three cell types: red blood cells, white blood cells, and platelets, as shown in Figure 1. The supply of oxygen from the heart to all tissues is often the responsibility of red blood cells [3]. They account for up to half of the total volume of blood. Likewise, the white blood cells play a pivotal role in the immune system of the human body and

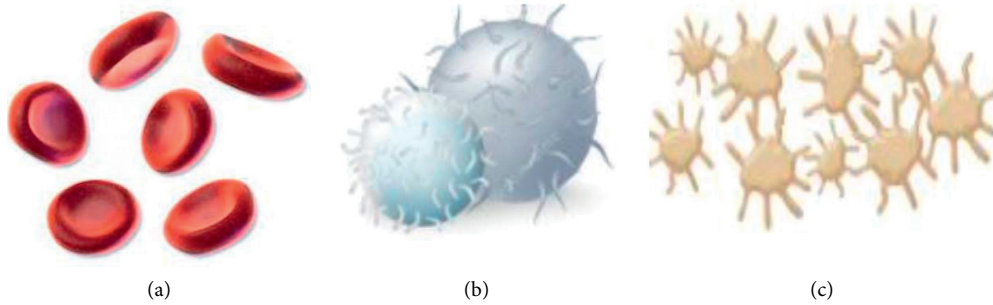


FIGURE 1: Types of blood cells in the human body. (a) Red blood cells. (b) White blood cells. (c) Platelets.

act as a defense wall from numerous infections and diseases [4]. As a result, the correct categorization of these white blood cells is critical to determine the nature of the disease. They are divided according to the composition of the cytoplasm. Lymphocytes are one of the categories of white blood cells and their disorders caused Acute Lymphoblastic Leukemia (ALL) [5]. Generally, leukemia is categorized into two subtypes known as acute leukemia and chronic leukemia. Without any particular treatment, the overall recovery rate of acute leukemia is barely three months while the onset period of chronic leukemia is more than acute leukemia. Acute lymphocytic leukemia (ALL) is one of the widespread types of acute leukemia responsible for about 25% of all childhood cancers [6]. It originates in the lymphatic system, which generates the blood cells. At the beginning stage, it appears in the bone marrow and is subsequently disseminated throughout the human body. In a healthy individual, the growth of WBC is dependent on the requirements of the body, but in the context of leukemia, they are formed abnormally while becoming ineffective.

Usually, the dark-purple-like color of these leukemic cells makes it easy to identify them but the assessment and further processing become extremely sophisticated due to the pattern and texture-based variations. Leukocytes are a class of cells that vary dramatically from each other. They might be recognized by their shape or size, but one problematic factor is that they are flanked by some other elements of the blood which includes red blood cells and platelets. The shape of lymphocytes is somewhat regular, and their nuclei have uniform and flat borders. The lymphocytes also called lymphoblast in patients of ALL have a quite minimal uniform border and tiny cavities in the cytoplasm known as vacuoles as well as inside the nuclei spherical particles are referred to as nucleoli. The disease becomes more acute as the stated morphology becomes more prominent. This might also result in premature death if the intervention is neglected and if its diagnosis is done later in the disease's progression. The age of a patient has a vital risk factor influencing prognosis because the probability of having ALL is greater in children aged 7–8 years. This probability is eventually reduced up to the age of 20 and starts to rise again around the age of 50. According to information reported by Ref. [7], 5930 new cases had the disease ALL in the United States in 2018, and around 1500 individuals, including both children and adults, are likely to die from ALL. Furthermore,

according to data reported in Ref. [8], in 2015, there were around 876,000 individuals who experienced ALL worldwide, and it triggered 111,000 deaths. The medication of acute lymphoblastic leukemia has evolved to make great development in the past 50 years. The survival rate of patients has increased up to 70% with early assessment and intervention [9]. Hence, at the earlier stages of acute lymphoblastic leukemia, its diagnosis and effective treatment are very essential. One of the important tools employed by the medical operators to diagnose acute lymphoblastic leukemia is referred to as morphology. With this diagnostic tool, it can be observed that a patient is suffering from acute lymphoblastic leukemia whenever the bone marrow has a considerable amount of cancer cells (B-lymphoblast cells). The fundamental factor to diagnose acute lymphoblastic leukemia is precisely discerning of cancer cells from normal cells (B-lymphoid precursors). On the contrary, the visual appearance of cancer cells is somewhat very similar to normal cells in microscopic images, which makes it hard to distinguish between them. Furthermore, it is very crucial for the hematologist to diagnose the presence of leukemia along with its specific form to prevent medical problems and determine the optimal treatment of leukemia disease. The screening of leukemia by a specialist through human blood samples is a critical and time-consuming task.

To tackle such challenges, quantifiable analysis of different blood samples is performed in the computer-aided-diagnosis (CAD) systems that are designed by employing either machine learning or deep learning approaches. There exist numerous research studies in which leukemic cancer detection is performed. With regard to traditional machine learning methods, a discriminative set of leukemic cells' features are first extracted followed by the process of classification [10]. Some researchers have suggested the segmentation process so that the accurate features are extracted from the region of interest, i.e., segmented lymphocyte images [11]. These segmentation methods include k-means, watershed, as well as HSV color-based segmentation [11]. In these segmentations, the extra elements present in the blood are eliminated and thus only details related to WBC involving lymphocytes and lymphoblast are drawn [10]. Specifically, for leukemic disease, the segmentations are generally divided into pixel-based, region-based, as well as shape-based approaches [12]. It has been observed that segmentation strategies based on K-means and edge-based

are widely used to segment out the cells of a blast from various smears of blood [13]. Recently, it is reported that by combining thresholding and morphological techniques superior segmentation is achieved [14]. Furthermore, the complex images having variations such as low contrast, noise-sensitivity are challenging to segment accurately using these approaches [15]. Furthermore, a lot of feature extraction approaches are employed in leukemic cell analysis. These include morphological features, such as shape, and edge features. In addition to these, textural, color, and GLCM, as well as geometrical and statistical features are also employed [10]. Some research studies have performed the hybridization of these features to further enhance the performance [16, 17]. Likewise, different classifiers have been exploited to perform the classification among leukemic and normal cells. These include Support Vector Machines (SVM), K -nearest neighbor (KNN), Random Forest (RF), Naive-Bayes, etc. [10, 18, 19]. All of these traditional machine learning approaches show significant results; however, these approaches require a lot of parametric steps as well as accurate analysis and feature engineering before the classification phase. When opposed to a good data representation, a bad data portrayal frequently results in worse performance [20].

Subsequently, with the emergence of deep learning, a lot of problems and challenges in image analysis have been solved as these approaches employed automated feature engineering. In the recent past, the automated diagnosis of several diseases with the science of computer vision emerges as a potential research area [21, 22]. Image recognition and segmentation using deep learning are some of the imperative elements in the technology of computer vision [23]. One of the most frequently used deep neural networks in computer vision is Convolutional Neural Networks (CNNs) [24–27]. These CNNs possess a great deal of self-learning capability, adaptability, and generalization power and are heavily used in medical imaging problems and IoT-based systems [28, 29]. Conventional image identification techniques need hand-crafted features extraction followed by categorization, while the CNN-based methods only require the image data which are given as an input to the network, and the task of image classification is achieved by their self-learning property [30]. Besides this, they have also required a substantial amount of data as well as computing power to train. In many circumstances, the total number of data samples is inadequate for a CNN to train from the beginning. In such situations, transfer learning is employed to exploit the potential of CNNs, while minimizing the computing cost.

Particularly, for the diagnosis of leukemia cancer, a lot of research studies have been proposed based on deep learning frameworks. In such methods, some research studies have suggested CNN-architectures with different depth levels and the setting of layers to perform leukemia cancer detection [31, 32]. It has been observed that deep learning through transfer learning method is the most widely used approach in leukemia cancer detection [33]. Several different pre-trained models including AlexNet, MobileNet, ResNet, Vgg16, etc., have been exploited [34, 35]. In addition, it is

indicated that deep learning methods work better than traditional machine learning methods in leukemia cancer detection [36]. However, in terms of feature learning, accuracy, and effectiveness, these techniques still have some shortcomings and need to be addressed. The emergence of new CNN architectures is a difficult engineering endeavor that often necessitates the choice of several new hyper-parameters and layer settings. Furthermore, in existing studies, the feature discriminability among leukemic and normal cells is not well-considered; hence, what if the learning or feature representation of deep learning algorithm is improved by adding more discriminating power to further boost up the performance? Secondly, most approaches are based on transfer learning methods and have reported very accurate results. Is there, however, a method other than transfer learning such as to increase the performance of a simple deep learning algorithm? This research study attempts to answer these questions, by suggesting a deep learning algorithm whose representational power is improved by incorporating squeeze-and-excitation learning. The main aim of this article is to provide a deep learning solution with the goal of addressing different challenges such as assisting timely as well as an accurate diagnosis by empowering the feature discriminability among leukemic and normal cells. Furthermore, it is worth noting that improving deep learning algorithms is an ongoing research challenge among numerous researchers. Convolutional neural networks (CNNs) and traditional deep learning models are excellent algorithms for solving a wide range of visual problems. Recent research [37], however, indicates that the representational power of traditional CNN architecture can be improved by adding modules that accurately describe dynamic and nonlinear relationships among channels using global details. Further, these modules aid in the learning of the model and considerably improve its accuracy. Hence, deploying and suggesting better deep learning solutions is one of the secondary objectives of this study. In addition, the proposed technique does not require a prior segmentation and all its parametric steps adjusted by the user to further perform the leukemia detection, rather it defines a fully automated solution to leukemia cancer detection.

More specifically, in this research article, we proposed an effective learning-based deep learning model for leukemia disease detection using microscopic blood samples-based image modality. The feature representation at every layer of feature extraction and representation is improved by emphasizing the interdependencies among channels [37]. This can be accomplished by the squeeze and excitation learning process in which we first squeeze the features acquired by convolution layers from microscopic blood samples to generate the channel descriptor. This descriptor combines the wide-range distribution of outcomes provided by channel-wise features and causes the feature details global receptive field of the model to be utilized by bottom layers. Similarly, after the squeeze, the excitation process further enhances the features in which the activations related to samples are being learned for every channel by a self-gating process depending on channel reliance, by regulating the

excitation of each channel. Both types of learning operations empower the feature representation of blood samples of leukemia disease which ultimately results in an inaccurate diagnosis of leukemia cancer. In addition, these operations also help in improving the feature discriminability among leukemic and normal cells. Furthermore, the total number of blood samples in both of the datasets is not appropriate for the training of the model; therefore, an excessive augmentation is also performed to boost the performance. Besides, we have demonstrated the results of leukemia diagnosis by the proposed Model using both cropped and full-size microscopic images, respectively. Some samples images from ALL_IDB1 and ALL_IDB2 are shown in Figure 2. The research has the following contributions:

- (i) An all-inclusive efficient and improved representational power-based deep learning model is proposed to diagnose the leukemia disease from microscopic blood samples
- (ii) A feature discriminability among leukemic and normal cells in blood samples is enhanced by global information embedding in squeeze operation and recursive recalibration using the excitation process
- (iii) During the feature extraction process, the proposed improved deep-learning model emphasizes relevant features of leukemic cells while suppressing irrelevant ones, resulting in improved performance
- (iv) The proposed model shows significant improvements over the traditional deep learning model and can be integrated with any Internet of Medical Things (IoMT)-based systems

The rest of the paper is partitioned into several sections: Section 2 presents some existing work, Section 3 describes the proposed method in detail, Section 4 explains experimentation results, and Section 5 concludes the paper followed by future direction.

2. Related Work

Over the decades, several strategies for automated leukemia identification on microscopic images have been established in the literature. These strategies include the traditional machine learning classifiers and deep learning algorithms. However, some approaches have employed ensemble machine learning as well as hybrid deep learning methods for leukemia cancer detection.

2.1. Conventional Machine Learning Approaches. In the existing literature, machine learning methods are extensively employed for leukemia cancer detection. These methods are generally categorized into several steps such as pre-processing, feature extraction, followed by classifications. However, some methods also involve segmentation and feature selection procedures to further improve the performance. For instance, Singhal et al. employed the Support Vector Machine (SVM)-based approach for automated diagnosis of Acute Lymphoblastic Leukemia (ALL) [13]. This

diagnosis can be accomplished by extracting the geometric features as well as texture features using Local Binary Patterns (LBP). The experimental outcomes of their proposed method demonstrate that texture features surpass the geometric features and exhibit an accuracy of 89.72%, which is a little bit high than the 88.79% accuracy given by geometric features. Similarly, Mohamed et al. proposed another method in which the color space of every microscopic image is transformed into YCbCr followed by acquiring the values of Gaussian distribution of Cb and Cr [38]. Later on, different sets of features are computed including morphological, texture, and size to train the classifier. Their designed strategy attained 94.3% accuracy by using the Random Forest as a classifier for the detection of two classes of leukemia (ALL and AML) and Myeloma. Mohapatra et al. suggested a framework for screening acute leukemia in pigmented blood samples and microscopic images of bone marrow [39]. After the extraction of features from microscopic images, a model based on the ensemble approach is trained for classification. In contrast to other traditional classifiers, such as naive Bayesian (NB), K -nearest neighbor, radial basis functional network (RBFN), multilayer perceptron (MLP), and SVM, their proposed ensemble attained 94.73% performance accuracy along with above 90% resultant values of average sensitivity and specificity. Subsequently, Patel and Mishra designed the framework using unsupervised learning in which leukemia identification is performed using k -means clustering [40]. With the help of this, leukemia detection is estimated by computing the proportion. Bhattacharjee et al. suggest an approach for the identification of acute lymphoblastic leukemia that employs the watershed transforms preceded by morphological transformations for segmentation. After extraction of morphological features, the Gaussian Mixture Model (GMM) and Binary Search Tree (BST) are employed to carry out the classification. Their proposed approach shows 95.56% accuracy. Mishra et al. proposed a model based on Linear Discriminant Analysis (LDA) for the classification of leukemia disease by employing Discrete Orthogonal Stockwell Transform (DOST) [41] for feature extraction from blood sample images [42].

2.2. Deep Learning Approaches. In the context of deep learning, many researchers adopt and design several architectures for the automated classification of leukemia cancer. These deep learning methods are further classified into traditional standalone deep learning models or the transfer of learning-based approaches. For instance, Shaheen et al. suggested the AlexNet-based deep learning model to diagnose Acute Myeloid Leukemia (AML) using blood samples in the form of microscopic images [34]. They have compared the performance of their presented approach with the LeNet-5 model in terms of accuracy, quadratic loss, recall, and precision. Their proposed method shows 98.58% accuracy along with 88.9% of the microscopic images being accurately classified with 87.4% accuracy. Rehman et al. suggested a CNN architecture comprising several convolutional and max-pooling layers for leukemia cancer

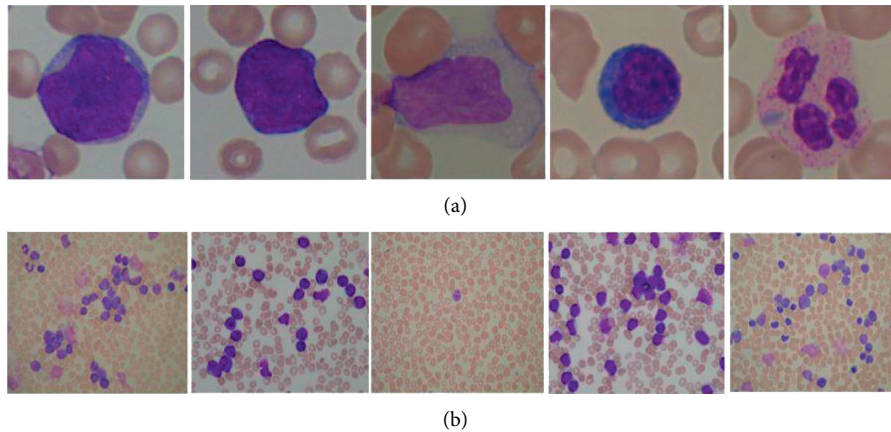


FIGURE 2: Some samples from ALL_IDB1 and ALL_IDB2 databases. (a) The first row corresponds to cropped microscopic images of blood samples. (b) The second row corresponds to full-size microscopic images of blood samples of “Acute lymphoblastic leukemia” and “non acute lymphoblastic leukemia”.

detection [31]. Prior to providing data samples as an input to the CNN algorithm, all microscopic samples are first pre-processed to be converted into HSV color space followed by a segmentation process to obtain the required region-of-interest. In their work, an accuracy of 97.98% is reported for leukemia cancer detection. Zakir Ullah et al. suggest the attention-based deep learning model to extract the most relevant features of leukemic cells [43]. However, the designed model is based on VGG16, which is one of the pretrained deep learning models. Their proposed method utilizes the segmented leukemic (malignant) and normal cell images and validation is performed using a 7-fold cross-validation. Pansombut et al. suggested a CNN model called ConVNet to detect ALL and all its subtypes [44]. They have compared their designed framework with traditional machine learning techniques including SVM, multi-layer perceptron (MLP), and Random Forest (RF). They employed two kinds of datasets with a total number of images in the collection being 363. Shafique and Tehsin designed a deep learning model to categorize leukemia disease into six different classes [2]. They employed a pretrained AlexNet to undertake binary classification on 368 images to avoid having to train from the beginning. A classification algorithm for WBC employing both transfer and deep learning is designed by Habibzadeh et al. [45] In the first stage, they have performed the preprocessing steps on the dataset followed by the process of feature extraction. In the last stage, the classification procedure is carried out through Inception and ResNet model. A total of 352 images are used in their work to validate the model’s accuracy. Ahmed et al. also designed an efficient approach for the categorization of White Blood Cell Leukemia [46]. In their work, the deep features are extracted using VGGNet and reduced by Swarm Optimization. This bio-inspired optimization technique plays a pivotal role in optimizing the deep features for accurate and reliable classification of White Blood Cell Leukemia. This work also reports encouraging results. One of the latest research in leukemia detection is the study by Bibi et al. [47]. They proposed an Internet of Medical things (IOMT)-based framework [48] along with the assistance of cloud computing and diagnostic devices that are connected through Internet resources. The designed system

enables real-time synchronization for screening and treatment of leukemia in patients as well as medical operators and professionals, thereby potentially decreasing the work and effort for patients and doctors. Their automated system is based on Dense Convolutional Neural Network (DenseNet-121) [49] and Residual Convolutional Neural Network (ResNet-34). The performance of the proposed method is validated on two different benchmark datasets referred to as LL-IDB and ASH image bank and the reported results are exceptional.

2.3. Hybrid Deep Learning Approaches. Other than employing standalone deep learning models, some research studies designed the hybrid deep learning frameworks to perform the leukemia cancer detection. For instance, Yu et al. proposed a hybrid method in which ResNet50 [50], VGG16 [51], and VGG19 [51], based on state-of-the-art convolutional neural networks (CNNs), are employed to carry out the automated identification of cells [52]. The outcomes of their proposed approach are compared with conventional machine learning approaches, i.e., K -Nearest Neighbors (KNN), Logistic Regression (LR), Support Vector Machine (SVM), and Decision Tree (DT). Their proposed technique shows 88.50% accuracy for cell recognition. Mourya et al. also design a hybrid model based on deep learning architecture in which dual CNN architectures are employed to enhance performance accuracy [53]. The proposed approach is validated on 636 blood samples of healthy and ALL cells and exhibits 89.70% accuracy. Furthermore, Jiang et al. employed the ViT-CNN referred to as vision-transformer CNN based on ensemble learning [54]. The proposed technique is able to distinguish the normal and cancer cells that are helpful in the detection of Acute Lymphoblastic Leukemia (ALL). In their work, both vision transformer and CNN-based model are integrated to draw the extensive set of cells features into distinct ways to obtain the improved classification outcomes. They have also enhanced the data by employing enhancement-random sampling (DERS) to overcome the challenges of the unbalanced

dataset. Their proposed algorithm shows outstanding results of 99.03% which proves the effectiveness of the proposed method as a CAD system for Acute Lymphoblastic Leukemia (ALL). Kassani et al. designed a hybrid approach in which VGG16 and MobileNet are combined to extract the deep features followed by classification of Leukemic B-lymphoblast [55]. Their proposed approach is enriched with various data augmentation methods and attained 96.17% accuracy, 95.17% sensitivity, and 98.58% specificity. Furthermore, Zoph et al. merge the two deep learning models, namely, NASNetLarge [56] and VGG19 to categorize the leukemic B-lymphoblast cells and normal B-lymphoid precursor cells, with a detection performance of 96.58% [57]. Their proposed model effectively diagnoses acute lymphoblastic leukemia and illustrated that in contrast to a single model, the ensemble learning is much better.

In addition, optimization-based algorithms are also employed for Leukemia disease classification. Krishna et al. proposed Chronological Sine Cosine Algorithm (SCA)-based deep learning model to detect the acute lymphocytic leukemia from the blood sample images and attained a 98.70% value of accuracy [58]. For instance, Tuba et al. employed the Generative Adversarial optimization (GAO) [59] for the detection of acute lymphocytic leukemia and achieved a 99.66% resultant value of accuracy [60]. Similarly, Saif et al. employed both Artificial Neural Network (ANN) and Genetic Algorithm (GA) [61] and carried out the segmentation of acute lymphoblastic leukemia utilizing local pixel information and reported 97.07% accuracy [15]. Acharya et al. proposed to design an acute lymphoblastic leukemia diagnosis by employing image segmentation and data mining techniques [62] and achieved 98.60% accuracy. Our suggested simple deep learning model employs squeeze-and-excitation learning and addresses the problem of morphological similarity among leukemic and normal cells, thereby increasing the accuracy of the traditional deep learning model and categorizing the images as healthy or unhealthy blood samples.

3. Methodology

The design overview of the proposed methodology is depicted in Figure 3. The proposed framework begins with the acquisition of microscopic images of blood samples. Later on, the data augmentation techniques are employed to overcome the problem of fewer data since in deep neural networks more data are required for their training and superior performance. Lastly, a deep CNN architecture-based squeeze and excitation learning is proposed to diagnose leukemia from the inputted microscopic images of blood samples. Each step is explained in-depth in the following subsections of methodology:

3.1. Acquisition of Data. The data utilized in this work to evaluate the model's performance were obtained from the Acute Lymphoblastic Leukemia Image Database for image processing (ALL-IDB). We used both of the datasets given by this database, ALL-IDB1 and ALL-IDB2. These are

publicly available datasets that comprise microscopic images of blood samples. The database focuses on Acute Lymphoblastic Leukemia (ALL), which is a potentially deadly type of leukemia. It is most frequently found in childhood, with the highest prevalence between the ages of 2 and 5 years. In the datasets, the labeling of ALL lymphoblast is annotated by experienced oncologists. All microscopic images are captured by a Canon Power Shot G5 camera which was used in conjunction with an optical laboratory microscope. The range of magnifications of the microscope is from 300 to 500 during data collection. All microscopic blood sample images are in the jpg. format, along with a 24-bit color depth. More precisely, the first dataset ALL-IDB1 is comprises 108 images including 39000 components of blood, wherein the lymphocytes have been annotated. Similarly, in the second dataset, the regions of cells are cropped from the whole microscopic image. Except for the image dimensions, ALL-IDB2 images have comparable grey level features to ALL-IDB1 images.

3.2. Data Augmentation. CNN exhibited cutting-edge performance in a variety of tasks. However, the amount of the training data has a significant influence on CNN performance [24, 25, 63]. Acquiring sufficient clinical images is a challenging task, due to data privacy concerns especially in the field of medical imaging. On the other hand, if machine learning and deep learning models were trained using the original images as well as with augmented samples, they might be more generalizable. In various image-based studies with CNNs, several ways of data augmentation have diminished the error rate of the network by providing speculation. The dataset used in this research includes a wide variety of microscopic blood sample images, but the quantity of blood samples in both datasets is quite limited. Hence, to tackle the problem of small dataset size, and overcome the problem of overfitting, we have employed several types of data augmentation to artificially increase the data for training of the model. In this research study, the augmentation type of rotation at 60 degrees, 90 degrees, and random shift in range (-1.0, 1.0) is employed.

3.3. Proposed CNN Architecture. The proposed deep CNN architecture is described below in detail:

3.3.1. Convolutional Layers and Max-Pool Layers. Generally, the two fundamental operations of the convolutional neural networks (CNNs) include the convolution and max-pooling layers mimicking a variety of substantially complex cells in the visual cortex. Besides this, CNNs have a localized perceptive area, hierarchical organization, feature extraction, and classification phase that can automatically learn the appropriate feature and categorization process, and has a significant implication in the domain of computer vision. In the proposed model, microscopic images of blood samples are preprocessed by data augmentation and directly fed into the proposed deep learning model design to craft the localized features. Consider a microscopic blood sample

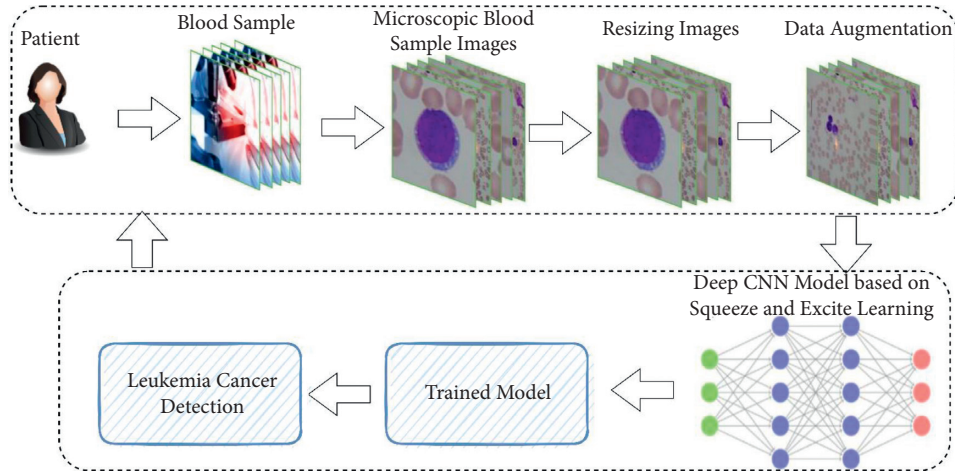


FIGURE 3: An overview of the proposed methodology.

image of a patient of dimension $M \times N$ with a kernel size of $w \times h$ which is convolved over the image to generate a collection of features maps of dimensions $o_w \times o_h$, as shown in equations (1) and (2):

In equations (1) and (2), the zero-padding in the direction of both width and height is denoted by p_w and p_h while the value of stride in both vertical and horizontal directions is denoted by s_h and s_w , respectively. The input image of size $224 \times 224 \times 3$ is provided as an input to the first convolutional layer. Continuing to follow the convolution layers, a pooling layer is also used which contributes to diminishing the computing and spatial necessities of the activation function and makes the proposed model to be translation invariant and functions interdependently on each layer of the input data and spatially downscales it.

3.3.2. Integrating Squeeze-and-Excitation Learning. In order to raise a network's representational potential, we have employed the Squeeze-and-Excitation learning to strengthen the spatial encoding of the model [37]. This learning emphasizes the channels' relationships by recursively recalibrating the outputs of channel-wise features as well as simultaneously considering channel interdependencies. It can be added as a computing unit that can be formed for any particular transformation such as $F_{tr}: X \rightarrow U, X \in \mathbb{R}^{H \times W \times C}, U \in \mathbb{R}^{H \times W \times C}$. For the sake of clarity, F_{tr} is supposed to be a convolutional function in the accompanying notation. Consider the $V = [v_1, v_2, v_3, \dots, v_C]$, which signifies the learned set of filter kernels, wherein v_c indicates the c^{th} filter's parameters. Therefore, the outcomes of the F_{tr} can be specified as $U = [u_1, u_2, u_3, \dots, u_C]$, in which the value of u is defined in equation (3):

$$u_c = v_c * X = \sum_{s=1}^C v_c^s * X^s. \quad (3)$$

In equation (3), $*$ represents the convolution, $v_c = [v_c^1, v_c^2, \dots, v_c^C]$ and $X = [x^1, x^2, x^c]$ (the bias terms are ignored for simplicity), while v_c^s denotes the filter

of size 2D, and hence indicates the singular v_c channel that operates on the equivalent X channel. The correlations among channels are implicitly encoded in v_c because the result can be computed by summing all the channels but they are jumbled with the spatial correlation acquired by the filters. Here the main objective is to assure that the model is able to improve its sensitivity to relevant features so that they can be accessed by some transformations while silencing less relevant ones. This can be accomplished by modeling the channel interdependencies explicitly and recalibrating kernel outputs in two stages, squeeze and excitation operation, before passing them into the succeeding transformation. The pictorial representation of squeeze and excite operations is shown in Figure 4.

$$o_w = \frac{M - w + 2p_w}{s_w} + 1, \quad (1)$$

$$o_h = \frac{N - h + 2p_h}{s_h} + 1, \quad (2)$$

(1) Embedding of Global Information by Squeeze Operation. The signal to every channel is taken into account in final features to address the channel dependencies. Since every learned kernel applies with a local receptive field, each component of the transformation result U is not capable of employing the context-related information beyond this region. The bottom layers of the model where the sizes of the receptive fields are smaller also become the major cause of this problem. This can be addressed by adding spatial details of squeeze global into a descriptor channel. More precisely, this can be accompanied by adding the average pooling, i.e., global to produce the statistics of channel-wise information. Mathematically, a statistic $z \in \mathbb{R}^C$ is formed by contracting U through $H \times W$ spatial dimensions in which the c^{th} component of z is computed by the following equation:

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^H u_c(i, j). \quad (4)$$

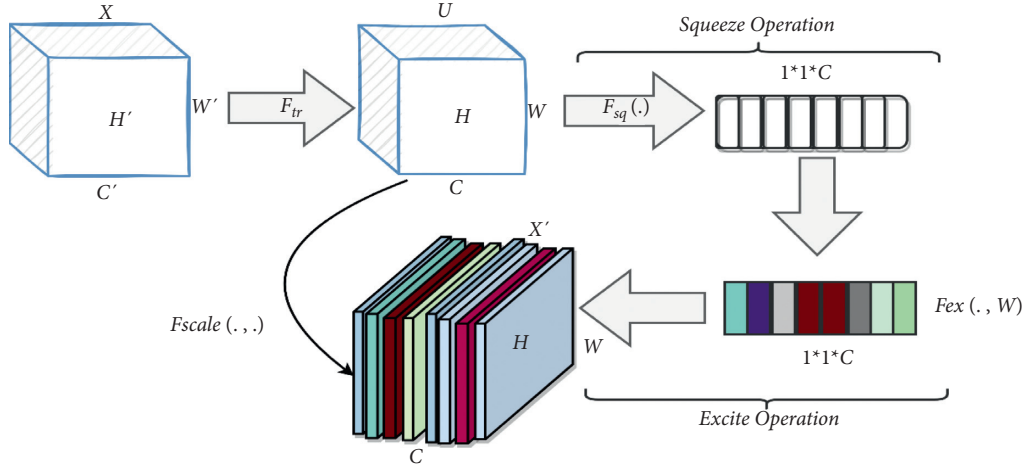


FIGURE 4: A pictorial representation of squeeze and excite operation.

The output of transformation U could well be interpreted as a group of descriptors that are local as well as for those whose statistics represent the complete microscopic image of the blood sample. Usually, in the feature engineering part, such information is useful [64–66]. The aggregation technique used here is the global average pooling.

(2) *Recalibration through Excitation Operation.* In order to completely acquire the channel-wise dependencies, another operation, namely, excitation is done to take advantage of the information attained in the squeeze operation. This aim can be accomplished by fulfilling two important conditions. The first one is flexibility indicating that the model should be able to comprehend the non-linear relationships among channels. Similarly, the second criteria are that the model should learn a nonexclusive relationship because we would like to guarantee that different channels are allowed to be noticed in contrast to one-hot activation. For this purpose, a gating mechanism is employed along with sigmoid activation.

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)). \quad (5)$$

In equation (5), the term δ denotes the ReLu activation, $W_1 \in \mathbb{R}^{C \times cr}$ and $W_2 \in \mathbb{R}^{C \times cr}$. To incorporate the generalization and reduce the complexity of the model, the gating process is parameterized by designing a bottleneck along with two FC layers with nonlinearity, i.e., the parameters W_1 and r denoting reduction ratio forms a layer of dimensionality reduction followed by ReLu activation. Lastly, by the use of parameters W_2 , a dimensionality growing layer is added. The block's final result is computed by rescaling the result U of the transformation with the following activations:

$$\hat{x}_c = F_{scale}(u_c, s_c) = s_c \cdot u_c, \quad (6)$$

where $\hat{X} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_c]$ and $F_{scale}(u_c, s_c)$ denote channel-wise multiplication among the feature maps $u_c \in \mathbb{R}^{H \times W}$ and the scalar s_c . The activations are served as channel weights that are tuned to the particular descriptor z . In this manner, the squeeze and excite operation assists to

improve the feature learning discriminability by introducing dynamics that are conditional on the input.

3.3.3. *Network Architecture.* The architecture of the proposed model is amalgamated with convolution, max-pool, as well as squeeze and excite operations, as indicated in the above sections. It consists of a stack of these layers configured with the best set of parameters to efficiently perform leukemia cancer detection. The network begins at the input layer, where microscopic blood samples of dimension $224 \times 224 \times 3$ are provided as input. Later on, this input is propagated to convolutional and max-pool layers. This convolution layer operated on the image with a kernel size of 3×3 to provide low-level optimum interpretations of the image that are effective for any image classification task. As depicted in Figure 5, this process of obtaining advantageous representations including both mid and high levels is further strengthened by employing subsequent convolution layers of the same kernel size. Furthermore, the total number of filters in each of the nonpadded convolution layers is configured to 32, 64, and 128, respectively. Following each convolution layer, a ReLu [67] activation function is implanted to bring the nonlinearity in the network learning. In addition, we have supplemented the network with batch normalization after every convolution operation to normalize the data and improve the performance. More specifically, there are three convolution layers, with each followed by ReLu [67] activation and batch normalization. After batch-normalization, the input is passed through squeeze and excite block. The squeeze operation enables the global information embedding while the recalibration process is attained through excite operation as described in the above sections. Subsequently, the max-pooling layer with a window size of 2×2 is used after every batch normalization layer. The stride size during pooling is set to 1. The sizes of feature maps after every Convolution \rightarrow ReLu \rightarrow BatchNormalization \rightarrow MaxPool are $222 \times 222 \times 16$, $52 \times 52 \times 32$, and $26 \times 26 \times 64$, respectively. After this, a global max-pooling layer is added to reduce the extracted feature dimensions followed by two dense layers with hidden units set to 128 and 1. The activation function on the second last dense layer is

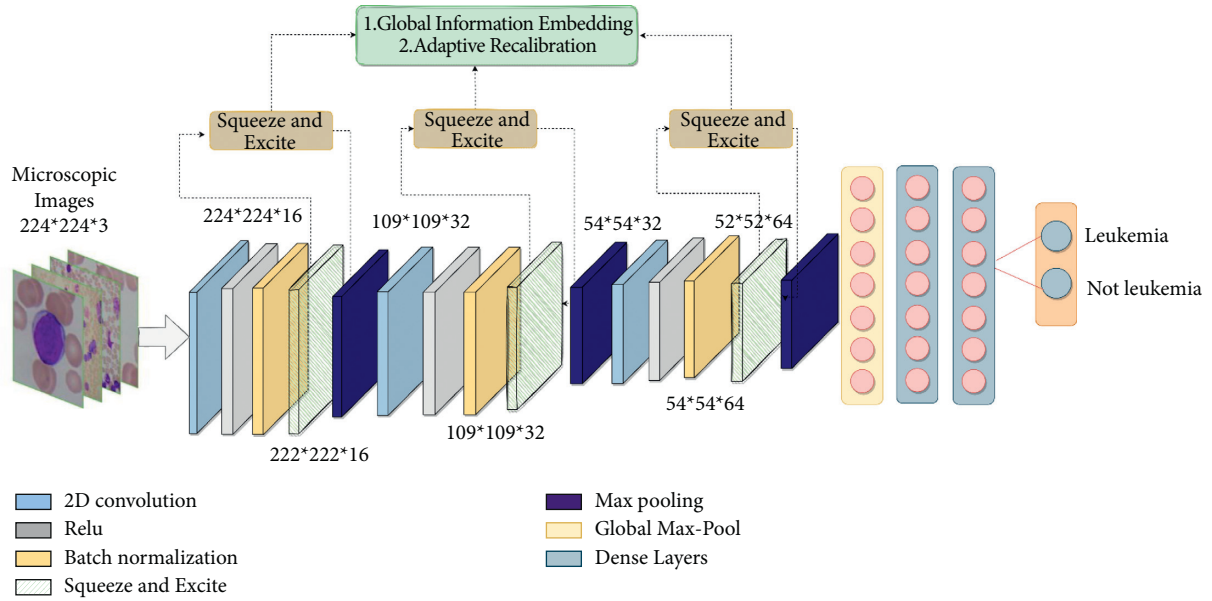


FIGURE 5: An architecture diagram of the proposed deep learning model.

ReLU while on the last layer it is sigmoid. In addition, the model is trained with loss function “binary_crossentropy” as well as weight optimizer Adam with a learning rate of 0.001. The graphical representation of the traditional deep learning model is depicted in Figure 6 while Figure 4 shows the architecture of the proposed model. All of the architecture aspects stated above are included in the traditional deep learning model illustrated in Figure 6 to increase its performance.

4. Experiments and Results

This section discusses the findings of the designed model in various experimental scenarios, followed by discussions and comparisons. In addition, the proposed model is implemented using Python with Keras deep learning framework and all simulations are run on Google Colab with a 12GB NVIDIA Tesla K80 GPU. The experimental setup includes two datasets, and performance is validated both alone and in combination. All of the parameters are defined by trial and error procedure, and the results are reported with the best set of parameter settings.

4.1. Evaluation Criteria. The criteria used to evaluate the performance of the proposed model are determined by the following metrics:

4.1.1. Accuracy. This metric measures the total number of classes accurately predicted by the trained model out of all categories, i.e., an Acute Lymphoblastic Leukemia (ALL) and not Acute Lymphoblastic Leukemia (ALL). This measure indicates how many patients are diagnosed with leukemia and those who are not. The higher the value of accuracy, the more accurate is the model [68–71]. The equation of accuracy is shown in the following equation:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (7)$$

4.1.2. Precision. Out of all positive cases, this metric measures the proportion of true positives [72]. In the instance of leukemia disease, it is the ability of the model to accurately highlight those patients who have leukemia disease. Mathematically, it is defined as in the following equation:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (8)$$

4.1.3. Recall. The recall assesses how the model is correctly highlighting the leukemia disease patients based on the overall relevant data. It is computed by the following equation:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (9)$$

4.1.4. FScore. This metric measures the overall efficiency of the model by integrating both values of recall and precision.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

In equations (7)–(10), the term TN represents the True negative, TP represents the true positive, FP represents the false positive, and FN denotes the false negative.

4.2. Results of the ALL_IDB1 Database. To assess the performance of the proposed model, we first validate this model with the ALL_IDB1 database. The whole database is partitioned into two nonoverlapping collections of train and test samples with a ratio of 80:20. As previously stated, the total

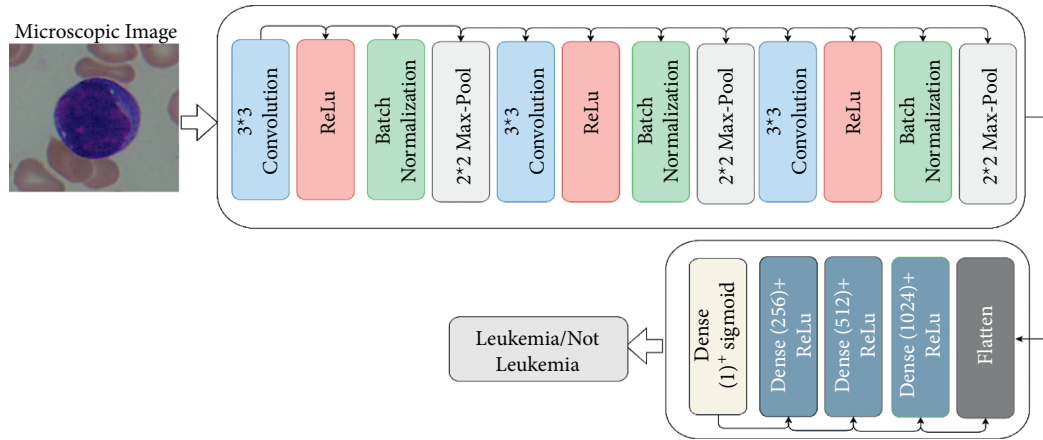


FIGURE 6: Leukemia cancer detection using traditional deep learning model.

number of microscopic blood samples of both ALL and without ALL is very less for training. Hence, we have employed data augmentation techniques to increase the total number of blood samples for training, as shown in Figure 7. The total number of train and test blood samples for both ALL and without ALL classes is provided in Table 1. Subsequently, after data augmentation, the train set sufficiently contains a large number of samples that are used to train the model. These augmented images are used to train the model. The proposed model extracts the features of leukemic cells from convolution and max-pool layers. At each feature level representation of images, the squeeze and excitation learning is incorporated to improve the representational capacity of the model by modeling the interdependencies among the channels explicitly with the help of these extracted convolution operations. The results on the ALL-IDB1 dataset are shown in Table 2. As illustrated in Table 2, it is observed that the proposed model is performing very efficiently in diagnosing the patients having leukemia disease with 100% accuracy.

Furthermore, the additional evaluation indicators that comprise precision, recall, and F Score values are 100%, 100%, and 100%, respectively. In addition, we have also verified the reliability of the model class-wise on the ALL-IDB1 database. In this examination, the performance is analyzed on individual classes, i.e., patients with ALL and without ALL. It has been revealed that the model is also performing very accurately in individual instances, as shown in Table 2. Furthermore, in the ALL-IDB1 dataset, the number of test samples is very limited and does not contain diverse variations. Hence, we validate the model three different times, and each time we formed the train and test set differently by random shuffling. After division, we augmented the train set only. Alternatively, we have also charted the confusion matrix of this experiment, which is depicted in Figure 8 (first image). For each class category present in the dataset, the confusion matrix illustrates the overall efficiency of the model. It is evident from the outcomes that the proposed model shows better performance in classifying the microscopic blood samples into ALL and without ALL classes. The model learns well due to an adaptive

recalibration of channel responses by considering interdependencies among channels. The squeeze and excitation learning brings the dynamics in the input to empower the feature discrimination. Both operations of squeeze and excitation can be included by global information embedding and recursive recalibration.

4.3. Results of the ALL_IDB2 Database. In the second phase of validation, we have employed the second dataset, namely, ALL_IDB2. In this dataset, the total number of microscopic blood samples is also very insufficient for the training of the proposed model. Hence, the same procedure of data augmentation is applied to this dataset. Later on, the train set with an excessive type of variations in the images is used to train the model. The total number of training and testing instances for this experimental setup is given in Table 3. Furthermore, all of the hyperparameters of the proposed model are the same as we set with the first database. The results of the proposed model on this database are shown in Table 2. As done previously with the first experiment, the features of microscopic images of blood samples are drawn from the convolutional and pooling layers whose learning improves by incorporating the squeeze and excitation operations. Table 2 shows that the model is also exhibiting good scores on this dataset. The overall accuracy obtained by the model in the first run is about 96% while values of precision, recall, and F -Score are 96%, respectively. Similarly, the results of the second and third runs in which we divided the train and test sets with different random shuffling are also encouraging, i.e., accuracy with the second run is 98% while with the third run it is 99.98%. Besides this, the performance of the model is also examined by demonstrating the class-wise performance of the model, as shown in Table 2. Another noteworthy thing to be mentioned here is that in this dataset, the regions of cells are cropped from the microscopic images while in the first experiment we have employed the full microscopic images of blood samples. The proposed model shows the best results on both types of image settings. Subsequently, the confusion matrix is also drawn for the experiment in which cropped cell images are used. Figure 8

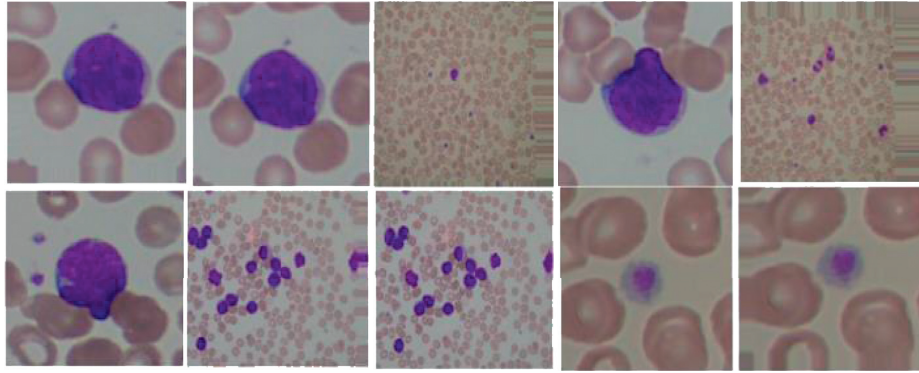


FIGURE 7: Results of data augmentation on microscopic images.

TABLE 1: Training samples and testing samples on the ALL_IDB1 dataset.

No#	NOT ALL	ALL	Total
Training	800	920	1720
Test	13	9	22
Total	813	929	1742

TABLE 2: Results of the proposed model on both ALL1_IDB1 and ALL_IDB2.

Exp#	Class-wise performance	Run#	Dataset	Accuracy (%)	Precision (%)	Recall (%)	FScore (%)
01	ALL	01	ALL_IDB1	100	100	100	100
02	Not ALL	01	ALL_IDB1	100	100	100	100
03	ALL	01	ALL_IDB2	96	96	96	96
04	Not ALL	01	ALL_IDB2	96	97	97	97
05	ALL	02	ALL_IDB1	100	100	100	100
06	Not ALL	02	ALL_IDB1	100	100	100	100
07	ALL	02	ALL_IDB2	98	100	96	98
08	Not ALL	02	ALL_IDB2	98	96	100	98
09	ALL	03	ALL_IDB1	100	100	100	100
10	Not ALL	03	ALL_IDB1	100	100	100	100
11	ALL	03	ALL_IDB2	99.98	99.03	99.87	99.44
12	Not ALL	03	ALL_IDB2	99.98	99.24	99.63	99.43
Results by integrating both datasets i-e ALL_IDB1 and ALL_IDB2							
12	Not ALL	01	ALL_IDB1 + ALL_IDB2	97.06	97.12	97.01	97.06
13	ALL	01	ALL_IDB1 + ALL_IDB2	97.06	97.03	97.21	97.11
14	Not ALL	02	ALL_IDB1 + ALL_IDB2	99	100	97.00	99.00
15	ALL	02	ALL_IDB1 + ALL_IDB2	99.24	97.00	100	99.00
16	ALL	03	ALL_IDB1 + ALL_IDB2	99.33	99.3	99.24	99.26
17	Not ALL	03	ALL_IDB1 + ALL_IDB2	99.01	99.36	99.00	99.17

(second image) shows the confusion matrix for this experiment.

4.4. Results of Combining Both Datasets. Furthermore, in the third experiment, we have combined the microscopic images of both datasets to create more diversity and increase the number of test images. Similarly, in this experiment, we have also performed the data augmentation to increase the training size. The total number of testing and training samples of both ALL and not ALL classes is given in Table 4. As done previously for both of the datasets separately, the train set with extensive data augmentation is given as an

input to the proposed model. The model extracts the features of leukemic and normal cells from cropped and full-size, respectively. The accuracy achieved in this scenario is also encouraging. Similarly, the recall, precision, and F-Score values are also better. The recall value of 99.24% is achieved on ALL classes with the third experiment while it is 97.01% in the first experiment, respectively.

Besides this, the confusion matrix of this experiment is also plotted, as shown in Figure 8 (third image). Moreover, the training loss and accuracy curves are also plotted for all of the experiments, as shown in Figure 9. The learning curves indicate that the model performance on epoch 5, in terms of accuracy is leading towards the best values of the accuracy.

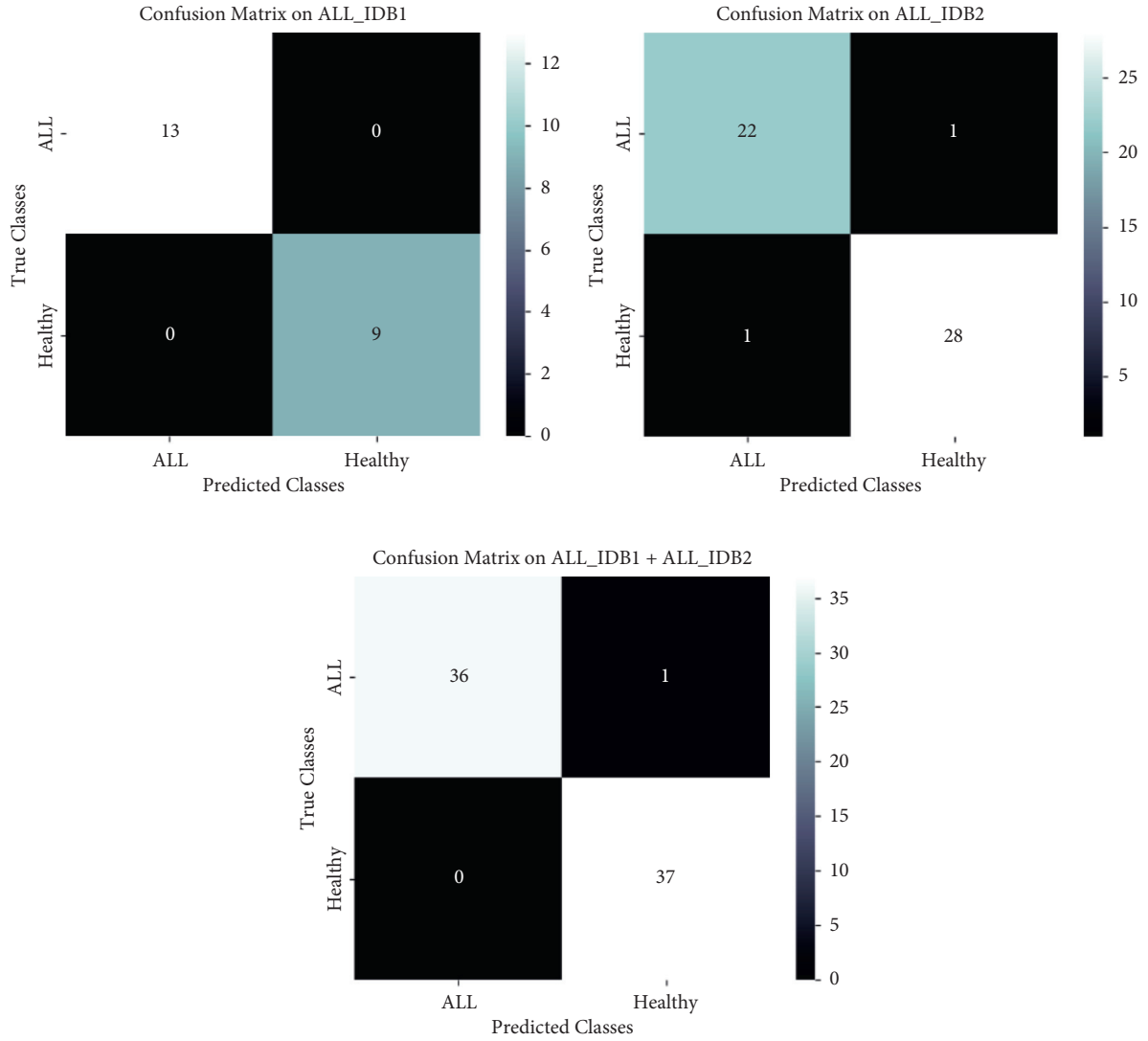


FIGURE 8: Confusion matrices of experiments performed on ALL_IDB1 and ALL_IDB2 datasets.

TABLE 3: Training samples and testing samples on the ALL_IDB2 dataset.

No#	Not All	ALL	Total
Training	1030	1050	2080
Test	27	25	52
Total	1057	1075	2132

TABLE 4: Training and testing samples by combining both datasets.

No#	Not ALL	ALL	Total
Training	1036	1022	2058
Test	41	33	74
Total	1077	1055	2132

Similarly, on epoch 10, the loss values are approximating near to zero. This behavior demonstrates the effectiveness of the proposed model during the learning process. In addition, the receiver operating curves (ROC) are also drawn for both of the databases. It is a likelihood curve that shows

a true-positive rate (TPR) versus a false-positive rate (FPR) at various thresholds. ROC is a very accurate metric to examine the efficiency of the binary classifier, but it can also be plotted for multi-class problems [73]. The ROC curve demonstrates the trade-off between sensitivity (or TPR) and

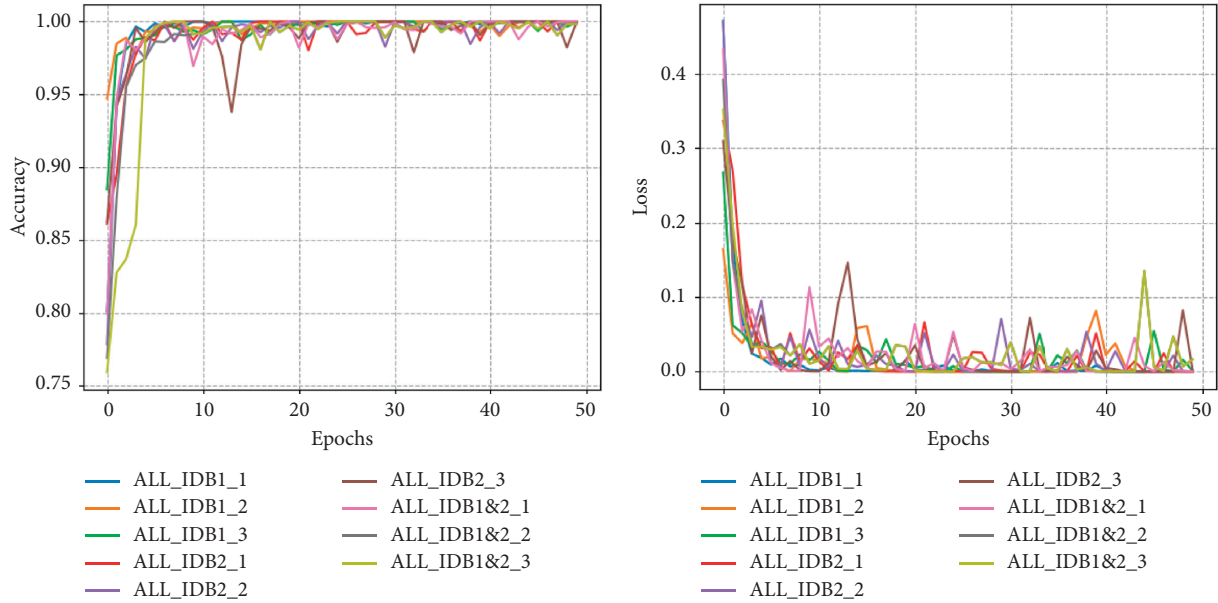


FIGURE 9: Lose and accuracy curves of training under different experiments on both datasets.

specificity ($1 - \text{FPR}$). Classification methods that produce curves nearer to the top-left corner function are the best. The nearer the curve gets to the ROC space's 45-degree diagonal, the slower and less reliable the assessment. The ROC curves are depicted in Figure 10 for all of the experiments.

In addition to the above results, we have compared the results of the proposed CNN architecture with the traditional deep learning model, as shown in Figure 6. In a traditional deep learning model, we generally have several convolution, max-pool, and dense layers. Here, in this study, we empower the representational capabilities of this traditional deep learning model by incorporating squeeze and excite operations. The results shown in Table 5 provide the details regarding improvements in terms of accuracy, precision, recall, and F Score values of the proposed model over the traditional deep learning model. We tested both models three times, each time randomly shuffling the whole dataset to create the train and test sets. It has been demonstrated that the proposed model shows 5.5% average accuracy improvement over the traditional deep learning model.

Furthermore, the precision, recall, and F Score values are also encouraging and high than traditional deep learning models. In addition, the average loss value on the test set for traditional CNN is 1.44 while for the proposed this loss value is 0.117. Furthermore, we have also examined the effect of data augmentation on the performance of both models. Generally, the effective training of deep learning models requires a very large amount of data. On the contrary, with less data, the underlying model is less generalizable and more prone to overfitting issues. Hence, in order to show the influence of data augmentation for this particular problem, the results are listed in Table 6. In the first experiment, we train the deep learning model three times by random shuffling of the data without any form of data augmentation. It is observed that without data augmentation the results are less in terms of accuracy. More specifically, the accuracy of

the traditional deep learning model is 88.6% while with the proposed model, it is 94%. However, when the data augmentations are performed over the data, the results are increased and models learn better, exhibiting accuracies of 92.8% and 98.43%, respectively.

4.5. Comparison with Existing Works. Finally, we have compared the results of the proposed model with the existing work on leukemia cancer detection, as shown in Table 7. For instance, Ahmed et al. proposed a CNN-based architecture to classify the different types of leukemia, both acute and chronic [32]. Their proposed architecture achieved an average accuracy of 88.25. In their work, the comparison is also performed with some traditional machine learning approaches such as Naive Bayes, decision tree, K -nearest neighbor, and support vector machines (SVM). Furthermore, the authors Shafique and Tehsin have suggested the transfer learning approach using the AlexNet model for the detection of acute lymphocytic leukemia (ALL) and its subtypes [2]. For only ALL detection, their proposed approach exhibits 99.50% accuracy, which is remarkable. Jothi et al. performed the ALL classification in which they employed the optimization-based backtracking algorithm to segment the leukemic cells from a given microscopic image [74]. Later, a different set of features are extracted such as morphological, color, and statistical, etc., followed by a feature selection. Finally, the classification between healthy and leukemic cells is done by the Jaya algorithm, which is population-based meta-heuristic optimization. Their proposed frameworks exhibit 99% accuracy. Subsequently, Mishra et al. also performed the ALL classification by improved feature extraction method using 2D-discrete orthonormal S-transform [42]. This method extracts an extensive set of relevant texture features which is further reduced by PCA and LDA-based algorithms. Finally, one of

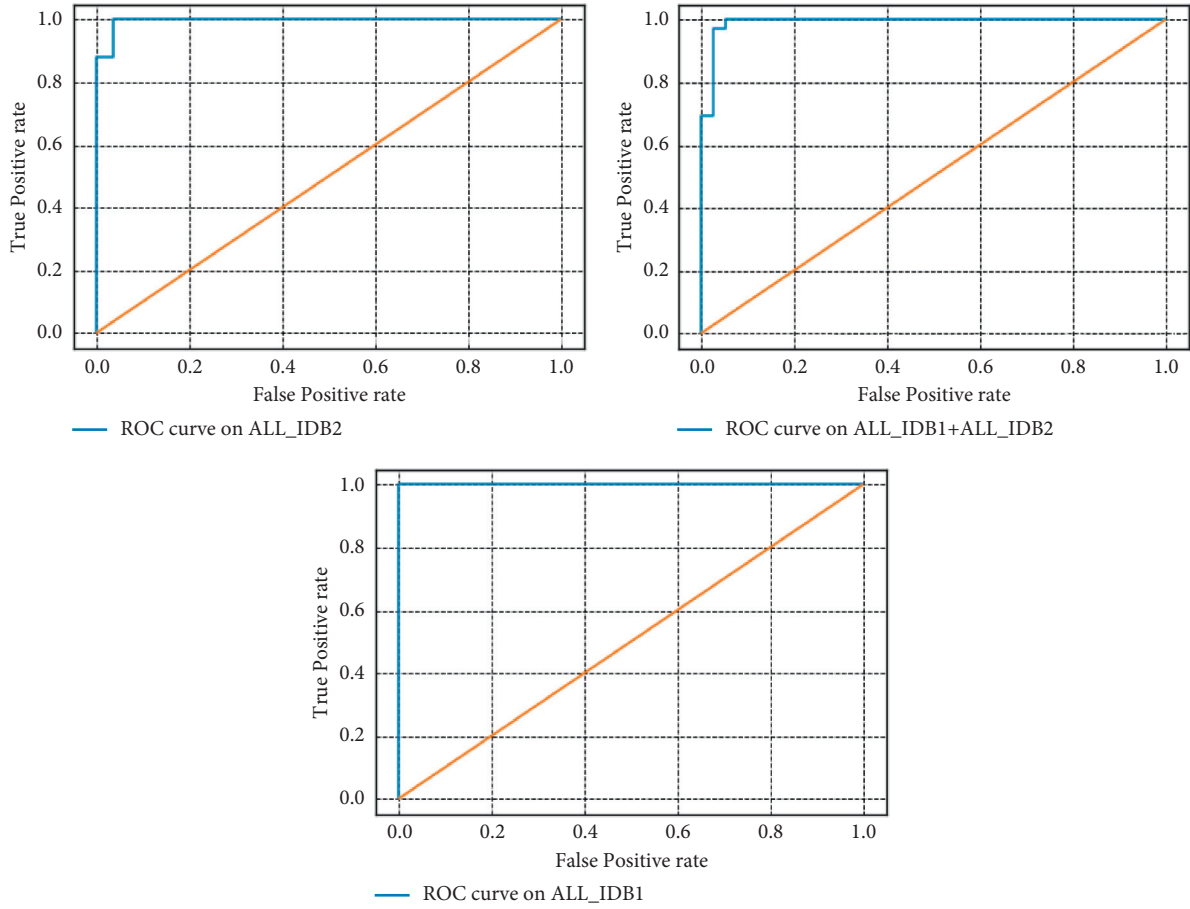


FIGURE 10: ROC curves of ALL_IDB1, ALL_IDB2, and ALL_IDB1+ALL_IDB2 dataset.

TABLE 5: Comparison with traditional deep learning model using ALLIDB1+ ALLIDB2 database.

Comparison of traditional deep learning model with the proposed model						
Run#	Model	Loss value	Accuracy (%)	Precision (%)	Recall (%)	FScore (%)
1	CNN	0.92	95	95	94.5	94.5
2	CNN	1.67	92	92	92	91.5
3	CNN	1.73	91.4	90.5	92	91.5
Average	CNN	1.44	92.8	92.5	95.28	92.5
1	Proposed	0.16	97	97	97	97
2	Proposed	0.092	99	98.5	98.5	98.5
3	Proposed	0.099	99	99	99	99
Average	Proposed	0.117	98.3	98.16	98.16	98.43

TABLE 6: Effect of data augmentation on both models.

Effect of data augmentations						
Augmentation	Model	Loss value	Accuracy (%)	Precision (%)	Recall (%)	FScore (%)
Yes	CNN	1.44	92.8	92.5	95.28	92.5
No	CNN	0.57	88.6	90.1	88.1	88.3
Yes	Proposed	0.117	98.3	98.16	98.16	98.43
No	Proposed	0.15	94	94	94	94

the popular classifiers Adaboost is used to classify the leukemic and normal cells with an accuracy of 99.66%. Furthermore, Jiang et al. proposed a vision transformer-based

convolutional neural network (ViT-CNN) for acute lymphocytic leukemia detection [54]. To get superior classification results, their proposed ViT-CNN ensemble model can

TABLE 7: Comparison with existing work.

Sr. No.	Authors	Methods	Dataset	Accuracy (%)
1	Ahmed et al. [32]	CNN	ALL_IDB	88.25
2	Shafique and Tehsin [2]	AlexNet-based transfer learning	ALL_IDB	99.50
3	Jothi et al. [74]	Jaya, SVM	ALL_IDB	99
4	Mishra et al. [42]	DOST, PCA, LDA	ALL_IDB1	99.66
5	Jiang et al. [54]	Vision transformer- based CNN	ISBI2019	99.03
6	Agaian et al. [75]	SVM with cell energy feature	ALL_IDB1	94
7	Tuba and Tuba [60]	Gao-based methods	ALL_IDB2	93.84
8	Jha and Dutta [58]	SCA-based deep CNN	ALL_IDB2	98.70
9	Proposed	Squeeze and excitation based CNN	ALL_IDB1	100
10	Proposed	Squeeze and excitation based CNN	ALL_IDB2	99.98
11	Proposed	Squeeze and excitation based CNN	ALL_IDB1 + ALL_IB2	98.3

extract features from cell images in two fundamentally distinct approaches. Their proposed method achieves an accuracy of 99.03%, respectively. In addition, Agaian et al. designed the cell energy feature-based approach to ALL feature extraction followed by SVM classifier to perform the classification [75]. Their proposed framework shows 94% accuracy in ALL detection. Tuba and Tuba perform acute lymphocytic detection using five shapes and six texture-based features. [60]. The classification is performed by SVM whose parameters are tuned with a generative adversarial-based optimization algorithm. Their proposed techniques show 93.84% accuracy. Moreover, Jha and Dutta proposed a chronological sine-cosine algorithm (SCA)-based deep CNN model to classify the ALL images [58]. The SCA algorithm is employed to find the best weights of the deep learning model to classify the microscopic images. Their proposed techniques demonstrate 98.70% accuracy. In comparison with all these approaches, some studies utilize the deep learning models, some are based on transfer learning mechanisms, some utilized optimization-based methods, while some have employed the traditional machine learning approaches. All of them perform excellently well, but the results of deep learning-based methods are more accurate and better. Hence, the proposed framework is also based on the deep learning method in which performance is boosted up by incorporating the squeeze and excitation learning. The results are presented in the above sections as well as according to comparison made in Table 7, and it is observed that the proposed approach is good in classifying leukemia cancer.

The major reason behind the improvements is the representational power of the model, as indicated in Ref. [37], i.e., the representational power of traditional CNNs is enhanced by explicitly considering the interdependencies among convolutional features' channels. This mechanism is accomplished by adding squeeze-and-excitation operations into the layers of deep learning. More precisely, the squeeze operations consolidate the widespread distribution of outputs acquired from channel-wise features. This is followed by an excitation operation, which takes the extracted information by squeeze operation as input to completely learn channel-associations with the recalibration process. These operations strengthen the model's representational power. They also improve its feature learning method in order to extract more compact and discriminative features, which is

a critical prerequisite for microscopic image analysis. Furthermore, the proposed model is also light-weighted in terms of network depth and layers as well as a number of trainable parameters. Furthermore, the suggested method is a simple and enhanced deep learning approach that does not require any post-processing or pre-processing techniques to identify leukemia cancer. On the contrary, when there are more different and complicated differences or variations in microscopic blood samples, it may be necessary to upgrade network configurations as well as network depth, since the model presently does not have deeper depths as it is a light-weighted model.

5. Conclusion

Leukemia is a form of blood cancer that is one of the principal causes of cancer-related death. Recent research studies propose deep learning-based strategies for leukemia cancer detection, including transfer learning approaches, and show incredibly precise outcomes. However, improving deep learning algorithms is a continuing research problem for various researchers. Hence, in this research study, an improved deep learning model based on squeeze and excitation learning is proposed to diagnose leukemia cancer from a given microscopic blood sample of patients. In the proposed model, the representation ability is improved at every level of feature representation by permitting it to undertake periodic channel-wise feature recalibration. The squeeze and excitation operations enable the model to extract strong, relevant, and discriminative features from leukemic and normal cells. The proposed model has been validated on publicly available datasets and shows promising results when compared to the traditional deep learning model. In the future, the proposed technique can be validated on the different subtypes of acute lymphocytic leukemia.

Data Availability

The dataset is publicly available and can also be obtained by contacting the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] S. Agaian, M. Madhukar, and A. T. Chronopoulos, "Automated screening system for acute myelogenous leukemia detection in blood microscopic images," *IEEE Systems journal*, vol. 8, no. 3, pp. 995–1004, 2014.
- [2] S. Shafique and S. Tehsin, "Acute lymphoblastic leukemia detection and classification of its subtypes using pretrained deep convolutional neural networks," *Technology in Cancer Research & Treatment*, vol. 17, Article ID 1533033818802789, 2018.
- [3] A. Biondi, G. Cimino, R. Pieters, and C.-H. Pui, "Biological and therapeutic aspects of infant leukemia," *Blood*, vol. 96, no. 1, pp. 24–33, 2000.
- [4] R. D. Labati, V. Piuri, and F. Scotti, "All-IDB: the acute lymphoblastic leukemia image database for image processing," in *Proceedings of the 2011 18th IEEE International Conference on Image Processing*, pp. 2045–2048, IEEE, Brussels, Belgium, Sept. 2011.
- [5] T. Tran, O.-H. Kwon, K.-R. Kwon, S.-H. Lee, and K.-W. Kang, "Blood cell images segmentation using deep learning semantic segmentation," in *Proceedings of the 2018 IEEE International Conference on Electronics and Communication Engineering (ICECE)*, pp. 13–16, IEEE, Xi'an, China, Dec. 2018.
- [6] T. C. Fujita, N. Sousa-Pereira, M. K. Amarante, and M. A. E. Watanabe, "Acute lymphoid leukemia etiopathogenesis," *Molecular Biology Reports*, vol. 48, no. 1, pp. 817–822, 2021.
- [7] "Key statistics for acute lymphocytic leukemia," 2019, <https://www.cancer.org/cancer/acute-lymphocytic-leukemia/about/key-statistics.html>.
- [8] R. Lipton, T. Schwedt, and B. Friedman, "GBD 2015 Disease and Injury Incidence and Prevalence Collaborators. Global, regional, and national incidence, prevalence, and years lived with disability for 310 diseases and injuries, 1990–2015: a systematic analysis for the Global Burden of Disease Study 2015," *Lancet*, vol. 388, no. 10053, pp. 1545–1602, 2017.
- [9] L. Li and Y. Wang, "Recent updates for antibody therapy for acute lymphoblastic leukemia," *Experimental Hematology & Oncology*, vol. 9, pp. 33–11, 2020.
- [10] M. Ghaderzadeh, F. Asadi, A. Hosseini, D. Bashash, H. Abolghasemi, and A. Roshanpour, "Machine learning in detection and classification of leukemia using smear blood images: a systematic review," *Scientific Programming*, vol. 2021, Article ID 9933481, 2021.
- [11] P. Jagadev and H. Virani, "Detection of leukemia and its types using image processing and machine learning," in *Proceedings of the 2017 International Conference on Trends in Electronics and Informatics (ICEI)*, pp. 522–526, IEEE, Tirunelveli, India, May 2017.
- [12] A. Khashman and E. Al-Zgoul, "Image segmentation of blood cells in leukemia patients," *Recent advances in computer engineering and applications*, vol. 2, pp. 104–109, 2010.
- [13] V. Singhal and P. Singh, "Local binary pattern for automatic detection of acute lymphoblastic leukemia," in *Proceedings of the 2014 Twentieth National Conference on Communications (NCC)*, pp. 1–5, IEEE, Kanpur, India, March 2014.
- [14] Y. Li, R. Zhu, L. Mi, Y. Cao, and D. Yao, "Segmentation of white blood cell from acute lymphoblastic leukemia images using dual-threshold method," *Computational and mathematical methods in medicine*, vol. 2016, Article ID 9514707, 2016.
- [15] S. S. Al-jaboriy, N. N. A. Sjarif, S. Chuprat, and W. M. Abdullah, "Acute lymphoblastic leukemia segmentation using local pixel information," *Pattern Recognition Letters*, vol. 125, pp. 85–90, 2019.
- [16] V. Shankar, M. M. Deshpande, N. Chaitra, and S. Aditi, "Automatic detection of acute lymphoblastic leukemia using image processing," in *Proceedings of the 2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, pp. 186–189, IEEE, Coimbatore, India, Oct. 2016.
- [17] G. Singh, G. Bathla, and S. Kaur, "Design of New Architecture to detect leukemia cancer from medical images," *International Journal of Applied Engineering Research*, vol. 11, pp. 7087–7094, 2016.
- [18] M. Alamgir Sarder, M. Maniruzzaman, and B. Ahammed, "Feature selection and classification of leukemia cancer using machine learning techniques," *Machine Learning Research*, vol. 5, no. 2, p. 18, 2020.
- [19] S. Mishra, B. Majhi, P. K. Sa, and L. Sharma, "Gray level co-occurrence matrix and random forest based acute lymphoblastic leukemia detection," *Biomedical Signal Processing and Control*, vol. 33, pp. 272–280, 2017.
- [20] S. Pouyanfar, S. Sadiq, Y. Yan et al., "A survey on deep learning: algorithms, techniques, and applications," *ACM Computing Surveys*, vol. 51, pp. 1–36, 2018.
- [21] H. S. Basavegowda and G. Dagnev, "Deep learning approach for microarray cancer data classification," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 1, pp. 22–33, 2020.
- [22] C. Kaushal and A. Singla, "Automated segmentation technique with self-driven post-processing for histopathological breast cancer images," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 4, pp. 294–300, 2020.
- [23] M. Maqsood, S. Yasmin, I. Mehmood, M. Bukhari, and M. Kim, "An efficient DA-net architecture for lung nodule segmentation," *Mathematics*, vol. 9, no. 13, p. 1457, 2021.
- [24] M. Hammad, M. H. Alkinani, B. Gupta, and A. A. Abd El-Latif, "Myocardial infarction detection based on deep neural network on imbalanced data," *Multimedia Systems*, pp. 1–13, 2021.
- [25] A. Sedik, M. Hammad, F. E. Abd El-Samie, B. B. Gupta, and A. A. Abd El-Latif, "Efficient deep learning approach for augmented detection of Coronavirus disease," *Neural Computing & Applications*, pp. 1–18, 2021.
- [26] M. Hammad, A. M. Iliyasu, A. Subasi, E. S. Ho, and A. A. Abd El-Latif, "A multitier deep learning model for arrhythmia detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2020.
- [27] M. Maqsood, M. Bukhari, Z. Ali et al., "A residual-learning-based multi-scale parallel-convolutions- assisted efficient CAD system for liver tumor detection," *Mathematics*, vol. 9, no. 10, p. 1133, 2021.
- [28] R. Gad, M. Talha, A. A. El-Latif et al., "Iris recognition using multi-algorithmic approaches for cognitive internet of things (CIoT) framework," *Future Generation Computer Systems*, vol. 89, pp. 178–191, 2018.
- [29] A. Shabbir, A. Rasheed, A. Rasheed et al., "Detection of glaucoma using retinal fundus images: a comprehensive review," *Mathematical Biosciences and Engineering*, vol. 18, no. 3, pp. 2033–2076, 2021.
- [30] W. Zhao, F. Chen, H. Huang, D. Li, and W. Cheng, "A new steel defect detection algorithm based on deep learning," *Computational Intelligence and Neuroscience*, vol. 2021, 2021.
- [31] A. Rehman, N. Abbas, T. Saba, S. I. u. Rahman, Z. Mehmood, and H. Kolivand, "Classification of acute lymphoblastic leukemia using deep learning," *Microscopy Research and Technique*, vol. 81, no. 11, pp. 1310–1317, 2018.

- [32] N. Ahmed, A. Yigit, Z. Isik, and A. Alpkocak, "Identification of leukemia subtypes from microscopic images using convolutional neural network," *Diagnostics*, vol. 9, no. 3, p. 104, 2019.
- [33] P. K. Das and S. Meher, "An efficient deep convolutional neural network based detection and classification of acute lymphoblastic leukemia," *Expert Systems with Applications*, vol. 183, p. 115311, 2021.
- [34] M. Shaheen, R. Khan, R. Biswal et al., "Acute myeloid leukemia (AML) detection using AlexNet model," *Complexity*, vol. 2021, 2021.
- [35] K. K. Anilkumar, V. J. Manoj, and T. M. Sagi, "Automated detection of leukemia by pretrained deep neural networks and transfer learning: a comparison," *Medical Engineering & Physics*, vol. 98, pp. 8–19, 2021.
- [36] A. Abhishek, R. K. Jha, R. Sinha, and K. Jha, "Automated classification of acute leukemia on a heterogeneous dataset using machine learning and deep learning techniques," *Biomedical Signal Processing and Control*, vol. 72, p. 103341, 2022.
- [37] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, IEEE, Salt Lake City, UT, USA, June 2018.
- [38] H. Mohamed, R. Omar, N. Saeed et al., "Automated detection of white blood cells cancer diseases," in *Proceedings of the 2018 First International Workshop on Deep and Representation Learning (IWDRL)*, pp. 48–54, IEEE, Cairo, Egypt, March 2018.
- [39] S. Mohapatra, D. Patra, and S. Satpathy, "An ensemble classifier system for early diagnosis of acute lymphoblastic leukemia in blood microscopic images," *Neural Computing & Applications*, vol. 24, no. 7-8, pp. 1887–1904, 2014.
- [40] N. Patel and A. Mishra, "Automated leukaemia detection using microscopic images," *Procedia Computer Science*, vol. 58, pp. 635–642, 2015.
- [41] Y. Wang and J. Orchard, "Fast discrete orthonormal Stockwell transform," *SIAM Journal on Scientific Computing*, vol. 31, no. 5, pp. 4000–4012, 2009.
- [42] S. Mishra, B. Majhi, and P. K. Sa, "Texture feature based classification on microscopic blood smear for acute lymphoblastic leukemia detection," *Biomedical Signal Processing and Control*, vol. 47, pp. 303–311, 2019.
- [43] M. Zakir Ullah, Y. Zheng, J. Song et al., "An attention-based convolutional neural network for acute lymphoblastic leukemia classification," *Applied Sciences*, vol. 11, no. 22, Article ID 10662, 2021.
- [44] T. Pansombut, S. Wikaisuksakul, K. Khongkrapan, and A. Phon-On, "Convolutional neural networks for recognition of lymphoblast cell images," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 7519603, 2019.
- [45] M. Habibzadeh, M. Jannesari, Z. Rezaei, H. Baharvand, and M. Totonchi, "Automatic white blood cell classification using pre-trained deep learning models: ResNet and Inception," *Tenth international conference on machine vision (ICMV 2017)*, vol. 10696, p. 1069612, 2018.
- [46] A. T. Sahlol, P. Kollmannsberger, and A. A. Ewees, "Efficient classification of white blood cell leukemia with improved swarm optimization of deep features," *Scientific Reports*, vol. 10, pp. 1–11, 2020.
- [47] N. Bibi, M. Sikandar, I. Ud Din, A. Almogren, and S. Ali, "IoMT-based automated detection and classification of leukemia using deep learning," *Journal of Healthcare Engineering*, vol. 2020, 2020.
- [48] S. Vishnu, S. J. Ramson, and R. Jegan, "Internet of medical things (IoMT)-An overview," in *Proceedings of the 2020 5th international conference on devices, circuits and systems (ICDCS)*, pp. 101–104, IEEE, Coimbatore, India, March 2020.
- [49] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, IEEE, Honolulu, HI, USA, July 2017.
- [50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, IEEE, Las Vegas, NV, USA, June 2016.
- [51] K. Simonyan and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, Cornell University, Ithaca, New York, 2014.
- [52] W. Yu, J. Chang, C. Yang et al., "Automatic classification of leukocytes using deep neural network," in *Proceedings of the 2017 IEEE 12th international conference on ASIC (ASICON)*, pp. 1041–1044, IEEE, Guiyang, China, Oct. 2017.
- [53] S. Mourya, S. Kant, P. Kumar, A. Gupta, and R. Gupta, *LeukoNet: DCT-Based CNN Architecture for the Classification of normal versus Leukemic Blasts in B-ALL Cancer*, Cornell University, Ithaca, New York, 2018.
- [54] Z. Jiang, Z. Dong, L. Wang, and W. Jiang, "Method for diagnosis of acute lymphoblastic leukemia based on ViT-CNN ensemble model," *Computational Intelligence and Neuroscience*, vol. 2021, 2021.
- [55] S. H. Kassani, P. H. Kassani, M. J. Wesolowski, K. A. Schneider, and R. Detersad, "A hybrid deep learning architecture for leukemic B-lymphoblast classification," in *Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 271–276, IEEE, Jeju, Korea (South), Oct. 2019.
- [56] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710, IEEE, Salt Lake City, UT, USA, June 2018.
- [57] P. H. Kasani, S.-W. Park, and J.-W. Jang, "An aggregated-based deep learning method for leukemic B-lymphoblast classification," *Diagnostics*, vol. 10, no. 12, p. 1064, 2020.
- [58] K. K. Jha and H. S. Dutta, "Mutual information based hybrid model and deep learning for acute lymphocytic leukemia detection in single cell blood smear images," *Computer Methods and Programs in Biomedicine*, vol. 179, Article ID 104987, 2019.
- [59] Y. Tan and B. Shi, "Generative adversarial optimization," *Lecture Notes in Computer Science*, vol. 11655, pp. 3–17, 2019.
- [60] M. Tuba and E. Tuba, "Generative adversarial optimization (Goa) for acute lymphocytic leukemia detection," *Studies in Informatics and Control*, vol. 28, no. 3, pp. 245–254, 2019.
- [61] S. Mirjalili, "Genetic algorithm," in *Studies in Computational Intelligence* vol. 780, , pp. 43–55, Springer, 2019.
- [62] V. Acharya and P. Kumar, "Detection of acute lymphoblastic leukemia using image segmentation and data mining algorithms," *Medical, & Biological Engineering & Computing*, vol. 57, no. 8, pp. 1783–1811, 2019.
- [63] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, pp. 1–48, 2019.
- [64] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: theory and practice," *International Journal of Computer Vision*, vol. 105, pp. 222–245, 2013.

- [65] L. Shen, G. Sun, Q. Huang, S. Wang, Z. Lin, and E. Wu, "Multi-level discriminative dictionary learning with application to large scale image classification," *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3109–3123, 2015.
- [66] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proceedings of the 2009 IEEE Conference on computer vision and pattern recognition*, pp. 1794–1801, IEEE, Miami, FL, USA, June 2009.
- [67] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 807–814, ACM, Washington, DC, USA, June 2010.
- [68] M. Bukhari, K. B. Bajwa, S. Gillani et al., "An efficient gait recognition method for known and unknown covariate conditions," *IEEE Access*, vol. 9, pp. 6465–6477, 2020.
- [69] R. Ashraf, S. Afzal, A. U. Rehman et al., "Region-of-Interest based transfer learning assisted framework for skin cancer detection," *IEEE Access*, vol. 8, pp. 147858–147871, 2020.
- [70] H. Nawaz, M. Maqsood, S. Afzal, F. Aadil, I. Mehmood, and S. Rho, "A deep feature-based real-time system for Alzheimer disease stage detection," *Multimedia Tools and Applications*, vol. 80, no. 28-29, pp. 35789–35807, 2020.
- [71] W. Ilyas, M. Noor, and M. Bukhari, "An efficient emotion recognition frameworks for affective computing," *The Journal of Contents Computing*, vol. 3, no. 1, pp. 251–267, 2021.
- [72] A. Latif, A. Rasheed, U. Sajid, J. Ahmed, N. Ali, N. I. Ratyal et al., "Content-based image retrieval and feature extraction: a comprehensive review," *Mathematical Problems in Engineering*, vol. 2019, Article ID 9658350, 2019.
- [73] N. Wang, Q. Li, A. A. Abd El-Latif, T. Zhang, and X. Niu, "Toward accurate localization and high recognition performance for noisy iris images," *Multimedia Tools and Applications*, vol. 71, no. 3, pp. 1411–1430, 2014.
- [74] G. Jothi, H. H. Inbarani, A. T. Azar, and K. R. Devi, "Rough set theory with Jaya optimization for acute lymphoblastic leukemia classification," *Neural Computing & Applications*, vol. 31, no. 9, pp. 5175–5194, 2019.
- [75] S. Agaian, M. Madhukar, and A. T. Chronopoulos, "A new acute leukaemia-automated classification system," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 6, no. 3, pp. 303–314, 2018.