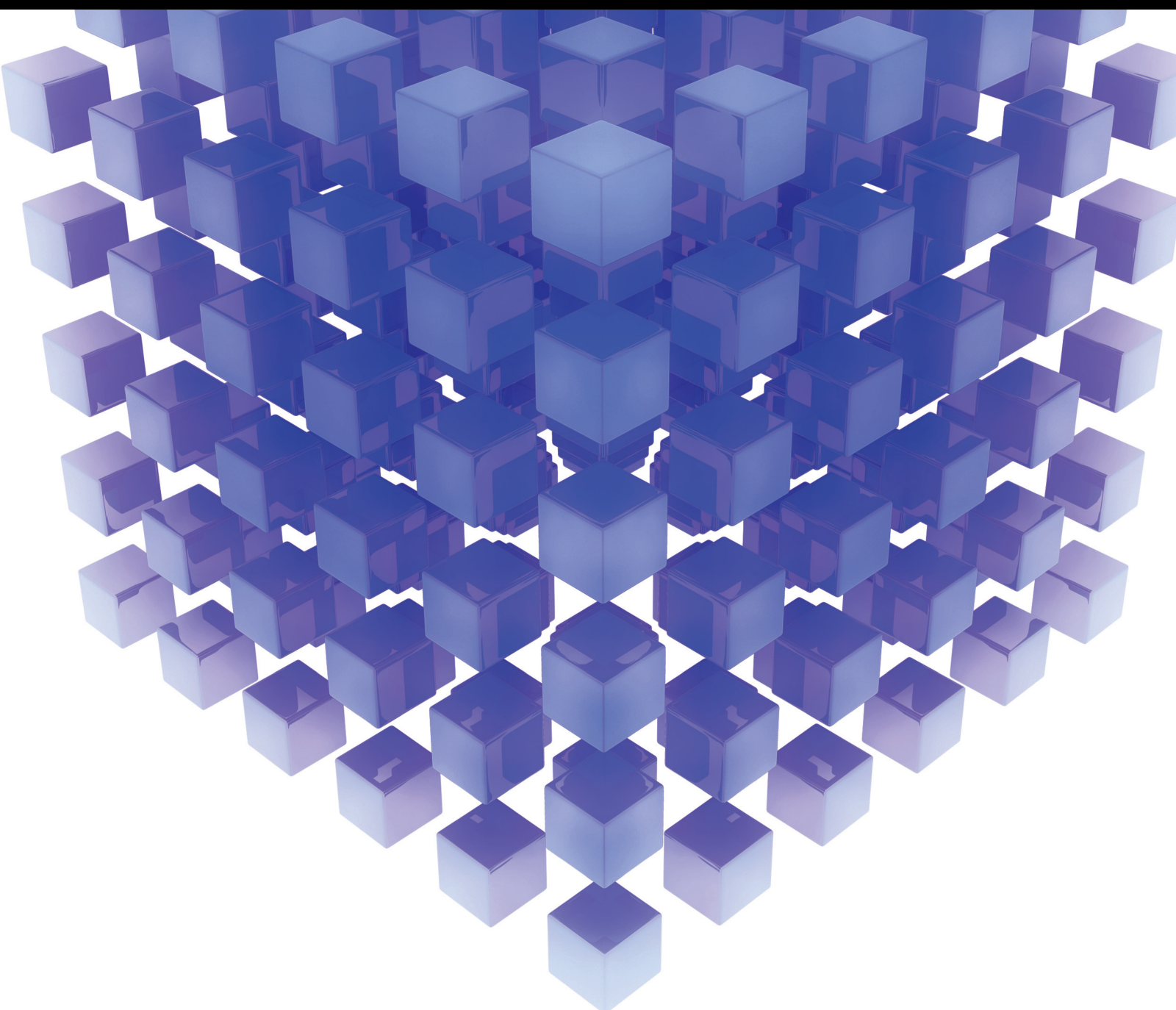


# Deep Learning and Optimization Approaches in Signal Processing

Lead Guest Editor: Saeid Jafarzadeh Ghouschi

Guest Editors: Araz Darba, Dimitar V. Bozalakov, and Ramin Ranjbarzadeh





---

# **Deep Learning and Optimization Approaches in Signal Processing**

Mathematical Problems in Engineering

---

## **Deep Learning and Optimization Approaches in Signal Processing**

Lead Guest Editor: Saeid Jafarzadeh Ghouschi  
Guest Editors: Araz Darba, Dimitar V. Bozalakov,  
and Ramin Ranjbarzadeh




---

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 Addressi , 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 Bacciocchi , 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 Eftekhari Azam, USA  
Said El Kafhali , Morocco  
Antonio Elipse , 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  
Stylianios 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

## **Review of Deep Learning Approaches for Thyroid Cancer Diagnosis**

Shokofeh Anari , Nazanin Tataei Sarshar , Negin Mahjoori, Shadi Dorosti , and Amirali Rezaie   
Review Article (8 pages), Article ID 5052435, Volume 2022 (2022)

## **Detection and Counting Method of Pigs Based on YOLOV5\_Plus: A Combination of YOLOV5 and Attention Mechanism**

Zishun Zhou   
Research Article (16 pages), Article ID 7078670, Volume 2022 (2022)

## **Urban Rail Transit Network Planning Based on Particle Swarm Optimization Algorithm**

Ning Yu   
Research Article (9 pages), Article ID 2401333, Volume 2022 (2022)

## **Optimized Neural Network for Research Evaluation of Mineral Resources Carrying Capacity in Southern Shaanxi**

Li YiCan, Wei JunHao, and Ali Arshaghi   
Research Article (10 pages), Article ID 7311245, Volume 2022 (2022)





## **Recognizing the Damaged Surface Parts of Cars in the Real Scene Using a Deep Learning Framework**

Mahboub Parhizkar  and Majid Amirfakhrian   
Research Article (7 pages), Article ID 5004129, Volume 2022 (2022)



## **Denoising Seismic Data via a Threshold Shrink Method in the Non-Subsampled Contourlet Transform Domain**

Yu Yang, Qi Ran, Kang Chen, Cheng Lei, Yusheng Zhang, Han Liang, Song Han, and Cong Tang   
Research Article (12 pages), Article ID 1013623, Volume 2022 (2022)

## **Partition KMNN-DBSCAN Algorithm and Its Application in Extraction of Rail Damage Data**

Yujun Li , Zhi Yang , Shangbin Jiao , and Yuxing Li   
Research Article (10 pages), Article ID 4699573, Volume 2022 (2022)





## **Recognition of Persian/Arabic Handwritten Words Using a Combination of Convolutional Neural Networks and Autoencoder (AECNN)**

Sara Khosravi  and Abdolah Chalechale   
Research Article (15 pages), Article ID 4241016, Volume 2022 (2022)


## **Skin Lesion Segmentation Based on Edge Attention Vnet with Balanced Focal Tversky Loss**

Majid Nour , Hakan Öcal, Adi Alhudhaif , and Kemal Polat   
Research Article (10 pages), Article ID 4677044, Volume 2022 (2022)

## **A New Generation Communication System Based on Deep Learning Methods for the Process of Modulation and Demodulation from the Modulated Images**



Nihat Daldal , Zeynel Abidin Sezer , Majid Nour, Adi Alhudhaif , and Kemal Polat   
Research Article (13 pages), Article ID 9555598, Volume 2022 (2022)

### **Microalgae Classification Using Improved Metaheuristic Algorithm**

Liu Wei, Su XiaoPan, and Faezeh Heydari 

Research Article (10 pages), Article ID 3783977, Volume 2022 (2022)

### **Novel Based Ensemble Machine Learning Classifiers for Detecting Breast Cancer**


Taarun Srinivas , Aditya Krishna Karigiri Madhusudhan, Joshuva Arockia Dhanraj , Rajasekaran

Chandra Sekaran, Neda Mostafaeipour , Negar Mostafaeipour , and Ali Mostafaeipour 

Research Article (16 pages), Article ID 9619102, Volume 2022 (2022)


### **A Novel Faster RCNN with ODN-Based Rain Removal Technique**

Purnachand Kollapudi, Mydhili K Nair, S. Parthiban, Abbas Mardani, Sachin Upadhye, Vinaykumar Nassa,

and Alhassan Alolo Abdul-Rasheed Akeji 





Research Article (11 pages), Article ID 4546135, Volume 2022 (2022)

### **PSO-Based ANN for Studying on Carrying Capacity of Iron Ore Resources-Economy-Environment in Southern Shaanxi, China**

Li YiCan, Wei JunHao, Pan Na, and Hakimeh Morabbi Heravi 



Research Article (15 pages), Article ID 4685257, Volume 2022 (2022)

### **Image Semantic Segmentation Method Based on Deep Learning in UAV Aerial Remote Sensing Image**

Min Ling , Qun Cheng , Jun Peng , Chenyi Zhao , and Ling Jiang


Research Article (10 pages), Article ID 5983045, Volume 2022 (2022)

### **The Offset Azimuth Prediction of Light Buoy Based on Markov Chain Optimization Multiplicative Seasonal Model**

Zhizheng Wu , Lüzhen Ren , Shibo Zhou, Yuqi Zhang, Wenpeng Xu, and Heyang Zhang



Research Article (12 pages), Article ID 1250206, Volume 2022 (2022)


### **Chimp Optimization Algorithm to Optimize a Convolutional Neural Network for Recognizing Persian/Arabic Handwritten Words**

Sara Khosravi  and Abdolah Chalechale 

Research Article (12 pages), Article ID 4894922, Volume 2022 (2022)





### **Federated Learning Algorithms to Optimize the Client and Cost Selections**

Ali Alferaidi, Kusum Yadav, Yasser Alharbi , Wattana Viriyasitavat, Sandeep Kautish , and Gaurav

Dhiman 

Research Article (9 pages), Article ID 8514562, Volume 2022 (2022)

### **Discrete Generalized Inverted Exponential Distribution: Case Study Color Image Segmentation**

Mohamed Abd Elaziz , Nahla S. Abdelrahman , N. A. Hassan , and M. O. Mohamed 

Research Article (17 pages), Article ID 3029932, Volume 2022 (2022)



## Contents


---

**Distributed Deep CNN-LSTM Model for Intrusion Detection Method in IoT-Based Vehicles**

Ali Alferaidi, Kusum Yadav, Yasser Alharbi , Navid Razmjooy , Wattana Viriyasitavat , Kamal Gulati , Sandeep Kautish , and Gaurav Dhiman 

Research Article (8 pages), Article ID 3424819, Volume 2022 (2022)

**Identification of Chaos in Financial Time Series to Forecast Nonperforming Loan**

Farid Ahmadi , Mohammad Pourmahmood Aghababa, and Hashem Kalbkhani

Research Article (11 pages), Article ID 2055655, Volume 2022 (2022)

## Review Article

# Review of Deep Learning Approaches for Thyroid Cancer Diagnosis

Shokofeh Anari <sup>1</sup>, Nazanin Tataei Sarshar <sup>2</sup>, Negin Mahjoori,<sup>3</sup> Shadi Dorosti <sup>4</sup>,  
and Amirali Rezaie <sup>3</sup>

<sup>1</sup>Department of Accounting, Economic and Financial Sciences, Islamic Azad University, South Tehran Branch, Tehran, Iran

<sup>2</sup>Department of Engineering, Islamic Azad University, Tehran North Branch, Tehran, Iran

<sup>3</sup>Independent Researcher, Tehran, Iran

<sup>4</sup>Department of Mechanical Engineering, Urmia University of Technology (UUT), P.O. Box: 57166-419, Urmia, Iran

Correspondence should be addressed to Shadi Dorosti; [shadi.dorosti@gmail.com](mailto:shadi.dorosti@gmail.com)

Received 23 June 2022; Revised 14 July 2022; Accepted 22 July 2022; Published 25 August 2022

Academic Editor: Araz Darba

Copyright © 2022 Shokofeh Anari 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.

Thyroid nodule is one of the common life-threatening diseases, and it had an increasing trend over the last years. Ultrasound imaging is a commonly used diagnostic method for detecting and characterizing thyroid nodules. However, assessing the entire slide images is time-consuming and challenging for the experts. For assessing ultrasound images in a meaningful manner, there is a need for automated, trustworthy, and objective approaches. The recent advancements in deep learning have revolutionized many aspects of computer-aided diagnosis (CAD) and image analysis tools that address the problem of diagnosing thyroid nodules. In this study, we explained the objectives of deep learning in thyroid cancer imaging and conducted a literature review on its potential, limits, and current application in this area. We gave an overview of recent progress in thyroid cancer diagnosis using deep learning methods and discussed various challenges and practical problems that might limit the growth of deep learning and its integration into clinical workflow.

## 1. Introduction

Thyroid cancer has been more common during the last three decades [1]. The most recent estimation for thyroid cancer reported by the American Cancer Society for 2022 is approximately 43,800 new cases and about 2,230 deaths [2]. Thyroid cancer is a solid tumor that usually shows up as a nodule or mass at the front base of the throat in the thyroid gland [3]. Thyroid cancer happens when rogue cells reproduce too rapidly for the immune system to control [4]. Generally, cancer results from gene mutation or changes to genes are responsible for controlling the cell function. Therefore, cells reproduce uncontrollably and spread into surrounding tissues [5]. Several types of thyroid cancer exist, but two types are by far the most common types that are responsible for 95% of thyroid cancers. These types include follicular and papillary thyroid cancer [6].

Treating early detected malignant thyroid nodules before the thyroid gland's cancerous cells spreading can result in effective treatment and less harm [7]. Thyroid cancer screening is a procedure for the early detection of malignant thyroid nodules [8]. Thyroid cancer is detected using two major methods: (1) palpation of the neck during a physical examination and (2) ultrasonography, which can detect palpable and nonpalpable nodules, especially those less than 1 cm in diameter [9]. Ultrasonography is used to identify the characteristics of thyroid nodules as the primary diagnostic tool. These identified characteristics help to classify nodules into benign or malignant type [10–12].

Over the last decades, computer-aided diagnosis (CAD) is employed as a new technique for automatic thyroid nodules diagnosis. Implementing artificial intelligence in CAD tools makes them smarter and increases the accuracy and consistency of the ultrasonography features interpretation,

ultimately decreasing the unnecessary biopsy. Machine learning and deep learning are the underlying techniques of AI-based CAD systems that greatly impact the medical field [13]. These methods rely on experts' knowledge to choose the essential features from a set of predefined specified characteristics collected from the region of interest [14]. In thyroid ultrasound images, features such as margin, shape, echogenicity, calcifications, and composition and have been used in many studies to develop CAD systems. The efficiency of these systems has been indicated previously [15–17]. Previous research has indicated how the traditional machine learning and deep learning algorithms, such as the support vector machines [18], GoogLeNet [19], and convolutional neural network (CNN) [20, 21], have changed the thyroid nodule diagnosis. The development of machine learning and artificial intelligence removed the constraints of employing CAD tools in the everyday routine of physicians and experts have been overcome significantly [22, 23].

This paper presents a comprehensive review of the deep learning approaches used for diagnosis of thyroid cancer. Most of the papers were published after 2018, indicating that the deep learning algorithm had a good performance for thyroid nodules classification; therefore, it gained much attention over the last years. In the following part of this paper, a review of the deep learning methods that previously have been applied for thyroid nodule classification is presented in Section 2. A comprehensive explanation of deep learning methods such as CNNs, generative adversarial networks (GANs), autoencoders, long short-term memory (LSTM), deep belief network (DBN), and recurrent neural networks (RNNs) was provided, and the investigations that applied these approaches for thyroid cancer classification were introduced. The rest of the paper provides discussion and conclusion.

## 2. Deep Learning Methods Reviews

Deep learning is a part of artificial intelligence that uses artificial neural networks. It is a machine learning technique for extracting patterns and making predictions from large datasets. The growing deep learning model application in health care, combined with the availability of well-characterized cancer datasets, has pushed research into deep learning's utility in analyzing cancer cells. In the following, a comprehensive review of deep learning models that are used for thyroid cancer classification is provided.

**2.1. Convolutional Neural Networks.** Convolutional neural network (CNN) is a type of neural network that has one or more convolutional layers. The use case of these layers is to process images, classify data, segment data, etc [24]. CNN is similar to traditional artificial neural networks and is made up of neurons that learn to optimize themselves. Each neuron receives data and performs an operation, forming the foundation for many artificial neural networks. The complete network expresses a single perceptual scoring function from the input raw image vectors through the final

output class score [25, 26]. CNN as a feed-forward neural network uses a grid-like layout to evaluate visual images, process data, and detect and categorize items in an image. CNN does three steps to make image classification practical, including the following:

- (1) Reduce the number of input nodes
- (2) Tolerate small shifts in where the pixels are in the image
- (3) Take advantage of the correlations observed in complex images

Figure 1 indicates the simple architecture of CNN. The CNN architecture is made up of several layers (known as multibuilding blocks). In the following, the description of each layer and its functions are provided in detail:

- (1) Convolutional layer: Convolutional layers are the fundamental component of a CNN structure. It is made up of several convolutional filters. The output feature map is generated by convolving the input image (represented as N-dimensional matrices) with these filters.
- (2) Pooling layer: Pooling is a technique used in convolutional neural networks to enable the network to recognize features regardless of their location in the image by generalizing characteristics retrieved by convolutional filters
- (3) Fully connected layer: In a neural network, fully connected layers are layers in which all of the inputs from one layer are connected to each activation unit of the next layer

CNN has been widely applied to ultrasound images to classify thyroid cancer. CNN is proved to be efficient in thyroid disease diagnosis based on medical imaging [19, 27, 28]. CNN had the greatest accuracy rates among other models for thyroid cancer diagnosis according to the latest researchers. Table 1 lists the most recent 20 articles that used CNN to diagnose thyroid cancer. The specificity, sensitivity, and accuracy rates are used to measure the method's functionality.

In this paper, we chose 20 articles out of 170 initial selected publications. These publications proved remarkable growth in implementing CNNs in the thyroid nodules' assessment in the past years (Table 1). CNNs are mainly focused on identifying suspicious nodules and diagnosing diseases like cancerous cells by the classification of nodules into malignant and benign types. This characteristic led to the growth of using CNNs over the last years. Lee et al. introduced a CAD system that works based on a deep learning approach for patients with thyroid cancer. Eight different CNN models were used to compare the accuracy of methods in classifying thyroid cancer tumors. As shown in Table 1, the ResNet50 had a better performance with higher accuracy, sensitivity, and specificity rate [29]. The other method that was proved to have a good performance is VGG16. Lin et al. proposed a deep learning approach based on VGG16 and used the whole slide images (WSIs) database

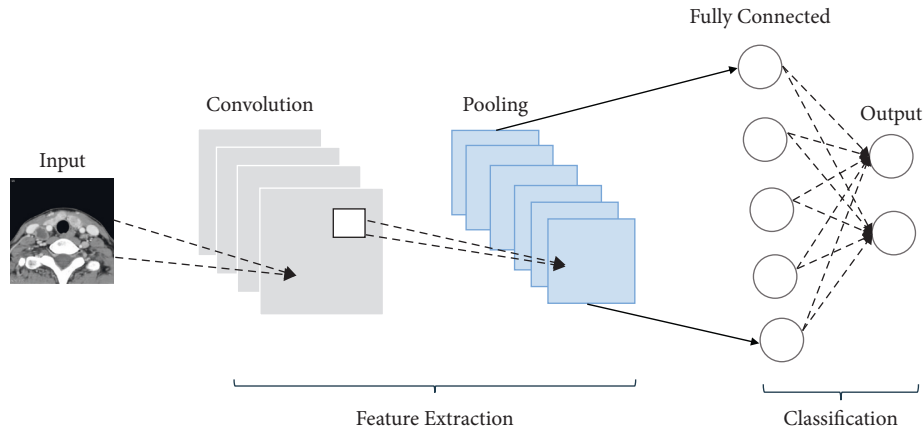


FIGURE 1: Simple structure of CNN.

for this purpose. The result of the proposed approach indicated 99% accuracy and 94% sensitivity [31]. Xception neural network also demonstrated high accuracy in diagnosing brain tumors. Zhang et al. indicated the high accuracy of this approach by applying Xception neural network to CT images [38]. CascadeMaskR-CNN utilized ultrasound images of thyroid cancer to diagnose benign from malignant tumors [46]. The experimental results indicated 94% of accuracy.

**2.2. Generative Adversarial Networks.** Generative adversarial networks (GANs) were introduced as a type of generative model and have gained much attention among artificial intelligence researchers since their introduction. GANs are inspired by the idea of two-player zero-sum games. These models estimate the potential distribution of the given dataset and then generate new samples from the estimated distribution [48]. GANs techniques have been widely applied in various fields due to their exceptional capability for dealing with a variety of types of problems, including image processing, computer vision, speech processing, and language processing [49]. Typically, GANs are made up of a generator and a discriminator learning simultaneously. The generator has the role of recording the probability distribution of the given datasets and then generating new data samples based on that distribution [50]. The discriminator is responsible for distinguishing real data and fake data and usually is a binary classifier. The generator and the discriminator can use a deep neural network structure. GAN utilizes minimax game optimization with the goal of reaching Nash equilibrium, where the generator is to capture the distribution of given datasets [51]. Figure 2 indicates the GANs model and its simple structure.

GANs generate fair data that are very close to the real data [48]. This method is the second most used approach that has been used for thyroid nodules classification. For example, Zhang et al. proposed an adversarial learning-based approach for tissue recognition from medical images by synthesizing medical images. The synthetic model is based on Wasserstein, deep convolutional GANs, and boundary equilibrium GANs approaches. The researchers

reported a 98.83% accuracy for tissue recognition synthetic images [52]. In another paper written by Yang and Qianqian, a semisupervised learning model proposed integrated domain knowledge in training dual path conditional GANs. Also, a semisupervised support vector machine is suggested for classifying thyroid nodules. After putting the model to the test, they found that it successfully avoids the mixed outcomes that might arise when using a limited dataset [53]. Zhao et al. introduced a novel thyroid cancer classification approach based on multimodal domain adaption. To deal with visual discrepancies between modal data, the researchers created semantic consistency GANs and used adversarial learning between dual domains, which is based on the self-attention mechanism. The rate of accuracy of this research for classifying benign and malignant nodules was 94.30 percent [54]. Shi et al. presented an adversarial augmentation technique that is knowledge-guided to synthesize medical images. They designed term and image encoders for extracting domain knowledge based on radiologists' ideas. Then, for high-quality thyroid nodule images and to constrain the auxiliary classifier GANs, domain knowledge is used as a condition. The researchers tested the proposed model on the classification of the ultrasonography thyroid nodule. The accuracy of the model is reported to be 91.46% [55]. The effectiveness of GANs for creating high-resolution pathology images was investigated by Levine et al. The researchers looked at ten different forms of cancer histologically, including five cancer types from the five primary histological subtypes of ovarian carcinoma and the Cancer Genome Atlas. They showed that histotype-classified actual and synthetic images had similar accuracies [56].

**2.3. Other Deep Learning Approaches.** There are other deep learning approaches that have been applied to ultrasound images for thyroid cancer diagnosis. In the following, a review of deep learning applications is presented.

**2.3.1. Autoencoders.** Autoencoders (AE) are a subtype of neural networks. They are primarily meant to encode the input, i.e., represent the input in a compressed and meaningful manner, then decode it, i.e., reconstruct the encoded

TABLE 1: Top 20 papers used convolutional neural networks for thyroid cancer.

No	Reference	Method	Data	Sensitivity	Specificity	Accuracy
1	[29]	ResNet50	CT images	90%	90%	90%
		Inception v3		88%	90%	89%
		Xception		86%	92%	89%
		VGG19		89%	94%	92%
		InceptionResNetV2		69%	94%	81%
		DenseNet121		69%	98%	84%
		DenseNet169		81%	98%	89%
VGG16	86%	83%	85%			
2	[30]	Xception	CT images	80.2%	83.0%	82.8%
		Inception v3		87%	73%	74%
		ResNet50		86%	69%	70%
		VGG19		85%	73%	74%
		InceptionResNetV2		76%	78%	78%
		DenseNet121		84%	81%	81%
		DenseNet169		79%	77%	77%
VGG16	84%	67%	68%			
3	[31]	VGG16	Whole slide imaging	94%	—	99%
4	[32]	Deep convolutional neural network (DCCN)	Sonographic images	93%	86%	89%
5	[33]	MFDN	Post-ablation whole-body planar scans (RxWBSs)	—	85%	93%
6	[34]	Multiprong CNN (MPCNN)	Sonographic images	88%	73%	—
7	[35]	Multi-input CNN	MRI	69%	97%	87%
8	[36]	Multi-input CNN	MRI	82%	—	88%
9	[37]	Inception v3	Ultrasound images	93.3%	87.4%	~95%
10	[38]	Xception neural network	Ultrasound imaging and computed tomography (CT)	94%	—	98%
11	[39]	ThyNet	Ultrasound images	94%	81%	—
12	[40]	R-CNN	Ultrasound images	81%	—	—
13	[41]	ThyNet	Ultrasound images	94%	81%	89%
14	[20]	SVM + CNN	Ultrasound images	96.4%	83.1%	92.5%
15	[42]	VGG16	Ultrasound images	63%	80%	74%
16	[43]	Inception v3	Ultrasound images	83.7%	83.7%	76.5%
		ResNet101		72.5%	81.4%	77.6%
		VGG19		66.2%	76.9%	76.1%
17	[44]	VGG16	Ultrasound images	70%	92%	—
18	[45]	Mask R-CNN	Ultrasound images	79%	—	—
19	[46]	CascadeMaskR-CNN	Ultrasound images	93%	95%	94%
20	[47]	Google inception v3	Histopathology images	—	—	95%

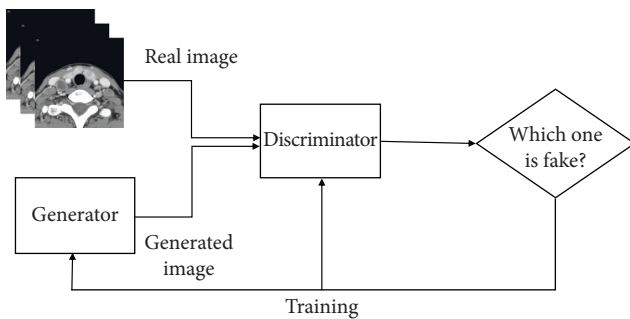


FIGURE 2: Simple structure of the GANS model.

input with the maximum possible similarity to the original input. Unsupervised learning and deep architectures rely heavily on autoencoders for transfer learning and other tasks. AE has been widely used in the medical field for tumor classifications. However, this method has not been applied widely for the classification of thyroid cancer. There are several studies that applied this method for this purpose. For example, Ferreira et al. applied six distinct AE types for thyroid nodules classification, as well as two different techniques to train the classification model. With an F1 score of  $99.61\% \pm 0.54$ , they conclude that combining a deeper classification network with the reconstruction of the input

space outperformed previous studies [57]. In another study, Ferreira et al. contributed to the literature by automatically classifying tumor samples by analyzing their gene expressions. The researchers tried to develop a methodology for distinguishing five different cancer types from RNA-Seq datasets, including thyroid, skin, stomach, breast, and lung cancers. In this research, they adopted autoencoders for initializing weights on deep neural networks and compared the performance of three different autoencoders. The results indicated an average F1 score of 99.03 for the RNA-Seq data [58]. Also, to categorize thyroid nodules, Li et al. employed the stacked denoising sparse autoencoder. This study used immune-related genes to build a classifier with a stacked denoising sparse autoencoder using data of gene expression from thyroid nodule tissues. The experimental results on distinguishing benign and malignant thyroid nodules demonstrated an accuracy of 92.9% [59].

**2.3.2. Long Short-Term Memory.** Long short-term memory (LSTM) is another deep learning approach that is able to learn order dependency in sequence prediction issues and is a kind of recurrent neural network (RNN). By default, these algorithms are designed to avoid the problem of long-term dependency and remember information for long periods of time. In order to consider the application of LSTM, Chen et al. proposed a new approach that divided the report into two layers: a word vector layer and a sentence presentation layer, with each layer employing the bidirectional long short-term memory and attention mechanism. Finally, they provided a model with good performance [60]. Wu et al. applied ML algorithms such as Gradient Boosting trees, k-nearest neighbor, decision trees, Naïve Bayes, logistic regression, random forest, and long short-term memory model using time-series tumor marker data on two large asymptomatic cohorts, including 163,174 records. Compared to the other ML models, the LSTM model proved the best at handling erratic data [61].

**2.3.3. Deep Belief Network.** The deep belief network or DBN is a type of deep neural network but is not the same and it is made up of multiple layers of restricted Boltzmann machines. These algorithms provide solutions for the limitations of training conventional neural networks in deep layered networks, including getting stuck in local minima due to poor parameters, slow learning, and requiring big training datasets. The only paper that applied the DBN method for thyroid nodule diagnosis is the research done by Pavithra and Parthiban. They presented a new pigeon inspired optimization (PIO) problem with the DBN model, named PIO-DBN, for the classification and diagnosis of thyroid disease. The PIO-DBN model reached the maximum accuracy of 98.91% and 96.28% on the two thyroid datasets used to evaluate the model [62].

**2.3.4. Recurrent Neural Networks.** Recurrent neural networks (RNNs) are a kind of artificial neural network used for dealing with sequential or time-series data. The distinguishing feature of these algorithms is their “memory.” In

RNNs, the information from prior inputs can influence the current input and output. These deep learning algorithms are commonly applied to ordinal or temporal problems. For thyroid cancer nodule diagnosis, Begum et al. utilized bidirectional RNN to evaluate the risk of getting thyroid illness in patients. The result of applying the proposed approach was a 98.72% rate of accuracy [63]. Also, Santillan et al. studied distinguishing malignant from benign thyroid lesions by applying five neural network approaches, and the results indicated that the RNN model performed better than the rest, having an accuracy of 98% [64].

### 3. Discussion

Ultrasound imaging has become one of the primary technologies for analyzing thyroid nodules due to its safety, cost-effectiveness, being noninvasive, and easily accessible. However, it is a challenging task to interpret ultrasound images, and the interpretation can be altered based on radiologists’ prior medical knowledge and observational skills. Therefore, the need for automated, reliable, and objective technologies for the interpretation of ultrasound images is significant. Progress in deep learning in recent years has revolutionized various areas of machine learning, such as computer vision and image processing. Although CAD systems that are based on artificial intelligence are evolving rapidly, there is no widespread adoption of any of these systems, and there are still conflicting issues. There is a significant need for AI-based CAD systems with better designs and practicality that provide consistent nodule management solutions in practice [65, 66]. In this paper, we reviewed the recent studies that deployed deep learning-based algorithms for analyzing medical images of thyroid nodules. The literature demonstrated that although CAD systems provide similar sensitivity to experienced radiologists, they still cannot reach the level of specificity and accuracy of experts [67]. Therefore, a probable option to consider is to combine the specificity and accuracy of radiologists with the sensitivity of CAD systems and use these systems as assistants for operators with less experience at primary care centers [7, 10–12]. Accordingly, it is necessary to apply deep learning approaches and develop models with high accuracy, specificity, and sensitivity [68, 69]. Future research should scrutinize the effectiveness of these methods and techniques. Moreover, developing more effective techniques for preprocessing images is necessary as they can alter the performance of deep learning models significantly. Other challenges that need to be addressed in future research include coping with data limitations, creating valid and public datasets, and developing standard evaluation measures. Furthermore, all deep learning approaches, including B-mode, Doppler, contrast-enhanced ultrasound, and SWE, should be used on multimodal images to get a complete picture of the lesions. Thyroid nodule diagnosis accuracy can be improved by registering, training, and evaluating thyroid nodules’ multimodal images. Besides, the lack of standard metrics for evaluating the suggested methods’ performance makes it difficult to compare their outcomes. Based on the recent publication, it can be concluded that among all deep

learning techniques, CNNs have been widely applied in order to diagnose thyroid cancer. The results yielded high sensitivity, specificity, and accuracy. However, other deep learning methods have not been applied widely, and there are not enough papers to make the comparison between methods reasonable. The second most used deep learning method to diagnose thyroid nodules is GANs. The high rate of sensitivity, specificity, and accuracy indicates that the application of this method on multimodal images can result in finding models with a better performance. The other popular deep learning approaches like RNN, DBN, and LSTM have not been used widely, and more research should be done to find out the rate of accuracy.

#### 4. Conclusion

As mentioned before, thyroid cancer begins when the cells divide rapidly and spread uncontrollably into surrounding tissues. Therefore, the early detection of cancerous nodules is essential for effective management of the disease besides reducing the number of deaths. The AI-based CAD systems development for processing thyroid images was very fast over the last decades. Thyroid nodule treatment will be improved if these technologies are thoroughly verified. This paper gives a comprehensive review of deep learning applications in assessing thyroid nodules. The overall conclusion of this study demonstrated that thyroid tumor classification and analysis would considerably benefit from the latest enhancements of deep learning approaches and new systematic deep learning techniques with high specificity, sensitivity, and accuracy. Currently, in comparison with the investigations that applied deep learning approaches for other cancer detection like breast cancer and brain cancer, it can be concluded that it is essential to conduct more investigations for developing systems with high accuracy. Despite the empirical strengths and successes of previous deep learning algorithms and methods in the assessment of ultrasound thyroid images, there still exist many deep learning methods that need to be applied to ultrasound images to investigate their performance. Currently, the number of the public dataset for thyroid cancer imaging is not enough. Therefore, the development of reliable and accessible datasets and the creation of uniform assessment metrics are issues that need to be covered in next researches. According to the results of specificity, sensitivity, accuracy, and rate of previously proposed approaches, it can be concluded that CNN is by far the most popular deep learning method for thyroid cancer diagnosis. According to Table 1, the VGG16 method is the technique that has been widely used for thyroid nodule classification. Moreover, GANs, RNNs, and LSTM methods have been utilized in some research. However, the number of published papers is not enough, and more investigations are required. Also, developing better preprocessing approaches for improving deep learning models' performance is mandatory.

#### Conflicts of Interest

All authors declare that they have no conflicts of interest.

#### References

- [1] E. D. Rossi, L. Pantanowitz, and J. L. Hornick, "A worldwide journey of thyroid cancer incidence centred on tumour histology," *Lancet Diabetes & Endocrinology*, vol. 9, no. 4, pp. 193–194, 2021.
- [2] R. L. Siegel, K. D. Miller, H. E. Fuchs, and A. Jemal, "Cancer statistics," *CA: A Cancer Journal for Clinicians*, vol. 72, no. 1, pp. 7–33, 2022.
- [3] M. E. Cabanillas, D. G. McFadden, and C. Durante, "Thyroid cancer," *The Lancet*, vol. 388, no. 10061, pp. 2783–2795, 2016.
- [4] B. Haugen, M. Tuttle, and L. Wartofsky, "Thyroid cancer," *Journal of Clinical Endocrinology & Metabolism*, vol. 91, no. 12, p. E1, Dec. 2006 E1.
- [5] A. Lujambio and S. W. Lowe, "The microcosmos of cancer," *Nature*, vol. 482, no. 7385, pp. 347–355, 2012.
- [6] M. L. Carcangiu, T. Steeper, G. Zampi, and J. Rosai, "Anaplastic thyroid carcinoma: a study of 70 cases," *American Journal of Clinical Pathology*, vol. 83, no. 2, pp. 135–158, 1985.
- [7] E. Olson, G. Wintheiser, K. M. Wolfe, J. Droessler, and P. T. Silberstein, "Epidemiology of thyroid cancer: a review of the national cancer database, 2000–2013," *Cureus*, vol. 11, no. 2, p. e4127, 2019.
- [8] L. Lamartina, G. Grani, C. Durante, S. Filetti, and D. S. Cooper, "Screening for differentiated thyroid cancer in selected populations," *Lancet Diabetes & Endocrinology*, vol. 8, no. 1, pp. 81–88, 2020.
- [9] J. S. Lin, E. J. A. Bowles, S. B. Williams, and C. C. Morrison, "Screening for thyroid cancer: updated evidence report and systematic review for the US preventive services task force," *JAMA*, vol. 317, no. 18, pp. 1888–1903, 2017.
- [10] E. G. Keramidas, D. K. Iakovidis, D. Maroulis, and S. Karkanis, "Efficient and effective ultrasound image analysis scheme for thyroid nodule detection," *Lecture Notes in Computer Science*, vol. 4633, pp. 1052–1060, 2007.
- [11] X. Yu, H. Wang, and L. Ma, "Detection of thyroid nodules with ultrasound images based on deep learning," *Current Medical Imaging Formerly Current Medical Imaging Reviews*, vol. 16, no. 2, pp. 174–180, 2020.
- [12] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.
- [13] Z. Jin, Y. Zhu, S. Zhang et al., "Ultrasound computer-aided diagnosis (CAD) based on the thyroid imaging reporting and data system (TI-RADS) to distinguish benign from malignant thyroid nodules and the diagnostic performance of radiologists with different diagnostic experience," *Medical Science Monitor*, vol. 26, Article ID e918452, 2020.
- [14] A. Hosny, C. Parmar, J. Quackenbush, L. H. Schwartz, and H. J. W. L. Aerts, "Artificial intelligence in radiology," *Nature Reviews Cancer*, vol. 18, no. 8, pp. 500–510, 2018.
- [15] Y. Chang, A. K. Paul, N. Kim et al., "Computer-aided diagnosis for classifying benign versus malignant thyroid nodules based on ultrasound images: a comparison with radiologist-based assessments," *Medical Physics*, vol. 43, no. 1, pp. 554–567, 2016.
- [16] B. Zhang, J. Tian, S. Pei et al., "Machine learning-assisted system for thyroid nodule diagnosis," *Thyroid Radiology and Nuclear Medicine*, vol. 29, no. 6, pp. 858–867, 2019, <https://home.liebertpub.com/thy>.
- [17] M. Sollini, L. Cozzi, A. Chiti, and M. Kirienko, "Texture analysis and machine learning to characterize suspected thyroid nodules and differentiated thyroid cancer: where do

- we stand?" *European Journal of Radiology*, vol. 99, pp. 1–8, 2018.
- [18] S. J. Chen, C. Y. Chang, K. Y. Chang et al., "Classification of the thyroid nodules based on characteristic sonographic textural feature and correlated histopathology using hierarchical support vector machines," *Ultrasound in Medicine and Biology*, vol. 36, no. 12, pp. 2018–2026, 2010.
- [19] J. Chi, E. Walia, P. Babyn, J. Wang, G. Groot, and M. Eramian, "Thyroid nodule classification in ultrasound images by fine-tuning deep convolutional neural network," *Journal of Digital Imaging*, vol. 30, no. 4, pp. 477–486, Aug. 2017.
- [20] C. Sun, Y. Zhang, Q. Chang et al., "Evaluation of a deep learning-based computer-aided diagnosis system for distinguishing benign from malignant thyroid nodules in ultrasound images," *Medical Physics*, vol. 47, no. 9, pp. 3952–3960, 2020.
- [21] H. P. Chan, L. M. Hadjiiski, and R. K. Samala, "Computer-aided diagnosis in the era of deep learning," *Medical Physics*, vol. 47, no. 5, pp. e218–e227, 2020.
- [22] S. Yoo, S. Lee, S. Kim, K. H. Hwang, J. H. Park, and N. Kang, "Integrating deep learning into CAD/CAE system: generative design and evaluation of 3D conceptual wheel," *Structural and Multidisciplinary Optimization*, vol. 64, no. 4, pp. 2725–2747, 2021.
- [23] M. A. Al-antari, M. A. Al-masni, M. T. Choi, S. M. Han, and T. S. Kim, "A fully integrated computer-aided diagnosis system for digital X-ray mammograms via deep learning detection, segmentation, and classification," *International Journal of Medical Informatics*, vol. 117, pp. 44–54, 2018.
- [24] J. Gu, Z. Wang, J. Kuen et al., "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, May 2018.
- [25] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, 2018.
- [26] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," 2015, <https://arxiv.org/abs/1511.08458v2>.
- [27] Z. Akkus and B. Arunitt, "Reduction of Unnecessary Thyroid Biopsies Using Deep Learning," in *Proceedings of the SPIE Medical Imaging*, vol. 10949, pp. 234–239, CA, USA, March 2019.
- [28] J. Yoon, E. Lee, J. S. Koo et al., "Artificial intelligence to predict the BRAFV600E mutation in patients with thyroid cancer," *PLoS One*, vol. 15, no. 11, Article ID e0242806, 2020.
- [29] J. H. Lee, E. J. Ha, and J. H. Kim, "Application of deep learning to the diagnosis of cervical lymph node metastasis from thyroid cancer with CT," *European Radiology*, vol. 29, no. 10, pp. 5452–5457, 2019.
- [30] J. H. Lee, E. J. Ha, D. Kim et al., "Application of deep learning to the diagnosis of cervical lymph node metastasis from thyroid cancer with CT: external validation and clinical utility for resident training," *European Radiology*, vol. 30, no. 6, pp. 3066–3072, 2020.
- [31] Y. J. Lin, T. K. Chao, M. A. Khalil et al., "Deep learning fast screening approach on cytological whole slides for thyroid cancer diagnosis," *Cancers*, vol. 13, no. 15, p. 3891, 2021.
- [32] X. Li, S. Zhang, Q. Zhang et al., "Diagnosis of thyroid cancer using deep convolutional neural network models applied to sonographic images: a retrospective, multicohort, diagnostic study," *The Lancet Oncology*, vol. 20, no. 2, pp. 193–201, 2019.
- [33] M. S. Kavitha, C. H. Lee, K. S. Shibudas, T. Kurita, and B. C. Ahn, "Deep learning enables automated localization of the metastatic lymph node for thyroid cancer on 131I post-ablation whole-body planar scans," *Scientific Reports*, vol. 10, no. 1, p. 7738, 2020.
- [34] V. Kumar, J. Webb, A. Gregory et al., "Automated segmentation of thyroid nodule, gland, and cystic components from ultrasound images using deep learning," *IEEE Access*, vol. 8, Article ID 63496, 2020.
- [35] A. Naglah, F. Khalifa, R. Khaled et al., "Novel MRI-based CAD system for early detection of thyroid cancer using multi-input CNN," *Sensors*, vol. 21, no. 11, p. 3878, 2021.
- [36] A. Naglah, F. Khalifa, R. Khaled, A. A. K. A. Razek, and A. El-Baz, "Thyroid cancer computer-aided diagnosis system using mri-based multi-input CNN model," in *Proceedings of the 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 1691–1694, Nice, France, April 2021.
- [37] Q. Guan, Y. Wang, J. Du et al., "Deep learning based classification of ultrasound images for thyroid nodules: a large scale of pilot study," *Annals of Translational Medicine*, vol. 7, no. 7, p. 137, 2019.
- [38] X. Zhang, V. C. S. Lee, J. Rong, F. Liu, and H. Kong, "Multi-channel convolutional neural network architectures for thyroid cancer detection," *PLoS One*, vol. 17, no. 1, Article ID e0262128, 2022.
- [39] Y. Liu, J. Liang, S. Peng, W. Wang, and H. Xiao, "A deep-learning model to assist thyroid nodule diagnosis and management – authors' reply," *Lancet Digit. Heal*, PDF, vol. 3, no. 7, pp. e411–e412, 2021.
- [40] H. Li, J. Weng, Y. Shi et al., "An improved deep learning approach for detection of thyroid papillary cancer in ultrasound images," *Scientific Reports*, vol. 8, no. 1, p. 6600, 2018.
- [41] S. Peng, Y. Liu, W. Lv et al., "Deep learning-based artificial intelligence model to assist thyroid nodule diagnosis and management: a multicentre diagnostic study," *The Lancet Digital Health*, vol. 3, no. 4, pp. e250–e259, 2021.
- [42] Y. Wang, W. Yue, X. Li et al., "Comparison study of radiomics and deep learning-based methods for thyroid nodules classification using ultrasound images," *IEEE Access*, vol. 8, Article ID 52017, 2020.
- [43] W. K. Chan, J. H. Sun, M. J. Liou et al., "Using deep convolutional neural networks for enhanced ultrasonographic image diagnosis of differentiated thyroid cancer," *Biomedicine*, vol. 9, no. 12, p. 1771, 2021.
- [44] S. W. Kwon, I. J. Choi, J. Y. Kang, W. I. Jang, G. H. Lee, and M. C. Lee, "Ultrasonographic thyroid nodule classification using a deep convolutional neural network with surgical pathology," *Journal of Digital Imaging*, vol. 33, no. 5, pp. 1202–1208, 2020.
- [45] F. Abdolali, J. Kapur, J. L. Jaremko, M. Noga, A. R. Hareendranathan, and K. Punithakumar, "Automated thyroid nodule detection from ultrasound imaging using deep convolutional neural networks," *Computers in Biology and Medicine*, vol. 122, Article ID 103871, 2020.
- [46] Y. Lu, Y. Yang, and W. Chen, "Application of deep learning in the prediction of benign and malignant thyroid nodules on ultrasound images," *IEEE Access*, vol. 8, Article ID 221480, 2020.
- [47] P. Tsou and C. J. Wu, "Mapping driver mutations to histopathological subtypes in papillary thyroid carcinoma: applying a deep convolutional neural network," *Journal of Clinical Medicine*, vol. 8, no. 10, p. 1675, 2019.
- [48] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: an overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.



- [49] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F. Y. Wang, "Generative adversarial networks: introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, Oct. 2017.
- [50] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: a review," *Medical Image Analysis*, vol. 58, Article ID 101552, 2019.
- [51] Y. Chen, X. H. Yang, Z. Wei et al., "Generative adversarial networks in medical image augmentation: a review," *Computers in Biology and Medicine*, vol. 144, Article ID 105382, May 2022.
- [52] Q. Zhang, H. Wang, H. Lu, D. Won, and S. W. Yoon, "Medical image synthesis with generative adversarial networks for tissue recognition," in *Proceedings of the 2018 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 199–207, NY, USA, June 2018.
- [53] W. Yang and D. Qianqian, "DScGANs: integrate domain knowledge in training dual-path semi-supervised conditional generative adversarial networks and S3VM for ultrasonography thyroid nodules classification," *Lecture Notes in Computer Science*, vol. 11767, pp. 558–566, 2019.
- [54] J. Zhao, X. Zhou, G. Shi et al., "Semantic consistency generative adversarial network for cross-modality domain adaptation in ultrasound thyroid nodule classification," *Applied Intelligence*, vol. 52, no. 9, Article ID 10383, 2022.
- [55] G. Shi, J. Wang, Y. Qiang et al., "Knowledge-guided synthetic medical image adversarial augmentation for ultrasonography thyroid nodule classification," *Computer Methods and Programs in Biomedicine*, vol. 196, Article ID 105611, 2020.
- [56] A. B. Levine, J. Peng, D. Farnell et al., "Synthesis of diagnostic quality cancer pathology images by generative adversarial networks," *The Journal of Pathology*, vol. 252, no. 2, pp. 178–188, 2020.
- [57] M. F. Ferreira, R. Camacho, and L. F. Teixeira, "Autoencoders as weight initialization of deep classification networks applied to papillary thyroid carcinoma," in *Proceedings of the 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 629–632, CA, USA, November 2019.
- [58] M. F. Ferreira, R. Camacho, and L. F. Teixeira, "Using autoencoders as a weight initialization method on deep neural networks for disease detection," *BMC Medical Informatics and Decision Making*, vol. 20, no. S5, p. 141, 2020.
- [59] Z. Li, K. Yang, L. Zhang, C. Wei, P. Yang, and W. Xu, "Classification of thyroid nodules with stacked denoising sparse autoencoder," *International Journal of Endocrinology*, vol. 2020, Article ID 9015713, 8 pages, 2020.
- [60] D. Chen, J. Zhang, and W. Li, "Thyroid nodule classification using two levels attention-based Bi-directional LSTM with ultrasound reports," in *Proceedings of the 2018 9th International Conference on Information Technology in Medicine and Education (ITME)*, pp. 309–312, Hangzhou, China, October 2018.
- [61] X. Wu, H. Y. Wang, P. Shi et al., "Long short-term memory model – a deep learning approach for medical data with irregularity in cancer predication with tumor markers," *Computers in Biology and Medicine*, vol. 144, Article ID 105362, 2022.
- [62] R. Pavithra and L. Parthiban, "Pigeon Inspired Optimization With Deep Belief Network For Thyroid Disease Diagnosis And Classification," *Computer Science, Medicine*, Article ID 231815424, 2020.
- [63] A. M. Begum, M. I. Tresa, A. Professor, and K. Ramakrishnan, "Machine learning based dysfunction thyroid cancer detection with optimal analysis," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 7, pp. 818–823, 2021.
- [64] A. Santillan, R. C. Tomas, R. Bangaol et al., "Discrimination of malignant from benign thyroid lesions through neural networks using FTIR signals obtained from tissues," *Analytical and Bioanalytical Chemistry*, vol. 413, no. 8, pp. 2163–2180, 2021.
- [65] Q. Zhang, S. Zhang, J. Li et al., "Improved Diagnosis of Thyroid Cancer Aided with Deep Learning Applied to Sonographic Text Reports: A Retrospective, Multi-Cohort, Diagnostic Study," *Cancer Biology & Medicine*, vol. 19, no. 5, pp. 733–741, 2021.
- [66] M. E. Tschuchnig, P. Grubmüller, L. M. Stangassinger et al., "Evaluation of multi-scale multiple instance learning to improve thyroid cancer classification," in *Proceedings of the 2022 Eleventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1–6, IEEE, Salzburg, Austria, April 2022.
- [67] S. Baseri Saadi, N. Tataei Sarshar, S. Sadeghi, R. Ranjbarzadeh, M. Kooshki Forooshani, and M. Bendecheche, "Investigation of effectiveness of shuffled frog-leaping optimizer in training a convolution neural network," *Journal of Healthcare Engineering*, vol. 2022, Article ID 4703682, 11 pages, 2022.
- [68] R. Ranjbarzadeh, S. Dorosti, S. Jafarzadeh Ghouschi et al., "Nerve optic segmentation in CT images using a deep learning model and a texture descriptor," *Complex & Intelligent Systems*, vol. 8, no. 4, pp. 3543–3557, 2022.
- [69] S. B. Saadi, R. Ranjbarzadeh, A. Amirabadi et al., "Osteolysis: A Literature Review of Basic Science and Potential Computer-Based Image Processing Detection Methods," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 4196241, 21 pages, 2021.

## Research Article

# Detection and Counting Method of Pigs Based on YOLOV5\_Plus: A Combination of YOLOV5 and Attention Mechanism

Zishun Zhou 

*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610000, China*

Correspondence should be addressed to Zishun Zhou; 2019110102030@std.uestc.edu.cn

Received 15 April 2022; Revised 16 June 2022; Accepted 29 June 2022; Published 22 August 2022

Academic Editor: Saeid Jafarzadeh Ghouschi

Copyright © 2022 Zishun Zhou. 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.

Information-based pig detection and counting is the trend in smart animal husbandry development. Cameras can efficiently collect farm information and combine it with artificial intelligence technology to assist breeders in real-time monitoring and analysis of farming. In order to improve the speed and accuracy of pig detection and counting, an advanced improved YOLO\_v5 method for pig detection and counting based on the attention mechanism is proposed. The model is named as YOLOV5\_Plus. This article utilizes a series of data augmentation methods, including translation, color augmentation, rescaling, and mosaic. The proposed model performs feature extraction on the original image with a backbone network, detects pigs of different sizes with three detection heads, and counts the detected anchor frames. Different versions of YOLOV5 are compared, and YOLOV5x is selected as the baseline model for the best performance. Attention modules are smartly combined with the model so that the model can better handle overlapping and misidentification. YOLOV5\_Plus can achieve an accuracy of 0.989, a recall of 0.996, mAP@.50 of 0.994, and mAP@.50:.95 of 0.796, which outperforms all competing models. The inference time per image during detection is only 24.1 ms. YOLOV5\_Plus model achieves real-time pig number and location detection, which is meaningful for promoting smart animal husbandry and saving labor costs in farming enterprises.

## 1. Introduction

Pork is one of the most important food sources for humans. According to the food and Agriculture Organization of the United Nations (FAO) [1], pork is rich in proteins, amino acids, and vitamins B6 and B12, which are the main ingredients of many traditional diets. In recent years, farm breeding has been expanding, and more and more farms want to monitor farm conditions through digital means accurately. For instance, real-time and accurate monitoring of the number of pigs can help farmers monitor the feeding density scientifically and rationally. In addition, pig counting can assist in accurate feeding, avoid feed waste, improve feed conversion rate, and increase economic efficiency.

Pig counting is facing many difficulties in large-scale agricultural production management. The traditional pig counting method relies on the visual observation of the breeder. Due to the large size of pig herds and overlap,

manual counting of the number of pigs is easy to miss, which is both time-consuming and costly. Pig counting based on deep learning also faces several challenges. For instance, light variations can have an impact on pig detection. Different conditions such as side light, strong light, and backlight can cause more intraclass variance. The varying postures of pigs also make object detection more challenging. Besides, a large scale of overlapping can easily result in low recall in the detection process.

With the rapid development of computer hardware and related theories, deep learning and deep neural networks have significantly improved in accuracy and speed in the past few years. Among them, the progress of target detection algorithms in deep learning is particularly impressive. It mainly includes one-stage algorithms and two-stage algorithms. One-stage algorithms include YOLOV1 [2], SSD [3], YOLOV2 [4], YOLOV3 [5], YOLOV4 [6], YOLOV5, etc. Two-stage algorithms include R-CNN [7], Fast R-CNN [8], Faster R-CNN [9], Mask R-CNN [10], etc. Deep learning-

based target detection algorithms have been widely used in agriculture and animal husbandry. Many algorithms have been designed to be applied to target counting, tracking, and other upper-layer applications. For example, a two-route convolutional neural network (CNN) was proposed by Ramin et al. [11] to detect and classify COVID-19 infection from CT images. To improve classification accuracy, fuzzy c-means clustering and local directional pattern (LDN) encoding methods were used to represent the input image, respectively, to find more complex patterns in the image. In the area of separation of breast cancer lesions, Jafarzadeh Ghouschi et al. [12] designed a well-designed CNN consisting of an autoencoded stacking (SAE) model with a logistic regression layer at the top of the network to monitor the flow. They applied CNNs, VGGs, and residual networks, respectively, to the Breast Cancer Database (BCDR-DM), and the result showed that CNNs outperformed the other two models. Yu et al. [13] proposed a deep learning network model based on multi-modules and attention mechanism (MAN) to realize the counting of cultured fish, which consists of a feature extraction module, attention module, and density map estimation module. Among them, the feature extraction module was composed of three parallel convolutional networks, which are used to extract the general feature map of the image and serve as the input of the subsequent module. The density map estimation module represents the distribution and the number of fishes in the image. The experimental results for MAN showed that the counting accuracy is about 97.12% and the deviation is 3.67. In the point pattern analysis based on YOLO object detection algorithms, Petso et al. [14] treated each animal as a point and identified five animal species by the behavioral pattern of those points. Animal features are harder to detect at higher altitudes and in the presence of environmental camouflage, animal occlusion, and shadows. The point pattern algorithms produced an  $F1$ -score above 96% across all drone altitudes.

Pig detection and counting are essential for accurate and fast counting of livestock. Ahrendt et al. [15] designed a computer vision system based on support maps to track loose housed pigs. It can achieve tracking of at least three pigs at the same time. Tian et al. [16] proposed a new solution for pig counting on the farm using deep learning. The network they designed was based on the combination of counting CNN and ResNeXt model, which achieved an improved high accuracy with low computational cost. The results demonstrated that in real-world data, this method got a mean absolute error of 1.67. Riekert et al. [17] used a deep learning system to detect the position and pose of pigs and achieved 84% mAP@.50 for the day and 58% mAP@.50 for the night. Although scholars have studied pig identification, the accuracy and speed of detection for pig identification are not satisfactory. In addition, pig detection is the basis of pig behavior analysis. Based on the detection and tracking of the pig's location, it is possible to analyze its behavior. For example, Kashiha et al. [18] proposed an ellipse fitting algorithm based on image pattern recognition to detect the position of a pig in a pen and analyze the specific behavior of the pig. Pigs could be identified with an average accuracy of

88.7%. Although the abovementioned researchers have reported studies on pig identification, the accuracy rate for pig identification is still below 95%. Therefore, improving the accuracy of pig identification is still relevant, especially in highly overlapping pig breeding environments.

However, no researcher has yet used YOLO-based algorithms for pig counting. YOLOV5, the latest generation of the YOLO family, is an advanced, fast, and accurate detector. Unlike density map-based counting methods, YOLOV5-based counting can directly detect the location and size of each target and then count the number of pigs based on the identified targets. It allows counting the number of pigs and annotating the pigs directly in the original image, which can better visualize the behavior of the pigs and facilitate movement detection. In recent studies, YOLOV5 has been used in various target detection tasks. Wang and Yan [19] used YOLOV5 for leaf detection and verified that the detection speed of YOLOV5 was significantly higher than Faster R-CNN with no significant difference in mAPs. Wang et al. [20] combined YOLOV5 and Siamrpn++ to propose a high-precision fish detection and tracking algorithm that achieves 76.7% accuracy and enables real-time tracking. Dong et al. [21] proposed a novel lightweight YOLOV5 network designed for vehicle detection. A convolutional block attention module (CBAM) was introduced to the backbone network to improve the model performance to select critical information, and CIoU\_Loss was applied to the bounding box regression loss function to accelerate the bounding box regression rate. The proposed model achieved considerably better results compared to relevant methods. Given the excellent performance of YOLOV5 in terms of detection accuracy and speed, we decided to transfer YOLOV5 to the field of pig detection.

The rest of this article is organized as follows: Section 2 describes the data and algorithms used in the article, including data enhancement, the YOLO series of target detection algorithms, and the attention mechanism. Section 3 shows the model comparison results and validates the performance of the improved algorithm-YOLOV5\_Plus. Section 4 discusses the advantages and limitations of the model. Section 5 summarizes the whole article and provides an outlook on prospects. The main contributions of this study are as follows:

- (1) Color augmentation, translation, shear, rescaling, and mosaic processing for images taken in a pig feeding environment.
- (2) To improve an improved YOLOV5 network architecture based on a convolutional block attention module (CBAM) for the problem of pig aggregation, which tends to lead to poor pig recognition accuracy.

## 2. Materials and Methods

*2.1. Data Description.* The dataset used in this experiment is public [22]. The original data files include 700 images in JPG format, and each photo contains multiple images of one pigpen. The annotation files include 700 files in JSON format. The annotation files and the photos are one-to-one

correspondence. The annotation information includes the coordinates of each pig's top left and bottom right corners of the box rectangle in the picture and the corresponding picture name. Figure 1 shows the samples of annotated pictures. The experiment randomly divided the dataset into training and validation sets of 549 and 151 images, respectively. The test set contains 220 images without annotation information. Compared with the training set, the test set includes more objects than pigs, mainly feeders, troughs, and light cords.

We convert the annotation information of the training set from json format to txt format. The converted annotation format is <id>, <center x>, <center y>, <width>, and <height>.

There are roughly 20 pigs in each image. All annotations were carefully checked manually to mark all pigs in the images.

**2.2. Image Augmentation.** Data augmentation can artificially expand the dataset, increase the diversity of the data, and improve the robustness of the model [23]. In recent years, it has been widely used in various research fields. Common data augmentation methods include geometric transformations such as cropping [24, 25], flipping [24–26], rotating [25, 26], scaling [24, 25], and warping [24, 25], as well as pixel scrambling [27], adding noise [25], illumination adjustment [25, 26], and contrast adjustment [25, 26]. Mosaic was first applied to YOLOV4 and can significantly improve the average precision (mAP). It stitches four images into one mosaic image, which can show more detection of target objects, speed up the effect of training the model, and improve the generalization of the model. Figure 2 shows the results of some augmented images. During the training process of this study, all data enhancement methods are turned on probabilistically.

### 2.3. Detection Principle

**2.3.1. The Development of YOLO.** YOLO was firstly proposed in 2016. Before that, two-stage algorithms usually consisted of two parts: (1) generating candidate regions and (2) classifying the candidate regions using a classification network. Although the two-stage algorithm can achieve high accuracy, the model is often complicated and large with low speed, which makes it hard to enable deployment on mobile devices. The YOLO algorithm transforms the target detection problem into a regression problem. The images are first divided into  $S \times S$  grids, predicting  $B$  bounding boxes. Each grid point predicts the target whose centroid lies at that grid point, its confidence score, and  $C$  conditional category probabilities  $pr(\text{class}_i|\text{object})$ . The confidence equation is as follows:

$$\text{Confidence} = \frac{\Pr(\text{class}_i|\text{object}) \times \Pr(\text{object}) \times IOU_{\text{pred}}^{\text{truth}}}{\Pr(\text{object}) \in \{0, 1\}}, \quad (1)$$

where  $\Pr(\text{object})$  is equal to 1 only if the grid contains one object.

The loss function calculates the sum of the position, width, height, and confidence errors of the prediction box with respect to the ground truth using the mean square error. The non-maximum suppression (NMS) is used to select the best bounding box if multiple bounding boxes detect the same target when detection is performed. YOLOV2 introduces the anchor mechanism, which uses k-means clustering to generate the width and height of the anchor to better match objects of different sizes. The backbone network of YOLOV2 is DarkNet-19, which is quite fast with fewer parameters. YOLOV3 makes some incremental improvements based on YOLOV2. It uses DarkNet-53 as the backbone network, extracting the most advanced techniques for target detection at that time, such as ResNet, DenseNet, and FPN. Compared with ResNet-152, DarkNet-53 has a faster speed (78 FPS compared to 37 FPS in ImageNet) and similar accuracy. YOLOV3 sets nine prediction boxes at each grid point and uses logistic regression to calculate the object score for each bounding box, ignoring the IOU samples more enormous than the threshold but not the best, which can significantly reduce the computational effort.

In YOLOV4, a new backbone network, CSPDarkNet53, is adopted, and the Mish activation function is used instead of LeakyReLU. In the neck network, PANet (path aggregation network) is used for feature fusion instead of FPN. Moreover, the spatial attention module (SAM), an attention mechanism, is introduced. In terms of data augmentation, YOLOV4 proposes the mosaic for the first time, where several images are cropped and stitched together to form new training set elements. These methods made YOLOV4 the state-of-the-art target detection algorithm at that time.

**2.3.2. YOLOV5.** The structure of YOLOV5 is similar to that of YOLOV4. It contains four main parts: input, backbone, neck, and head. The YOLOV5 used in this article is YOLOV5 v6.0, released on 12 October 2021. The structure of YOLOV5s is shown in Figure 3.

The input end includes mosaic data enhancement, image size processing, and adaptive anchor box calculation. Mosaic data enhancement enriches the background and the number of small objects in the dataset by combining four images. Image size processing adaptively adds minimum black borders to the original images of different lengths and widths and uniformly scales them to a standard size. The adaptive anchor box calculation compares the output predicted boxes with the real boxes based on the initial anchor boxes, calculates the gap, and then updates it in reverse, continuously iterating the parameters to obtain the most suitable anchor box value.

The backbone network of YOLOV5 mainly consists of BottleneckCSP(C3), Focus, and SPP (SPPF) modules. The original CSPNet [28] (Cross Stage Partial Network) splits the feature map into two parts. One part is convolved directly, and the other part goes through the dense block. Then the two parts are concatenated together. With CSPNet,

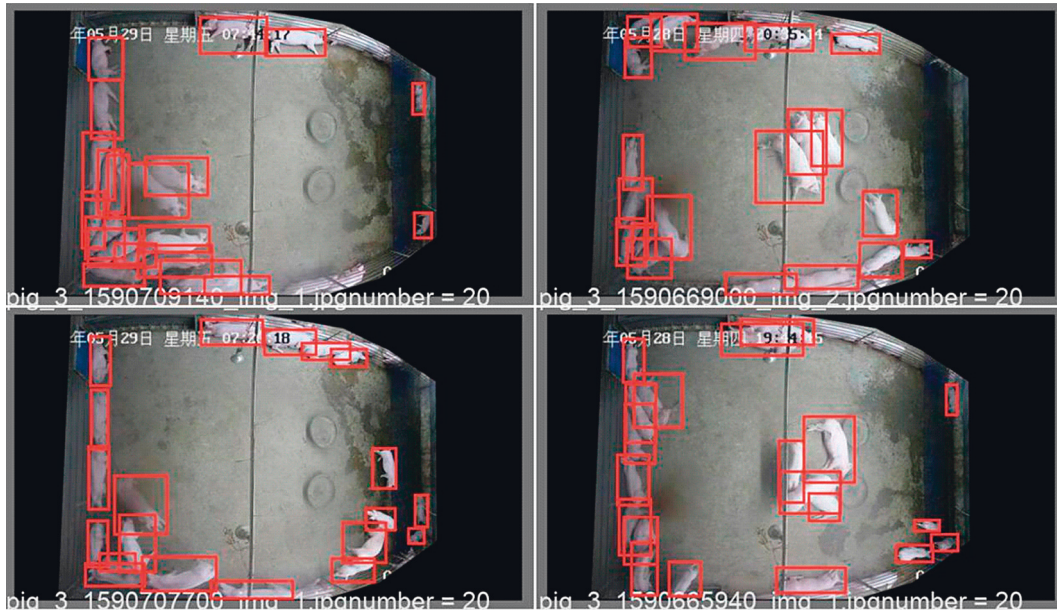


FIGURE 1: Annotation of pigs in the images.

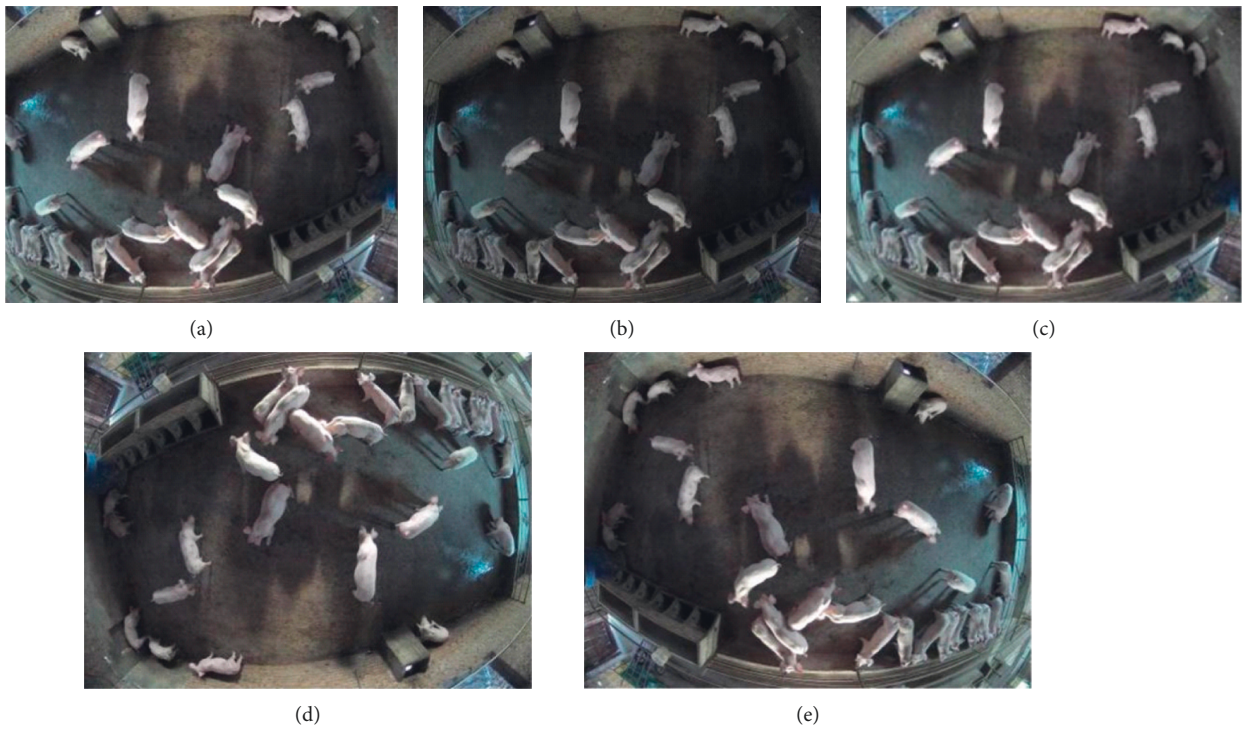


FIGURE 2: Continued.

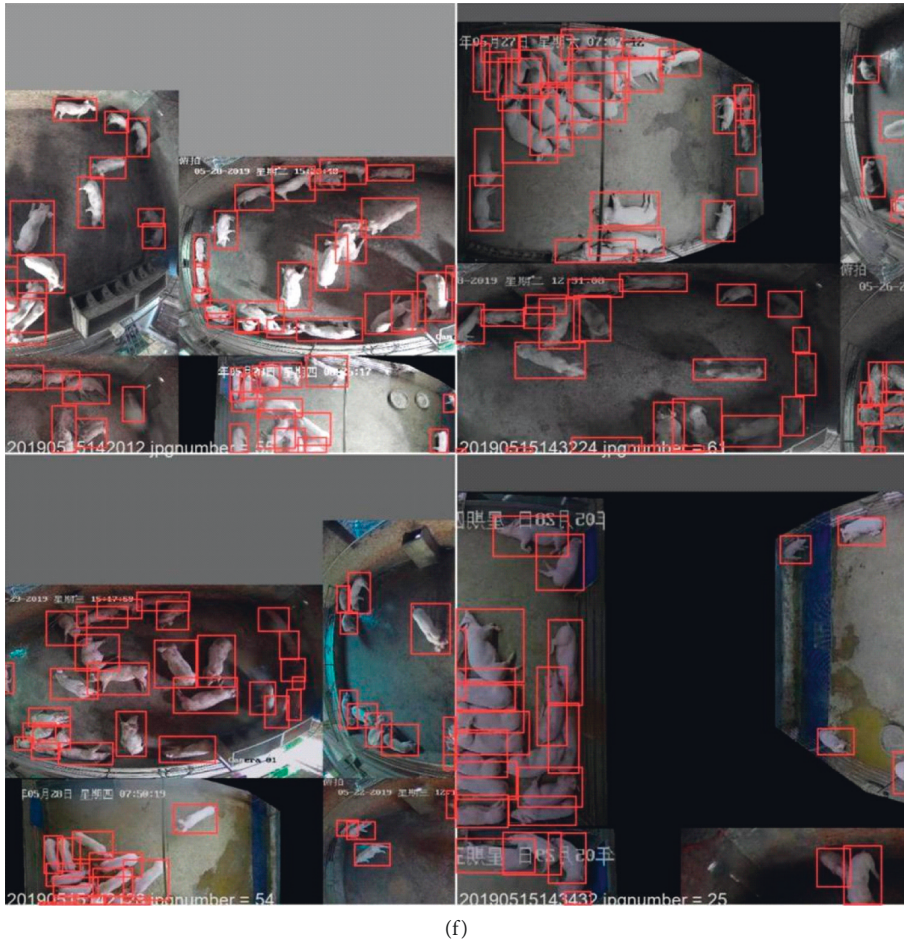


FIGURE 2: Examples of image augmentation. (a) Translate. (b) Color augmentation. (c) Rescaling. (d) Flip up-down. (e) Flip right-left. (f) Mosaic augmentation.

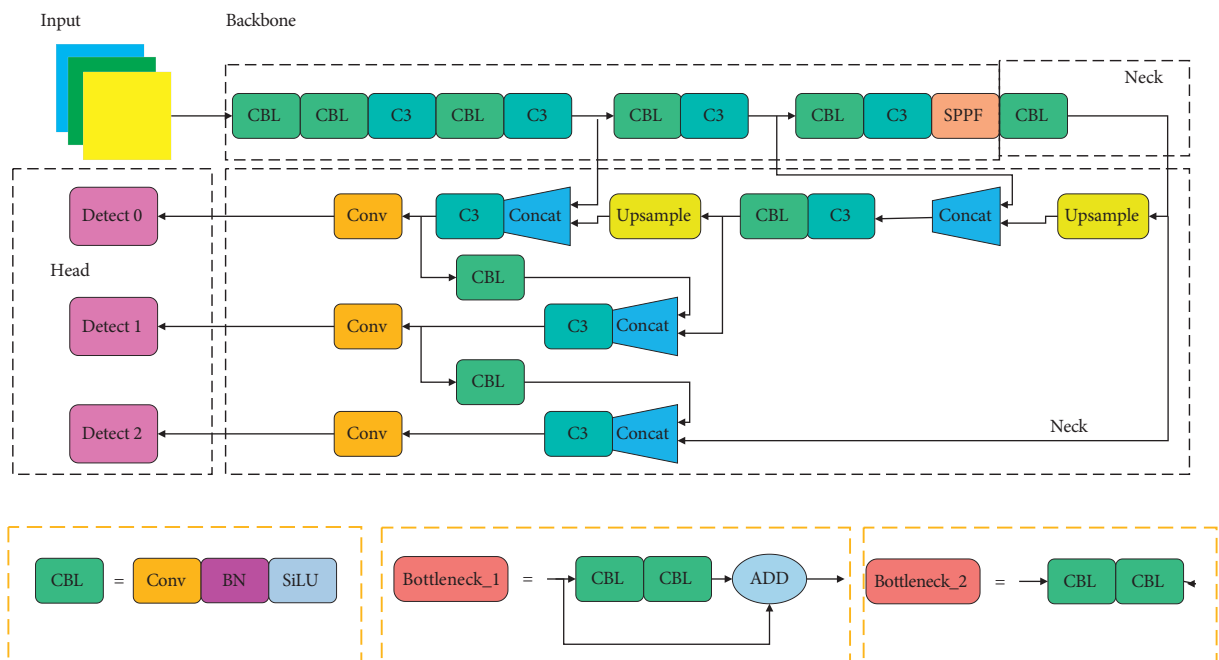


FIGURE 3: The structure of YOLOV5s.

computational costs can be greatly reduced, and accuracy can be improved to a certain extent. In the latest YOLOV5, the C3 module is mainly applied instead of BottleneckCSP. Compared with BottleneckCSP, C3 contains only three convolutions except for the Bottleneck part. It is simpler, faster, and lightweight with similar performance to BottleneckCSP. The bottleneck is one of the components of BottleneckCSP(C3). YOLOV5 uses two different bottlenecks in the backbone and neck: Bottleneck\_1 and Bottleneck\_2. The structure of both bottlenecks is shown in Figure 3. The structure of BottleneckCSP and C3 is shown in Figures 4 and 5, respectively.

The Focus module divides the input image data into four pieces. The four pieces are generated by changing the width and height to 1/2 of the original input data and the number of channels to four times, obtaining the two-fold downsampling feature map without information loss. The Focus module is usually located at the beginning of the backbone network. These four pieces of data are spliced together in the channel dimension and then convolved to obtain a binary downsampled feature map without information loss. The Focus module realizes downsampling while increasing the channel dimension, reducing FLOPs, and increasing speed. The operation of the focus slice is shown in Figure 6.

The SPP module is located at the end of the backbone. The spatial pyramid pooling module executes the maximum pooling with different kernel sizes and fuses the features by concatenating them. It combines different resolutions into the features by pooling the images with different sizes (kernel size = 5, 9, 13). A more lightweight structure SPPF is used in YOLOV5 v6.0 with all kernels of size 5. The structure of SPP and SPPF is shown in Figures 7 and 8, respectively.

YOLOV5 adopts the FPN and PAN structure in the neck. FPN is top-down, passing down the strong semantic features from the top layer to augment the pyramid. However, FPN only enhances the semantic information, not the localization information. PAN adds a bottom-up enhancement behind FPN. The feature map at the top layer can also enjoy the rich location information brought by the bottom layer, which improves the accuracy of the model. The structure of the neck is shown in Figure 9.

The head end outputs a vector with the category probability of the target object, the object score, and the position of the bounding box for that object. The detection network consists of three detection layers with different size feature maps used to detect target objects of different sizes. Each detection layer outputs the corresponding vector and finally generates the prediction bounding box and category of the object in the original image and marks it.

YOLOV5 contains four network structures due to different widths and depths: YOLOV5s, YOLO V5m, YOLOV5l, and YOLOV5x. YOLOV5s have the smallest width and depth with the fastest speed; YOLOV5x has the largest width and depth but runs relatively slower.

**2.3.3. Convolutional Block Attention Module (CBAM).** CBAM [29] was proposed by Woo et al. in 2018, which consists of two submodules: the channel attention module and the spatial attention module.

**(1) Channel Attention Module.** Channel attention focuses on exploring the relationship between feature maps of different channels. To enhance the representational power of the network, the channel attention module uses both average-pooled and max-pooled features, which are fed into a shared MLP with only one hidden layer to generate the channel attention map  $M_C \in R^{C \times 1 \times 1}$ . The final output is obtained by fusing the two feature vectors through element-wise summation. The channel attention formulation is shown in equation (2). Figure 10 shows the structure of the spatial attention module

$$\begin{aligned} M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\ &= \sigma((W_1 W_0 F_{avg}^c) + (W_1 W_0 F_{max}^c)). \end{aligned} \quad (2)$$

**(2) Spatial Attention Module.** Different from channel attention, spatial attention is mainly concerned with where the information is concentrated. The algorithm first performs the average-pooling and max-pooling operations on the channel-refined feature in the extended channel direction to obtain two feature maps with dimensions  $H \times W \times 1$ . Then these two feature maps are stitched together and performed a convolution operation to generate a spatial attention map  $M_S(F) \in R^{H \times W}$ . The formula of spatial attention is presented in Eq. (3). Figure 11 shows the structure of the spatial attention module.

$$\begin{aligned} M_S(F) &= \sigma(f^{7 \times 7}([AvgPool(F); MaxPool(F)])) \\ &= \sigma(f^{7 \times 7}([F_{avg}^S; F_{max}^S])). \end{aligned} \quad (3)$$

**(3) Arrangement of Attention Modules.** Previous research shows that sequential arrangement of channel attention and spatial attention modules can perform better than parallel arrangement. Figure 12 shows the overall structure of CBAM.

**2.3.4. The Improved YOLOV5\_Plus.** YOLOV5\_Plus is an improved model based on YOLOV5x. Previously, attention mechanisms have been tried in other fields combined with YOLOV5. For example, Yan et al. [30] combined the SE module with YOLOV5 to improve the accuracy of coal-gangue classification. Qi et al. [31] achieved high-accuracy recognition of tomato virus disease and improved detection speed based on a YOLOV5 and SE module model. However, how the attention mechanism can be applied to overlapping and dense target recognition and how it performs in the counting domain are unknown. Based on this, we combine the attention mechanism with YOLOV5 and conduct a series of experiments to find the best solution. YOLOV5\_Plus can better extract channel and spatial features in the feature map by utilizing the attention mechanism.

The attention mechanism has been used in various target detection algorithms to obtain better results. Attention mechanisms can make models more robust to objects of different locations and sizes and avoid overfitting problems. For example, Ranjbarzadeh et al. [32] combined cascaded convolutional neural network and DWA (distance-wise

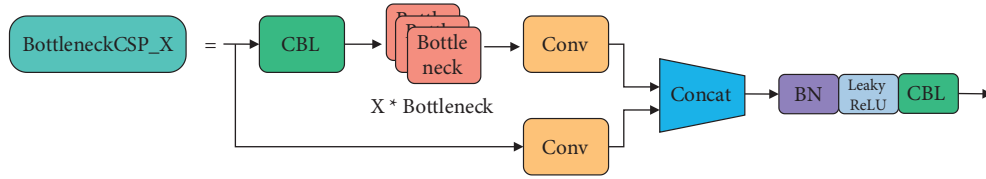


FIGURE 4: The structure of BottleneckCSP\_X.

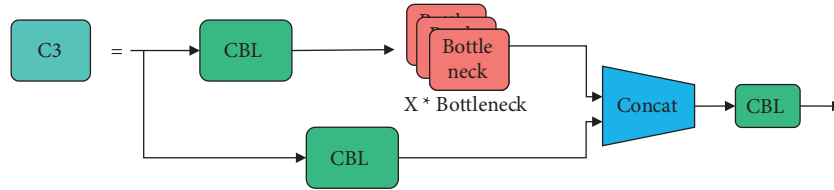


FIGURE 5: The structure of CSP2\_X.

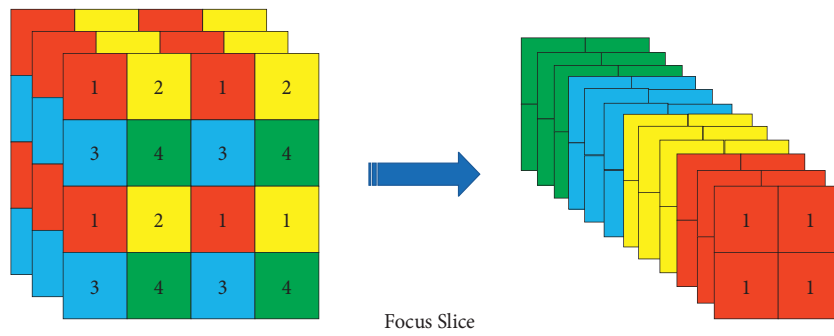


FIGURE 6: Focus slice operation.

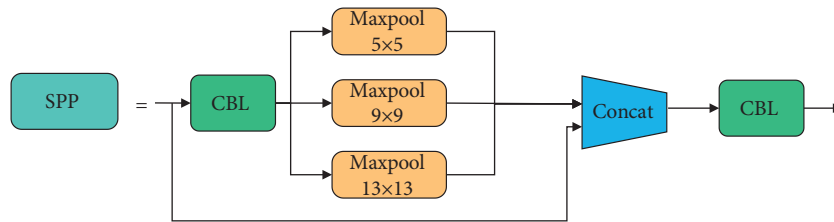


FIGURE 7: The structure of SPP.

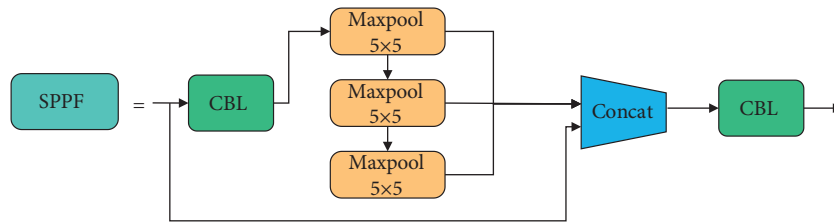


FIGURE 8: The structure of SPPF.

attention) to develop a new brain tumor segmentation architecture. The attention mechanism greatly improved the performance of the model by applying key location features of the image to the fully connected layer. This study carried out a comprehensive and systematic investigation of how the attention module was added. We experimented with different attention modules, including CBAM, SE [33], and

CoordAtt [34]. Different ways of adding attention modules to the model have been investigated. One option is to replace the CBL module in the backbone with an attention module. It is also the way chosen by YOLOV5\_Plus. Another way is to replace the CBL module in BottleneckCSP (C3) with the attention module. Figure 13 shows the improved network structure. From Figure 13(a), it can be seen that the four



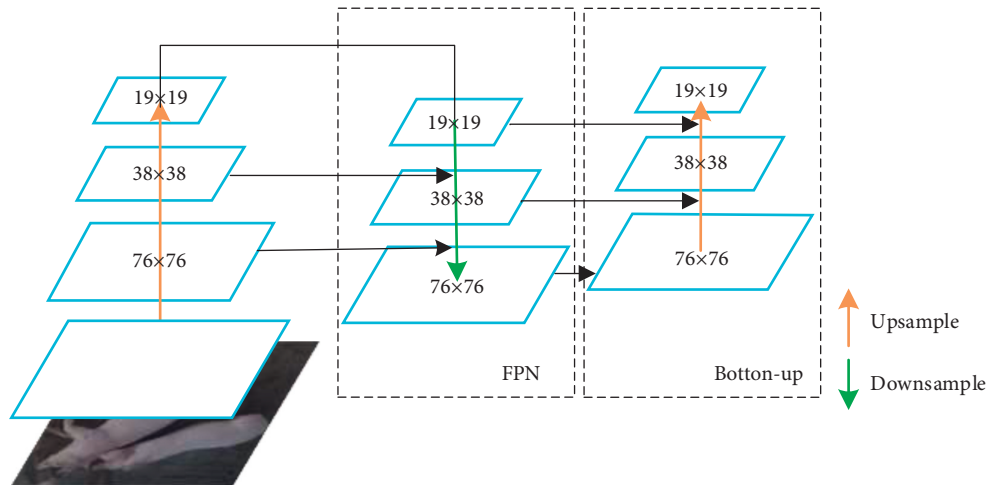


FIGURE 9: The structure of FPN and PAN.

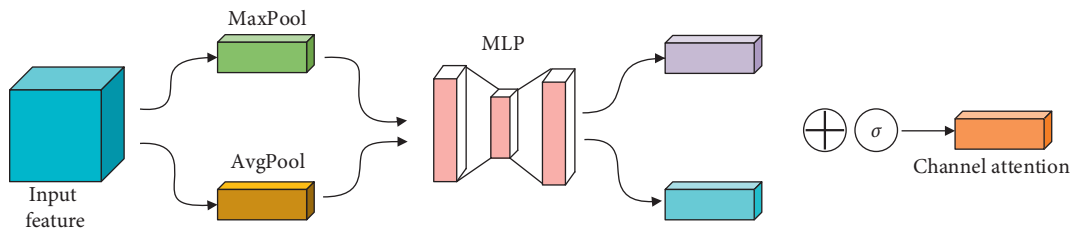


FIGURE 10: Channel attention module.

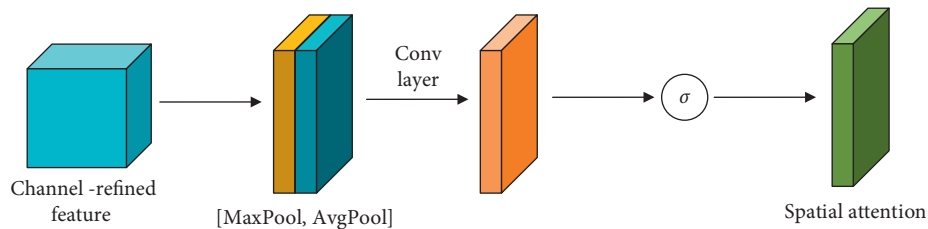


FIGURE 11: Spatial attention module.

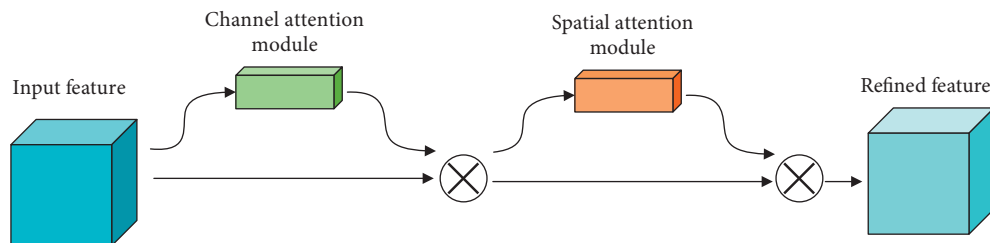


FIGURE 12: The overview of CBAM.

CBLs in the backbone are replaced with attention modules with no increase in the depth of the network. Figure 13(b) shows our attempt to replace the CBL module in BottleneckCSP (C3) with an attention module, but this proposal was not adopted in the end due to poor results. In addition, the attention module is used to replace CBL modules in the backbone, head, and the entire network, respectively, ultimately choosing to replace only the CBL in the backbone.

**2.3.5. Evaluation Indicators in This Article.** In this article, the pig detection model uses precision (equation (4)), recall (equation (5)), average precision (equations (6) and (7)), F1-score (equation (8)), and mAP as evaluation metrics [35]. The confusion matrix is introduced first in Table 1.

Among them, TP stands for true positive (there is a pig in the image, and the algorithm makes a correct prediction), FP stands for false positive (there is no pig in the image, but the

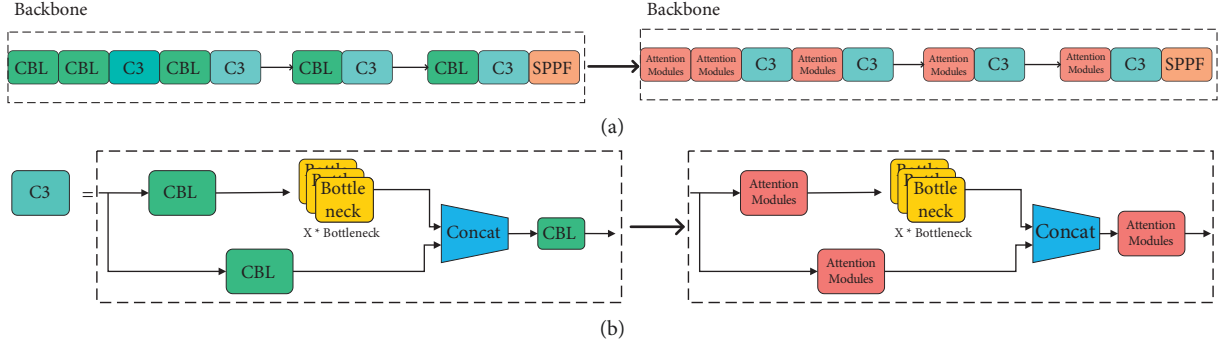


FIGURE 13: Two methods of combining attention modules with YOLOV5. (a) CBLs in the backbone are replaced by attention modules (our proposed model architecture.). (b) CBLs in BottleneckCSP (C3) are replaced by attention modules.

algorithm detects one), and FN stands for false negative (the algorithm fails to detect the pig in the image). Objects whose IOU with ground truth (e.g., IOU = 0.5) is greater than the threshold are considered TP in the algorithm. Objects with IOU less than the threshold are considered FP; those not correctly identified are considered FN.

Precision represents the proportion of positive samples to those predicted to be positive. The recall represents the proportion of detected positive samples to all positive samples. Average-Precision (AP) refers to the Precision-Recall (PR) curve area. It is relatively difficult to calculate AP with integration, so the interpolation method is commonly used as an alternative. AP@.50 represents the average precision when IOU = 0.5. Similarly, AP@.55 represents the average precision when IOU = 0.55. AP@.50: 95 represents the average of AP@.50, AP@.55, . . . , AP@.95. F1-score is the harmonic mean of accuracy and recall. The mAP is the average AP of all classes of objects.

$$\text{precision} = \frac{TP}{TP + FP}, \quad (4)$$

$$\text{recall} = \frac{TP}{TP + FN}, \quad (5)$$

$$AP = \int_0^1 P(R)dR, \quad (6)$$

$$AP_{50:95} = \frac{1}{10} (AP_{50} + AP_{55} + \dots + AP_{90} + AP_{95}), \quad (7)$$

$$F1 - \text{score} = 2 \times \frac{P \times R}{P + R}. \quad (8)$$

### 3. Results

The model is mainly trained on Google Colaboratory with a Tesla P100 GPU, and the hyperparameters of the model are based on scratch.yaml. Table 2 records some necessary parameter settings of the model.

**3.1. Experimental Configuration.** In the model's training process, 541 images are used in the training set, and 149

TABLE 1: Confusion matrix.

	Labeled positive	Labeled negative
Predicted positive	TP	FP
Predicted negative	FN	TN

TABLE 2: The settings of hyperparameters in training.

Parameters	Value
Optimization algorithm	SGD
Epoch	1000
Patience	100
Learning rate	0.01
weight_decay	0.0005
Image size	640
Batch	16 for YOLOV5x and 32 for YOLOV5s
Loss function	BCE loss

images are used in the validation set. The model parameters are divided into three groups for optimization: weights, bias, and batch-norm. The optimizer uses stochastic gradient descent (SGD) by default. One-cycle learning rate optimization strategy is used during training [36]. In the initial three epochs, the algorithm performs warmup epochs with a larger learning rate (0.01). The learning rate becomes 1/10 of the original one at the end of the warmup epochs. The weight decay ratio is 0.0005. In consideration of the limited memory of the GPU, the batch size is set to 16 in most cases. All algorithms are expected to execute 1000 epochs. However, the algorithm will be stopped immediately if the mAP does not rise within 100 rounds. The formula for mAP calculation here is given in equation (9). To avoid the influence of the pretrained model on the results, the pretrained YOLOV5x model on the MS COCO dataset was not used in the experiments.

$$mAP = 0.1 \times mAP_{50} + 0.9 \times mAP_{50:95}. \quad (9)$$

**3.2. Model Selection.** The model selection approach is taken to explore the best solution. A suitable deep learning model is selected from the alternatives. By comparing the performance of different models, we choose the model with the best performance in each metric as the final one.

In this section, we first compare different versions of YOLOV5 and compare the effects of different hyperparameters, including data augmentation and learning rate, on the performance of YOLOV5. The best-performing version of the YOLOV5 model is selected as the baseline model, and it is combined with the attention mechanism to design a new attention mechanism-based YOLOV5 model. We name this model as YOLO\_Plus and explore its performance.

First, the four YOLOV5 models were compared. Complex models are generally considered more effective for feature extraction and more susceptible to overfitting. Different versions of YOLOV5 have different depth-multiple and width-multiple to control the width and depth of the network. YOLOV5 has several versions, such as YOLOV5s, YOLOV5l, YOLOV5x, and YOLOV5m. According to the official figures, each model's performance on MS COCO is shown in Table 3.

Table 4 shows the results of YOLOV5s, YOLOV5l, and YOLOV5x on the test images. All models are trained based on `scratch.yaml`, using SGD as the optimizer and a batch size of 16. It can be seen from the table that YOLOV5x achieves an  $\text{mAP@.50:.95}$  of 0.764, outperforming YOLOV5s and YOLOV5l, which are 0.757 and 0.761, respectively. Figure 14 shows the samples of YOLOV5s, YOLOV5l, and YOLOV5x on the same data. All models show good detection performance and can detect most pigs, which confirms the effectiveness of YOLOV5. However, there are still some misses in the detector. For example, in Figure 14(a), YOLOV5s missed two pigs in the top left corner of the image and two pigs in the bottom left corner of the image; in Figure 14(b), one pig in the top left corner of the image was missed by YOLOV5l. Overall, YOLOV5x achieves the best results: it makes fewer mistakes and achieves a high confidence score for the bounding box produced by the same pig. In a nutshell, YOLOV5x can be considered the best model and thus is chosen as the baseline model. Although YOLOV5s have the fastest inference speed of 5.5 ms, their accuracy does not meet the application requirements.

Proper hyperparameter settings are critical to the effectiveness of training. There are more than 30 hyperparameters in YOLOV5 for different training settings, including learning rate, optimizer, loss function, data augmentation, etc. YOLOV5 provides different hyperparameters such as `hyp.scratch.yaml`, `hyp.finetune.yaml`, and `hyp.finetune_objects365.yaml`. `hyp.scratch.yaml` is the default parameter of YOLOV5 and is optimized for training from scratch on the MS COCO dataset. `hyp.finetune.yaml` is based on a genetic algorithm evolving 306 generations on the COCO dataset. Table 5 shows the results of the model with different hyperparameters of YOLOV5x. It can be seen from the table that the model trained with `hyp.scratch.yaml` as hyperparameter outperforms `hyp.finetune.yaml` in terms of recall, precision,  $\text{mAP@.50}$ , and  $\text{mAP@.50:.95}$ . These differences are significant. It verifies the great influence of the choice of hyperparameters on the results of the experiments. `hyp.scratch.yaml` is used for subsequent experiments.

Although YOLOV5x has achieved a high level of precision and recall and can accurately identify pigs, there are still some issues in dealing with specific situations. These problems can be divided into three main categories, as shown in Figure 15. Category A is overlapping, where multiple pigs are not identified due to clustering and stacking. Category B is missing detection mainly caused by the diversity of light conditions and target size. Category C detects backgrounds such as light ropes, feeders, and walls. In general, categories A and C are the most common errors in detection.

According to our conjecture, the primary reason for errors A and B is that the model fails to extract sufficient features with insufficient learning ability. For error C, the main reason is that the model learns mostly color information but not the shape of the pigs. Therefore, it is easy to misidentify the troughs and feeders which share similar colors. It is also challenging to deal with many pigs being stacked together.

Different attention mechanisms have been combined with YOLOV5x to address the above problems, including CBAM, SE, and CoordAtt. Attention mechanisms can emphasize the important information of the object and suppress some irrelevant details, thus better handling the stacking and misidentification cases. Table 6 shows the results of adding different attentions to the model. The CBAM attention module replaces the CBL module in the backbone network. The SE attention is added at the end of the backbone, and the CoordAtt is located in front of the C3 module in the backbone network. As can be seen from the table, in most cases model combined with the attention mechanism can achieve better performance. The best results for all metrics are obtained using YOLOV5x + CBAM. The recall, precision, and  $\text{mAP@.50:.95}$  are 0.4%, 0.7%, and 3.2% higher than that of YOLOV5x, respectively.

We name the model as YOLOV5\_Plus. Figure 16 shows the changes in the relevant metrics during the training process. To better compare the performance of YOLOV5\_Plus and YOLOV5x, the samples of detection images obtained using YOLOV5\_Plus and YOLOV5x are visualized, as shown in Figure 17. It can be seen that YOLOV5\_Plus has made significant improvements in detection capabilities, greatly reducing the number of missed and false detections. The default detection image size by YOLOV5 is 640. However, during the detection process, we found that resizing the image size to 960 can better avoid the false detection of troughs at the edge of the image. Figure 18 shows the sample images after adjusting the image size. In general, the performance of YOLOV5\_Plus<sub>img\_size = 960</sub> is so impressive. Compared with YOLOV5x, YOLOV5\_Plus achieved higher accuracy and recall. By resizing the image to 960 at detection time, the resolution of the image is increased, so it is more effective in detecting overlapping objects and can better avoid false detection of troughs located at the edge of the image. The study attempted to use larger or smaller resolutions for training or detection, but it did not work well. One possible reason is that too large or too small resolutions do not match the preset anchor sizes in YOLOV5x.

TABLE 3: Performance of YOLOV5s, YOLOV5m, YOLOV5l, and YOLOV5x.

Model	Size (pixels)	mAP(Val) 0.5:0.95	mAP(Val) 0.5	Speed V100 b32 (ms)	Params (M)	FLOPs @640(B)
YOLOV5s	640	37.2	56	0.9	7.2	16.5
YOLOV5m	640	45.2	63.9	1.7	21.2	49
YOLOV5l	640	48.8	67.2	2.7	46.5	109.1
YOLOV5x	640	50.7	68.9	4.8	86.7	205.7

TABLE 4: Performance metrics for the test images using YOLOV5s, YOLOV5x, and YOLOV5l.

Metric	YOLOV5s	YOLOV5x	YOLOV5l
Recall	0.99	0.992	0.994
Precision	0.981	0.982	0.98
mAP .50	0.993	0.993	0.993
mAP .50:.95	0.757	0.764	0.761
F1-score	0.985	0.987	0.987
Inference speed (ms)	5.5	24.1	13

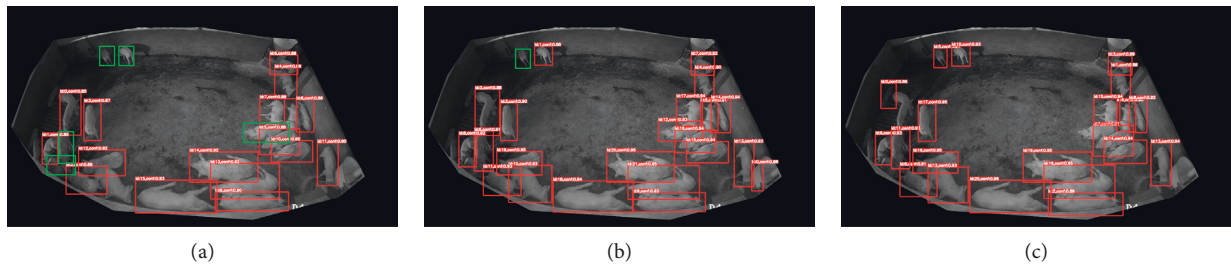


FIGURE 14: Examples of pig detection by YOLOV5s, V5l, and V5x with confidence threshold = 0.85. The pigs detected are marked with red boxes, and those not detected or incorrectly detected are marked with green boxes. (a) YOLOV5s. (b) YOLOV5l. (c) YOLOV5x.

TABLE 5: Performance metrics for the test images using hyp.scratch.yaml and hyp.finetune.yaml.

Metric	hyp.scratch.yaml	hyp.finetune.yaml
Recall	0.992	0.984
Precision	0.982	0.973
mAP .50	0.993	0.991
mAP .50:.95	0.764	0.704
F1-score	0.987	0.978
Inference speed	24.1 ms	23.7 ms

However, an interesting phenomenon can be found in Table 6. The combination of YOLOV5x + CoordAtt is shown a decline in performance on almost all metrics. The reason may lie in the potential limitation that the design of the attention mechanism changes with different data structures, and it may be difficult to couple between two different types of attention mechanisms [37].

Table 7 shows the experimental results when the CBAM modules are added to different locations in the YOLOV5x. The attention module replaces the CBL module in the backbone, head, and the whole network of YOLOV5x. The table shows that the best results are obtained when only the CBAM module is applied to the backbone. After adding the attention module to head and backbone + head, the mAP@.50:.95 drops to 0.753 and 0.763, respectively.

We use the CBAM module to replace the CBL module in the backbone and C3 module, respectively, for comparison. The results are shown in Table 8. It can be seen from the table that using the CBAM module to replace the CBL module in the backbone gives better performance and faster inference speed.

Table 9 shows the experimental training results using YOLOV5x pretrained on MS COCO. The results indicate that transfer datasets can significantly improve the performance of the model, with 0.2%, 0.4%, and 3.1% improvement in recall, precision, and mAP@.50: .95, respectively. The results were only slightly lower than those of the untrained YOLOV5\_Plus. However, we could not pretrain our model on MS COCO due to limited computing power and time.

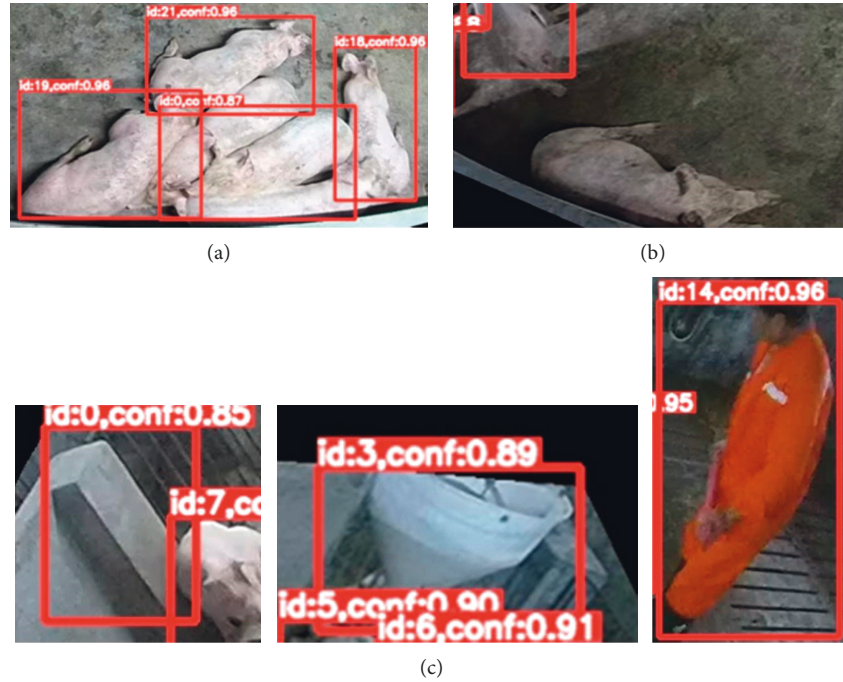


FIGURE 15: Three typical errors in pig detection by YOLOV5x. (a) Overlap. (b) False negative. (c) False positive.

TABLE 6: Comparison of different attention modules combined with YOLOV5x.

Algorithms	Recall	Precision	mAP@.50	mAP@.50: 95	F1-score	Inference speed (ms)
YOLOV5x	0.992	0.982	0.993	0.764	0.987	24.1
YOLOV5x + CBAM (YOLOV5_Plus)	<b>0.996</b>	<b>0.989</b>	<b>0.994</b>	<b>0.796</b>	<b>0.992</b>	24.1
YOLOV5x + SE	0.996	0.985	0.994	0.779	0.99	21.6
YOLOV5x + CoordAtt	0.995	0.982	0.994	0.762	0.988	24.9
YOLOV5x + CBAM + SE	0.996	0.988	0.994	0.778	0.992	25.2

The bold values indicate that the best results for all metrics are obtained using YOLOV5x + CBAM.

Other methods, including focal loss and soft-NMS, have been experimented with but have not yielded good results.

**3.3. Comparison of Target Detection Algorithms.** We compare our YOLOV5\_Plus model with YOLOV3 and YOLOV4. Table 10 shows the performance of the above three algorithms. From the table, we can see that YOLOV5\_Plus achieves the best results in all metrics. In particular, for the mAP@.50: 95, YOLOV5\_Plus achieves a result of 0.796, which improves by 7.1% and 7.9% compared to YOLOV3 and YOLOV4, respectively. These improvements are mainly attributed to innovative modules in YOLOV5, such as Bottleneck\_1, Bottleneck\_2, Focus, and FPN + PAN for better feature fusion. As a result, YOLOV5 can achieve more efficient information extraction at a very fast speed.

## 4. Analysis and Discussion

**4.1. Findings.** Pig detection is an important part of smart animal husbandry. Fast and accurate counting can help farmers improve feeding efficiency and avoid feed waste.

However, lighting changes, shooting perspectives, large scale of overlapping, and varying postures of pigs pose many challenges for pig detection. Our results show that using YOLOV5 combined with the attention mechanism can improve accuracy and recall. Overlapping pigs can be better detected, and misidentified backgrounds can be avoided as much as possible.

Before deep learning, counting livestock usually relied on traditional machine learning methods such as RF and SVM. However, deep learning outperformed other methods in most recent cases and showed great potential [38]. Few papers have applied some of the current state-of-the-art detectors to the field. Therefore, our study can be seen as a new attempt at pig detection.

We chose YOLOV5x as the most suitable baseline model for our task in model selection. It outperforms YOLOV5s and YOLOV5l by 0.7% and 0.3%, respectively, in mAP@.50: .95. It is shown that the scale and complexity of YOLOV5x match the task.

Hyperparameters play an essential role in model performance. Inappropriate hyperparameter selection can cause up to 6% degradation in mAP. We suppose that the

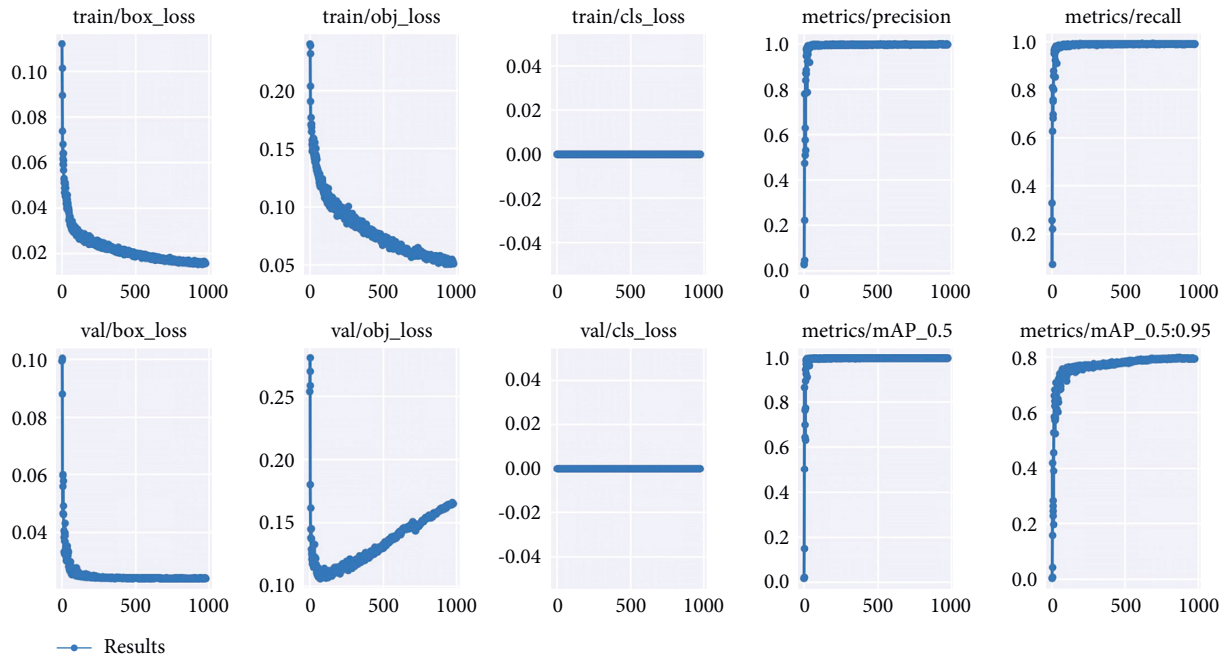


FIGURE 16: Changes in relevant metrics of training YOLOV5\_Plus.

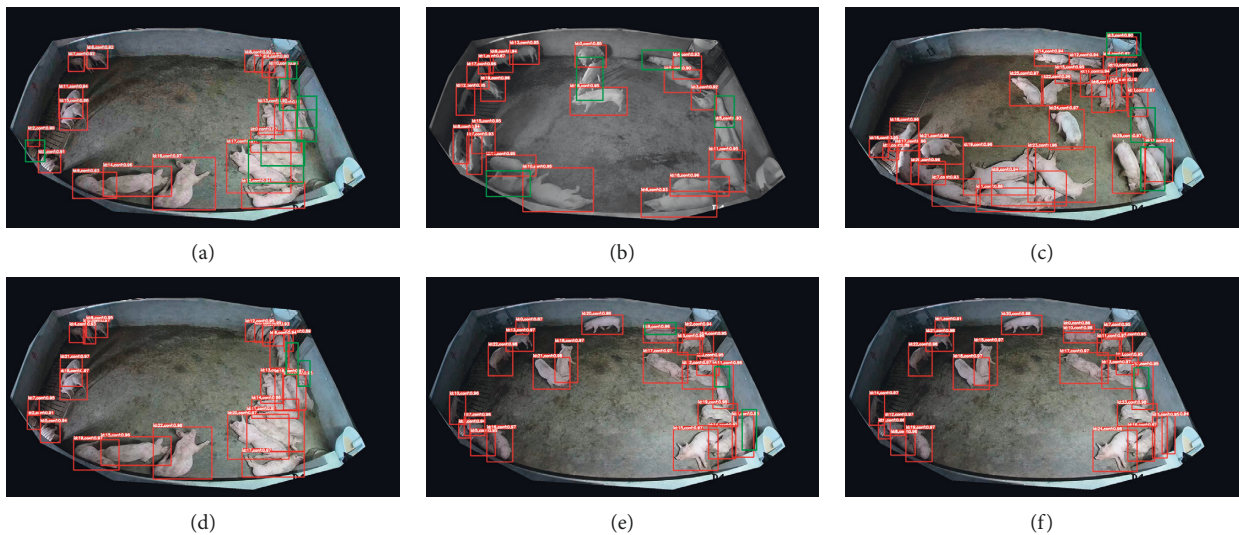


FIGURE 17: Examples of pig detection by YOLOV5x and YOLOV5\_Plus. The first row shows the pictures detected by YOLOV5x. The second row shows the pictures detected by YOLOV5\_Plus. The two pictures in the same column are the same. The pigs detected are marked with red boxes, and those not detected or incorrectly detected are marked with green boxes. (a) YOLOV5x\_1. (b) YOLOV5x\_2. (c) YOLOV5x\_3. (d) YOLOV5\_Plus\_1. (e) YOLOV5\_Plus\_2. (f) YOLOV5\_Plus\_3.

degradation of mAP is mainly attributed to the difference in data augmentation. Therefore, appropriate data augmentation is of great importance for the target detection task.

Attention mechanisms can improve the performance of CNNs and YOLO [39]. We tried popular attention mechanisms, including CBAM, SE, and CoordAtt. The results show that most of attention mechanisms can enhance the performance of the model. The experimental results show that the mAP@.50:.95 can be improved up to 3.2% by attention mechanisms.

Pretraining models are widely used in NLP and other AI-related fields. The AI community has become a consensus to use pretrained models instead of training models from scratch [40]. We compared models pretrained on MS COCO with models without pretraining. The results show that pretraining can substantially improve mAP by approximately +3%.

In addition, the inference speed of our model takes only 24.1 ms per image, which means that we can achieve

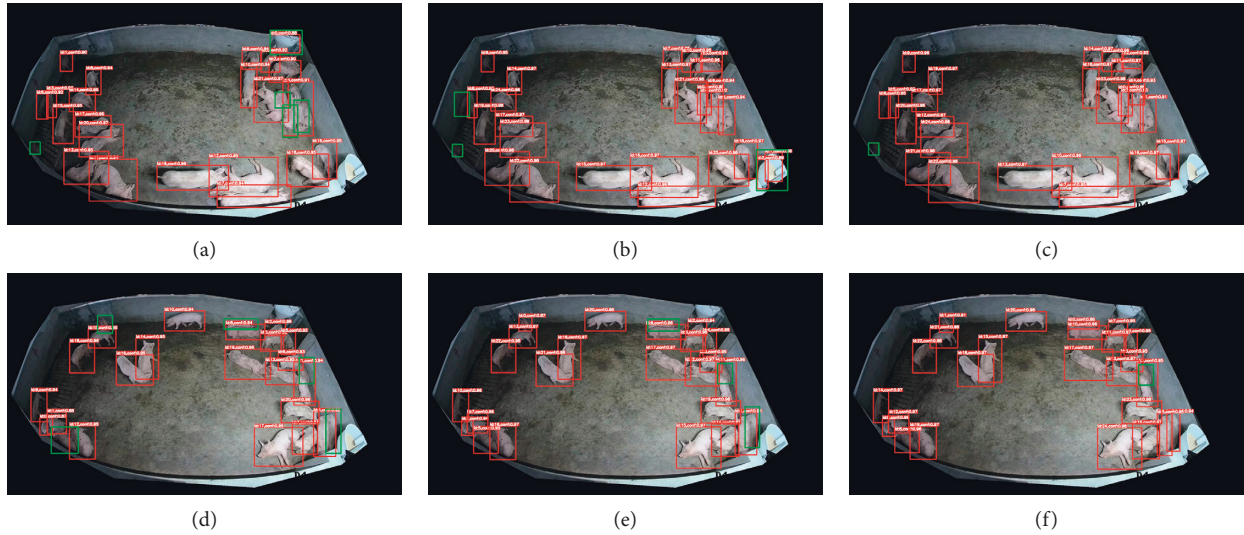


FIGURE 18: Examples of pig detection by YOLOV5x, YOLOV5\_Plus, and YOLOV5\_Plus img\_size = 960 with a confidence threshold of 0.85. The pigs detected are marked with red boxes, and those not detected or incorrectly detected are marked with green boxes. Top row: compared with YOLOV5x and YOLOV5\_Plus, YOLOV5\_Plus img\_size = 960 has fewer false positive and false negative errors. Bottom row: 20 pigs are detected by YOLOV5x, 22 by YOLOV5\_Plus, and 24 by YOLOV5\_Plus img\_size = 960. (a) YOLOV5x. (b) YOLOV5\_Plus. (c) YOLOV5\_Plus img\_size = 960. (d) YOLOV5x. (e) YOLOV5\_Plus. (f) YOLOV5\_Plus img\_size = 960.

TABLE 7: Comparison of adding CBAM in different parts of YOLOV5x.

Algorithms	Recall	Precision	mAP@.50	mAP@.50:.95	F1-score	Inference speed (ms)
YOLOV5x	0.992	0.982	0.993	0.764	0.987	24.1
YOLOV5x + CBAMbackbone (YOLOV5_Plus)	<b>0.996</b>	<b>0.989</b>	<b>0.994</b>	<b>0.796</b>	<b>0.992</b>	24.1
YOLOV5x + CBAMhead	0.993	0.98	0.993	0.753	0.986	24.4
YOLOV5x + CBAMbackbone + head	0.996	0.983	0.994	0.763	0.989	24.2

TABLE 8: Comparison of YOLOV5x with different modules replaced by CBAM.

Algorithms	Recall	Precision	mAP@.50	mAP@.50: 95	F1-score	Inference speed (ms)
YOLOV5x	0.992	0.982	0.993	0.764	0.987	24.1
YOLOV5x + CBAMCBL (YOLOV5_Plus)	<b>0.996</b>	<b>0.989</b>	<b>0.994</b>	<b>0.796</b>	<b>0.992</b>	24.1
YOLOV5x + CBAMC3	0.994	0.987	0.994	0.791	0.99	26.7

TABLE 9: Comparison of transfer models using MS COCO transfer dataset with original YOLOV5x.

Algorithms	Recall	Precision	mAP@.50	mAP@.50:.95	F1-score	Inference speed (ms)
YOLOV5x	0.992	0.982	0.993	0.764	0.987	24.1
YOLOV5x transfer models	0.994	0.986	0.994	0.795	0.99	23.0

TABLE 10: Comparison of recall, precision, mAP@.50, and mAP@.50:.95 in methods.

Algorithms	Recall	Precision	mAP@.50	mAP@.50:.95	F1-score
YOLOV3	0.971	0.989	0.992	0.725	0.980
YOLOV4	0.988	0.835	0.992	0.717	0.905
YOLOV5x_Plus	0.996	0.989	0.994	0.796	0.992

detection speeds of over 40Fps, perfectly meeting the needs of farm applications.

**4.2. Limitations.** Our work still has some limitations. First, although our model combined with the attention mechanism has significantly improved in handling overlap

situations, it still has difficulty handling test images where types of objects are not presented in the training set, such as feeders and troughs. However, since the objective frames identified by YOLOV5\_Plus generally have a higher confidence score, we can screen out most of the troughs by raising the confidence threshold and resizing the test images.

However, there are still some troughs and feeders that cannot be eliminated. We believe this is because the model tends to emphasize the color of the target rather than the shape. The similar posture of pigs and feeders is also one of the possible reasons.

Second, we failed to pretrain our model on MS COCO due to limited computing power and time. The prior knowledge of the pretraining parameters could help our model be trained better and faster. We demonstrated the effectiveness of pretraining using the YOLOV5x model pretrained on MS COCO provided by GitHub.

## 5. Conclusion

This study presents a deep learning method for pig detection on farms. The YOLOV5\_Plus model is a lightweight and efficient model with high accuracy, which is mainly built by combining YOLOV5 and the attention mechanism. The results show that the model can achieve 98.9% recall, 99.6% accuracy, and 79.6% mAP@.50:.95, which are the best results among the competing models. In addition, the size of the detected image can be increased from 640 to 960, which is very effective in avoiding the false detection of troughs. It is worth mentioning that the attention mechanism enables YOLOV5\_Plus to improve a lot compared to other algorithms when dealing with dense pigs and falling pigs. Compared with YOLOV3 and YOLOV4, our proposed model achieves higher precision, recall, and mAP. More data augmentation methods will be explored in future research to improve image quality. Meanwhile, detection heads of different scales can be added to meet the difficulties caused by different degrees of overlapping and relatively small images, increasing the generalizability and robustness of the model.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no competing interests.

## References

- [1] Aga info, [https://www.fao.org/ag/againfo/themes/en/pigs/HH\\_nutrition.html](https://www.fao.org/ag/againfo/themes/en/pigs/HH_nutrition.html), 2013.
- [2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection[C]," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegas NV, June 2016.
- [3] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: Single Shot Multibox detector," in *European Conference on Computer Vision* Springer Cham, New York, NY, USA, 2016.
- [4] J. Redmon and A. Farhadi, "YOLO9000: better faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, Honolulu, HI, July 2017.
- [5] J. Redmon and A. Farhadi, "YOLOV3: An Incremental improvement," 2018, <https://arxiv.org/abs/1804.02767>.
- [6] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOV4: Optimal Speed and Accuracy of Object detection," 2020, <https://arxiv.org/abs/2004.10934>.
- [7] 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, June 2014.
- [8] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, Santiago Chile, December 2015.
- [9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 39, no. 6, Montreal, Canada, December 2015.
- [10] K. He, G. Gkioxari, P. Dollár, and P. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, Venice, Italy, October 2017.
- [11] R. Ranjbarzadeh, S. Jafarzadeh Ghouschi, M. Bendeche et al., "Lung infection segmentation for COVID-19 pneumonia based on a cascade convolutional network from CT images," *BioMed Research International*, vol. 2021, Article ID 5544742, 16 pages, 2021.
- [12] S. Jafarzadeh Ghouschi, R. Ranjbarzadeh, S. A. Najafabadi, E. Osgooei, and E. B. Tirkolaei, "An extended approach to the diagnosis of tumour location in breast cancer using deep learning," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–11, 2021.
- [13] X. Yu, Y. Wang, D. An, and Y. Wei, "Counting method for cultured fishes based on multi-modules and attention mechanism," *Aquacultural Engineering*, vol. 96, Article ID 102215, 2022.
- [14] T. Petso, R. S. Jamisola, D. Mpoeleng, E. Bennitt, and W. Mmereki, "Automatic animal identification from drone camera based on point pattern analysis of herd behaviour," *Ecological Informatics*, vol. 66, Article ID 101485, 2021.
- [15] P. Ahrendt, T. Gregersen, and H. Karstoft, "Development of a real-time computer vision system for tracking loose-housed pigs," *Computers and Electronics in Agriculture*, vol. 76, no. 2, pp. 169–174, 2011.
- [16] M. Tian, H. Guo, H. Chen, Q. Wang, C. Long, and Y Ma, "Automated pig counting using deep learning," *Computers and Electronics in Agriculture*, vol. 163, Article ID 104840, 2019.
- [17] M. Riekert, S. Opderbeck, A. Wild, and E. Gallmann, "Model selection for 24/7 pig position and posture detection by 2D camera imaging and deep learning," *Computers and Electronics in Agriculture*, vol. 187, Article ID 106213, 2021.
- [18] M. Kashiha, C. Bahr, S. Ott et al., "Automatic identification of marked pigs in a pen using image pattern recognition," *Computers and Electronics in Agriculture*, vol. 93, pp. 111–120, 2013.
- [19] L. Wang and W. Q. Yan, "Tree Leaves Detection Based on Deep learning," in *Communications in Computer and Information Science*, pp. 26–38, Springer Cham, New York, NY, USA, 2021.
- [20] H. Wang, S. Zhang, S. Zhao, Q. Wang, D. Li, and R. Zhao, "Real-time detection and tracking of fish abnormal behavior based on improved YOLOV5 and SiamRPN++," *Computers and Electronics in Agriculture*, vol. 192, Article ID 106512, 2022.
- [21] X. Dong, S. Yan, and C. Duan, "A lightweight vehicles detection network model based on YOLOv5," *Engineering*



- Applications of Artificial Intelligence*, vol. 113, Article ID 104914, 2022.
- [22] AI Studio, “Pig Inventory Challenge,” 2021, <https://aistudio.baidu.com/aistudio/datasetdetail/96116>.
- [23] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, “Albumentations: fast and flexible image augmentations,” *Information*, vol. 11, no. 2, p. 125, 2020.
- [24] J. Wu, Z. Zhao, C. Sun, R. Yan, and X. Chen, “Reliability engineering and system safety pin Lyu, Hanbin Zhang, Wenbing Yu, Chao Liu. A novel model-independent data augmentation method for fault diagnosis in smart manufacturing,” *Procedia CIRP*, vol. 107, pp. 949–954, 2022.
- [25] P. Lyu, H. Zhang, W. Yu, and C. Liu, “A novel model-independent data augmentation method for fault diagnosis in smart manufacturing,” *Procedia CIRP*, vol. 107, pp. 949–954, 2022.
- [26] F. Gao, L. Fu, X. Zhang et al., “Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN,” *Computers and Electronics in Agriculture*, vol. 176, Article ID 105634, 2020.
- [27] M. Demirtaş, “A new RGB color image encryption scheme based on cross-channel pixel and bit scrambling using chaos,” *Optik*, vol. 265, Article ID 169430, 2022.
- [28] C. Y. Wang, H. Y. M. Liao, and Y. H. Wu, “CSPNet: A New Backbone that Can Enhance Learning Capability of CNN,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 390–391, Seattle, WA, USA, November 2019.
- [29] S. Woo, J. Park, J. Y. Lee, and S. Kweon, “Cbam: Convolutional Block Attention module,” *Computer Vision - ECCV 2018*, Springer Cham, New York, NY, USA, 2018.
- [30] P. Yan, Q. Sun, N. Yin, L. Hua, S. Shang, and C. Zhang, “Detection of coal and gangue based on improved YOLOv5.1 which embedded scSE module,” *Measurement*, vol. 188, Article ID 110530, 2022.
- [31] J. Qi, X. Liu, K. Liu et al., “An improved YOLOv5 model based on visual attention mechanism: application to recognition of tomato virus disease,” *Computers and Electronics in Agriculture*, vol. 194, Article ID 106780, 2022.
- [32] R. Ranjbarzadeh, A. Bagherian Kasgari, S. Jafarzadeh Ghouschi, S. Anari, M. Naseri, and M. Bendecheche, “Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images,” *Scientific Reports*, vol. 11, no. 1, Article ID 10930, 2021.
- [33] 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, Salt Lake City UT, June 2018.
- [34] Q. Hou, D. Zhou, and J. Feng, “Coordinate attention for efficient mobile network design,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Article ID 13713, Nashville, TN, USA, June 2021.
- [35] J. Wang, J. Wei, and S. Mei, “Improved YOLOv3 for small object detection in remote sensing images,” *Computer Engineering and Applications*, vol. 57, no. 20, pp. 133–141, 2021.
- [36] T. He, Z. Zhang, H. Zhang, J. Xie, and M. Li, “Bag of tricks for image classification with convolutional neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 558–567, Long Beach CA, June 2019.
- [37] J. Kim, S. Lee, E. Hwang et al., “Limitations of deep learning attention mechanisms in clinical research: empirical case study based on the Korean diabetic disease setting,” *Journal of Medical Internet Research*, vol. 22, no. 12, Article ID e18418, 2020.
- [38] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: a survey,” *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.
- [39] W. Li, K. Liu, L. Zhang, and F. Cheng, “Object detection based on an adaptive attention mechanism,” *Scientific Reports*, vol. 10, no. 1, Article ID 11307, 2020.
- [40] X. Han, Z. Zhang, and N. Ding, “Pre-trained Models: Past, Present and future,” *AI Open*, vol. 2, pp. 225–250, 2021.

## Research Article

# Urban Rail Transit Network Planning Based on Particle Swarm Optimization Algorithm

**Ning Yu** 

*School of Architecture and Civil Engineering, Qiqihar University, Qiqihar 161006, China*

Correspondence should be addressed to Ning Yu; [yun@qqhru.edu.cn](mailto:yun@qqhru.edu.cn)

Received 6 June 2022; Revised 11 July 2022; Accepted 15 July 2022; Published 21 August 2022

Academic Editor: Ramin Ranjbarzadeh

Copyright © 2022 Ning Yu. 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 order to solve the problem that the urban rail transit network is affected by a large number of signals, resulting in poor control effect, and improve the living comfort of residents near urban rail transit, a study on urban rail transit network planning based on particle swarm optimization algorithm is proposed. The learning factor is dynamically adjusted according to the inertia weight parameters, and the particle swarm optimization parameters are selected in combination with the setting of the maximum velocity parameters. The individual optimal particle is selected by using the dominant relationship between the individual particles, and the optimization of the optimal particle is completed by combining the selection requirements of the global optimal particle. We design the v2x communication implementation scheme, obtain the traffic flow information of urban rail transit, build the signal input and output model based on particle swarm optimization algorithm, obtain the output feedback signal, and determine the planning scale of urban rail transit network, so as to build the urban rail transit network planning model and complete the urban rail transit network planning. The experimental results show that the proposed method can improve the utilization rate of urban rail transit network planning, effectively control the change of network signal amplitude, and reduce the repetition rate of urban rail transit network planning.

## 1. Introduction

The layout of the rail transit network should be consistent with the urban form, land use layout, and development direction. The planning and layout of the urban rail transit network should comply with the needs of the future land development of the city. The two restrict and depend on each other, and finally form a coincidence [1, 2]. Urban rail transit network planning is a special plan in urban comprehensive transportation planning, which refers to a system to determine the overall rationality and scientificity of urban rail transit system on the basis of urban overall planning and urban comprehensive transportation planning. Its significance and role are enormous. The backbone system of urban rail transit network shall conform to the dominant passenger flow direction of the city, form a complementary relationship with the urban road system, and smoothly connect with other urban public transport network systems to facilitate their conversion. The network planning shall reflect the

unity of stability, flexibility, and continuity. The network planning of the central urban area should be relatively stable, the urban fringe should leave room for development, and the entire network should be able to expand and develop with the adjustment and expansion of the city scale. The choice of urban rail transit mode should be based on local conditions, and the composition of functional levels should be reasonable and unified [3]. Just as the urban road system planning needs to define the road functions at different levels, the rail transit network also needs to be divided into different functional levels. The rail transit lines at this level adopt different standards and adopt flexible construction, operation, and traffic organization schemes to meet the traffic needs at all levels. The planning and construction of urban rail transit network should be combined with the current situation, taking into account the economic and social benefits and energy consumption, and should be based on the network planning, implementation, and gradual improvement [4–7]. Due to the nonrepeatability of

construction and huge project investment, the construction of each line must be based on the line network planning, carefully considered and decided according to the urban structure, urban function positioning, and construction needs, so as to avoid waste.

Relevant scholars have studied this, and reference [8] put forward the use of NetLogo to achieve the agent optimization matching of passenger demand and service supply of urban rail transit network. Passenger demand and service supply are one of the most important factors determining the performance of urban rail transit system. In this paper, the multi-agent dynamic interaction technology is used to transform the multi-criteria problem into a distributed artificial intelligence optimization problem. In terms of demand, the dynamic passenger transport demand with agent is modeled from the perspective of bounded rational travel decision. At the supply end, a train traffic dynamic service supply model based on Agent is established. The simulation results emphasize the importance of jointly/interactively representing the supply side and the demand side. These results are of great significance for formulating effective capacity utilization strategies of urban rail transit networks and policies to provide passengers with a high level of service. Reference [9] is put forward based on the influence of rail transit station, sustainability overlap multi-objective optimization of land use distribution, considering the multiple rail transit site overlapping influence the characteristic of land use, this article developed a new multi-objective land use allocation optimization model, to make sure every undeveloped land area city land use type and intensity. The pareto optimal land development scheme with minimum deviation is obtained by using non-dominated ranking genetic algorithms and ranking technology similar to ideal solution. The areas of influence of the two sites in the study area partially overlap, with many plots undeveloped. The results show that considering the comprehensive effect of multiple rail transit stations, the newly developed optimal land use allocation model can reasonably achieve the multi-objective coordination of urban sustainable development to the maximum extent.

Although some progress has been made in the above research, the tracking interval of urban transport trains is shortened significantly and the density of trains is increased significantly. In this situation, the difficulty of urban rail transit network planning and operation adjustment is further increased. Because of the great urban rail transit project investment, long construction period, especially the circuit and layout of urban land use, development pattern, and urbanization play an important role, so the research on urban rail transit network planning is of great importance, is proposed based on particle swarm optimization algorithm of urban rail transit line network planning study. Compared with other intelligent algorithms, particle swarm optimization algorithm has a simple algorithm principle, fast optimization speed, strong global and local convergence ability, and is very suitable for solving large-scale network planning problems. The experiment proves that the research content has good effect and strong feasibility. Using the dominant relationship between individual particles to select

the individual optimal particle, combined with the global optimal particle selection requirements to complete the optimization of the optimal particle; The V2X communication implementation scheme was designed to obtain the traffic flow information of urban rail, and the signal input and output model based on particle swarm optimization algorithm was constructed to obtain the output feedback signal and determine the planning scale of urban rail transit network, so as to complete the planning of urban rail transit network. The experimental results show that the proposed method can reduce the repetition rate of urban rail transit network planning.

## 2. Urban Rail Transit Network Planning Based on Particle Swarm Optimization Algorithm

*2.1. Particle Swarm Optimization Algorithm.* Particle swarm optimization algorithm is also translated into particle swarm optimization algorithm, particle swarm optimization algorithm, or particle swarm optimization algorithm. It is a stochastic programming algorithm based on group cooperation developed by simulating the foraging behavior of birds [10]. It is generally considered as a kind of cluster intelligence, which can be incorporated into multi-agent optimization system. The particle swarm optimization algorithm is initialized as a group of random particles (random solutions), and then the optimal solution is found through iteration. In each iteration, the particles update themselves by tracking two "extreme values" [11]. The first is the optimal solution found by the particle itself. This solution is called the individual extreme value  $p_{best}$ . The other extreme value is the optimal solution found by the whole population. This extreme value is the global extreme value  $g_{best}$ . In addition, we cannot use the whole population but only the neighbors of some of the optimal particles, so the extremum in all neighbors is the local extremum.

*2.1.1. Selecting Particle Swarm Optimization Parameters.* The design of urban rail transit network planning under particle swarm optimization algorithm needs to control some parameters, such as particle swarm size, learning factor, inertia weight, population topology, and maximum speed [12]. How to select particle swarm optimization parameters will have a direct impact on the optimization effect of urban rail transit network planning.

*(1) Inertia Weight Parameters of Particle Swarm Optimization.* The particle swarm inertia weight parameter  $G_c$  can measure the planning performance of the particle swarm optimization algorithm and make the algorithm distribute reasonably in the planning process of the whole urban rail transit network. As the inertia weight of particle swarm increases, the planning ability of the algorithm will be enhanced. As the inertia weight of particle swarm decreases, the algorithm will only have local planning ability [13]. Generally, the optimal range of particle swarm inertia weight parameter is  $0 \sim 1.4$ , but the convergence speed of the

algorithm will be accelerated when the value of particle swarm inertia weight parameter is  $0.8 \sim 1.2$ . When the inertia weight parameter of particle swarm optimization is set in the range of  $0.8 \sim 1.2$ , the convergence speed of the algorithm will be improved. Therefore, in order to enable the algorithm to have the ability of global planning of urban rail transit network at the initial stage of calculation and local planning at the end of calculation, the inertia weight of particle swarm optimization is set to decrease with the calculation of the algorithm [14]. Because the algorithm has two factors of calculation cost and effect, the linear reduction strategy of inertia weight is adopted to reduce the inertia weight value to about 0.4 as the optimal parameter of particle swarm inertia weight.

Particle swarm optimization algorithm has many advantages, but it will make the particle swarm follow the optimal particle to fly. After several iterations of the algorithm, the particle swarm will have a strong convergence, resulting in the slow convergence of the algorithm [15]. In order to overcome the shortcomings of the algorithm in iterative calculation, and considering the influence of inertia weight parameters on the algorithm performance, the inertia weight value is set to 0.7, and the inertia weight parameters are reduced to 0.4 with the increase of iterative calculation times. The change of inertia weight parameters is determined by

$$G_c = \frac{M_{\max} - M_{\min}}{T_{\max}} \quad (1)$$

In formula (1),  $M_{\max}$  represents the maximum weight parameter,  $M_{\min}$  represents the minimum weight parameter, and  $T_{\max}$  represents the longest calculation time.

(2) *Learning Factors  $x_1$  and  $x_2$* . The introduction of inertia weight parameter can improve the retrieval performance of the algorithm, but the learning factor also affects the planning performance of particle swarm optimization algorithm. The selection of learning factors  $x_1$  and  $x_2$  is through the trajectory of particle swarm, and  $x_1 + x_2 \leq 4$  must be met at the same time. The learning factors  $x_1$  and  $x_2$  are the factors that control the particles to learn from the optimal individual, so as to control the approach to the best advantage in the particle swarm. If  $x_1 = 0$ , the particles do not have their own experience, and the convergence speed of the algorithm is relatively fast. However, when the algorithm is applied to problems with high complexity, it will make the algorithm very easy to enter the local optimal state. If  $x_2 = 0$ , it means that the particles only have their own experience, and there is no information shared by the particle swarm. Because there is no communication between the individual particles, the probability of the algorithm getting the solution is very small.

The two parameters of learning factor  $x_1$  and  $x_2$  have their own adjustment functions, but the adjustment of the two parameters is separated from each other, which weakens the unity of the algorithm in the process of urban rail transit network planning and is not conducive to the optimization and retrieval performance of the algorithm. Select the learning factor as the nonlinear function of the inertia weight parameter, and the calculation formula is as follows:

$$\begin{cases} x_1 = G_c^2 + G_c + 0.4, \\ x_2 = 0.4 + x_1. \end{cases} \quad (2)$$

The optimization algorithm of particle swarm optimization algorithm changes with the change of learning factors  $x_1$  and  $x_2$ , and the speed of particles will also change. When optimizing the parameters of learning factors  $x_1$  and  $x_2$ ,  $x_1 = x_2$  is usually selected, and the range of parameters is set between 0 and 4. In order to improve the unity of the algorithm, it is generally used to dynamically adjust the learning factor and set the maximum speed parameter through the inertia weight parameters to complete the selection of particle swarm optimization parameters.

2.1.2. *Optimizing Optimal Particles*. Particle Swarm Optimization (PSO) algorithm often ignores the improvement of updating the individual extremum of particles, and usually uses the dominant relationship between particles to extract the optimal individual of particles. During the evolution of particle swarm, the extreme value of particles cannot be updated. First, calculate the intensity  $Q_d$  of a single particle in the particle swarm, the fitness value  $S_y$  of a single particle, and the intensity  $Q_d$  of a single particle in the particle swarm can be calculated according to

$$Q_d = \frac{m_i}{O \times (x_1 + x_2)} \quad (3)$$

In formula (3),  $m_i$  represents the number of individual particles dominated by the  $i$ th particle in the particle swarm, and  $O$  represents the size of the particle swarm.

In a particle swarm, the fitness value  $S_y$  of a single particle is determined by the individual strength of all particles dominated by the population. The calculation formula is

$$S_y = \frac{Q_d}{R_{ab}} \quad (4)$$

In formula (4),  $R_{ab}$  represents the dominant relationship between particle individual  $a$  and particle  $b$ .

In addition to the selection of individual particles, the optimization of optimal particles also considers the selection of global optimal particles. Generally, a solution is randomly selected in the non-inferior solution set of particle swarm optimization as the global optimal particle. In order to improve the diversity of algorithm planning, the global optimal particles are optimized by dynamic weighting method. Calculate the fitness of each particle in the particle swarm according to formula (5). The calculation formula is

$$F_H = \frac{1}{V_b \times K} \quad (5)$$

In formula (5),  $V_b$  represents the position of the  $b$ th particle, and  $K$  represents the final position of the particle.

Using the dominant relationship between individual particles, combined with the selection requirements of individual optimal particles and global optimal particles, the optimization of optimal particles is completed, so as to

improve the update of individual extremum of particles; The specific process of optimizing optimal particles is shown in Figure 1.

According to the optimized particle flow, the implementation steps of particle swarm optimization algorithm are analyzed in detail:

Step 1: initialize the specific position of particles in particle swarm;

Step 2: calculate the average optimal position of particle swarm;

Step 3: calculate the fitness value of the current position of the particle, compare the fitness value calculated in the previous iteration, if it is less than the result of the previous iteration, it will be updated to the current position of the particle according to the historical position of the particle, and calculate the current global optimal position of the particle swarm;

Step 4: compare the results of the previous iteration with the current global optimal position of the particle. If the value of the current global optimal position of the particle is better than that of the previous iteration, the global optimal position of the particle swarm is used to update the position of the particle;

Step 5: calculate the optimal position of each dimension of the particle to obtain the random point position of a particle; Calculate the latest position of the particle;

Step 6: output the results to complete the optimal particle optimization.

To sum up, through the selection of parameters such as the learning factor and inertia weight of the particle swarm, the particle swarm optimization and the selection of the individual particle optimal position and the global optimal particle of the particle swarm are carried out to complete the optimization of the optimal particle; Finally, by optimizing the optimal particle flow design, the application of particle swarm optimization algorithm in urban rail transit network planning is realized.

**2.2. Intelligent Control of Urban Rail Transit Signal Based on V2X Communication Technology.** The intelligent control of urban rail transit signals based on V2X (vehicle to everything) network uses V2X technology to obtain vehicle operation status information, and controls urban rail transit signals according to the obtained information. Multiplexing cellular network is the main means for V2X communication system to realize information transmission. The structure of V2X communication system is shown in Figure 2.

When the data transmitter of a V2X system is in the uplink transmission mode, it will affect the fixed base station, resulting in that the signals received by the base station are all interference signals with the same source. At this time, multiplexing problems are prone to occur in the network, resulting in data disorder in the transmission process [16]. In order to solve the above problems, the urban rail transit signal intelligent control method calculates the signal to interference ratio of the communication link, and analyzes

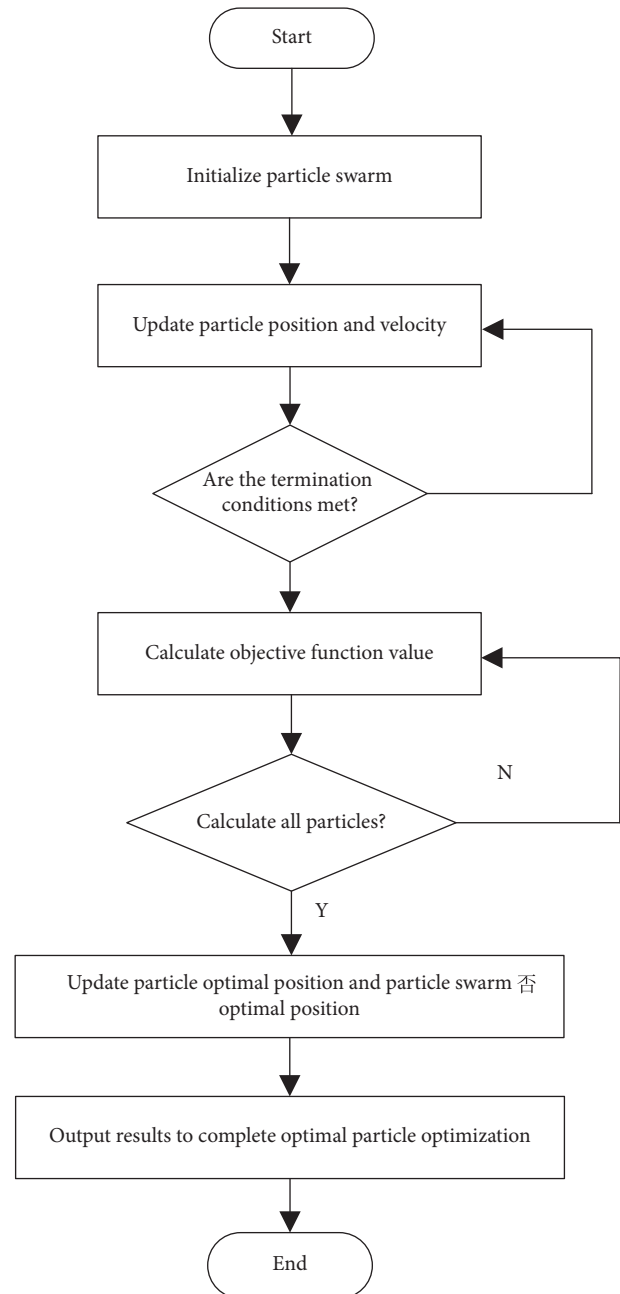


FIGURE 1: Flow chart of optimal particle optimization.

the multiplexing problem of the link in the communication process according to the calculation results.

If the data link multiplexing phenomenon in the communication network interferes with the signal transmission, most communication links will take measures to improve the signal reception quality, including reducing the bit error rate and improving the signal-to-noise ratio, which will affect other multiplexing terminals and interfere with the communication of other links. At this time, the original multiplexing terminal will receive the rebound interference, and such repetition will have an adverse impact. In order to solve the above problems, a communication implementation scheme is designed under the V2X communication technology, as shown in Figure 3.

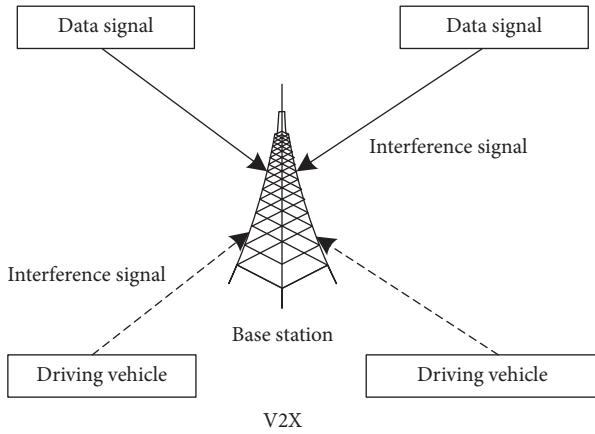


FIGURE 2: V2X communication system structure.

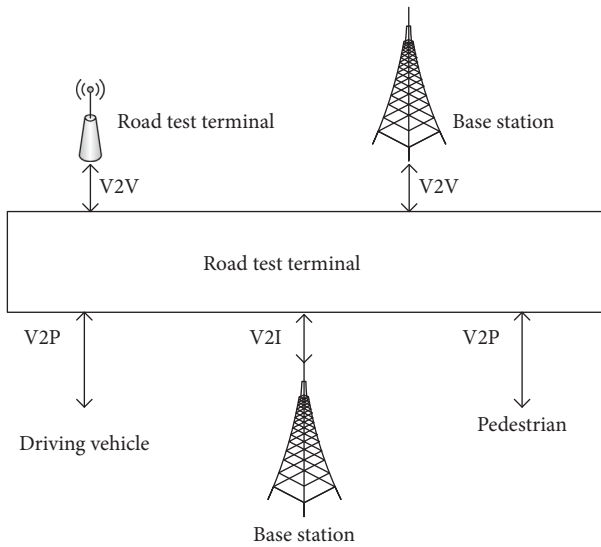


FIGURE 3: V2X communication implementation scheme.

According to Figure 3, the V2X communication implementation scheme establishes a wireless multi hop connection by combining the global positioning system and wireless communication technologies, such as wireless LAN, cellular network, to provide high-speed data access services for vehicles in high-speed mobile state, so as to realize the information interaction between V2X.

### 2.3. Research on Urban Rail Transit Network Planning

**2.3.1. Determining the Planning Scale of Urban Rail Transit Network.** Before determining the planning scale of urban rail transit network, it is necessary to establish the planning scale index of urban rail transit network, including the total length index of urban rail transit route, traffic density index, and transportation capacity index [17]. The total length of urban rail transit lines is

$$L_z = \sum L_i \times F_H. \quad (6)$$

In formula (6),  $L_i$  represents the length of the  $i$  traffic line, and  $L_z$  can indirectly describe the overall size of urban rail transit planning scale, which is a key index to evaluate the passenger flow transmission capacity and traffic equipment demand of urban rail transit.

The formula for calculating the transportation capacity index of urban rail transit is

$$H_J = \sum_{i=1}^m M_i \times L_z. \quad (7)$$

In formula (7),  $M_i$  represents the passenger flow transmission capacity of the  $i$  traffic line, and the urban rail transit transmission capacity index  $H_J$  can reflect the impact of urban rail transit on the passenger transport system.

In the process of determining the total length  $L_z$  of urban rail transit lines according to urban population size and land resources, considering that there is a certain close correlation between the scale of urban rail transit and land area and total population [18], a signal input-output model based on particle swarm optimization algorithm can be established as shown in Figure 4, the model refers to the structure that describes the characteristics of the system with the external characteristics of the input and output variables of the system.

It can be seen from Figure 4 that on this basis, the asynchronous in-phase control method is adopted to limit the control target parameters. When the output signal of the model satisfies the Gaussian stationarity, the feedback result at the output is consistent with the filter transmission result of the model signal.

The results of the signal input-output model based on particle swarm optimization algorithm are used to calculate the total amount of urban rail transit. The calculation formula is as follows:

$$Z_l = Z_z \times Q_{cb} \times B_{xc} \times H_J. \quad (8)$$

In formula (8),  $Z_z$  represents the total amount of urban rail transit,  $Q_{cb}$  represents the per capita travel rate, and  $B_{xc}$  represents the expected value of the population.

The planning scale of urban rail transit network is obtained through the determination of three indicators, namely, the total length indicator of urban rail transit route, the transmission capacity indicator, and the total amount indicator of urban rail transit, so as to eliminate the impact of traffic structure on urban rail transit network planning and determine the planning scale of urban rail transit network.

**2.3.2. Construction of the Urban Rail Transit Network Planning Model.** Just as previous studies on urban space [19–23], the passenger flow distribution points shall be marked on the urban rail transit lines, and the alternative stations shall be marked one by one along the passenger flow distribution points at the urban rail transit stations. The final planning scheme of the transportation network shall be determined according to the passenger flow coverage of each transportation station. If there are alternative stations in the

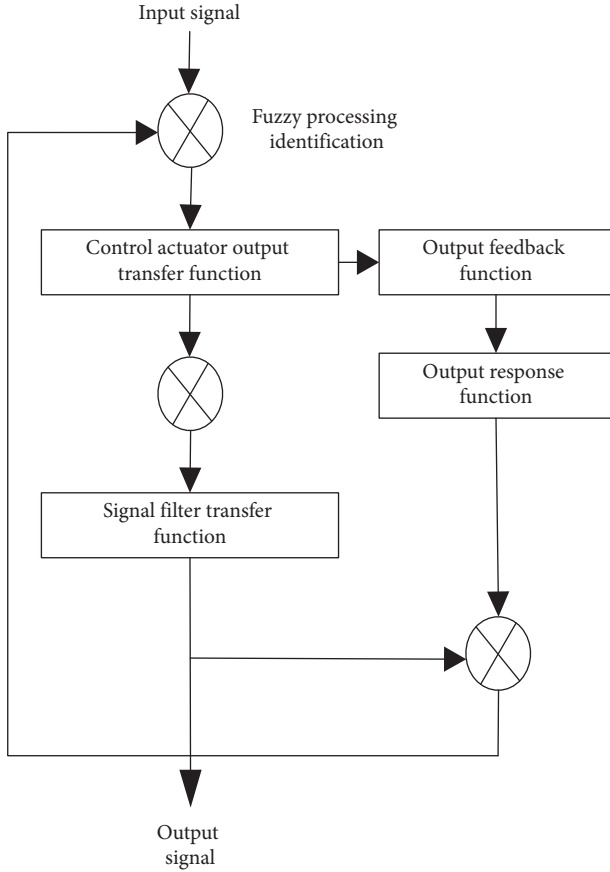


FIGURE 4: Signal input and output model based on particle swarm optimization algorithm.

urban rail transit station set, if stations are selected from the central city, the total number of stations can be expressed as

$$Z_{ZD} = \ln\left(\frac{L}{\bar{l}_i}\right) \times Z_l. \quad (9)$$

In formula (9),  $Z_{ZD}$  represents the total number of stations on the urban rail transit line,  $\ln$  represents the rounding function, and  $\bar{l}_i$  represents the average length of the distance between the optimal traffic stations.

Take the starting point of the urban rail transit line as the center of the circle and  $d_{\min}$  and  $d_{\max}$  as the radius to make a circle. When the urban rail transit line intersects at two points, the traffic station between the two intersections is the alternative station. If there is only one traffic alternative station, the selection of this station is unique. If there are multiple alternative stations, it indicates that there are multiple urban rail transit network planning schemes. Each alternative station is rounded in the above way until the last traffic station, and then the alternative stations and multiple alternative schemes are obtained, as shown in Figure 5.

Although urban rail transit stations can be selected in the above steps, the determination of  $d_{\min}$  and  $d_{\max}$  are difficult, and there is no fixed value between  $d_{\min}$  and  $d_{\max}$ . When selecting urban rail transit stations, the distance difference between  $d_{\min}$  and  $d_{\max}$  are very large, which increases the number of options finally obtained.

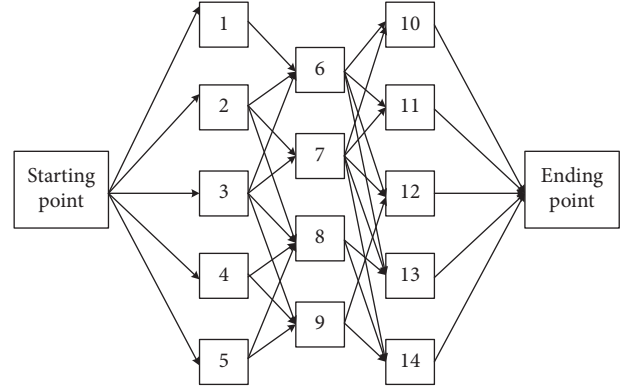


FIGURE 5: Schematic diagram of the formation of urban rail transit stations.

Based on the passenger flow distribution range of urban rail transit stations, the urban rail transit network planning model is calculated as follows:

$$E_{ij} = P \times (1 - r) \times (d_{\max} - d_{\min}). \quad (10)$$

In formula (10),  $P$  represents a constant, and  $r$  represents the weighted sum of population density and travel density.

According to the established model, the traffic station coverage in urban rail transit network planning can be obtained, so as to analyze the coverage of bus stations and select the planning scheme with the largest traffic station coverage. Therefore, the passenger flow coverage of urban rail transit lines with the best planning scheme can be obtained.

Here, the optimal planning scheme is found among the planning schemes with the same number of transport stations, and the urban rail transit network planning is realized by calculating the passenger flow coverage of urban rail transit planning stations.

### 3. Experimental Analysis

In order to test the application performance of urban rail transit network planning based on particle swarm optimization algorithm, the simulation experiment is carried out. The experiment is designed by Matlab, and a simulation platform is established. The signal acquisition frequency on the platform is 25 KHz, the sampling time is 1500 s, and the baud interval width is 15 BPs. Set the node coverage of particle swarm to  $500 \times 500$ , the optimal number of sub nodes is 64, the information communication coverage radius of relay transmission node is  $R = 1.25$ , the total node size is 1000, the improvement duration is set to 60 min (lasting for 15 sampling points), the time sampling interval of sampling points is 10 min, and the maximum iteration round is 1500. See Table 1 for other parameter settings.

According to the above environment and the parameter settings in Table 1, the urban rail transit network planning test under the particle swarm optimization algorithm is carried out. Taking a city rail train as the research object, the city rail supports the interconnection between equipment of different signal manufacturers. The total length of the urban

TABLE 1: Experimental parameter settings.

Parameter name	Parameter size	Parameter name	Parameter size
Iterations	30	Inertia weight	0.7298
Spatial dimension	6	Simulation area	200 × 200
Routing and forwarding protocol	IEE802.15.4	Penalty coefficient	2000
Smoothness weight	0.3	Traffic line weight	0.7

rail transit line is 16 kilometers, and the airport can be reached through the railway station.

The studied urban rail transit network planning method based on particle swarm optimization algorithm, reference [8] method and reference [9] method are used for comparison, and the comparison results of urban rail transit network planning utilization are obtained, as shown in Figure 6.

It can be seen from the experimental results in Figure 6 that when reference [9] method is used to plan the urban rail transit network, the method does not know the overall scale of the urban rail transit network planning in advance, which makes the utilization rate of the urban rail transit network planning low; Reference [8] method in the planning process, the utilization rate of urban rail transit planning is higher than that of the planning optimization method in reference [9], and is relatively stable. With the increase of urban rail transit network planning area, the utilization rate of urban rail transit network planning tends to be stable, but the utilization rate is still not ideal; When using the planning optimization method in this paper to plan the urban rail transit network, this method not only determines the network planning scale but also establishes the urban rail transit network planning model, which simplifies the calculation process and effectively improves the utilization rate of urban rail transit network planning. The reason is that before the planning scale of urban rail transit network is determined, the planning scale index of urban rail transit network is established, including the total length index of urban rail transit route, traffic density index, and transport capacity index, which is beneficial to improve the planning of urban rail transit network line to a certain extent.

In this experimental case, the signal amplitude change of the original integrated communication network of urban rail transit network planning is shown in Figure 7.

It can be seen from Figure 7 that the signal amplitude changes regularly under normal running conditions, and the fluctuation range is [-1, 1]. Based on this, the urban rail transit network planning method based on particle swarm optimization algorithm is used to analyze whether the signal amplitude change of urban rail transit dedicated communication integrated network is consistent with the normal running of trains. The results are shown in Figure 8.

It can be seen from Figure 8 that the network signal amplitude changes regularly using the urban rail transit network planning method based on particle swarm optimization algorithm, in which the change amplitude is [-1, 1]. It is basically consistent with the signal amplitude change of the original communication integrated network, and it is more perfect at 1.2 s and 1.5 s. The comparison results show that the research planning method

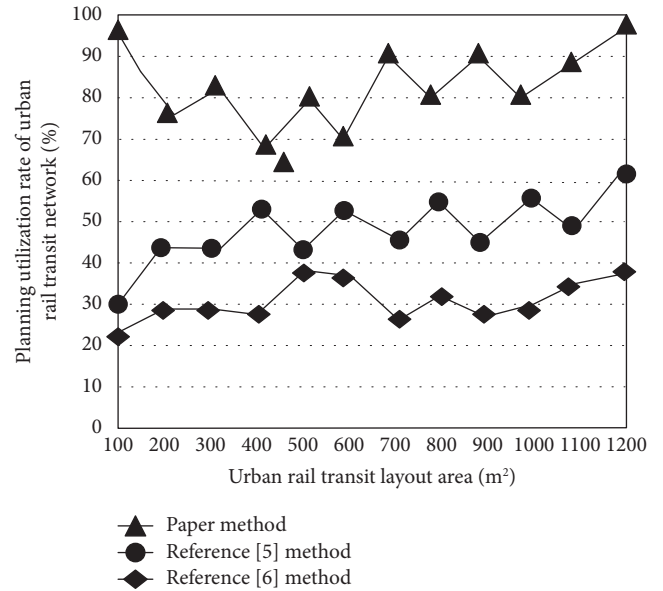


FIGURE 6: Planned utilization rate of urban rail transit network.

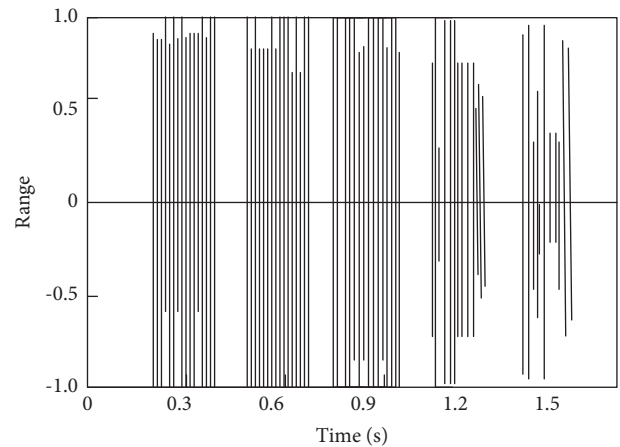


FIGURE 7: Amplitude change of original communication integrated network signal.

can effectively control the amplitude change of network signal. The reason is that the method in this paper adopts V2X technology to obtain vehicle operating status information, and control urban rail transit signals based on the obtained information, which is conducive to controlling signal amplitude changes to a certain extent.

In order to further improve the feasibility of the research method, the research method, reference [8] method, and reference [9] method are, respectively, used to carry out the



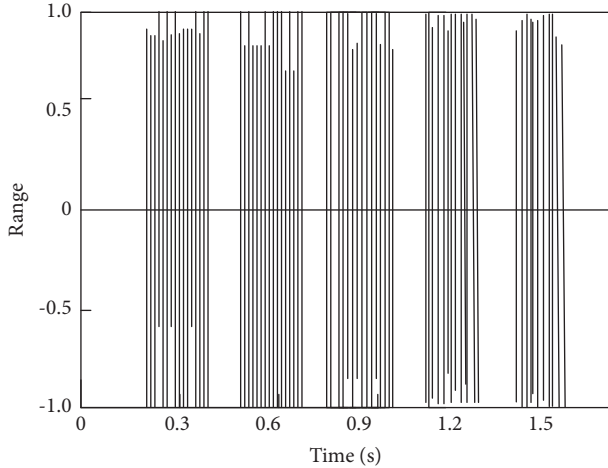


FIGURE 8: Signal amplitude change analysis.

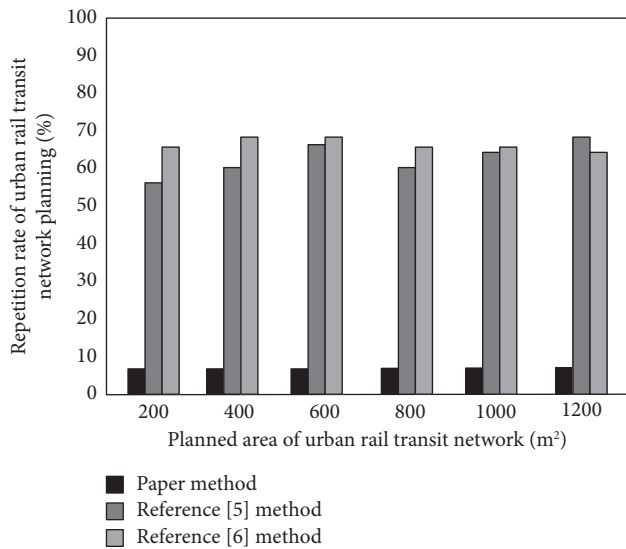


FIGURE 9: Comparison results of repetition rate of urban rail transit network planning.

comparison experiment on the repetition rate of urban rail transit network planning. The results are shown in Figure 9.

It can be seen from the experimental results in Figure 9 that when the planning optimization method in this paper is adopted, the repetition rate of urban rail transit network planning is far lower than that of the other two methods with the change of urban rail transit network planning area. In contrast, the planning methods in reference [8] and reference [9] have high repetition rate in urban rail transit network planning, resulting in poor planning effect of urban rail transit network layout. The reason is to mark the passenger flow distribution points on the urban rail transit line, mark the alternative stations along the passenger flow distribution points one by one in the urban rail transit station, and determine the final planning scheme of the transportation line network according to the coverage of the passenger flow of each traffic station, so as to benefit and reduce the planning repetition rate of the transportation line network.

Based on the above experimental results, it can be seen that the proposed method has a good effect on improving the utilization rate of urban rail transit network planning and the repetition rate of urban rail transit network planning, comprehensively improving the planning effect of urban rail transit network, and the network signal amplitude changes regularly.

## 4. Conclusion and Prospect

### 4.1. Conclusion

- (1) When using the planning optimization method in this paper to plan the urban rail transit network, this method not only determines the network planning scale but also establishes the urban rail transit network planning model, which simplifies the calculation process and effectively improves the utilization rate of urban rail transit network planning.
- (2) The amplitude variation of network signal using the method studied is regular, and the amplitude variation is  $[-1, 1]$ . It is basically consistent with the signal amplitude change of the original communication integrated network. It is more perfect at 1.2 s and 1.5 s to effectively control the signal amplitude change of the network.
- (3) When using the planning optimization method in this paper, with the change of urban rail transit network planning area, the repetition rate of urban rail transit network planning is low, which has a good effect.

4.2. *Prospect.* In the research of urban rail transit network planning based on particle swarm optimization algorithm studied in this paper, there are still the following problems to be solved, which need to be further studied and improved:

- (1) Considering the complexity of urban rail transit network planning, some factors need to be simplified in the modeling process, and a single objective optimization model is established. In the next study, all factors and optimization indexes should be considered as much as possible to establish a more accurate multi-objective optimization model for urban rail transit network planning.
- (2) The train operation adjustment simulation system mainly starts from ATS system. It needs to realize the simulation function of urban rail transit network planning based on particle swarm optimization algorithm, and fully consider the information interaction with the planning system. In order to make it practical, more perfect function modules should be designed in combination with the actual situation on-site.
- (3) More practical applications are needed to prove that it is feasible to solve the urban rail transit network planning problem based on particle swarm optimization algorithm, and the performance comparison with other intelligent methods will be realized in further research.

## Data Availability

Requests for access to the data used to support the findings of this study should be made to Ning Yu (yun@qghru.edu.cn).

## Conflicts of Interest

The author declares no conflicts of interest.

## Authors' Contributions

Ning Yu performed conceptualization and formal analysis and developed the methodology and wrote the draft.

## Acknowledgments

This work was supported by Young Innovative Talents Project of Basic Scientific Research Business Expenses of Heilongjiang Provincial Undergraduate Universities (145109236).

## References

- [1] M. Li, X. Zhou, Y. Wang, L. Jia, and M. An, "Modelling cascade dynamics of passenger flow congestion in urban rail transit network induced by train delay - ScienceDirect," *Alexandria Engineering Journal*, vol. 61, no. 11, pp. 8797–8807, 2022.
- [2] Y. Ge and B. Wang, "Predicting the impact of urban rail transit construction on employment attractiveness based on the GM (1, 1) model: a case study of Guangzhou," *IOP Conference Series: Earth and Environmental Science*, vol. 647, no. 1, pp. 012215–13135, 2021.
- [3] J. Na, J. Zhu, J. Zheng, S. Di, H. Ding, and L. Ma, "Cellular automata based land-use change simulation considering spatio-temporal influence heterogeneity of light rail transit construction: a case in nanjing, China," *ISPRS International Journal of Geo-Information*, vol. 10, no. 5, pp. 308–320, 2021.
- [4] C. Deng, X. Zhang, Y. Huang, and Y. Bao, "Equipping seasonal exponential smoothing models with particle swarm optimization algorithm for electricity consumption forecasting," *Energies*, vol. 14, no. 13, pp. 4036–4114, 2021.
- [5] W. H. Zhang, L. C. Chou, and M. Chen, "Consumer perception and use intention for household distributed photovoltaic systems," *Sustainable Energy Technologies and Assessments*, vol. 51, Article ID 101895, 2022.
- [6] J. Zhang and W. Huang, "A pilot assessment of new energy usage behaviors: the impacts of environmental accident, cognitions and new energy policies," *Frontiers in Environmental Science*, vol. 10, 2022.
- [7] M. Chen and W. H. Zhang, "Purchase intention for hydrogen automobile among Chinese citizens: the influence of environmental concern and perceived social value," *International Journal of Hydrogen Energy*, vol. 46, no. 34, pp. 18000–18010, 2021.
- [8] J. Zhang, "Agent-based optimizing match between passenger demand and service supply for urban rail transit network with NetLogo," *IEEE Access*, vol. 9, no. 99, pp. 32064–32080, 2021.
- [9] X. Feng, Z. Tao, X. Niu, and Z. Ruan, "Multi-objective land use allocation optimization in view of overlapped influences of rail transit stations," *Sustainability*, vol. 13, no. 23, pp. 13219–13314, 2021.
- [10] Y. Zhang, T. Zuo, M. Zhu, C. Huang, J. Li, and Z. Xu, "Research on multi-train energy saving optimization based on cooperative multi-objective particle swarm optimization algorithm," *International Journal of Energy Research*, vol. 45, no. 2, pp. 2644–2667, 2021.
- [11] D. Wen, C. Shi, K. Liao, J. Liu, and Y. Zhang, "Fast backfire double annealing particle swarm optimization algorithm for parameter identification of permanent magnet synchronous motor," *Progress In Electromagnetics Research M*, vol. 104, no. 1, pp. 23–38, 2021.
- [12] K. E. Purushothaman and N. Velmurugan, "Multiobjective optimization based on self-organizing Particle Swarm Optimization algorithm for massive MIMO 5G wireless network," *International Journal of Communication Systems*, vol. 34, no. 5, pp. 1–15, 2021.
- [13] S. Kaya, A. Gümüüşcü, İ. B. Aydilek, I. H. Karacizmeli, and M. E. Tenekeci, "Solution for flow shop scheduling problems using chaotic hybrid firefly and particle swarm optimization algorithm with improved local search," *Soft Computing*, vol. 25, no. 10, pp. 7143–7154, 2021.
- [14] A. Kartono, D. Anggraini, S. T. Wahyudi, and A. A. Setiawan, "Study of parameters estimation of the three-compartment pharmacokinetic model using particle swarm optimization algorithm," *Journal of Physics: Conference Series*, vol. 1805, no. 1, pp. 012032–32, 2021.
- [15] T. Dya, B. B. Blaise, G. Betchewe, and M. Alidou, "Implementation of particle swarm optimization algorithm in Matlab code for hyperelastic characterization," *World Journal of Mechanics*, vol. 11, no. 07, pp. 146–163, 2021.
- [16] S. Muhammad-Arif, A. Hussain, T. Tjing-Lie, S. Muhammad-Ahsan, and H. Abbas-Khan, "Analytical hybrid particle swarm optimization algorithm for optimal siting and sizing of distributed generation in smart grid," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 6, pp. 1221–1230, 2020.
- [17] H. Li, Q. Xu, and Y. Liu, "Method for diagnosing the uneven settlement of a rail transit tunnel based on the spatial correlation of high-density strain measurement points," *Sustainability*, vol. 13, no. 16, pp. 9245–9256, 2021.
- [18] H. Marouani, "Optimization for the redundancy allocation problem of reliability using an improved particle swarm optimization algorithm," *Journal of Optimization*, vol. 2021, no. 2021, pp. 1–9, Article ID 6385713, 2021.
- [19] Q. Yang, H. Zhan, and J. Huang, "Urban green service equity in Xiamen based on network analysis and concentration degree of resources," *Open Geosciences*, vol. 14, no. 1, pp. 304–315, 2022.
- [20] Y. Bai, L. Chou, and W. Zhang, "Industrial innovation characteristics and spatial differentiation of smart grid technology in China based on patent mining," *Journal of Energy Storage*, vol. 43, Article ID 103289, 2021.
- [21] T. Qiu, D. Zhou, J. Wang, and V. Lingamuthu, "Accessibility of rehabilitation facility: evaluation based on spatial big data in xiamen," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–13, Article ID 4008472, 2022.
- [22] X. Zhu, J. Dai, H. Wei, D. Yang, W. Huang, and Z. Yu, "Application of the fuzzy optimal model in the selection of the startup hub," *Discrete Dynamics in Nature and Society*, vol. 2021, pp. 1–9, Article ID 6672178, 2021.
- [23] T. Qiu, D. Zhou, and W. Li, "Fitness culture and green space equity: accessibility evaluation of shanghai communities," *Frontiers in Environmental Science*, vol. 10, 2022.

## Research Article

# Optimized Neural Network for Research Evaluation of Mineral Resources Carrying Capacity in Southern Shaanxi

Li YiCan,<sup>1</sup> Wei JunHao,<sup>1</sup> and Ali Arshaghi <sup>2</sup>

<sup>1</sup>School of Earth Resources, China University of Geosciences, Wuhan, Hubei 430074, China

<sup>2</sup>Department of Electrical Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran

Correspondence should be addressed to Ali Arshaghi; [ali.arshagi.eng@iauctb.ac.ir](mailto:ali.arshagi.eng@iauctb.ac.ir)

Received 17 March 2022; Revised 19 April 2022; Accepted 27 June 2022; Published 11 August 2022

Academic Editor: Araz Darba

Copyright © 2022 Li YiCan 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.

Evaluation of the carrying capacity of mineral resources is one of the important research content in the implementation of sustainable development. Based on analyzing the metallogenic geological characteristics, distribution, and resource status of mineral resources in southern Shaanxi, this paper establishes an analysis model of mineral resources and mineral advantages based on the analytic hierarchy process and applies them to evaluate the advantages of mineral resources. To provide optimal and efficient results, an improved model of an artificial neural network based on the bat optimization algorithm has been utilized. Through model analysis, the potential value and carrying capacity of mineral resources in three major prefecture-level cities in southern Shaanxi are comprehensively evaluated and analyzed. The results show that the main dominant minerals in southern Shaanxi are gold, lead zinc, and molybdenum ore. There are three grades of mineral resources carrying capacity: Shangluo City is an excellent grade, Hanzhong City is a good grade, and Ankang City is a general grade.

## 1. Introduction

Mineral resources are a kind of important material basis for human survival and social development and important pillars of the national economy. The sustainable utilization of mineral resources plays a vital role in the sustainable development strategy and high-quality development of the entire national economy. The mineral resources carrying capacity refers to the carrying capacity of the economically recoverable reserves of mineral resources (or their production capacity) for social and economic development under the prevailing scientific and technological natural environment and social and economic conditions in a foreseeable period. It is an important indicator to measure the sustainable guarantee supply of mineral resources in a country or region and the degree to which it meets the needs of social and economic development and can also embody and reflect the guarantee and carrying capacity of its regional mineral resources to a certain extent.

Southern Shaanxi is rich in mineral resources, characterized by wide distribution, variety, and obvious regional

characteristics. It is a key area of mineral resources in Shaanxi Province and even the whole country. Southern Shaanxi is dominated by nonferrous metals, precious metals, ferrous metals, and various nonmetallic minerals. Regional economic development also mainly comes from the development and utilization of mineral resources. The rational development and utilization of mineral resources is an important foundation for the sustainable development of the regional economy in southern Shaanxi. Due to environmental protection and other reasons, some mining enterprises have stopped production, and the mining economy has experienced a serious decline, affecting regional economic development to a certain extent. The contradiction between the rapid economic and social development in southern Shaanxi and the fragile mineral resources carrying capacity and the contradiction between the extensive growth mode and the shortage of resources are increasingly prominent, and it is urgent to carry out research and evaluation work on the mineral resources carrying capacity [1]. Based on the current situation of resources in southern Shaanxi, this paper puts forward a set of scientific mineral

resources carrying capacity evaluation system and proposal in order to improve the guarantee capacity of mineral resources and scientifically and rationally develop and utilize mineral resources, which will be of extremely great significance to promote the high-quality development of mining economy in southern Shaanxi [2, 3]. To provide optimal and efficient results, an improved model of an artificial neural network based on the bat optimization algorithm has been utilized. The paper has been organized as follows: in Section 2, an overview of the mineral resources is explained in southern Shaanxi. Section 3 describes the analysis of advantageous minerals for mineral resources in southern Shaanxi. In section 4, the artificial neural network and its optimization by the bat algorithm have been explained. Section 5 represents a comprehensive analysis of the potential value and carrying capacity of mineral resources in southern Shaanxi. Finally, the paper has been concluded in Section 6.

## 2. Overview of Mineral Resources in Southern Shaanxi

*2.1. Overview.* Shaanxi Province is located in the inland area, with a land area of about 205,600 square kilometers, about 870 kilometers long from north to south, and 200 to 500 kilometers wide from east to west. The region is long and narrow, with the topographical feature of the north-south high and the middle-low. The mountainous area such as plateaus, mountains, and valleys accounts for about 81% of the total area. From north to south, the area under the jurisdiction is the Loess Plateau in northern Shaanxi, the Weihe Plain in Guanzhong, and the Qinba Mountains in southern Shaanxi. The Qinba Mountains in southern Shaanxi include the Qinling Mountains, Bashan Mountains, and Hanjiang Valley, which constitute a diverse topography of alpine canyons and alluvial plains. The Hanjiang River flows between the Qinling Mountains in the north and the Bashan Mountains in the south, forming a unique topographical feature of “two mountains sandwiching one river.” The Qinling Mountains are about 400–500 kilometers long from east to west and about 300 kilometers wide from north to south in the province. Among them, the altitudes of Zhongnan Mountain, Taibai Mountain, and Huashan Mountain are 2,604 meters, 3,767 meters, and 2,160 meters, respectively; Bashan Mountains are located in the southernmost, with an altitude of 1,500–2,000 meters. The mountainous area of the region accounts for 82.25% of the total area of southern Shaanxi.

Southern Shaanxi is located in the southern part of Shaanxi Province, referring to the area south of the Qinling Mountains, mainly including Hanzhong, Ankang, and Shangluo cities, adjacent to Henan and Hubei in the east; Gansu in the west; Sichuan, Chongqing, and Hubei in the south; and connected to three cities of Baoji, Xi’an, and Weinan in Guanzhong in the north of Shaanxi Province. In 2020, the total population in the administrative region of southern Shaanxi is about 7.75 million, accounting for 19.60% of the total population of Shaanxi Province, and the land area is about 70,000 square kilometers, accounting for

34% of the total land area of Shaanxi Province. The development and utilization of minerals have promoted industrial development and consolidated the regional economic foundation. According to statistics, in 2017, the gross industrial output value of industrial enterprises above the designated size in Shaanxi Province was RMB 2.485443 trillion, an increase of 18.8% over the previous year. The mining and related processing and manufacturing industries completed RMB 1.419827 trillion, accounting for 57.1% of the province’s gross industrial output value above the designated size (according to the Shaanxi Provincial Department of Land and Resources, 2018), of which southern Shaanxi is the main contributor to Shaanxi’s mining economy.

Southern Shaanxi is located in the Qinling Mountains metallogenic belt, with complex geological structure and rich mineral resources [4]. It is an important metallogenic belt in my country; 83 kinds of minerals have been discovered, mainly nonferrous metals, precious metals, ferrous metals, and nonmetallic minerals [5]. Among the 2,044 mineral deposits in the province, 1,071 are located in southern Shaanxi [6–8], accounting for 52.40% of the total number of mineral deposits in the province. Among them, Shangluo City has the largest number of deposits (425), followed by Hanzhong City (344) [6] and Ankang City (302) [7]. It is the most important polymetallic mineral resource enrichment area among the proven resources in the province. It has laid a solid resource foundation for the economic development of southern Shaanxi. Due to the differences in metallogenic geological background and geological conditions, there are obvious differences in the types and quantities of main minerals in different regions. It can be seen from Table 1 that gold, iron, manganese, copper, nickel, lead, zinc, phosphorus, sulfur, serpentine, and other minerals are mainly distributed in the Hanzhong area; gold, iron, antimony, mercury, tungsten, titanium, vanadium, barite, lead, zinc, and other minerals are mainly distributed in Ankang area; and iron, molybdenum, silver, antimony, copper, potassium feldspar, vanadium, titanium, lead-zinc, crystal, and other minerals are mainly distributed in Shangluo area. In recent years, the reserves of gold, copper, lead, zinc, iron, molybdenum, antimony, manganese, and other important minerals in southern Shaanxi have been declining year by year, showing negative growth, and resource crisis mines have continued to emerge. The contradiction between the situation of mineral resources and the need for economic and social development is becoming increasingly prominent.

Figure 1 shows the distribution figure of major mineral resources in southern Shaanxi.

Table 2 indicates the list of weights of second-level indicators.

*2.2. Analysis of Metallogenic Characteristics.* Southern Shaanxi is located in the Qinling Mountains and the Daba Mountains. It is a very important polymetallic metallogenic belt in my country, and the mine resources are very rich. The three cities in southern Shaanxi span the Xiaoqinling Mountains, North Qinling Mountains, South Qinling

TABLE 1: Distribution table of main mineral resources in southern Shaanxi.

Serial number	Area	Mainly distributed minerals
1	Hanzhong	Gold, iron, manganese, copper, nickel, lead, zinc, phosphorus, sulfur, serpentine, etc.
2	Ankang	Gold, iron, antimony, mercury, tungsten, titanium, vanadium, barite, lead-zinc, etc.
3	Shangluo	Iron, molybdenum, silver, antimony, copper, potassium feldspar, vanadium, titanium, lead-zinc, crystal, etc.

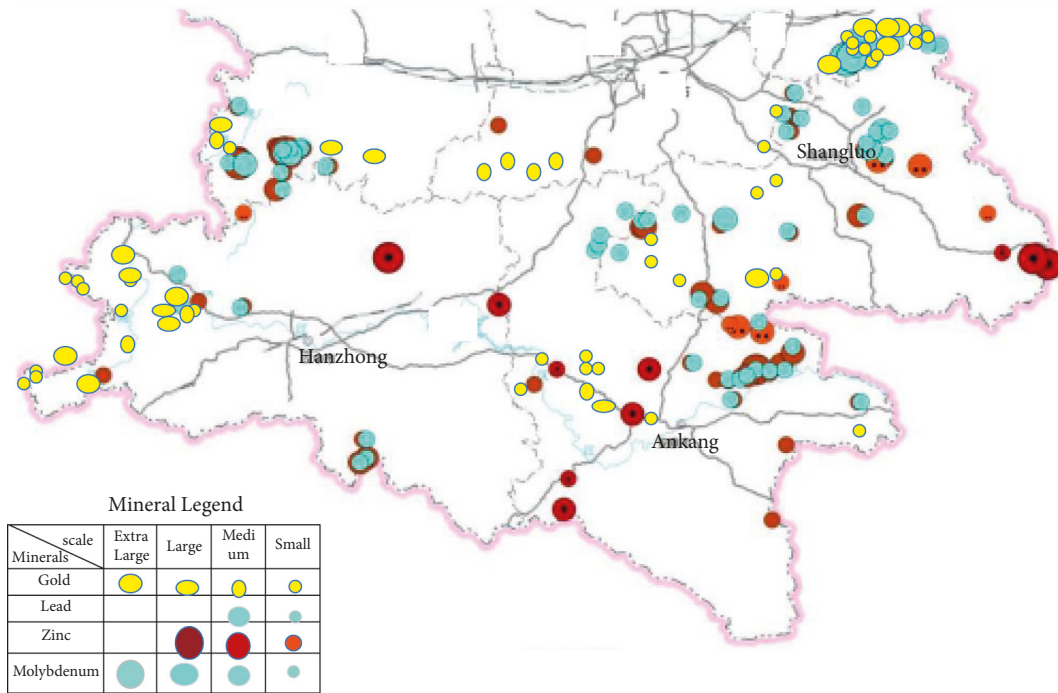


FIGURE 1: Distribution figure of major mineral resources in southern Shaanxi.

TABLE 2: List of weights of second-level indicators.

Serial number	Second-level indicator B	Weight
1	Mineral resources themselves B1	0.1796
2	Mineral resources survey and exploration B2	0.2496
3	Development and utilization of mineral resources B3	0.2364
4	Socioeconomic benefits B4	0.3343

Mountains, Motianling, and the edge of the Yangtze platform [9–11]. They have excellent metallogenic geological conditions and are rich in nonferrous metals, precious metals, ferrous metals, and various nonmetallic minerals [12–14]. The distribution of mineral resources in the three cities in southern Shaanxi is slightly different: Hanzhong is rich in iron, manganese, titanium, nickel, metallurgical quartz stone, solvent limestone, and other minerals; Ankang is rich in mercury, pyrite, barite, and other minerals resources; and Shangluo is rich in iron, titanium, lead, zinc, gold, and other mineral resources. Figure 2 shows the hierarchical structure model of comprehensive evaluation of advantageous mineral resources in southern Shaanxi.

2.2.1. *Gold Ore.* As the dominant mineral in the Qinling Mountains metallogenic belt, the output of gold ore is mainly controlled by the tectonic-magmatic belt. There are

two regional gold metallogenic belts in the north and south. The gold ores in the belt follow the distribution of regional large faults and own the characteristics of close magmatism in the Indo-Chinese-Yanshan period. The dominant types of gold ores are mainly structural altered rock type and quartz vein type, which are concentrated in the West Qinling Mountains and Xiaoqinling Mountains areas.

The gold ores in Shaanxi Province are mainly distributed in the areas under the jurisdiction of cities such as Xi’an, Baoji, Weinan, Hanzhong, Ankang, and Shangluo. Among them, the areas under the jurisdiction of Hanzhong City and Shangluo City have a relatively large number of mineral fields, accounting for 21.28% and 31.38% of the total gold mining fields in the province, respectively, while the reserves of gold mineral resources are mainly concentrated in the areas under the jurisdiction of Baoji City and Shangluo City, accounting for 23.77% and 32.86% of the province’s total identified gold reserves, respectively.

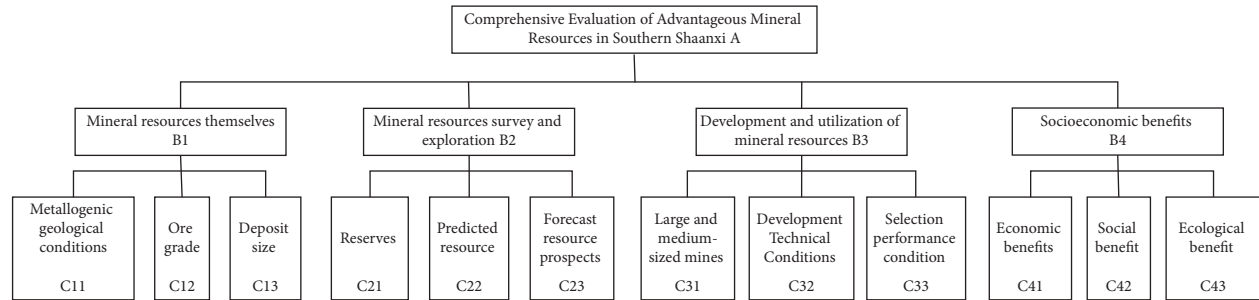


FIGURE 2: The hierarchical structure model of comprehensive evaluation of advantageous mineral resources in southern Shaanxi.

**2.2.2. Lead-Zinc Ore.** The large-scale mineralization of lead-zinc ore in the West Qinling Mountains area generally includes two stages: Devonian tectonic rifting-contemporaneous depositional mineralization and Triassic tectonic-magmatic superimposed mineralization.

The geographical location of lead-zinc ore in Shaanxi Province is mainly distributed in the areas under the jurisdiction of Baoji, Ankang City, and Shangluo City in southern Shaanxi. Lead-zinc resource reserves account for 86.93% of the province's total lead-zinc resource reserves. Among them, Baoji City's lead and zinc resource reserves account for 26.44% and 51.58% of the province's total lead and zinc resource reserves, respectively, Shangluo City's lead resource reserves account for 47.74% of the province's total lead resource reserves, and Ankang City's zinc resource reserves account for 23.58% of the province's total zinc resource reserves. The area of Fengxian-Taibaimiao in Baoji City, Shanyang-Zhashui and Zhen'an in Shangluo City, and the northern part of Xunyang in Ankang City are relatively concentrated areas of lead-zinc ore in Shaanxi Province.

**2.2.3. Iron Ore.** Iron ores are distributed in all prefecture-level cities in Shaanxi Province. However, iron ores with industrial significance are mainly distributed in the areas under the jurisdiction of Hanzhong City, Ankang City, and Shangluo City in southern Shaanxi [15]. Its mineral deposits account for 74.83% of the province's total iron deposits, and its iron ore resource reserves account for 96.55% of the province's total iron ore resource reserves. Among them, the Mian-Lue-Yang area in Hanzhong City and the Bijigou area in Yangxian County, the Ziyang-Zhenping area in Ankang City, and the Zhashui-Shanyang area in Shangluo City are areas with relatively more iron deposits. The iron resources reserves are mainly distributed into three types of deposits: magmatic segregation type, bed-controlled hydrothermal type, and volcanic-sedimentary metamorphic type.

**2.2.4. Copper Ore.** Copper ores are distributed in 6 prefecture-level cities in the southern Shaanxi and Guanzhong regions. Shangluo City has the largest number of copper mines, accounting for 52.17% of the province's total copper mines and 28.86% of the province's total copper resource reserves, followed by Hanzhong, accounting for 20.87% of the province's total copper mines, accounting for 20.82% of the province's total copper resource reserves. Weinan City

has a small number of ore deposits, but its copper resource reserves account for 25.03% of the province's total copper resource reserves. The Mian-Lue-Yang area of Hanzhong City, the Zhashui-Shanyang area of Shangluo City, and the Shangzhou-Danfeng area are the areas with relatively more copper deposits. The identified resources and reserves of copper deposits are mainly distributed in two types of deposits: porphyry type and tectonic hydrothermal type.

**2.2.5. Molybdenum-Tungsten Ore.** Molybdenum and tungsten ores are distributed in the northern margin of the Qinling Mountains metallogenic belt, and the main type is porphyry. In recent years, large-scale molybdenum deposits have been discovered in the West Qinling Mountains, and breakthroughs have been made in the prospecting of molybdenum ore. Wenquan, Jiangligou, and other porphyry molybdenum-tungsten (copper) ores and a series of prospecting clues have been discovered and several molybdenum deposits (points) such as Yueheping, Xigou and Guilin'gou in the Zhen'an-Ning-Shaan junction area of the South Qinling Mountains. Recently, good molybdenum mineralization has been discovered in the gold deposits of the Xiaoqinling Mountains gold ore concentration area, and the molybdenum resources of individual deposits can reach a medium scale. In the west of the molybdenum deposits in Jinduicheng, the molybdenum (lead) deposits accompanying silver and lead in Xigou, Huaxian County, have been discovered in the latest exploration, which further shows that the Qinling metallogenic belt has huge potential for molybdenum prospecting.

**2.2.6. Antimony Ore.** Mercury ore and antimony ore are mostly symbiotic, mainly in the epithermal medium-low temperature hydrothermal type. The distribution of deposits is controlled by stratigraphy and structure, and they are mainly distributed in the southern South Qinling Mountains. The Triassic rift basin that traverses the east and west of the Qinling Mountains metallogenic belt is an important mercury-antimony-gold polymetallic metallogenic belt in Qinling Mountains, in which the argillaceous fine clastic turbidite sedimentary rock series is an important ore-hosting rock series [16–18]. At present, dozens of large and medium-sized mercury-antimony deposits such as Yawan, Ganzhai, Gongguan, and Qingtonggou and numerous mineral fields have been discovered, which constitute a

veritable “golden belt” of the Qinling Mountains, and the prospecting practice in recent years has proved that its prospecting potential is huge.

### 3. Analysis of Advantageous Minerals of Mineral Resources in Southern Shaanxi

Southern Shaanxi is rich in mineral resources and has many kinds of minerals. This paper selects advantageous minerals and strategic minerals for follow-up evaluation. This paper collects and sorts out the information of related projects such as the utilization status of mineral resources in southern Shaanxi and the evaluation of the potential of mineral resources in Shaanxi Province and finally selects four advantageous minerals in southern Shaanxi, gold, lead, zinc, and molybdenum, for analysis.

This paper adopts an optimized version of an artificial neural network for this process. The optimized neural network is a network with optimal weights selection. In the following, the structure of the optimized neural network has been described. Table 3 shows the analysis list of the advantages of four main mineral resources in southern Shaanxi.

### 4. Optimized Neural Network Based on Bat Optimization Algorithm

Optimization is a topic in mathematics, computer science, and operations research in which the best element (according to a set of criteria) is selected from a set of options. Optimization has significant applications in “artificial intelligence”. Artificial neural networks (ANNs) are computational systems that are inspired by but not necessarily related to biological neural networks. Today, a variety of neural network algorithms are widely used to solve “machine learning” problems [19]. Indeed, the use of neural networks in various issues is of interest to researchers.

Today, perceptron neural networks are still of particular importance and are a fast and reliable solution to classification problems. In many complex mathematical problems that lead to the solution of complex nonlinear equations, a multilayer perceptron network can be used simply by defining appropriate weights and functions [20].

In this type of network, an input layer is used to apply the problem inputs to a hidden layer and an output layer ultimately provides the solutions to the problem. The model used in the simulation (input  $X_i$  is multiplied by its weight and enters the second layer (hidden layer) and then the sum of the coefficients of all  $X_i$  is calculated and a certain value is obtained) and again enters the layer as input. It is then, and this process continues until it reaches the output layer, and in the output layer, the output function is obtained by applying the output function to it.

**4.1. Multilayer Perceptron Neural Networks.** An artificial neural network (ANN) is an idea for information processing that is inspired by the biological neural system and processes information like the brain. The key element of this idea is the

new structure of the information processing system. This system is made up of a large number of highly interconnected processing elements called neurons that work together to solve a problem. In traditional computational methods, a series of logical expressions are used to operate. In contrast, neural networks use a set of nodes (as neurons) and edges (as synapses) to process data. In this system, inputs flow into the network and a series of outputs are generated.

A popular method to better minimize the error between the network output and the experimental data is to utilize the backpropagation (BP) approach. Based on the BP method, the idea is to the optimal selection of the weights of the neurons to minimize this error and to provide an efficient confirmation with the experimental data. Error minimization of the BP method is based on the gradient descent (GD) algorithm. A drawback of the GD method is that sometimes stuck in the local minimum point affects the network's success.

The network output for nodes can be achieved based on two stages. First, the input-weighted summation in the input has been obtained as follows:

$$z_i = \sum_{j=1}^n w_{ij} \alpha_j + \beta_i, \quad (1)$$

where  $\alpha_j$  specifies the input data,  $\beta_i$  signifies the bias of neuron number  $i$ , and  $w_{ij}$  describes the connected neurons between  $\alpha_j$  and the  $i^{\text{th}}$  hidden neuron.

The activation function is used to activate the neuron output. There are several activation functions for MLP networks. This study uses the sigmoid function for this purpose. The sigmoid function is achieved based on the following equation:

$$f_j(x) = \frac{1}{1 + e^{-y_j}}, \quad (2)$$

where  $f_j$  is the activation function and  $y$  is the network output.

At last, the network output is achieved as follows:

$$y_i = \sum_{k=1}^m w_{ki} \alpha_k + \beta_i. \quad (3)$$

**4.2. Bat Optimization Algorithm.** Today, there is a great tendency to use algorithms inspired by nature and based on the swarm intelligence of animals. Especially in the category of optimization techniques, these algorithms have been very popular [21]. With a close look at the nature and the great variety of animals that collectively follow certain methods in their lives, it is obvious that many algorithms can be inspired by them. For example, the genetic algorithm was first introduced by John Holland. This algorithm uses Darwin's principles of natural selection to find the optimal formula for predicting or matching the pattern.

Also, the particle swarm algorithm is based on the swarm behavior of birds and was proposed in 1975 by Kennedy and Eberhart. There are many algorithms in this field such as the

TABLE 3: Analysis list of the advantages of four main mineral resources in southern Shaanxi.

Indicators and weight minerals	B1 0.1796				B2 0.2496				B3 0.2364				B4 0.3343				Comprehensive weight	Ranking
	C11	C12	C13	C21	C22	C23	C31	C32	C33	C41	C42	C43	C44	C45				
Gold ore	0.2899	0.2010	0.1899	0.3211	0.2522	0.2011	0.2310	0.2115	0.2610	0.3511	0.2410	0.2360	0.2489	2				
Zinc ore	0.3989	0.4652	0.3656	0.4262	0.3856	0.3406	0.3232	0.3108	0.4121	0.4477	0.3862	0.4772	0.3949	1				
Lead ore	0.2010	0.2502	0.1501	0.3211	0.1542	0.1262	0.1846	0.2577	0.2017	0.2974	0.2120	0.2501	0.2171	3				
Molybdenum ore	0.1704	0.1356	0.1420	0.1761	0.1523	0.1785	0.1902	0.1403	0.1725	0.0985	0.0510	0.0621	0.1391	4				



world cup optimization algorithm, cuckoo algorithm, and firefly algorithm. It should be noted that one of the most important discoveries of modern science that is the focus of many scholars and scientists today and has led to the solution of many unsolved problems is the subject of artificial neural networks that are inspired by human neural nature. Based on the cases and examples mentioned, we conclude that it is being added to nature-inspired algorithms day by day.

The bat algorithm is a metaheuristic algorithm inspired by the swarm behavior of bats in the wild, introduced in 2010 by Yang. This algorithm is based on the use of sound reflection by bats. Bats find the exact path and location of their prey by sending sound waves and receiving reflections. When the sound waves return to the transmitter (bat), the bat can draw an acoustic image of the obstacles in front of its surroundings and see the surroundings well even in complete darkness. Using this system, bats can detect moving objects such as insects and immobile objects such as trees. The bat algorithm is based on the echo detection feature of microbats. In general, there are two types of bats, the first type is large bats, and the second type is called small bats. Microbats use this feature for night flying and hunting. Echo detection of microbats is in practice a perceptual system in which ultrasonic waves are generated to obtain echoes. The bat's brain and nervous system can create an image of its surroundings and details by comparing transmitted and reflected waves.

This ability allows microbats to detect their prey in complete darkness. The intensity of the wave produced by the bat is 130 decibels, and it uses the frequency of 15 kHz to 200 kHz for hunting prey. This is while the human hearing range is from 20 Hz to 20 kHz. In order to identify the data, the bat must be able to separate the sound produced by it from its echo. Microbats have two methods for this purpose. Echo detection with short time cycles: these bats can detect the sound sent by themselves from the reflected sound with the help of timing. Echo detection with long time cycles: these bats produce a continuous sound and separate the pulses and echoes by changing the frequency. They can change the pulse of any output frequency depending on the flight speed. In this way, the received echo is still in the appropriate hearing range.

The following three ideal rules are used to develop this algorithm:

- (i) Using bounce detection, all bats can estimate the distance and distinguish between prey and fixed obstacles.
- (ii) Bats are randomly searching for prey at speed  $v_i$  in position  $x_i$  with constant frequency  $f_{min}$  with variable wavelength  $\lambda$  and loudness  $A_0$ . They can automatically adjust the wavelength of their emitted pulses and adjust their pulse emission rate,  $r \in (0, 1)$ , according to the proximity of their prey.
- (iii) Although the volume can be changed in different ways, it is assumed that the volume changes from a large (positive) value of  $A_0$  to a constant minimum value,  $A_{min}$ .

- (iv) We can also use this approximation, which is generally the frequency  $f$  in a range  $[f_{min}, f_{Max}]$ , which corresponds to a wavelength spectrum as  $[\lambda_{min}, \lambda_{Max}]$ .

Each bat depends on the speed  $v_i^t$  and the location  $x_i^t$  by repeating  $t$  in the next  $d$ -dimensional search or the solution space.

Among all bats, there is only one optimal solution,  $x^*$ , so the three rules mentioned in the previous section can be calculated using the following formulas:

$$\begin{aligned} F_i &= f_{min} + (f_{Max} - f_{min}) \times \beta, \\ V_i^t &= v_i^{t-1} + (x_i^{t-1} - x^*) f_i, \\ X_i^t &= x_i^{t-1} + v_i^t, \end{aligned} \quad (4)$$

where  $\beta$  describes a random value between 0 and 1 which is generated by a uniform random distribution.

As mentioned, wavelength or frequency can also be used for implementation. Here,  $f_{min} = 0$  and  $f_{Max} = 1$ , which is interesting depending on the size of the problem. Each bat is first assigned a random frequency obtained uniformly from  $[f_{min}, f_{Max}]$ . For this reason, it can be said that the bat algorithm is a frequency scale algorithm so that it provides a balanced combination of exploration and exploitation. Pulse amplitude and rate provide an automatic mechanism for automatic control and magnification of the area leading to prescribed solutions. With these consumptions, we have the following:

$$\begin{aligned} A_i^{t+1} &= \alpha A_i^t, \\ r_i^{t+1} &= r_i^0 \times [-\exp(-\gamma t)], \end{aligned} \quad (5)$$

where  $\alpha$  and  $\gamma$  represent two constant variables.

The function  $\alpha$  is similar to the function of the temperature reduction factor in the simulated annealing algorithm. For  $0 < \alpha < 1$ ,  $\gamma > 0$ , when  $t \rightarrow \infty$ ,

$$A_i^{t+1} \rightarrow 0, r_i^{t+1} \rightarrow r_i^0. \quad (6)$$

In the simplest case,  $\gamma = \alpha$  can be considered. The simulations performed in this algorithm are considered:  $\gamma = \alpha = 0.98$ .

In this study, the bat optimization algorithm has been used to minimize mean square error (MSE) between  $y_i$  and  $z_i$ .

The equation of the MSE is formulated as follows:

$$\text{MSE} = \frac{1}{n} \sum_{j=1}^n \sum_{i=1}^m (y_i(k) - z_i(k))^2. \quad (7)$$

Therefore, the pseudo-code of the present study can be considered as follows:

- (1) Initializing the bat algorithm
- (2) Evaluate error value of the network
- (3) Update the algorithm to provide proper value for network parameters
- (4) Check the criteria condition

TABLE 4: Potential value table of retained reserves of three major minerals in three prefecture-level cities in southern Shaanxi in 2020.

Area	Total land area (KM <sup>2</sup> )	Total population (10,000 people)	Potential value (RMB hundred million)	Ranking	Unit area potential value (RMB ten thousand)	Ranking	Potential value per capita (RMB)	Ranking
Hanzhong city	27200	321	115.53	2	42.47	2	3599.06	2
Ankang city	23529	249	64.45	3	27.37	3	2588.35	3
Shangluo city	19292	205	339.87	1	176.17	1	16579.02	1

(5) If the criteria condition is not achieved, go to (3)

(6) If the criteria condition is reached, go to (7)

(7) End

## 5. Comprehensive Analysis of Potential Value and Carrying Capacity of Mineral Resources in Southern Shaanxi

This paper mainly evaluates the potential value of four kinds of advantageous mineral resources in three prefecture-level cities in southern Shaanxi. The potential value of mineral resources is essentially the value of a proven available mineral resource converted from the price of its primary products, reflecting and measuring the strength and potential of mineral resources in a study area from a macro level. The potential value of mineral resources is the value of a proven available resource converted from the price of its primary product. This indicator does not deduct the mining and selection loss of mineral resources and the element cost of exploration and mining and is used to reflect the strength of mineral resources in a specific area from a macro level. The evaluation of the potential value of mineral resources is a general assessment of the potential of mineral resources and the possible economic benefits of future development. It can not only reflect the abundance of mineral resources but also reflect the economic scale of future mineral development and provide a basis for analyzing the contribution of mineral exploitation to the regional economy, and it is also an important basis for resource economic zoning.

### 5.1. Analysis Method

(i) Begin the neural network training

(ii) Determine the network inputs including the following:

(iii) Mineral Reserves (R): the current situation of mineral resources utilization in southern Shaanxi and local mineral records are used as the data for this calculation. By the end of 2017, the retained reserves of four major mineral resources in three prefecture-level cities in southern Shaanxi are used.

(iv) The Price of Mineral Products (P): the price of mineral products selects the latest market price of the latest raw mineral products as the calculation parameter.

(v) Grade Adjustment Coefficient (G): the grade adjustment coefficient refers to the conversion coefficient between unit mineral reserves and unit mineral products, which is obtained by dividing the average grade of mineral reserves by the mineral product grade per unit price.

The conversion formula is as follows: grade adjustment coefficient = average grade of mineral reserves ÷ mineral product grade per unit price.

5.2. *Comprehensive Analysis and Evaluation of Potential Value and Economic Carrying Capacity of Major Mineral Resources in Southern Shaanxi.* Mining plays a pivotal role in the development of the national economy in Shaanxi Province and southern Shaanxi. The evaluation of the potential value of mineral resources and economic carrying capacity can embody and reflect the guarantee capacity of regional mineral resources and the contribution capacity of the mineral economy to a certain extent. The capacity of mineral resources to pay for the future economic development speed is to determine the economic development speed under the condition of available resources in the future by measuring the demand for resources of the economic development speed. The carrying capacity of mineral resources for social and economic development is the degree of support to social material production and people's life.

From Table 4 and Figures 3 and 4, it can be seen that by the end of 2020, the total potential value of retained resource reserves of the four major mineral resources in the three prefecture-level cities in southern Shaanxi was as high as RMB 51.985 billion.

The potential value, potential value per unit area, and potential value per capita of the retained reserves of the four major mineral resources in the three prefecture-level cities in southern Shaanxi are in the order of Shangluo, Hanzhong, and Ankang. Among them, the most potential value of Shangluo City is as high as RMB 33.987 billion, and the potential value per capita is RMB 16,500, ranking first in southern Shaanxi. This is mainly due to Shangluo City's super-large gold deposits, considerable reserves of lead-zinc ore, and molybdenum ore resource. Hanzhong City ranks second, with a potential value of RMB 11.553 billion and RMB 3,599.06 per capita, thanks to the reserves of lead-zinc ore resources in Hanzhong City. Figure 5 shows the evaluation figure of the carrying capacity of mineral resource in southern Shaanxi.

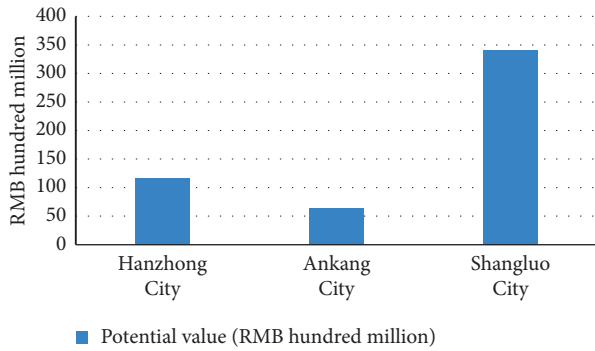


FIGURE 3: Histogram of potential value of mineral resources in three prefecture-level cities in southern Shaanxi.

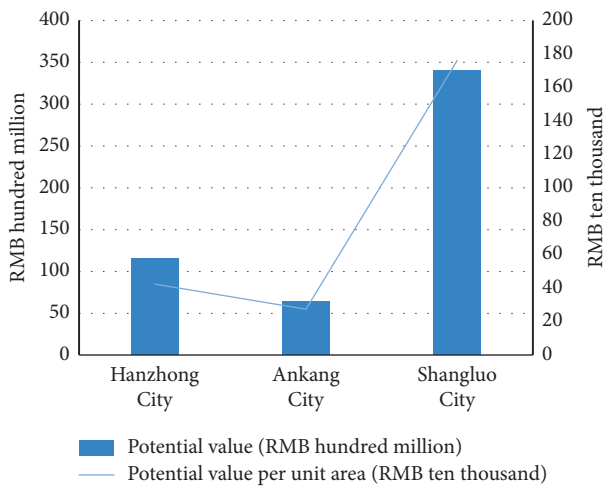


FIGURE 4: Comparison figure of the potential value of mineral resources and the potential value per unit area of three prefecture-level cities in southern Shaanxi.



Mineral resources carrying capacity level

- Superior grade
- Good grade
- General grade

FIGURE 5: Evaluation figure of resources carrying capacity of mineral resource in southern Shaanxi.

Based on the analysis of the mineral advantages of the four main minerals (gold, lead, zinc, and molybdenum ore) in southern Shaanxi and the analysis of the potential value carrying capacity of mineral resources in the three prefecture-level cities, it is concluded that the grades of mineral resources carrying capacity of the three prefecture-level cities in southern Shaanxi are divided into three grades, namely: Shangluo City is an excellent grade, Hanzhong City is a good grade, and Ankang City is a general grade.

### 6. Conclusion and Analysis

Based on the current situation of development and utilization of advantageous mineral resources in southern Shaanxi, this paper establishes a hierarchical structure model for comprehensive evaluation of advantageous mineral resources in southern Shaanxi and draws the following conclusions:

- (1) In general, the mineral resources in southern Shaanxi have an advantageous position in the follow-up carrying capacity of the national economy. Through the analysis of the distribution of minerals and the status quo of resources in this region, an analysis and evaluation model for mineral advantages based on the analytic hierarchy process is established, apply it to the research area, and the evaluation results are good, which is more in line with the actual situation.
- (2) Based on the evaluation and analysis of the advantageous mineral resources, the evaluation model equation of the potential value of mineral resources in southern Shaanxi is established, and the potential value of mineral resources in the three prefecture-level cities in the study area is calculated and analyzed. The mineral resources carrying capacity in southern Shaanxi is comprehensively evaluated and graded based on this. Among the three cities in southern Shaanxi, Shangluo City has the highest mineral resources carrying capacity, which is excellent, Hanzhong City has a good carrying capacity, and Ankang City has a general carrying capacity.
- (3) The development of the southern Shaanxi focuses on the coordinated development of resources, environment, and social economy, and one cannot ignore the other. It is unfeasible to only pursue economic development and ignore ecological and environmental protection, and it is unrealistic to simply protect resources and the environment without developing the economy. Therefore, seeking a reasonable development model and maximizing resource and environmental benefits are the solutions to the sustainable development of resources.

### Data Availability

The data used to support the findings of this study are included in the following: Shaanxi Province Yearbook 2010–2017, Shaanxi Local Chronicle 2010–2017, and Internal data from my company (the internal data of the unit

are confidential); they are accessible from the following link: <https://kdocs.cn/l/cruvypHxwghM>.

## Conflicts of Interest

The authors declare no conflicts of interest.

## References

- [1] J. Zhao, S. Chen, and R. Zuo, "Identifying geochemical anomalies associated with Au–Cu mineralization using multifractal and artificial neural network models in the Ningqiang district, Shaanxi, China," *Journal of Geochemical Exploration*, vol. 164, no. 4, pp. 54–64, 2016.
- [2] W. Dong, X. Li, and W. Li, *China Mineral Geology. Shaanxi Volume [R]*, pp. 30–89, Provincial Institute of Geological Survey, Xi'an City Shaanxi, 2020.
- [3] L. Guo, C. Yan, and L. Jing, *Report on Mineral Prediction Results of Important Minerals in Shaanxi Province [R]*, pp. 15–63, Provincial Institute of Geological Survey, Xi'an City Shaanxi, 2013.
- [4] W. Ning, B. Kuan-jiang, and Y. Zhi-fa, "Analysis and forecast of Shaanxi GDP based on the ARIMA Model," *Asian Agricultural Research*, vol. 2, pp. 34–41, 2010.
- [5] X. Yuan, W. Dong, and X. Li, *Research Report on Metallogenic Regularity of Important Mineral Species in Shaanxi Province [R]*, pp. 26–98, Provincial Institute of Geological Survey, Xi'an City Shaanxi, 2013.
- [6] X. Liu, T. Tan, Y. Bai, and L. C. Chou, "Restoration performance of regional soil and water resources in China based on index of coupling and improved assessment tool," *Alexandria Engineering Journal*, vol. 61, no. 7, pp. 5677–5686, 2022.
- [7] W. Cui, Q. Meng, W. Li, and Q. Feng, "Enrichment and Chemical Speciation of Vanadium and Cobalt in Stone Coal Combustion Products in Ankang, Shanxi Province, China," *International Journal of Environmental Research and Public Health*, vol. 19, no. 15, p. 9161, 2022.
- [8] X. Yu, Y. Liu, Z. Zhang, Y. Xiong, and M. Dang, "Urban spatial structure features in Qinling mountain area based on ecological network analysis—case study of Shangluo City," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 12829–12845, 2022.
- [9] Y. Hou, "Analysis of the carrying capacity of mineral resources in domestic primary energy [J]," *Modern Business Trade Industry*, vol. 10, no. 22, pp. 2–4, 2014.
- [10] B. Liu, J. Wang, and Ke Guo, "Guarantee degree of mineral resources based on geological survey evaluation—taking copper mines in central tibet as an example [J]," *Metal Mine*, vol. 35, no. 08, pp. 87–90, 2014.
- [11] X. Li, W. Jia, and X. Meng, *Research Report on the Transformation Strategy of Mineral Resources in Shandong Province*, Institute of Geological Sciences, China Academy of Land & Resources Economics, Jinan, Shandong, 2010.
- [12] Y. Yan and J. Cheng, "Evaluation of mineral resources carrying capacity in key mining economic zones [J]," *Science and Technology Management of Land and Resources*, vol. 31, no. 04, pp. 29–33, 2014.
- [13] Y. Wang and S. Bu, "Research on the economic carrying capacity of China's mineral resources [J]," *Coal Economic Research*, vol. 15, no. 12, pp. 15–18, 1998.
- [14] Q. Xu, "Discussion on several issues in the analysis of regional mineral resources carrying capacity [J]," *Journal of Natural Resources*, vol. 11, no. 02, pp. 135–141, 1996.
- [15] Y. Duan, *Research on the Carrying Capacity of Mineral Resources in Chenzhou City Based on Sustainable Development [D]*, Hunan Agricultural University, Changsha, China, 2011.
- [16] K. Wang, Na Li, and X. Yu, "Ecological environment carrying capacity of shandong peninsula [J]," *Science & Technology Progress and Policy*, vol. 21, no. 5, pp. 56–58, 2004.
- [17] Y. Liu, "Evaluation of mineral resources carrying capacity and analysis of environmental constraints—taking fujian province coal resources as an example [J]," *Journal of Minjiang University*, vol. 33, no. 3, pp. 43–48, 2012.
- [18] D. Xu, L. Qu, and J. Zhang, "Analysis of mineral resources carrying capacity in guizhou province [J]," *Science & Technology Progress and Policy*, vol. 21, no. 5, pp. 56–58, 2004.
- [19] Z. Guo, L. Xu, Y. Si, and N. Razmjoo, "Novel computer-aided lung cancer detection based on convolutional neural network-based and feature-based classifiers using metaheuristics," *International Journal of Imaging Systems and Technology*, vol. 31, no. 4, pp. 1954–1969, 2021.
- [20] D. Yu, Y. Wang, H. Liu, K. Jermsittiparsert, and N. Razmjoo, "System identification of PEM fuel cells using an improved Elman neural network and a new hybrid optimization algorithm," *Energy Reports*, vol. 5, pp. 1365–1374, 2019.
- [21] M. Ramezani, D. Bahmanyar, and N. Razmjoo, "A new improved model of marine predator algorithm for optimization problems," *Arabian Journal for Science and Engineering*, vol. 46, no. 9, pp. 8803–8826, 2021.

## Research Article

# Recognizing the Damaged Surface Parts of Cars in the Real Scene Using a Deep Learning Framework

**Mahboub Parhizkar**  and **Majid Amirfakhrian** 

*Department of Mathematics, Central Tehran Branch, Islamic Azad University, Tehran, Iran*

Correspondence should be addressed to Majid Amirfakhrian; amirfakhrian@gmail.com

Received 30 May 2022; Revised 10 July 2022; Accepted 18 July 2022; Published 10 August 2022

Academic Editor: Araz Darba

Copyright © 2022 Mahboub Parhizkar and Majid Amirfakhrian. 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.

Automatically recognizing the damaged surface parts of cars can noticeably diminish the cost of processing premium assertion that leads to providing contentment for vehicle users. This recognition task can be conducted using some machine learning (ML) strategies. Deep learning (DL) models as subsets of ML have indicated remarkable potential in object detection and recognition tasks. In this study, an automated recognition of the damaged surface parts of cars in the real scene is suggested that is based on a two-path convolutional neural network (CNN). Our strategy utilizes a ResNet-50 at the beginning of each route to explore low-level features efficiently. Moreover, we proposed new mReLU and inception blocks in each route that are responsible for extracting high-level visual features. The experimental results proved the suggested model obtained high performance in comparison to some state-of-the-art frameworks.

## 1. Introduction

The evaluation of car parts is a challenging task that mainly originates from the insurance industry. The issue of automated assessment of damaged parts of a car represents the foremost challenge in the damage assessment and auto repair industry [1, 2]. This field of study has a number of application areas ranging from accidental damage evaluation for car insurance companies to car evaluation companies such as body shops and car rentals [3]. In evaluating a vehicle, the damaged parts can take any form containing missing parts, minor and major dents, and scratches. Generally, the evaluation region has an important level of noise such as oil, grease, or dirt that makes an inaccurate recognition challenging [1, 4]. Also, the recognition of some specific parts is the first stage in the repair industry for having an accurate labor and parts evaluation where the presence of dissimilar car sizes, shapes, and models makes the task even more difficult for a model based on a machine learning strategy to perform well [5].

Nowadays, many machine learning algorithms have been broadly used in many industries to bring down the

charge of manual endeavors including object recognition [6, 7], exterior car body damage-detection [8], image encoding [9], and healthcare (organ, skeletal, body pose, tumor/cancer segmentation) [10, 11].

Detecting the damaged parts of the outer surface in various kinds of cars has received great attention in the field of machine vision in recent years, and many frameworks have been proposed for reducing the claim leak issue [2, 12, 13].

However, applying frameworks using ML strategies in this field is a very challenging task. This challenging task is because of some issues including light reflection, the presence of unidentified objects surrounding vehicles, scene illumination, and background detection [14, 15].

There are two main ML strategies for recognizing damaged outer parts of a car including hand-crafted feature extracting methods and deep learning models [8, 16].

Amirfakhrian et al. [3] proposed a clustering approach based on the fuzzy similarity criteria and changing the color space for recognizing the damaged parts of a car. They used a similarity score among two images that is computed using the color spectrum. Parhizkar et al. [8] suggested a cascade

convolutional neural network to recognize the damaged parts even in the presence of high illumination variation. Moreover, the Kirsch compass kernels are used to produce some edge maps for creating an encoded image. They used two textural descriptor approaches namely local binary pattern (LBP) and local directional number pattern (LDN) to obtain more informative features from the original image. Shirode et al. [17] suggested a deep learning model to recognize the damaged parts of a car using two separate CNN model. The first model (VGG16) is utilized to identify the damaged parts, locations, and their severities. The second model is able to mask out the precise damaged regions.

Our strategy to recognize the car damaged parts is based on learning a convolutional neural network (CNN) architecture utilizing the collected dataset, which consists of 3,000 images with different dimensions captured from different cameras.

The proposed CNN classifies all pixels inside the image into the background, damaged parts, and normal parts. All pixels including normal and damaged parts of the car have their own classes. For instance, we have two classes for normal Windshield and damaged Windshield. So, overall classes include 20 classes for car parts and one class for the background. All car parts have 10 categories including windshield, hood, front bumper, rear bumper, fenders, trunk lid, front doors, back doors, roof, and quarters.

To solve the problems of two-stage models (car detection and recognizing the damaged parts), we suggest a two-route CNN model for exploring both global and local features. The main contributions of this study are listed as follows:

- (1) A new two-route CNN model that automatically finds and localizes damaged parts of the car inside an input image
- (2) Employ a transfer learning approach to find more informative details from a real-scene image
- (3) Applying local and global patches to the CNN model for increasing the final performance of the model

## 2. Materials and Methods

In this part, our datasets and a detailed description of the model architecture are described. The suggested model is shown in detail in Figure 1.

*2.1. Proposed Convolutional Neural Network.* Convolutional neural network architectures are broadly utilized in the field of computer vision, such as medical image analysis, object detection, and action detection. In these networks, features and patterns inside the image can be explored by convolution operation [18–20]. The lower convolutional layers (Conv layers) can extract some features such as curves, lines, and edges. The deeper convolutional layers are able to learn more complex hidden patterns inside the image [21, 22].

The convolution operation in a Conv layer is implemented using a convolution filter (kernel), and its parameters are learned during the learning process. During the

convolving procedure, each filter is convolved with the input image to compute an informative feature map [23]. It should be mentioned that the dimension of the convolution filter is always smaller than the dimension of the input image. In another word, a convolution filter slides over an input image and calculates the dot product between the convolution filter and the input at each spatial position [1, 13, 24].

In this study, we employed two ResNet-50 models to explore low-level features, and three  $3 \times 3$  new.mReLU blocks in each route are responsible for extracting high-level visual features. A residual neural network (ResNet) stacks residual blocks sequentially is an artificial neural network (ANN). ResNet-50 is 50 layers deep including 48 Conv layers, one maximum pooling, and one average pooling layer [25]. In order to recognize each part of the car more precisely, we need a network that is able to detect more informative features. So, we employed a pretrained ResNet-50 architecture that trained on ImageNet at the beginning of the framework [26, 27].

The extracted features of the ResNet-50 model are used as the input of the  $3 \times 3$  mReLU block. In order to achieve a better segmentation result, three  $3 \times 3$  mReLU blocks are utilized sequentially.

As the performance of the ResNet-50 has been proved in the field of image classification (extracting high-quality features of images on ImageNet), we employed this model to explore some informative details about the damaged car and background. Moreover, by applying deeper layers (more feature extraction layers), a better result in detecting damaged parts of a car will be obtained.

The employed  $3 \times 3$  mReLU architecture is shown in detail in Figure 2. This model was inspired by the concept of learning multiple patterns using intermediate layers in a CNN model [25, 28, 29]. In other words, utilizing stacking up the mReLU pipeline is more efficient than a simple linear chain of convolution layers to classify a varying-scale object. In the suggested mReLU block, four  $3 \times 3$  filters (small receptive fields) are used to solve the problem of overfitting and allow the framework to use a deeper architecture [28]. Also, the scale/shift layer is applied after the second concatenation layer to apply some trainable biases and weights [30, 31].

A negation layer is employed to multiply  $-1$  into the output of the previous layer to enhance the exploration process of informative features. We applied three and two Conv layers before the negation layers for improving the performance of the feature extraction process. Moreover, some bottleneck layers ( $1 \times 1$  Conv layers) are used to reduce the computation procedure [31, 32].

Moreover, we proposed a new inception block that obtains the multiscale nature of car segmentation tasks and enhances the performance of the model when encountering a complex background [20, 30]. This proposed inception block in high detailed descriptions is indicated in Figure 3. This idea is inspired by some works conducted by [31, 32] that decreases the number of feature kernels in each layer. This decreasing the number of parameters leads to maintaining the sparsity of the architecture and improving the computational performance.

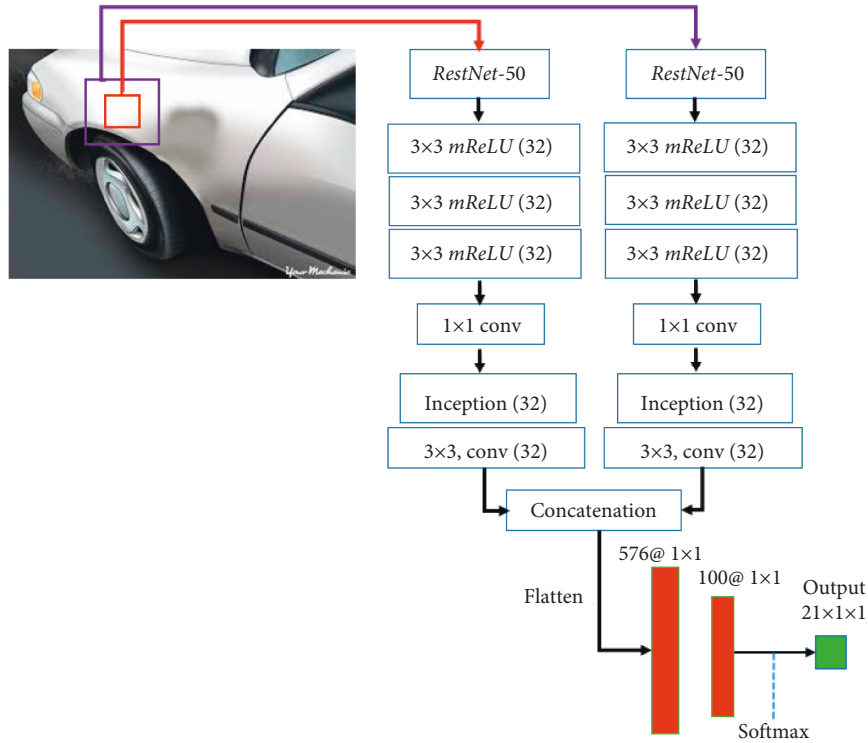


FIGURE 1: Our two-route CNN model using two ResNet-50.

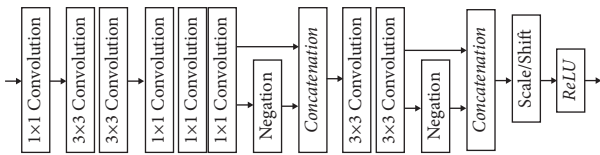


FIGURE 2: Our suggested  $3 \times 3mReLU$ .

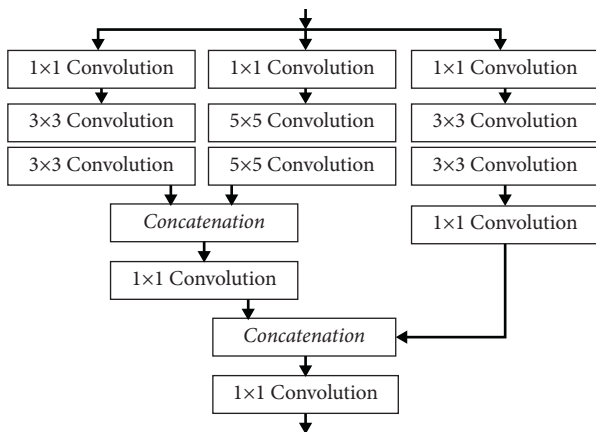


FIGURE 3: Our suggested inception block.

The inception block has three routes for extracting features. At the beginning of each route, a  $1 \times 1$  convolutional layer is used and its output fed into two convolution layers. Then, the output of the first two routes is concatenated and is fed into another bottleneck

layer. Next, the output of routes one and two is concatenated with the third route. Due to the use of inception and  $3 \times 3mReLU$  blocks, the overfitting problems have been satisfactorily addressed.

**2.2. Data Augmentation.** For effective learning and implementation, a CNN model needs to be trained with a large amount of the training data [18, 33]. The deep learning-based approaches need to be trained on large training datasets for avoiding overfitting and to maximize learning. Besides, the performance and learning accuracy of DL models are improved with ample and high-quality training data. Data augmentation (DA) techniques are used for changing or enhancing a dataset [34, 35].

In this paper, we use some DA techniques including geometric transformations (crop, rotate, and randomly flip), brightness transformation, mirror transformation, Gaussian noise, random elastic deformations, and random intensity variation [36–38].

### 3. Results and Discussion

The employing dataset includes 3,000 images of different sizes obtained using various cameras. All images were resized to the size of  $620 \times 620$  before applying to the CNN model. The experiments were carried out employing Python on the NVIDIA Tesla K80 GPU, 8 GB RAM, and Windows 10. Our technique is compared with some studies in terms of car damage detection. We assess the accuracy of the suggested strategy with different criteria that are defined as follows [39–41].

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{FP} + \text{TP}} \times 100\%, \\ \text{recall} &= \frac{\text{TP}}{\text{FN} + \text{TP}} \times 100\%, \\ \text{IoU} &= \frac{\text{area of overlap}}{\text{area of union}}, \end{aligned} \quad (1)$$

where false positive (FP) is the number of pixels that are incorrectly categorized as the body of a car while true positive (TP) implies the number of pixels that are correctly predicted by the suggested CNN framework. Also, a false negative (FN) indicates classified pixels that do not belong to the car; however, they are wrongly predicted as the car parts.

In Table 1, the results of detecting car parts are listed for 10 parts of the car. As it is clearly shown, the best outcomes for the precision are obtained for fenders, front doors, and windshield, and the worst results are related to quarters, roof, rear bumper, and back doors. For the recall assessment, the fenders, hood, and windshield gained the best results whereas the roof, front bumper, back doors, and quarters gained the worst results. Similar to the recall and precision outcomes, the best outcomes in terms of the IoU were achieved for front doors.

In Table 2, the comparison results for detecting car parts between our model and eight other approaches are listed. As it is clearly indicated, the proposed architecture gained the best results among all the eight other models. Moreover, the FCOMB [12] and VGG [1] strategies have the worst outcome among all the other techniques in terms of all evaluation criteria while the texture descriptor [8] pipeline is the second best model.

By looking deeply, we come to understand that the PANet [42] model indicates a very similar performance to the texture descriptor [8] model, and the VGG model [1] demonstrates an almost alike consequence to the combined feature (YOLOv3) [43] model. The differences between the obtained values of recall and precision criteria from the FCOMB [12] model and the suggested model are great numbers equal to 34% and 33%, respectively.

Table 3 exhibits the results of detecting car parts using the proposed model. The best outcomes in terms of the precision achieved for the trunk lid and front doors and the worst outcomes are related to the rear bumper. For the recall assessment criteria, the roof, trunk lid, windshield, and hood obtained the greatest results whereas the back doors and fenders achieved the worst results. Similar to the recall and precision outcomes, the best outcomes in terms of the IoU were achieved for the roof.

Table 4 indicates the comparison of results for detecting damaged parts of a car using our approach and some of the recently published papers in terms of Recall, Precision, and IoU. By comparing the gained results belonging to all models in Table 4, it is clear that the combined feature (YOLOv3) [43] and CNN [47] models gained the worst outcomes in terms of all criteria. In contrast, the HTC [46] and texture descriptor [8] pipelines obtained the next greatest values for all criteria. Nevertheless, the suggested two-route model

TABLE 1: Quantitative assessment based on recall, precision, and IoU for detecting different parts of a car.

Parts	Precision (%)	Recall (%)	IoU
Windshield	91	94	0.89
Trunk lid	88	92	0.87
Roof	87	89	0.86
Front bumper	88	90	0.83
Rear bumper	87	91	0.81
Front doors	92	92	0.90
Back doors	87	90	0.86
Quarters	87	90	0.85
Fenders	93	95	0.86
Hood	90	93	0.82

TABLE 2: Quantitative comparison between the proposed model and eight other models based on recall, precision, and IoU for detecting different parts of a car.

Architecture	Precision (%)	Recall (%)	IoU
PANet [42]	86	85	0.78
Combined feature (YOLOv3) [43]	66	62	0.61
VGG models [1]	64	62	0.60
FCOMB [12]	56	58	0.55
Improved mask RCNN [44]	69	72	0.64
Mask RCNN [45]	65	70	0.69
HTC [46]	83	84	0.76
Texture descriptor [8]	87	86	0.80
Proposed method	89	92	0.85

TABLE 3: Quantitative assessment based on recall, precision, and IoU for detecting different damaged parts of a car.

Parts	Precision (%)	Recall (%)	IoU
Windshield	93	94	0.91
Trunk lid	96	94	0.93
Roof	92	95	0.92
Front bumper	92	90	0.89
Rear bumper	91	92	0.88
Front doors	95	92	0.91
Back doors	94	90	0.91
Quarters	90	91	0.87
Fenders	93	90	0.88
Hood	92	94	0.89

TABLE 4: Quantitative comparison between the proposed model and eight other models based on recall, precision, and IoU for detecting different damaged parts of a car.

Architecture	Precision (%)	Recall (%)	IoU
PANet [42]	87	86	0.85
Combined feature (YOLOv3) [43]	67	68	0.64
CNN [47]	58	61	0.57
FFNN [48]	88	89	0.86
Mask RCNN [45]	84	81	0.78
VGG models [1]	85	84	0.81
HTC [46]	89	90	0.87
Texture descriptor [8]	90	91	0.88
Proposed method	93	92	0.90



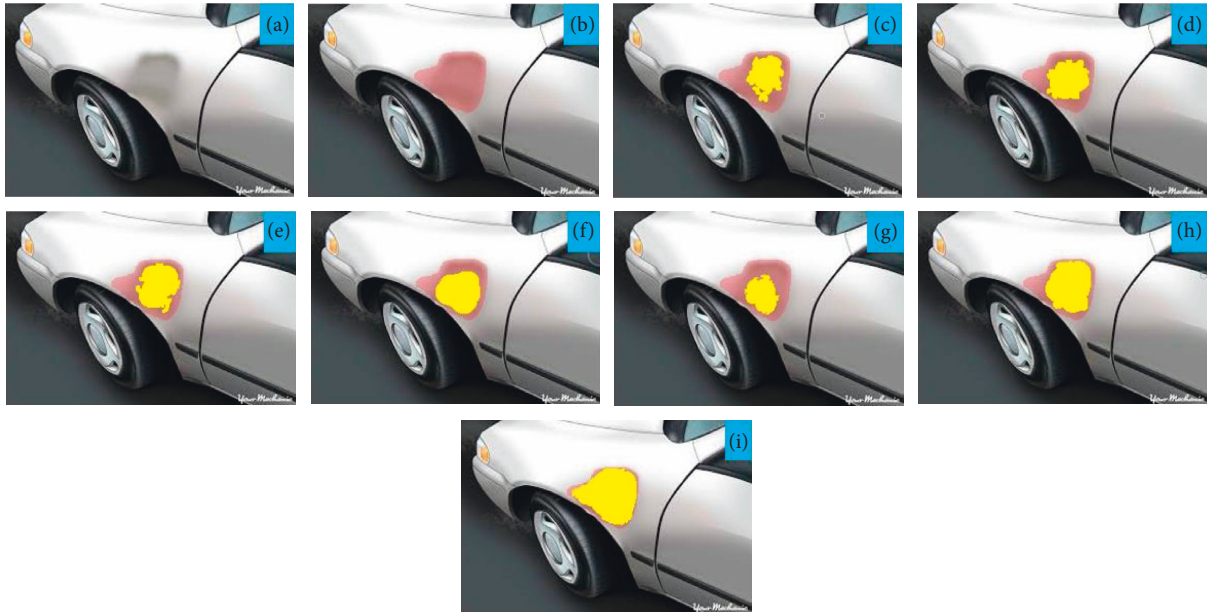


FIGURE 4: A visual comparison between our model and some other techniques. (a) Original image. (b) Labeled image. (c) YOLOv3. (d) Mask RCNN. (e) VGG. (f) FFNN. (g) CNN. (h) Texture descriptor. (i) Proposed model.

gained the best values among all compared models, which denotes its high effectiveness in the achievement of the desired objectives.

Figure 4 denotes the results of detecting a damaged part of a car using different models. As it is clearly shown, the damaged region of the car could not be recognized correctly by YOLOv3, Mask RCNN, FFNN, and CNN. For further explanation, the center region of the fender could not be appropriately recognized by the Mask RCNN, Yolov3, and CNN techniques. This difficulty was addressed during the implementation of the texture descriptor and VGG architectures. The segmentation outcomes obtained using FFNN and texture descriptor demonstrated an improvement in recognizing damaged regions in areas with high illumination variation and could reduce false positives.

#### 4. Conclusion

In this paper, we suggested a deep learning-based model for recognizing damaged parts of the car. We employed a two-route CNN model that is able to extract both global and local features from the input image. In order to minimize the efficiency of the model for segmenting target parts, a pre-trained ResNet-50 model was used at the beginning of the pipeline. Moreover, an inception block and a  $3 \times 3$  mReLU block were suggested to solve the overfitting. After analyzing the suggested framework, we realized that the outcomes of utilizing a pretrained CNN model that is applied to each route lead to a high segmentation performance with respect to some other models such as FFNN, YOLOv3, HTC, and VGG. To validate the suggested model, it was implemented on a private car collection database, and its outcomes from three measurement criteria were compared with some state-of-the-art techniques, including PANet [42], combined feature (YOLOv3) [43], CNN [47], FFNN [48], mask RCNN

[45], VGG models [1], HTC [46], and texture descriptor [8]. The results indicated that our framework demonstrated better achievements toward the comparative approaches in different terms of other comparative methodologies.

#### 5. Disclosure

The funding sources had no involvement in the study design, collection, analysis or interpretation of data, writing of the manuscript, or in the decision to submit the manuscript for publication.

#### Data Availability

Data will be available upon request to the corresponding author.

#### Conflicts of Interest

The authors declare that they have no conflicts of interest.

#### References

- [1] P. M. Woraratpanya and K. Woraratpanya, "Car damage detection and classification," in *Proceedings of the 11th International Conference on Advances in Information Technology*, Bangkok, Thailand, July 2020.
- [2] J. De Deijn, *Automatic Car Damage Recognition using Convolutional Neural Networks*, M.S. thesis, 2018.
- [3] M. Parhizkar and M. Parhizkar, "Integration of image segmentation and fuzzy theory to improve the accuracy of damage detection areas in traffic accidents," *Journal of Big Data*, vol. 8, no. 1, 147 pages, 2021.
- [4] K. Pasupa, P. Kittiworapanya, N. Hongngern, and K. Woraratpanya, "Evaluation of deep learning algorithms for semantic segmentation of car parts," *Complex & Intelligent Systems*, pp. 1–13, 2021.

- [5] P. Rakshata, "Car damage detection and analysis using deep learning," *Algorithm For Automotive*, vol. 5, no. 6, pp. 1896–1898, 2019.
- [6] A. Uhlmann, Z. Uhlmann, N. F. Glenn et al., "Tree canopy and snow depth relationships at fine scales with terrestrial laser scanning," *The Cryosphere*, vol. 15, no. 5, pp. 2187–2209, 2021.
- [7] M. Wilmanski, C. Kreucher, and J. Lauer, "Modern approaches in deep learning for SAR ATR," *Algorithms for Synthetic Aperture Radar Imagery XXIII*, vol. 9843, Article ID 98430N, May 2016.
- [8] M. Amirfakhrian and M. Amirfakhrian, "Car detection and damage segmentation in the real scene using a deep learning approach," *International Journal of Intelligent Robotics and Applications*, vol. 6, no. 2, pp. 231–245, 2022.
- [9] Y. Pourasad, R. Ranjbarzadeh, and A. Mardani, "A new algorithm for digital image encryption based on chaos theory," *Entropy*, vol. 23, no. 3, p. 341, 2021.
- [10] S. Jafarzadeh Ghoushchi, R. Ranjbarzadeh, S. A. Najafabadi, E. Osgoee, and E. B. Tirkolaei, "An extended approach to the diagnosis of tumour location in breast cancer using deep learning," *Journal of Ambient Intelligence and Humanized Computing*, 2021.
- [11] S. B. Ranjbarzadeh, R. Ranjbarzadeh, A. Amirabadi et al., "Osteolysis: a literature review of basic science and potential computer-based image processing detection methods," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–21, Article ID 4196241, 2021.
- [12] B. Balci, Y. Artan, B. Alkan, and A. Elihos, "Front-view vehicle damage detection using roadway surveillance camera images," *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems*, pp. 193–198, 2019.
- [13] C. T. Kaya and T. Kaya, "Car damage analysis for insurance market using convolutional neural networks," *Intelligent and Fuzzy Techniques in Big Data Analytics and Decision Making*, vol. 1029, pp. 313–321, 2020.
- [14] S. Jayawardena, *Image Based Automatic Vehicle Damage Detection*, 2013.
- [15] T. Kumar, J. Park, M. S. Ali, A. F. M. Uddin, J. H. Ko, and S.-H. Bae, "Binary-Classifiers-Enabled Filters for Semi-supervised Learning," *IEEE Access*, vol. 9, pp. 167663–167673, 2021.
- [16] C. Wang, G. Yan, S. Yu et al., "Fast vehicle and pedestrian detection using improved mask R-CNN," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–15, Article ID 5761414, 2020.
- [17] A. Shirode, T. Rathod, P. Wanjari, and A. Halbe, "Car damage detection and assessment using CNN," *2022 IEEE Delhi Section Conference (DELCON)*, vol. 2022, Article ID 9752971, 2022.
- [18] R. Dorosti, S. Jafarzadeh Ghoushchi, S. Safavi et al., "Nerve optic segmentation in CT images using a deep learning model and a texture descriptor," *Complex & Intelligent Systems*, pp. 1–15, 2022.
- [19] K. Kumar and A. Kumar, "Cross-spectral iris recognition using CNN and supervised discrete hashing," *Pattern Recognition*, vol. 86, pp. 85–98, Feb. 2019.
- [20] C. Szegedy, L. Wei, J. Yangqing et al., "Going Deeper with Convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, MA, USA, June 2015.
- [21] S. Vitale, G. Ferraioli, and V. Pascasio, "Multi-objective CNN-based algorithm for SAR despeckling," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 11, pp. 9336–9349, 2021.
- [22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the - IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, December 2016.
- [23] S. Baseri Saadi, N. Tataei Sarshar, S. Sadeghi, R. Ranjbarzadeh, M. Kooshki Forooshani, and M. Bendeche, "Investigation of effectiveness of shuffled frog-leaping optimizer in training a convolution neural network," *Journal of Healthcare Engineering*, vol. 2022, pp. 1–11, Article ID 4703682, 2022.
- [24] K. Yasaka, H. Akai, O. Abe, and S. Kiryu, "Deep learning with convolutional neural network for differentiation of liver masses at dynamic contrast-enhanced CT: a preliminary study," *Radiology*, vol. 286, no. 3, pp. 887–896, 2018.
- [25] F. Naiemi, V. Ghods, and H. Khalesi, "A novel pipeline framework for multi oriented scene text image detection and recognition," *Expert Systems with Applications*, vol. 170, pp. 114549–122021, 2021.
- [26] Z. Tian, M. Shu, P. Lyu et al., "Learning Shape-Aware Embedding for Scene Text Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [27] T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2529–2541, 2016.
- [28] S. Hong, B. Roh, K.-H. Kim, Y. Cheon, and M. Park, "PVANet: Lightweight Deep Neural Networks for Real-Time Object Detection," 2016, <http://arxiv.org/abs/1611.08588>.
- [29] A. Valizadeh, S. Jafarzadeh Ghoushchi, R. Ranjbarzadeh, and Y. Pourasad, "Presentation of a segmentation method for a diabetic retinopathy patient's fundus region detection using a convolutional neural network," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–14, Article ID 7714351, 2021.
- [30] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, "Rethinking the Inception Architecture for Computer Vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, June 2016.
- [31] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Inception recurrent convolutional neural network for object recognition," *Machine Vision and Applications*, vol. 32, no. 1, pp. 28–44, 2021.
- [32] R. F. Aljehane and N. O. Aljehane, "An optimal segmentation with deep learning based inception network model for intracranial hemorrhage diagnosis," *Neural Computing & Applications*, vol. 33, no. 20, pp. 13831–13843, 2021.
- [33] A. Chandio, Y. Shen, M. Bendeche, I. Inayat, and T. Kumar, "AUDD: audio Urdu digits dataset for automatic audio Urdu digit recognition," *Applied Sciences*, vol. 11, no. 19, p. 8842, 2021.
- [34] F. Zhan, H. Zhu, and S. Lu, "Scene Text Synthesis for Efficient and Effective Deep Network Training," 2019, <https://arxiv.org/abs/1901.09193v1>.
- [35] M. R. Matin and A. Matin, "Detection of COVID 19 from CT image by the novel LeNet-5 CNN architecture," in *Proceedings of the 2020 23rd International Conference on Computer and Information Technology (ICCIT)*, December 2020.
- [36] F. Wang, S. H. Zhong, J. Peng, J. Jiang, and Y. Liu, "Data augmentation for eeg-based emotion recognition with deep convolutional neural networks," *MultiMedia Modeling*, vol. 10705, pp. 82–93, 2018.

- [37] D. Abdelhafiz, C. Yang, R. Ammar, and S. Nabavi, "Deep convolutional neural networks for mammography: advances, challenges and applications," *BMC Bioinformatics*, vol. 20, no. S11, pp. 281–320, 2019.
- [38] C. Khoshgoftaar and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 60–48, Dec. 2019.
- [39] R. Jafarzadeh Ghouschi, S. Bendechache, M. Amirabadi et al., "Lung infection segmentation for COVID-19 pneumonia based on a cascade convolutional network from CT images," *BioMed Research International*, vol. 2021, pp. 1–16, Article ID 5544742, 2021.
- [40] N. Karimi, R. Ranjbarzadeh Kondrood, and T. Alizadeh, "An intelligent system for quality measurement of Golden Bleached raisins using two comparative machine learning algorithms," *Measurement*, vol. 107, pp. 68–76, Sep. 2017.
- [41] R. Saadi and S. B. Saadi, "Automated liver and tumor segmentation based on concave and convex points using fuzzy c-means and mean shift clustering," *Measurement*, vol. 150, Article ID 107086, 2020.
- [42] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, *Path Aggregation Network for Instance Segmentation*, pp. 8759–8768, 2018.
- [43] H. S. Malik, M. Dwivedi, S. N. Omakar, S. R. Samal, A. Rathi, and E. B. Monis, *EasyChair Preprint Deep Learning Based Car Damage Classification and Detection*, 2020.
- [44] Q. Zhang, X. Chang, and S. B. Bian, "Vehicle-damage-detection segmentation algorithm based on improved mask RCNN," *IEEE Access*, vol. 8, pp. 6997–7004, 2020.
- [45] A. Patil, "Car damage recognition using the expectation maximization algorithm and mask R-CNN," *Information and Communication Technology for Intelligent Systems*, vol. 196, pp. 607–616, May 2020.
- [46] K. Chen, "Hybrid task cascade for instance segmentation," pp. 4974–4983, 2019, <https://github.com/>.
- [47] K. Patil, M. Kulkarni, A. Sriraman, and S. Karande, "Deep learning based car damage classification," *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, vol. 2017, pp. 50–54, 2017.
- [48] A. F. Agarap, "Deep Learning Using Rectified Linear Units (ReLU) Deep Learning Using Rectified Linear Units (ReLU)," pp. 2–8, 2020, <https://arxiv.org/abs/1803.08375>.

## Research Article

# Denoising Seismic Data via a Threshold Shrink Method in the Non-Subsampled Contourlet Transform Domain

Yu Yang,<sup>1</sup> Qi Ran,<sup>2</sup> Kang Chen,<sup>2</sup> Cheng Lei,<sup>2</sup> Yusheng Zhang,<sup>3</sup> Han Liang,<sup>2</sup> Song Han,<sup>2</sup> and Cong Tang<sup>2</sup> 

<sup>1</sup>PetroChina Southwest Oil & Gasfield Company, Chengdu, Sichuan 610051, China

<sup>2</sup>Research Institute of Exploration and Development, PetroChina Southwest Oil & Gasfield Company, Chengdu 610051, Sichuan, China

<sup>3</sup>Oil and Gas Resources Division, PetroChina Southwest Oil & Gasfield Company, Chengdu 610051, Sichuan, China

Correspondence should be addressed to Cong Tang; [tangcong@petrochina.com.cn](mailto:tangcong@petrochina.com.cn)

Received 14 March 2022; Revised 31 May 2022; Accepted 3 June 2022; Published 8 August 2022

Academic Editor: Saeid Jafarzadeh Ghouschi

Copyright © 2022 Yu Yang 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 seismic exploration, effective seismic signals can be seriously distorted by and interfered with noise, and the performance of traditional seismic denoising approaches can hardly meet the requirements of high-precision seismic exploration. To remarkably enhance signal-to-noise ratios (SNR) and adapt to high-precision seismic exploration, this work exploits the non-subsampled contourlet transform (NSCT) and threshold shrink method to design a new approach for suppressing seismic random noise. NSCT is an excellent multiscale, multidirectional, and shift-invariant image decomposition scheme, which can not only calculate exact contourlet transform coefficients through multiresolution analysis but also give an almost optimized approximation. It has better high-frequency response and stronger ability to describe curves and surfaces. Specifically, we propose to utilize the superior performance NSCT to decomposing the noisy seismic data into various frequency sub-bands and orientation response sub-bands, obtaining fine enough transform high frequencies to effectively achieve the separation of signals and noises. Besides, we use the adaptive Bayesian threshold shrink method instead of traditional handcraft threshold scheme for denoising the high-frequency sub-bands of NSCT coefficients, which pays more attention to the internal characteristics of the signals/data itself and improve the robustness of method, which can work better for preserving richer structure details of effective signals. The proposed method can achieve seismic random noise attenuation while retaining effective signals to the maximum degree. Experimental results reveal that the proposed method is superior to wavelet-based and curvelet-based threshold denoising methods, which increases synthetic seismic data with lower SNR from  $-8.2293$  dB to  $8.6838$  dB, and  $11.8084$  dB and  $9.1072$  dB higher than two classic sparse transform based methods, respectively. Furthermore, we also apply the proposed method to process field data, which achieves satisfactory results.

## 1. Introduction

In recent years, high-precision seismic exploration has been a key subject in modern seismic exploration. This technique will be hindered if the noise in acquired seismic signals cannot be removed perfectly. The traditional seismic data denoising approaches can hardly meet the requirements of high-precision seismic exploration because the level and complexity of the accompanying noise in seismic signals have significantly increased due to the increasingly complex

exploration environment and the increase in exploration depth with the extension of field of seismic exploration. So, it is crucial to design new effective techniques to remarkably enhance the signal-to-noise ratio (SNR).

At present, many seismic denoising approaches have been proposed including the initial seismic data denoising method [1], traditional transform domain based denoising methods [2–4], sparse transform based methods [5–8] for solving multitasks, learning-based methods [9, 10], and other methods [11–14]. Actually, as the most common

seismic noise, random noise can penetrate the whole time domain and severely distort and interfere with effective seismic data. Thus, since Canales [1] first developed a random noise reduction approach, a lot of random noise attenuation methods have been presented on this basis, such as the sparse transform based approaches, the empirical mode decomposition (EMD) based approaches, and fast dictionary learning-based approaches [10]. Chen and Ma [4] removed random noise with predictive filtering of f-x empirical mode decomposition. Chen and Fomel [6] developed an EMD-Seislet transform based method to remove the seismic random noise. Liu et al. [7] presented variational mode decomposition for suppressing the random noise of seismic data. In reality, one type of the intensively used and most efficient seismic random noise attenuation approaches are based on the sparse transform of multiscale geometric analysis. Zhang and Lu [2] removed noise and improved the resolution of seismic data by using the applied wavelet transform. Neelamani et al. [3] attenuated random noise with the curvelet transform, and subsequently, several variants [8, 12] with good results have been reported. Lin [14] proposed a three-dimensional (3-D) steerable pyramid decomposition-based suppression method of seismic random noise. Sang et al. [15] presented an unconventional technique on the basis of a proximal classifier with consistency (PCC) in transform domain for attenuating seismic random noise, and they also proposed another seismic denoising approach [16] via the deep neural network and simultaneously suppressed seismic coherent and incoherent noises [17] based on the deep neural network.

High SNR data are the important guarantee of high-precision seismic exploration. But, the existing transform domain based methods are difficult to obtain higher SNR data due to not fine enough transform high frequencies such as wavelet transform or curvelet transform. Compared with the existing sparse transform based methods, that is, wavelet-based transform and curvelet-based transform methods, the NSCT presents multiscale, multidirectional, and shift-invariant decomposition scheme, which has better high-frequency response and stronger ability to describe curves and surfaces. Besides, they often conduct rough threshold operation by using manual threshold processing methods such as hard thresholding or soft thresholding. There is often a loss of effective signals. Therefore, to remarkably enhance SNRs and adapt to high precision seismic exploration, we exploit an effective seismic data denoising method in this paper. The contributions are as follows:

- (i) We propose to utilize the new sparse transform technique, non-subsampled contourlet transform (NSCT), to decomposing the noisy seismic data into various frequency sub-bands and orientation response sub-bands, obtaining fine enough transform high frequencies to effectively achieve the separation of signals and noises.
- (ii) We use the adaptive Bayesian threshold shrink method instead of the traditional handcraft threshold scheme for denoising the high-frequency sub-bands of NSCT coefficients, which pays more

attention to the internal characteristics of the signals/data itself and improve the robustness of method, which can work better for preserving richer structure details of effective signals.

- (iii) We conduct the experiments on synthetic and field data, which reveals that our approach is superior to the wavelet and the curvelet transform based classical ones, achieving higher signal-to-noise ratio (SNR) values.

The remainder of this paper is organized as follows. We present our method in Section 2. Experiments and performance evaluation are presented in Sections 3. Conclusion is drawn in Section 4.

## 2. Method

In this paper, we focus on transform domain based thresholding methods due to their good performance. The wavelet-based thresholding scheme is the most classic method for seismic data denoising. Wavelets can sparsely represent one-dimensional (1-D) digital data with smoothed point discontinuities and have been successfully used for representing digital signals [18]. However, wavelets cannot efficiently handle higher dimensional data because of the usual presence of other kinds of singularities. As a matter of fact, curvelets [19], contourlets [20], bandelets [21], and some other image/signal representations can take the advantages of the anisotropic regularity of a surface along edges, but these representations all have their own disadvantages, such as lack of a multiresolution geometry representation for curvelets, extremely limited clear directional features for contourlets, and computationally expensive geometry optimization for bandelets. The non-subsampled contourlet transform (NSCT) [22] is an excellent multiscale, multidirectional, and shift-invariant image decomposition scheme, which can not only calculate exact contourlet coefficients through multiresolution analysis but also give an almost optimized approximation. It has better high-frequency response and stronger ability to describe curves and surfaces. Therefore, we attempt to utilize NSCT to denoise seismic data in this paper.

*2.1. NSCT for Seismic Data.* The NSCT [22] primarily consists of a cascade of non-subsampled pyramid filter bank (NSPFB) and non-subsampled direction filter bank (NSDFB). First, the NSPFB is utilized to decompose an image and the sub-bands obtained are used as inputs of the NSDFB to generate decomposition results of the initial image in multiple directions and dimensions. The NSCT conducts  $K$ -level decomposition on an image to produce one low-frequency (LF) and several high-frequency (HF) sub-bands, and the size of all these sub-bands is identical with that of the original image. So, the full reconstruction of NSCT is possible since NSPFB and NSDFB can both be completely rebuilt.

The shift-invariant filtering structure of the NSCT results in its multiscale feature. By using a bank of non-subsampled 2-D two-channel filters, we have sub-band decomposition

like the Laplacian pyramid. Figure 1 shows a 3-stage non-subsampled pyramid (NSP) decomposition, whose expansion has an alike concept with the 1-D non-subsampled wavelet transform (NSWT) using the à trous algorithm [22]. For a  $J$ -stage decomposition, the redundancy will be  $J + 1$ . The region  $[-(\pi/2^j), (\pi/2^j)]^2$  and its complement are the ideal passband support of the low- and high-pass filter at the  $j$ -th stage, respectively. Upsampling the filter for the first stage can yield the filters for subsequent stages; thus, no additional filter is needed to give the multiscale property. Our structure differs from that of the separable NSWT. Particularly, our structure produces one bandpass image in each stage leading to  $J + 1$  redundancy. However, the NSWT generates three directional sub-bands in each stage, leading to  $3J + 1$  redundancy. The advantage of NSP lies in its ability to generate better filters due to its generality.

The NSCT has following steps. First, the NSPFB is used to multidimensionally decompose an image into an HF and an LF sub-band. In multilevel decomposition, an image is finally decomposed to an LF sub-band and a set of HF sub-bands if only its LF sub-band is further iteratively filtered. The redundancy of an  $X$ -level NSPFB decomposition will be  $X + 1$ . With regard to the  $X$ -level low-pass and the bandpass filter, the ranges of the ideal support of frequency domain are  $[-\pi/2^{X-1}, \pi/2^{X-1}]^2$  and  $[-\pi/2^{X-1}, \pi/2^{X-1}]^2 \cup [-\pi/2^X, \pi/2^X]^2$ , respectively. The decomposition does not extra filter during acquiring multidimension properties. So, the generated redundancy of  $X + 1$  will be generated in each stage with a bandpass image, and the structure is significantly superior to that of the wavelet transform. Then, these sub-bands can be decomposed along singular points and multiple directions in various dimensions, and the directions are integrated. The NSDFB also belongs to a two-channel filter bank which comprises decomposition filters  $U_i(z)$ , ( $i = 0, 1$ ) and synthesis filters  $V_i(z)$ , ( $i = 0, 1$ ) satisfying Bézout's identity:

$$U_0(z) + V_0(z) = U_1(z) + V_1(z). \quad (1)$$

To adopt ideal support of the frequency domain, two channels are decomposed by  $U_0(z)$  and  $U_1(z)$ . Then,  $U_1(z)$  and  $U_0(z)$  are upsampled instead of subsampled by all sampling matrices in each level to get direction filters in the subsequent levels. This completes the image decomposition and an NSCT transform is schematically presented in Figures 2 and 3. Figure 2 displays an overview of NSCT which has a filter bank for dividing the 2-D frequency plane into sub-bands plotted in the bottom left quarter of Figure 1. By using NSCT, we decompose the 2-D seismic signal data into two shift-invariant components: an NSP structure to ensure multiscale properties (Part 1 of Figure 2) and an NSDFB structure to give directionality (Part 2 of Figure 2). The obtained idealized frequency partitioning diagram is presented in Figure 3. The structure consists in a bank of filters that splits the 2-D frequency plane into several sub-bands. In this paper, we use this mode of non-downsampling to reduce the sampling distortion in the filters and obtain translation invariance, in which the size of the directional sub-band at each scale is the same as that of the original 2-D seismic signal matrix. The NSCT has more details to be preserved,

and the decomposition can better maintain the edge information and contour structure of the seismic signals.

Figure 4 shows processing results of implementing the two-level NSCT on the synthesized seismic signal data with noise (Figure 4(a)) to yield a low-pass sub-band (Figure 4(b)) and a set of high-pass sub-bands (Figures 4(c) and 4(d)). Here, two and four shearing directions are used for the coarser and the finer scale, respectively. We can see from Figure 4 that the LF record (Figure 4(b)) decomposed by the NSCT basically contains the effective synthesized seismic signals. For HF records (Figure 4(d)) of scale 1, all they contain is noise, while HF records (Figure 4(c)) of scale 2 contain partially effective signals and noise, which needs to be further processed via signal-noise separation.

*2.2. Denoising Seismic Data Using the Threshold Shrink in the NSCT Domain.* To remarkably suppress seismic random noise while not damaging the effective signal in the process of denoising, this paper proposes a novel NSCT-based scheme with an adaptive threshold value setting for suppressing seismic random noise. The NSCT with the properties of multiscale, multidirection, and relative optimized sparsity can not only calculate the exact contourlet coefficients through multiresolution analysis but also give an almost optimized approximation. As the NSCT is multidirectional, large coefficients can be obtained when the direction of the NSCT basic function is approximately the same as the direction of the seismic signals, while small coefficients can be obtained when they have large difference. Thus, the random noises are distributed on small coefficients, so we can remove smaller coefficients to achieve random noise attenuate using an appropriate thresholding operator.

We first analyze the sparsity of NSCT before giving the steps of denoising. It is known that the degree of approximation of the decomposed effective data determines the effect of noise suppression [23]. That is to say, the denoising effect depends on the sparsity of the approach. Figure 5 presents the reconstruction error on synthesized data (Figure 6(a)) in the wavelet transform domain, curvelet transform domain, and NSCT domain. Clearly, the construction error of NSCT is the smallest at the same percentage of coefficients, and it is approximate to zero at 6% coefficient, showing its optimal sparsity. In Figure 6, we compare the high-frequency coefficients of NSCT, curvelet transform, and wavelet transform, where we can clearly see that the NSCT represents the curvature more accurately.

Generally, one threshold is used for the whole image/signals (or sub-bands) in signal denoising techniques based on the threshold shrink. Obviously, the threshold value should be smaller if the signals contain more effective information, and it should be larger if the signals have more smooth regions. For the two cases, a larger threshold value should be correspondingly used for a higher noise level. Evidently, detailed information with an optimal threshold value does not function adequately for smooth regions and vice versa. Therefore, setting for threshold value can be further optimized by introducing adaptive threshold for different regions in seismic signals to exploit the fact that

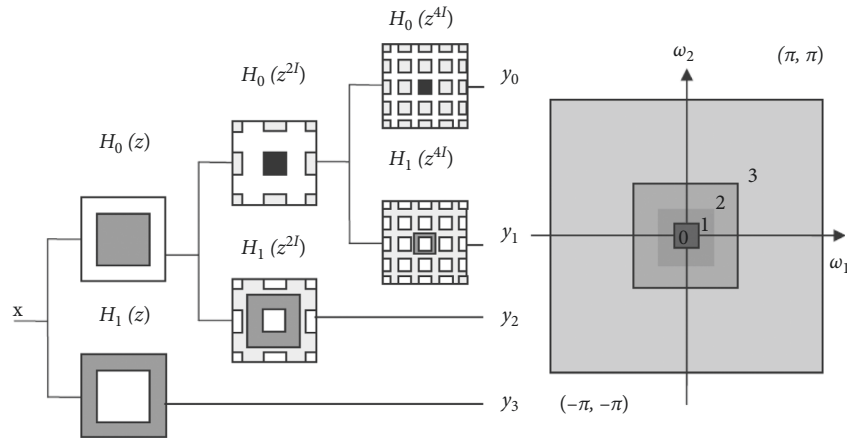


FIGURE 1: (a) Schematic diagram of 3-stage NSP decomposition. The aliasing due to upsampling is denoted with lighter gray. (b) Sub-bands on the 2-D frequency plane.

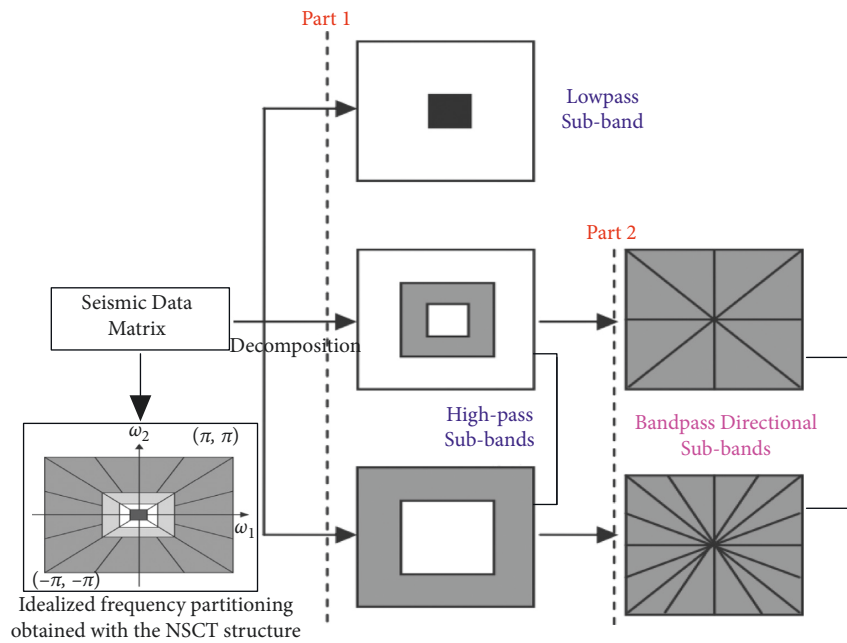


FIGURE 2: Schematic diagram of the NSCT of 2D seismic data.

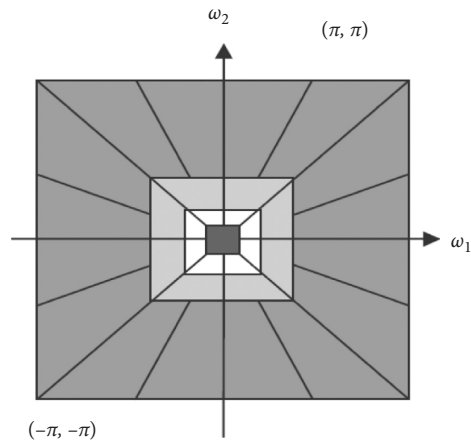


FIGURE 3: Ideal frequency division by the NSCT.

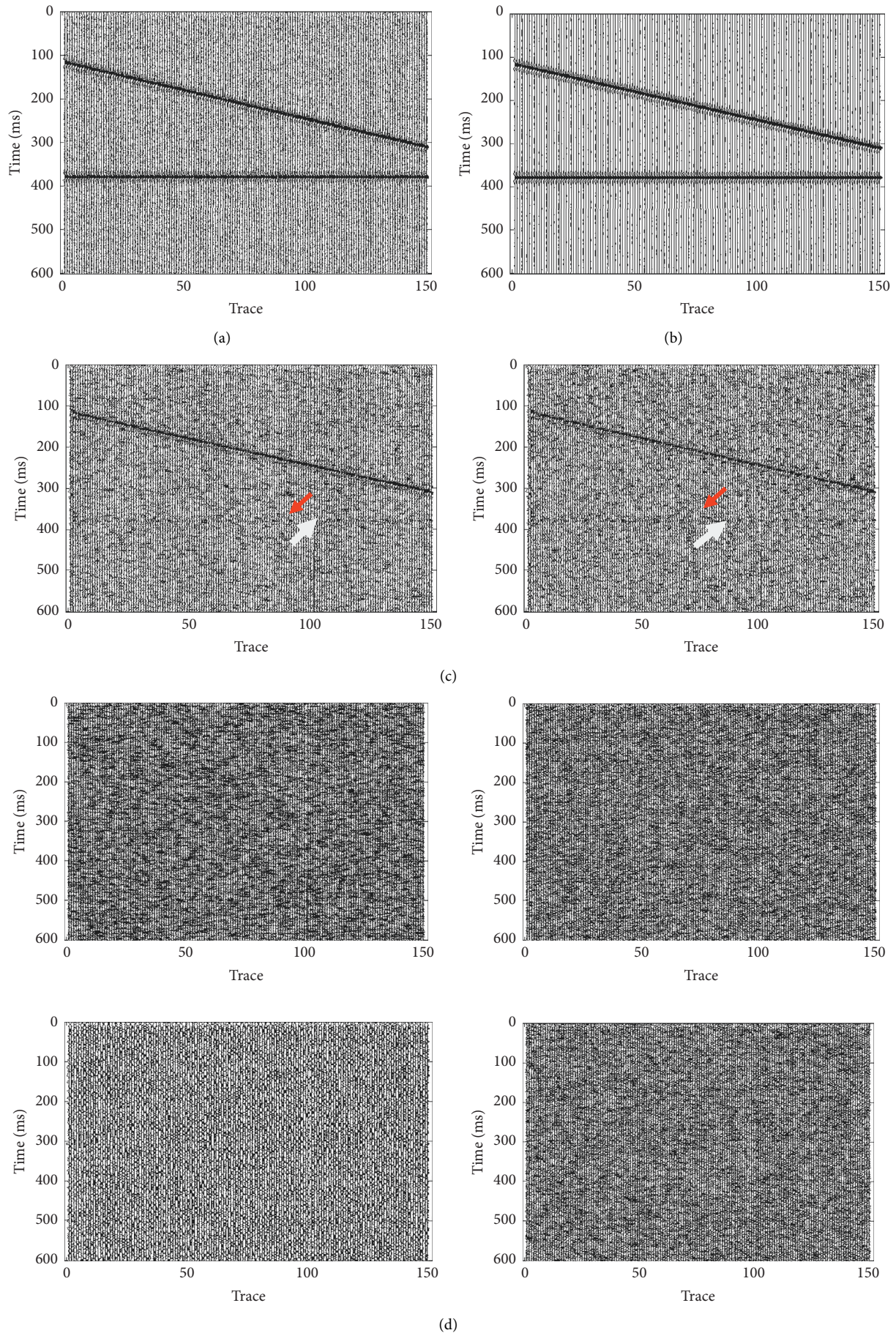


FIGURE 4: Example of NSCT processing. (a) Synthesized seismic signal data with noise. (b) Approximate NSCT coefficients. Seismic signals of the detailed NSCT coefficients at scale 2, 2 directions (c) and scale 1, 4 directions (d).



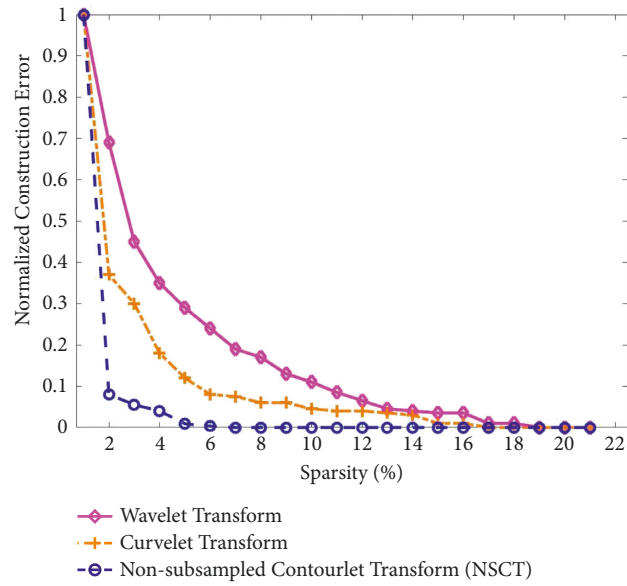


FIGURE 5: Reconstruction errors in three transform domains.

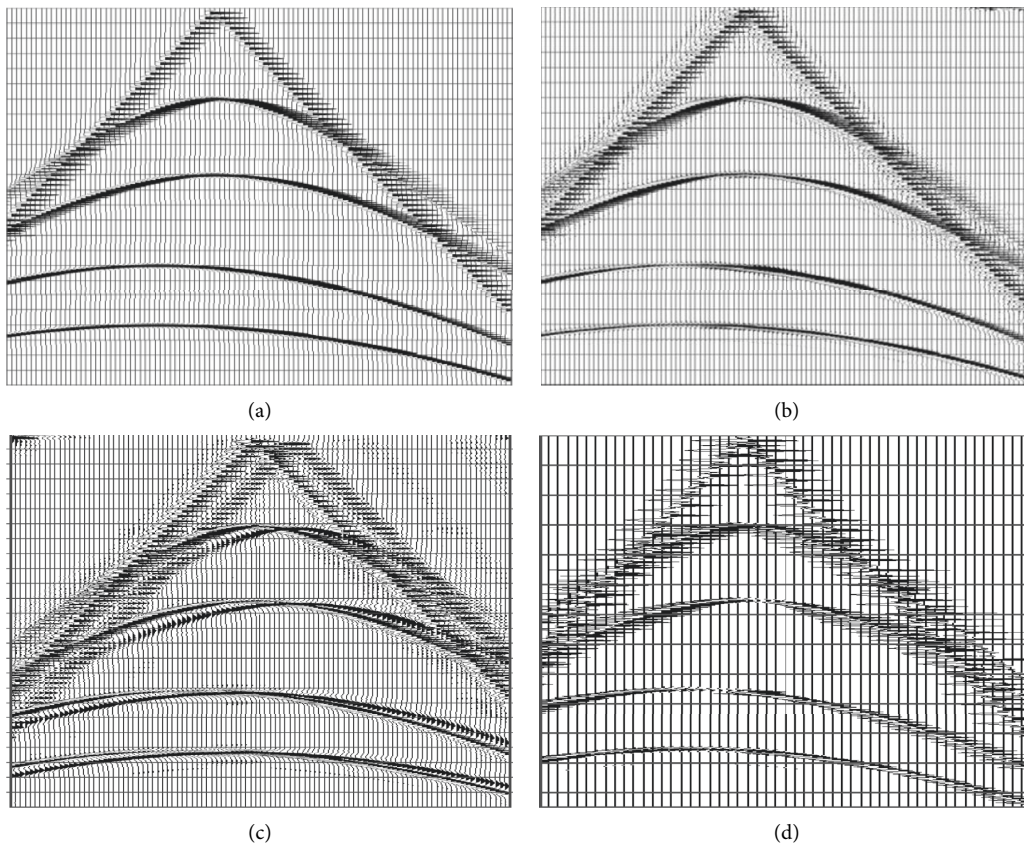


FIGURE 6: Comparison of the NSCT, curvelet transform, and WT coefficients on the synthetic seismic data. (a) The original synthetic seismic data, and the fusion of HF coefficients of (b) NSCT, (c) discrete curvelet transform, and (d) WT.

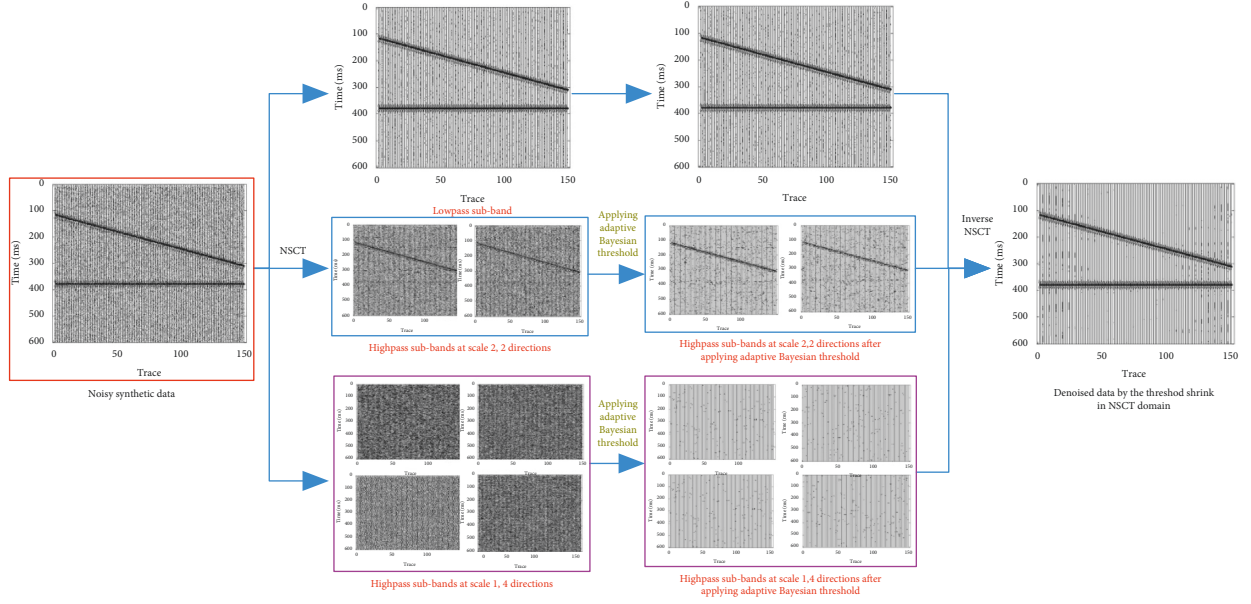


FIGURE 7: Demonstration for the analysis framework of seismic random denoising in the NSCT domain. (a) Before attack. (b) After attack.

most signals consist of smooth regions and effective seismic signal information.

Specifically, the two-dimensional noisy seismic data can be calculated by

$$f(t, g) = x(t, g) + n(t, g), \quad (2)$$

where  $t$  denotes time,  $g$  represents the trace number, and  $x(t, g)$ ,  $n(t, g)$ , and  $f(t, g)$  represent the effective seismic data, the additive random noise, and the noisy observed seismic signals, respectively. Signal  $x(t, g)$  will be recovered from  $f(t, g)$ .

In NSCT-based denoising approaches, a threshold is properly set for the NSCT coefficients so that seismic signals can be retrieved from the acquired noisy seismic data. The proposed seismic random noise attenuation has the following main steps.

*Step 1.* Decomposing noisy seismic data with a  $K$ -level NSCT to yield one low-pass sub-band and one set of high-pass sub-bands  $D_{k,j}$  ( $k = 1, 2, \dots, K; j = 1, 2, \dots, J$ ), with the current scale  $k$ , the decomposition orientation  $j$ , and the total number  $J$  of decomposition directions.

*Step 2.* Calculating denoising threshold values of all sub-bands  $D_{k,j}$ . The level adaptive Bayesian threshold [24] is used and calculated as below:

- (i) Using the robust median estimator to calculate noise variance  $\delta$  from sub-bands:

$$\delta = \frac{\text{Median}(|C(x, y)|)}{0.6745}, \quad C(x, y) \in D_{K,J}. \quad (3)$$

- (ii) Using the maximum likelihood estimator (MLE) [24] to estimate signal variance  $\delta_{k,j}$  for the noisy coefficients of each detail sub-band  $D_{k,j}$ :

$$\delta_{k,j} = \max\left(0, \frac{1}{mn} \sum_{x=1}^m \sum_{y=1}^n [C_{k,j}(x, y)]^2 - \delta^2\right), \quad (4)$$

where  $C_{k,j}(x, y) \in D_{k,j}$ , and  $m$  and  $n$  denote the size of seismic signals

- (iii) Calculating discriminating threshold  $\delta_{th}$  with the near exponential prior of NSCT coefficients across scales:

$$\delta_{th} = \frac{\delta \cdot \sum_k \delta_{k,j} \cdot 2^{-k}}{\sum_k k^2 \cdot 2^{-k}}, \quad (5)$$

where  $k$  denotes the current scale

- (iv) Calculating denoising threshold  $T(k, \delta_{k,j})$  of each sub-band for  $\delta_{k,j} < \delta_{th}$ :

$$T(k, \delta_{k,j}) = 2^{(k-(J/2))/J} \cdot \frac{\delta^2}{\delta_{k,j}}, \quad (6)$$

where  $\delta_{k,j}$  denotes the standard deviation of sub-band  $D_{k,j}$ .

*Step 3.* Processing the noise-related NSCT coefficients in high-frequency sub-bands  $D_{k,j}$  with the well-known soft-thresholding method [25]:

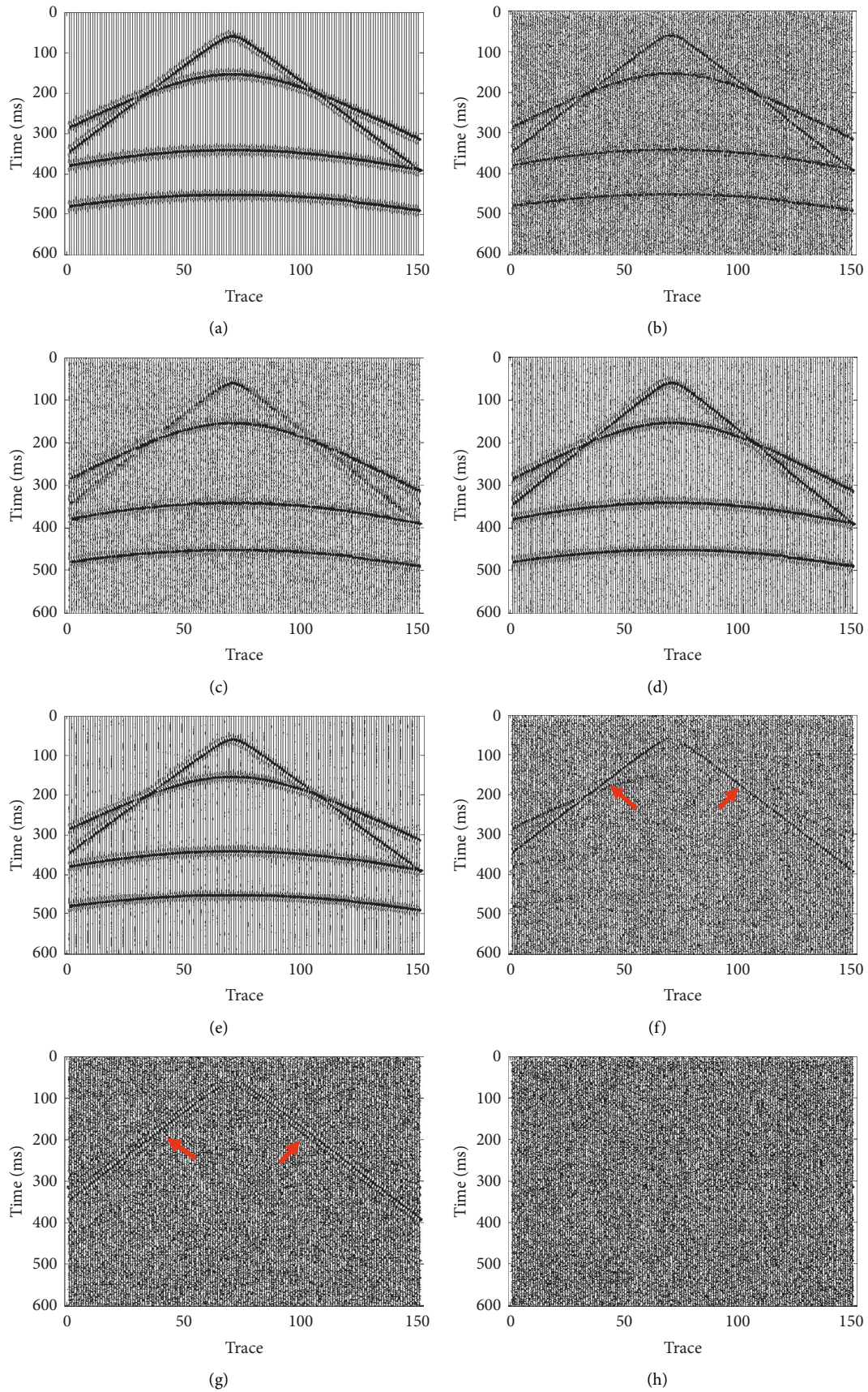


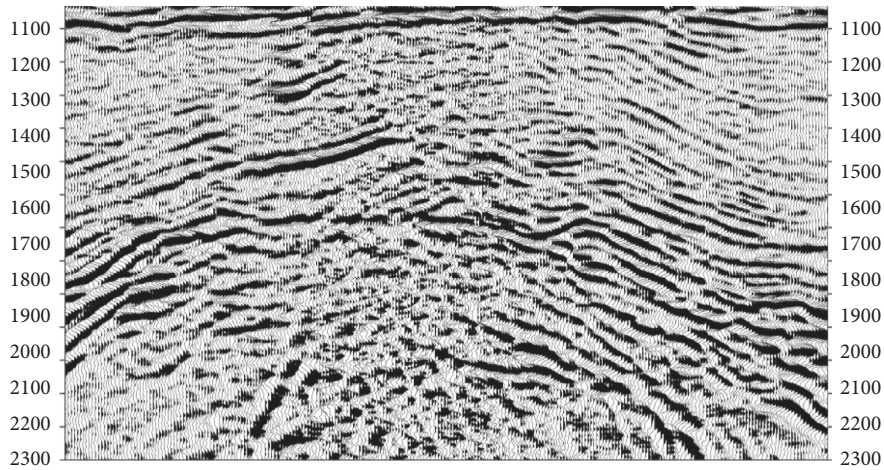
FIGURE 8: Processed results on synthesized data with noise for various approaches. (a) Noise-free. (b) Synthesized data with noise. Denoised data and removed noise by the wavelet-based method (c, f), the curvelet-based threshold denoising method (d, g), and our approach (e, h), respectively.

TABLE 1: Comparison of various SNRs before denoising and after denoising (dB).

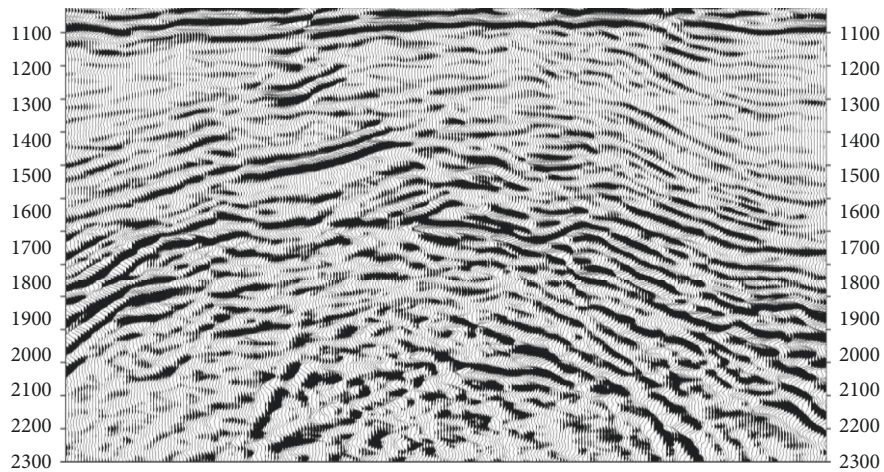
Noisy data	Wavelet-based denoising	Curvelet-based denoising	Our method
-8.2293	-3.1246	-0.4234	8.6838
-2.2063	5.8541	8.2496	13.2078
1.3116	7.7566	10.5394	15.9510
7.3169	11.9148	13.8784	19.4674

TABLE 2: Comparison of various SNRs before denoising and after denoising in the NSCT domain (dB).

Noisy data	Hard thresholding	Soft thresholding	Shrink thresholding
-6.4468	6.4325	6.8746	7.2314
0.4326	14.2376	14.3628	15.3071
5.6183	16.9283	17.1426	17.9552
8.6341	21.6875	22.4328	22.9457

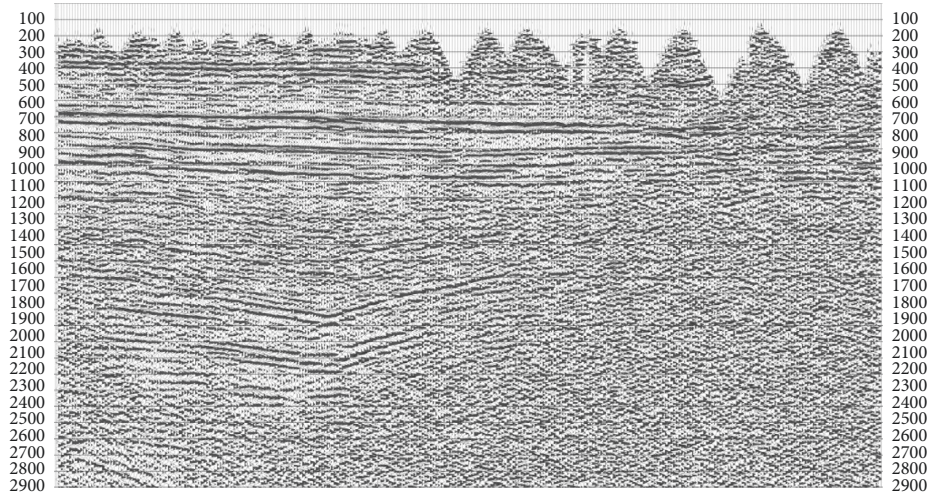


(a)

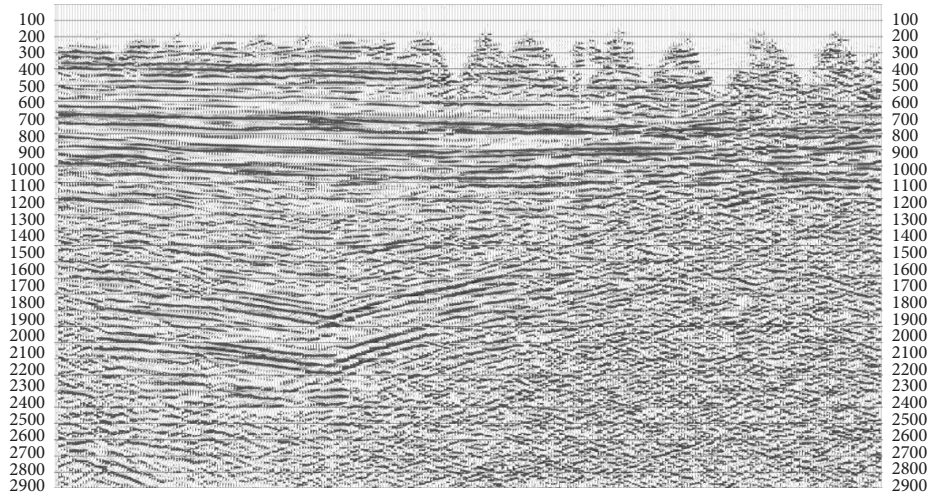


(b)

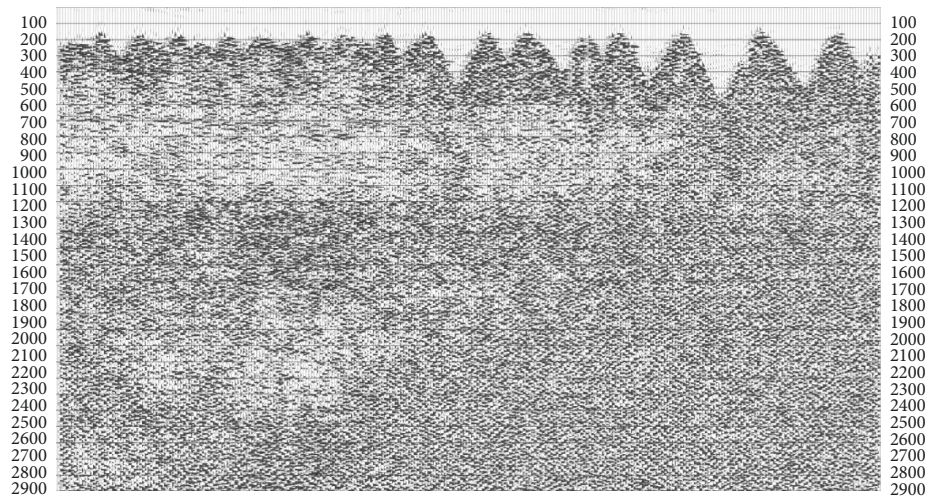
FIGURE 9: (a) Real migration profile. (b) Processed result by our approach.



(a)



(b)



(c)

FIGURE 10: (a) Real stacked profile. (b) Processed result by the proposed approach. (c) Noise removed by our approach.

$$\widehat{C}_{k,j}(x, y) = \begin{cases} 0, & \text{otherwise,} \\ \text{sgn}(C_{k,j}(x, y)) \cdot \left( |C_{k,j}(x, y)| - T(k, \delta_{k,j}) \right), & |C_{k,j}(x, y)| \geq T(k, \delta_{k,j}), \end{cases} \quad (7)$$

where  $\widehat{C}_{k,j}(x, y)$  and  $C_{k,j}(x, y)$  are the matrices of the coefficients after and before denoising in the NSCT domain, respectively.  $T(k, \delta_{k,j})$  denotes the adaptive Bayesian threshold

*Step 4.* Reconstructing the denoised seismic data by conducting an inverse NSCT on these denoised NSCT sub-bands.

The workflow of our method is figuratively presented in Figure 7, where a two-level NSCT is performed on synthetic data to yield a low-pass sub-band and a set of high-pass sub-bands, and two and four shearing directions are used for coarser and finer scale, respectively.

### 3. Experiments

In this section, we evaluate our proposed method with two classic sparse transform based methods (wavelet- and the curvelet-based thresholding schemes) on synthetic seismic data and field data. The hardware used in this experiment is Intel 6226R CPU @2.90 GHz processor, 93 GB memory and NVIDIA RTX 3090 24G graphics card. The software used is Matlab R2016b.

*3.1. Synthetic Seismic Example.* To demonstrate the performance, an example of hyperbolic-events synthetic data is used. Figure 8(a) presents the synthesized signals of 150 traces with 1 ms time sampling interval. The Ricker wavelet is expressed by

$$x(t) = \left(1 - 2\pi^2 f^2 t^2\right) \cdot e^{-\pi^2 f^2 t^2}. \quad (8)$$

Figure 8(b) is the corresponding noisy synthetic data, which is denoised by using two existing methods, namely, the wavelet- and the curvelet-based threshold denoising approach and the proposed approach. These three methods use the same threshold method. We set up  $K=2$  and  $J=2$  for the proposed approach. So, we can acquire the denoising threshold  $T(k, \delta_{k,j})$  by computing formulas (3-6) step by step; thus, we can further obtain the denoised high-frequency results by the soft-thresholding method. Figures 8(c)–8(e) show the obtained results by three methods, respectively. Obviously, the result of our approach is much superior to the ones by other two approaches. Concretely, the results are evaluated by using SNRs [26]:

$$\text{SNR} = 20 \cdot \log_{10} \frac{\|\mathbf{x}_0\|_2}{\|\mathbf{x}_1 - \mathbf{x}_0\|_2}, \quad (9)$$

where  $\mathbf{x}_0$  and  $\mathbf{x}_1$  represent the noise-free data and the noisy or denoised data, respectively. The resulted SNR values for Figures 8(b)–8(e) are  $-2.2063$  dB,  $5.8541$  dB,  $8.2496$  dB, and  $13.2078$  dB, respectively. Obviously, wavelet- and the curvelet-based approaches implement insufficient noise removal, while our approach conducts good performance in attenuating most random noise and the SNR value has been

significantly improved. Figures 8(f)–8(h) show the removed noise sections by these three approaches, respectively. The wavelet- and the curvelet-based approaches lose part of useful signals (red arrow). Obviously, the proposed approach does not harm any useful signals. Besides, Tables 1 and 2 present the summary of the results with various SNRs before and after denoising. Our approach shows the better denoising performance, especially for the low SNR seismic data.

In addition, to validate the processing result of our method, real noisy seismic data are measured with same excitation and reception in the identical data area. Figure 9 presents the acquired noisy signals (Figure 9(a)) and the denoising result with the proposed approach (Figure 9(b)). We can see that several highlighted effective signals, clearer interlayer structure, and improved event continuity can be observed from the patterns of the denoised data, which significantly improves the SNR value. Figure 10(a) shows the real stacked profile. Similarly, after processing using our proposed approach, effective signals are highlighted; information between layers is richer and noises are effectively suppressed, significantly improving the SNR (Figure 10(b)); it can be seen from the removed noise that there is basically no effective signal loss.

### 4. Conclusion

This article presents a novel NSCT-based seismic random noise denoising method. The superior performance NSCT with an appropriate thresholding operator brings excellent denoising results for seismic signals. The proposed method can achieve seismic random noise attenuation while retaining effective signals to the maximum degree. The experiments are performed with both synthesized and real seismic signals and the results demonstrate effectiveness of our approach compared with existing ones. In the future, we will consider deep learning based techniques to denoise seismic data with low SNR in view of the powerful learning ability and feature recognition ability, which aim to highlight effective signals and suppress false signals.

### Data Availability

The data included in this paper are available without any restriction.

### Conflicts of Interest

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

### References

- [1] L. L. Canales, "Random noise reduction," in *Proceedings of the 54th Annual International Meeting of SEG Technical Program Expanded Abstracts*, pp. 525–527, Atlanta, Georgia, December 1984.

- [2] J. H. Zhang and J. M. Lu, "Application of wavelet transform in removing noise and improving resolution of seismic data," *Journal of the University of Petroleum: Edition of Natural Science*, vol. 31, no. 12, pp. 1975–1981, 1997.
- [3] R. Neelamani, A. I. Baumstein, D. G. Gillard, M. T. Hadidi, and W. L. Soroka, "Coherent and random noise attenuation using the curvelet transform," *The Leading Edge*, vol. 27, no. 2, pp. 240–248, 2008.
- [4] Y. Chen and J. Ma, "Random noise attenuation byf-xempirical-mode decomposition predictive filtering," *Geophysics*, vol. 79, no. 3, pp. V81–V91, 2014.
- [5] B. Wang, R. S. Wu, X. Chen, and J. Li, "Simultaneous seismic data interpolation and denoising with a new adaptive method based on dremlet transform," *Geophysical Journal International*, vol. 201, no. 2, pp. 1182–1194, 2015.
- [6] Y. Chen and S. Fomel, "EMD-seislet transform," in *Proceedings of the 85th SEG Annual International Meeting*, pp. 4775–4778, New Orleans, LA, USA, October 2015.
- [7] W. Liu, S. Cao, Y. Chen, and S. Zu, "An effective approach to attenuate random noise based on compressive sensing and curvelet transform," *Journal of Geophysics and Engineering*, vol. 13, no. 2, pp. 135–145, 2016.
- [8] W. Liu, S. Cao, Y. Chen, and D. Zhang, "Application of variational mode decomposition in random noise attenuation and time frequency analysis of seismic data," in *Proceedings of the EAGE 78th Annual International Conference and Exhibition*, Houston, TX, USA, August 2016.
- [9] S. Beckouche and J. W. Ma, "Simultaneous dictionary learning and denoising for seismic data," *Geophysics*, vol. 79, no. 3, pp. A27–A31, 2014.
- [10] Y. Chen, "Fast dictionary learning for noise attenuation of multidimensional seismic data," *Geophysical Journal International*, vol. 209, no. 1, pp. 21–31, 2017.
- [11] D. Bonar and M. Sacchi, "Denoising seismic data using the nonlocal means algorithm," *Geophysics*, vol. 77, no. 1, pp. A5–A8, 2012.
- [12] Y. H. Yuan, Y. B. Wang, Y. K. Liu, and X. Chang, "Non-dyadic curvelet transform and its application in seismic noise elimination," *Chinese Journal of Geophysics*, vol. 56, no. 3, pp. 1023–1032, 2013.
- [13] Q. B. Zhou, J. H. Gao, Z. G. Wang, and K. X. Li, "Adaptive variable time fractional anisotropic diffusion filtering for seismic data noise attenuation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 4, pp. 1905–1917, 2016.
- [14] C. Lin, "Suppression of random noise in seismic data based on 3D Steerable Pyramid decomposition," *Progress in Geophysics*, vol. 33, no. 3, pp. 1081–1087, 2018.
- [15] Y. Sang, J. G. Sun, X. F. Meng, H. Jin, Y. Peng, and X. Zhang, "Seismic random noise attenuation based on PCC classification in transform domain," *IEEE Access*, vol. 8, Article ID 30368, 2020.
- [16] Y. Sang, J. G. Sun, D. Gao, and H. Wu, "Noise Attenuation Of Seismic Data Via Deep Multiscale Fusion Network," *Wireless Communications and Mobile Computing*, vol. 2021, no. 8, 8 pages, Article ID 6612346, 2021.
- [17] Y. Sang, B. Yu, Q. Wang, and S. Yin, "Suppressing seismic coherent and incoherent noise via deep neural network," in *Proceedings of the International Conference on Smart Internet of Things*, pp. 327–331, Jeju, Korea, August 2021.
- [18] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, Academic Press, Cambridge, MA, USA, 2008.
- [19] E. J. Candes and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewiseC2singularities," *Communications on Pure and Applied Mathematics*, vol. 57, no. 2, pp. 219–266, 2004.
- [20] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, 2005.
- [21] E. L. Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 423–438, 2005.
- [22] A. L. D. Cunha, J. Zhou, and M. N. Do, "The nonsubsampling contourlet transform: theory, design, and applications," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 3089–3101, 2006.
- [23] H. R. Shahdoosti and O. Khayat, "Image denoising using sparse representation classification and non-subsampling shearlet transform," *Signal, Image and Video Processing*, vol. 10, no. 6, pp. 1081–1087, 2016.
- [24] E. Levina and P. J. Bickel, "Maximum likelihood estimation of intrinsic dimension," *Advances in Neural Information Processing Systems*, vol. 17, pp. 777–784, 2005.
- [25] Y. Chen, Z. Ji, and C. Hua, "Spatial adaptive Bayesian wavelet threshold exploiting scale and space consistency," *Multidimensional Systems and Signal Processing*, vol. 19, no. 1, pp. 157–170, 2008.
- [26] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

## Research Article

# Partition KMNN-DBSCAN Algorithm and Its Application in Extraction of Rail Damage Data

Yujun Li <sup>1,2</sup>, Zhi Yang <sup>1</sup>, Shangbin Jiao <sup>1,2</sup> and Yuxing Li <sup>1,2</sup>

<sup>1</sup>School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, Shaanxi, China

<sup>2</sup>Shaanxi Province Complex System Control and Intelligent Information Processing Key Laboratory, Xi'an 710048, Shaanxi, China

Correspondence should be addressed to Zhi Yang; yangzhi1045549083@163.com

Received 24 May 2022; Revised 11 June 2022; Accepted 15 June 2022; Published 13 July 2022

Academic Editor: Saeid Jafarzadeh Ghouschi

Copyright © 2022 Yujun Li 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 order to realize intelligent identification of rail damage, this paper studies the extraction method of complete damage ultrasonic B-scan data based on the density-based spatial clustering of applications with noise algorithm (DBSCAN). Aiming at the problem that the traditional DBSCAN algorithm needs to manually set the Eps and Minpts parameters, a KMNN-DBSCAN (K-median nearest neighbor DBSCAN) algorithm is proposed. The algorithm first uses the dataset's own distribution characteristics to generate a list of Eps and Minpts parameters and then determines the optimal Eps and Minpts through an optimization strategy to achieve complete self-adaptation of the two parameters of Eps and Minpts. In order to further improve the clustering performance of the algorithm, the partition idea is introduced, and the partition KMNN-DBSCAN algorithm is proposed to solve the problem that the clustering results of the DBSCAN algorithm are inconsistent with the actual categories on datasets with uneven density. The experimental results show that the KMNN-DBSCAN algorithm has higher clustering accuracy and silhouette coefficient (SC) for the  $D_{037}$  dataset ultrasound information group (UIG) division; compared with the KMNN-DBSCAN algorithm, the proposed partition KMNN-DBSCAN algorithm has higher clustering accuracy, F-Measure, and SC values. The partition KMNN-DBSCAN algorithm achieves accurate division of all damage UIG on the damaged B-scan data with large density differences, and completes the effective extraction of complete damage data.

## 1. Introduction

Ultrasonic rail flaw detection vehicles are widely used in rail damage detection with the advantages of high detection sensitivity, good directionality, and accurate defect positioning [1]. The ultrasonic rail detection vehicle collects the damage B-scan data based on the multichannel ultrasonic probe, and the technicians classify the damage based on the damage B-scan data [2]. Rail damage identification is mainly divided into two parts: complete damage data extraction and damage identification. Effective extraction of damage data is an important prerequisite for accurate damage identification [3]. Complete damage data extraction refers to dividing the adjacent ultrasonic echo points in physical locations together to form an ultrasonic information group (UIG). UIG is the complete damage data and is the smallest unit of

damage identification [4]. The original B-scan data are stored in the form of data stream, which is difficult to extract manually. In this paper, the clustering algorithm is used to complete the damage data extraction.

Clustering algorithms are divided into partition-based clustering, hierarchical-based clustering, model-based clustering, and density-based clustering according to different clustering criteria [5]. The partition-based clustering must determine the final classification of the dataset before clustering, and the number of damages in the B-scan data file is uncertain, so the partition-based clustering is not suitable for rail damage classification [6]. The hierarchical-based clustering is only suitable for finding spherical or spherical clusters, while UIG is irregular in shape, so this method is not suitable for UIG division [7]. The model-based clustering algorithm assumes that the input dataset has a



potential probability distribution, and the clustering effect will be affected if the assumptions do not hold [8]. However, the distribution of B-scan data is random, so the model-based clustering algorithm is not suitable for UIG division of B-scan data. The density-based clustering algorithm uses density as the clustering criterion and can find clusters of any shape without presetting the number of clustering results. The representative algorithm is DBSCAN (density-based spatial clustering of applications with noise), which can filter out abnormal points as noise while clustering [9]. The DBSCAN algorithm satisfies the requirement that the number of damaged rails in ultrasonic B-scan data is unknown, the damage shape is irregular, and the noise needs to be filtered out. However, the traditional DBSCAN algorithm has two flaws. The first is that the neighborhood radius  $Eps$  and the minimum density threshold  $Minpts$  need to be manually set, which is prone to clustering failure. The second is that the traditional DBSCAN algorithm has errors when clustering datasets with large density differences. In view of the above two problems, some scholars have improved the DBSCAN algorithm. For example, the literature [10] proposed that the distance value corresponding to the region where the distance distribution curve of the input dataset “steepened” was taken as  $Eps$ . This method provides a criterion for determining  $Eps$  based on the characteristics of data distribution, which has certain guiding significance for the selection of  $Eps$ , but it needs to determine the “steepened” area through manual observation. The literature [11] proposes the AF-DBSCAN algorithm, which uses polynomial fitting to fit the distance curve corresponding to a certain  $K$  value in the input dataset, solves the inflection point of the fitted curve, and takes the maximum distance corresponding to the inflection point as the optimal value of  $Eps$ . However, the discussion on the selection of  $K$  value is lacking in the text. The literature [12] proposes the SA-DBSCAN algorithm, which uses inverse Gaussian fitting to fit the probability distribution curve of the dataset, and takes the distance value corresponding to the peak point of the distribution curve as the value of  $Eps$ . Under the same  $Eps$ , the noise points are the least when the corresponding  $Minpts$  are taken as the optimal value of  $Minpts$ . However, the SA-DBSCAN algorithm achieves complete adaptation of the two parameter values, but the algorithm makes assumptions about the distribution of the input dataset and is not applicable to all datasets. The literature [13–15] improves the DBSCAN algorithm based on the idea of grid division to achieve good clustering effect on datasets with uneven density distribution. However, the mesh size setting of the parameter adaptive algorithm based on mesh division lacks theoretical guidance and is difficult to set manually.

In view of this, this paper proposes a new parameter adaptive DBSCAN algorithm KMNN-DBSCAN, which automatically determines the optimal clustering parameters based on the distance distribution characteristics of the input dataset, and realizes the fully adaptive selection of  $Eps$  and  $Minpts$  parameters. After that, the partition idea was introduced to improve the KMNN-DBSCAN algorithm, and the partition KMNN-DBSCAN algorithm was proposed. The partition KMNN-DBSCAN algorithm is proposed to

solve the problem of errors in the traditional DBSCAN algorithm when clustering datasets with large density differences, and realizes the effective extraction of complete damage data.

## 2. Algorithm Principle

*2.1. Principle of DBSCAN Algorithm.* The DBSCAN (density-based spatial clustering of applications with noise) algorithm is a clustering algorithm based on high-density connected regions, which can filter out noise points while discovering any cluster. The related concepts of DBSCAN algorithm are as follows:

- (1) The  $Eps$  neighborhood of element points: for a certain element point  $p$  in a given dataset  $D$ , the  $Eps$  neighborhood of  $p$  refers to the set of all element points in the area with  $p$  as the center and  $Eps$  as the radius, denoted as  $Eps(p)$ .  $Eps(p)$  expression is as follows:

$$Eps(p) = \{q \in D | distance(p, q) \leq Eps\}, \quad (1)$$

where  $distance(p, q)$  is the Euclidean distance between the element point  $p$  and the element point  $q$  in the dataset  $D$ .

- (2) Density: DBSCAN defines the number of element points contained in the  $Eps$  neighborhood of an element point as the density of the point, and the expression is as follows:

$$Density = Num(Eps(p)). \quad (2)$$

- (3) Core point: if the  $Eps$  neighborhood of an element point contains the number of element points greater than or equal to the given minimum density threshold  $Minpts$ , the point is called a core point, and the expression is as follows:

$$Num(Eps(p)) \geq Minpts. \quad (3)$$

- (4) Boundary point: if the number of element points contained in the  $Eps$  neighborhood of an element point is less than the given minimum density threshold  $Minpts$ , the point is called a boundary point, and the expression is as follows:

$$Num(Eps(p)) < Minpts. \quad (4)$$

- (5) Direct density reachability: for any two element points  $p$  and  $q$  in dataset  $D$ , if  $q$  is in the  $Eps$  neighborhood of  $p$ , and  $p$  is the core point, then point  $q$  is said to be directly density reachable from point  $p$ .
- (6) Density reachable: for a set of element points  $p_1, p_2, \dots, p_i, \dots, p_n$  in dataset  $D$ , where  $p = p_1$ , and  $q = p_n$ . An element point  $q$  is said to be density-reachable from point  $p$  if it is directly density-reachable for any  $p_{i+1}$  to  $p_i$ .
- (7) Density connection: for an element point  $r$  in the dataset  $D$ , if the element point  $p$  and the element

point  $q$  are both density-reachable from point  $r$ , then point  $q$  is said to be density-connected from point  $p$ .

(8) Cluster ( $C$ ): for a nonempty subset  $C$  of the input dataset, if  $C$  meets the following conditions,  $C$  is called a cluster.

- (1) For any element point  $q$ , there is an element point  $p$  belonging to  $C$ . If  $q$  is density-reachable from  $p$ , then  $q$  belongs to  $C$ .
- (2) For any two element points in  $C$ , they are density connection.

Noise point: the element point that does not belong to any cluster is called noise point, denoted as noise, and the expression is as follows:

$$noise = \{o \in D | \forall i: o \notin Ci\}. \quad (5)$$

The clustering principle of DBSCAN algorithm is the maximum density connected sample set derived from the density reachability relationship [16].

**2.2. Principle of KMNN-DBSCAN Algorithm.** The core of KMNN-DBSCAN is to generate a list of Eps and Minpts parameter pairs based on the distance distribution characteristics of the dataset. The parameters are optimized based on the Eps and Minpts parameter lists, and the optimal parameter pair is obtained, so as to realize the complete adaptation of the Eps and Minpts parameters of DBSCAN. The algorithm flow of KMNN-DBSCAN is shown in Figure 1.

The steps of the KMNN-DBSCAN algorithm are as follows:

Step 1. Generate a list of Eps parameter values.

The Eps parameter list is generated by the K-median nearest neighbor (KMNN) algorithm. The main principle of this algorithm is to first calculate the K-nearest neighbor distance matrix of the input dataset and then find the median of the K-nearest neighbor distances of all element points, and constitute the K-median nearest neighbor set. Take the K-median nearest neighbor set as the Eps parameter list, and the specific steps are as follows:

- (1) Calculate the distance distribution matrix  $Dist_{n \times n}$

$$Dist_{n \times n} = \{dist(i, j) | 1 \leq i \leq n, 1 \leq j \leq n\}, \quad (6)$$

where  $n$  is the number of element points in the input dataset  $D$ ,  $dist(i, j)$  is the Euclidean distance between element point  $i$  and element point  $j$ , and  $Dist_{n \times n}$  is the distance matrix.

- (2) Calculate the K-nearest neighbor distance matrix  $KNN_{n \times n}$

$$KNN_{n \times n} = sort(Dist_{n \times n}). \quad (7)$$

Arrange each row of the distance distribution matrix in ascending order to obtain the K-nearest neighbor distance matrix  $KNN_{n \times n}$ . The  $i$ th row of

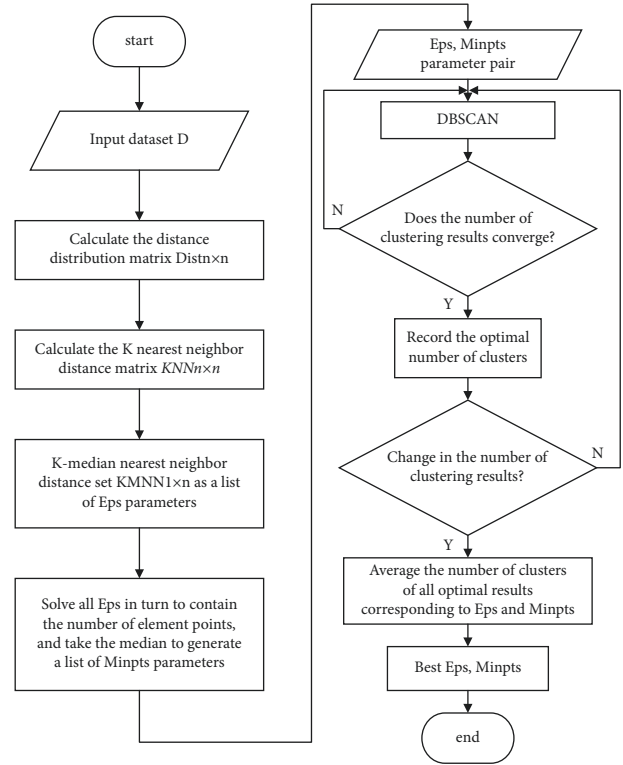


FIGURE 1: KMNN-DBSCAN algorithm flow.

$KNN_{n \times n}$  represents the distance between the element points closest to the  $i$ th element point  $1 \sim n$ , and the  $K$ -th column ( $K = 1, 2, \dots, n$ ) represents the  $K$ -nearest neighbor distance corresponding to all element points. Where the first column is the distance from the element point to itself, the first column of  $KNN_{n \times n}$  is all zero.

- (3) K-median nearest neighbor (KMNN) distance set  $KMNN_{1 \times n}$

Find the median of each column of  $KMNN_{1 \times n}$ , and all the medians form the K-median nearest neighbor distance set  $KMNN_{1 \times n}$ . Take the K-median nearest neighbor distance set  $KMNN_{1 \times n}$  as a list of Eps parameter values, where the Eps value corresponds to the  $K$ .

Step 2. Generate a list of Minpts parameter values.

Generates a list of Minpts parameter values using the median method and a given list of Eps parameter values. For the current Eps parameter list, the number of element points contained in the Eps neighborhood of all element points under different Eps is obtained in turn. The median of the number of element points contained in the Eps neighborhood of all element points is taken as the Minpts value corresponding to the current Eps value. The Minpts values corresponding to all Eps values are obtained to form the Minpts parameter list, and the Minpts values correspond to the Eps values. Different  $K$  corresponds to different Eps, Minpts parameter value pairs.

Step 3. Eps and Minpts parameter pair optimization.

The Eps and Minpts parameter value pairs corresponding to different  $K$  values are selected in turn as the clustering parameters of the DBSCAN algorithm, and the number of clustering results under the Eps and Minpts parameter values corresponding to different  $K$  values is obtained. This paper considers that when the number of clustering results does not change for three consecutive times, the number of clustering results is stable, and the current number of clustering results is recorded as the optimal number of clustering results. After the number of clustering results is stable, we continue the above operations until the number of clustering results changes. The interval from the optimal number of clustering results to the change of the number of clustering results is called the stable area. For the stable area corresponding to all the  $\bar{Eps}$  and  $\bar{Minpts}$  parameter value pairs, the expected  $\bar{EPS}$  and  $\bar{Minpts}$  are obtained respectively, and  $\bar{EPS}$  and  $\bar{Minpts}$  are taken as the optimal Eps and Minpts value.

**2.3. Partition KMNN-DBSCAN Algorithm Principle.** In order to solve the problem that the traditional DBSCAN algorithm has errors in the clustering results on datasets with large density differences, the idea of partitioning is introduced to improve the KMNN-DBSCAN algorithm [17], and the partition KMNN-DBSCAN algorithm is proposed. The implementation steps of the partition KMNN-DBSCAN algorithm are as follows:

Step 1. Partition the input dataset.

Project the element points of the input dataset on the  $X$ -axis and the  $Y$ -axis, respectively, to obtain the distribution of the input dataset on the  $X$ -axis and the  $Y$ -axis. The partition points are selected at the places where the density is the most sparse among different density centers, and sub-datasets with different densities are obtained.

Step 2. Perform cluster analysis on sub-datasets separately.

KMNN-DBSCAN is used to adapt the parameters to the sub-datasets, respectively, and the local optimal parameters are used to cluster the sub-datasets to obtain the clustering results of the sub-datasets.

Step 3. Merge local clustering results.

Merge the clustering results of the sub-datasets and finally get the clustering results of the original datasets.

The flow chart of the partition KMNN-DBSCAN algorithm is shown in Figure 2.

### 3. Ultrasound Information Group (UIG) Division Experiment

As shown in Figure 3, the complete damage data extraction for the original B-scan data includes the division of UIC and UIG, and UIG is the complete damage data.

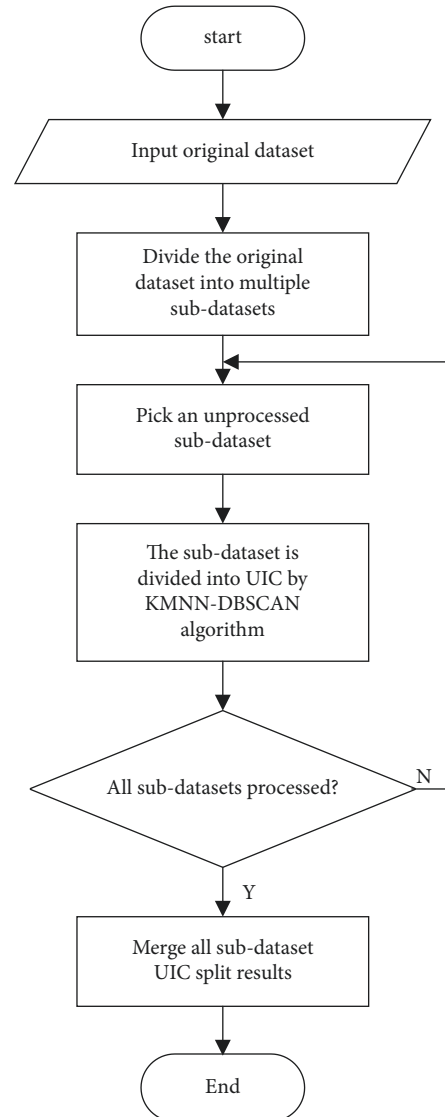


FIGURE 2: Flowchart of the partitioned KMNN-DBSCAN algorithm.

#### 3.1. Ultrasound Information Combination (UIC) Division.

Ultrasonic steel rail flaw detection vehicle adopts ultrasonic probe wheel as sensor. The ultrasonic probe wheel has built-in ultrasonic sensors with four angles:  $0^\circ$ ,  $37^\circ$ , straight  $70^\circ$ , and oblique  $70^\circ$ . Different angle sensors are used to detect different areas of the rail, and the combination of different ultrasonic sensor channels is used to detect different types of damage. If UIC is not divided, it will affect the complete data extraction of damage [18].

Figure 4 shows the detection range of ultrasonic sensors with different angles. The ultrasonic propagation range of the  $0^\circ$  probe is from the rail surface to the rail bottom, and the detection range is the projection area of the rail waist. The  $0^\circ$  probe is usually used to detect the level of crack from the rail surface to the rail bottom. The detection range of the  $37^\circ$  probe is from the rail surface of the rail waist projection area to the lower part of the rail waist. The  $37^\circ$  probe is usually used to detect the crack of the screw hole of the rail

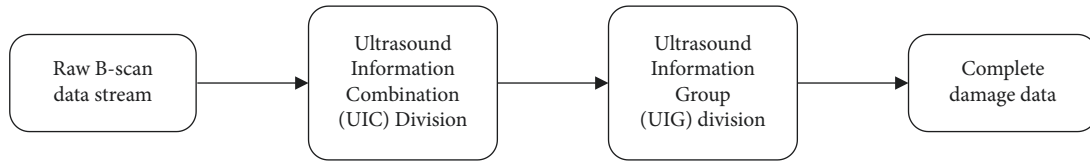


FIGURE 3: Complete damage data extraction process.

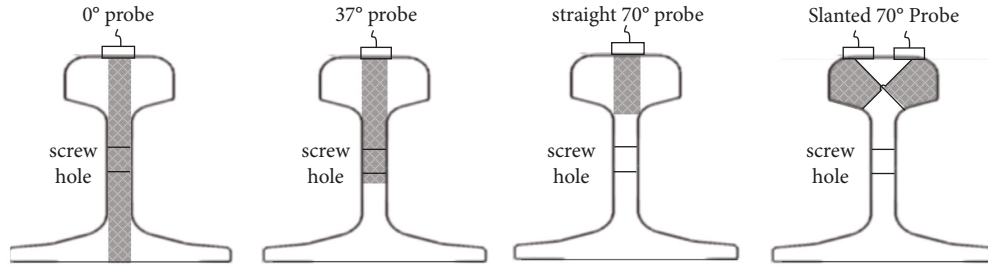


FIGURE 4: Detection range of ultrasonic sensors at different angles.

TABLE 1: Ultrasonic sensor combination corresponding to common rail damage.

Damage location	Damage type	Corresponding ultrasonic sensor combination
Rail surface	Oblique downward defect	37° probe
Rail head	Inner transverse hole	Slanted 70° probe
Rail head	Central nuclear injury	Straight 70° probe
Rail head	Outer lateral hole	Slanted 70° probe
Rail waist	Intact screw hole	0°, 37° probe
Rail waist	Horizontal cracks in screw holes	0°, 37° probe
Rail waist	Oblique crack in screw hole	0°, 37° probe
Rail bottom	Cross-hole	0° probe

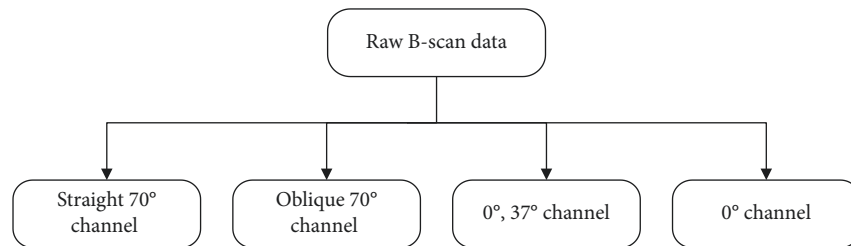


FIGURE 5: B-scan data UIC division.

waist. The detection range of the straight 70° probe is the central area of the rail head, which is generally used to detect nuclear damage in the middle of the rail head. The detection range of the oblique 70° probe is the area on both sides of the rail head and is generally used to detect the lateral holes on the inner and outer sides of the rail head.

As shown in Figure 4, there are related cases for sensors with different angles (such as 0° and 37° ultrasonic sensors), and there are also unrelated channels (such as 37° and oblique 70° ultrasonic sensors). If no distinction is made, the division of subsequent UIG will be affected, thereby affecting the effect of injury judgment. Table 1 shows the ultrasonic sensor combinations corresponding to seven common rail damages. From Table 1, it can be seen that different ultrasonic sensor combinations are used to detect different damages.

In this paper, the raw B-scan data are divided into UIC according to the ultrasonic sensor channel and position correlation, combined with the combination of ultrasonic sensor channels corresponding to each damage type. The UIC division results are shown in Figure 5.

3.2. *KMNN-DBSCAN Algorithm Applied to Ultrasonic Information Group (UIG) Division.* In order to verify the effectiveness of the KMNN-DBSCAN algorithm in this paper, the KMNN-DBSCAN algorithm is applied to the  $D_{037}$  UIC dataset for UIG division. The division results of the KMNN-DBSCAN algorithm are compared with those of the SA-DBSCAN and AF-DBSCAN algorithms.  $D_{037}$  is the 0° and 37° channel UIC data of the damage B-scan data collected in an experiment. The dataset contains 830 two-dimensional

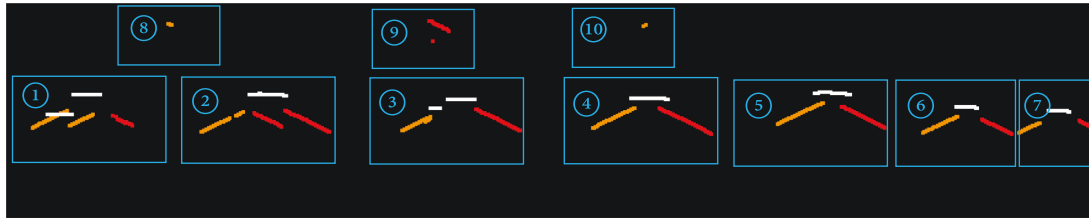


FIGURE 6:  $D_{037}$  dataset corresponds to B-scan.

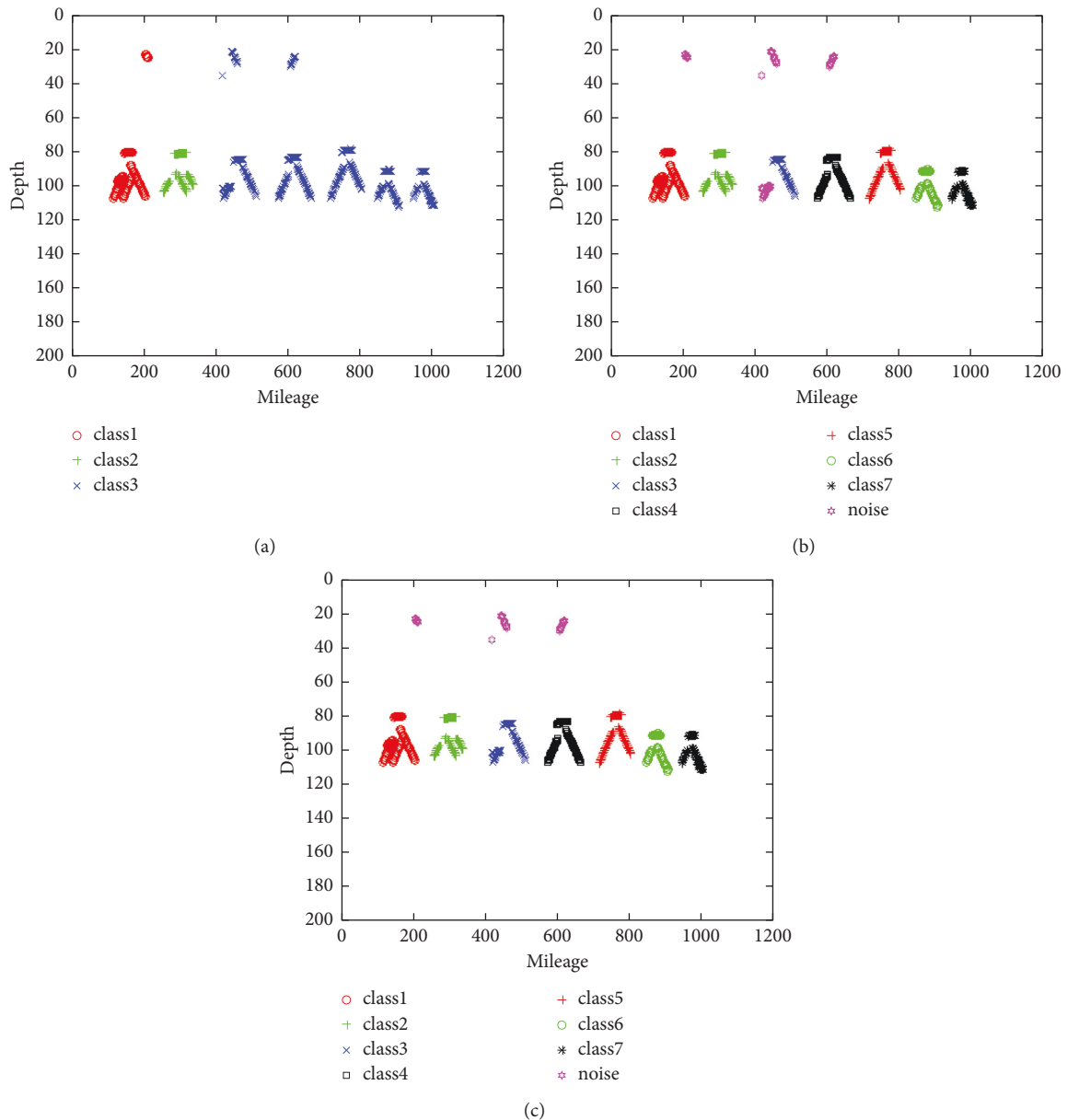


FIGURE 7: UIG division results of different parameter adaptive algorithms. (a) SA-DBSCAN. (b) AF-DBSCAN. (c) KMNN-DBSCAN.

ultrasonic echo data. Figure 6 shows the corresponding B-scan of the  $D_{037}$  dataset. The  $D_{037}$  dataset contains 10 damaged UIGs; that is, the  $D_{037}$  dataset contains 10 clusters.

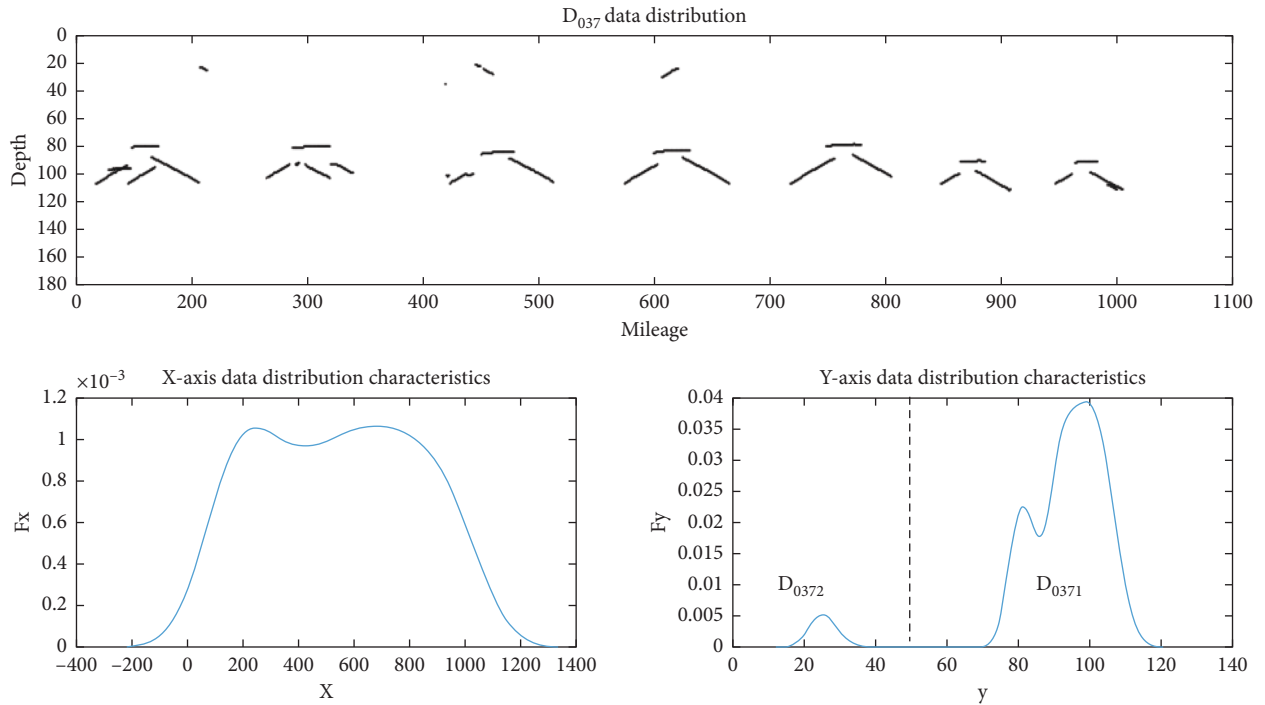
Figure 7(a)–7(c) are the UIG division results of the  $D_{037}$  dataset by the SA-DBSCAN algorithm, the AF-DBSCAN algorithm, and the KMNN-DBSCAN algorithm. All

elements with the same point type form a cluster class, elements with different point types represent different clusters, and noise points are points that do not belong to any cluster.

The UIG division results and evaluation indexes of different clustering algorithms are shown in Table 2.

TABLE 2: UIG division results and evaluation indexes of different clustering algorithms.

Algorithm name	Eps	Minpts	Actual number of classes	Number of clustering results	Number of correct clusters	F-Measure	SC
AF-DBSCAN	23.4	30.1	10	7	6	0.905	0.442
SA-DBSCAN	16.5	12.0	10	3	1	1	0.468
KMNN-DBSCAN	22.9	42.7	10	7	7	0.957	0.855

FIGURE 8: Data distribution characteristics of  $D_{037}$  dataset.

Among them, F-Measure reflects the ability of the algorithm to identify valid data points, and the larger the F value, the stronger the ability of the algorithm to identify valid data points [19]. The silhouette coefficient comprehensively reflects the compactness within the class and the separation between the classes. The value range of SC is  $[-1, 1]$ . Within the value range, the larger the value of SC, the better the clustering result [20].

Judging from the correct number of clusters in Table 2, the number of clustering results of the KMNN-DBSCAN algorithm is the closest to the real number of clusters in the  $D_{037}$  dataset, and the number of correct clusters is also the most, indicating that the KMNN-DBSCAN algorithm has the highest UIG division accuracy. From the F-Measure in Table 2, the F values of the AF-DBSCAN algorithm, the SA-DBSCAN algorithm, and the KMNN-DBSCAN algorithm are all above 0.9, indicating that the three algorithms have a strong ability to identify valid element points. According to the SC index in Table 2, the SC value of the UIG division result of the KMNN-DBSCAN algorithm is significantly higher than that of the comparison algorithm, indicating that the clusters divided by KMNN-DBSCAN have more compact elements within the cluster and more dispersed between different clusters, which shows that the UIG

division of the KMNN-DBSCAN algorithm is more reasonable.

By comparing the UIG division results of the KMNN-DBSCAN algorithm and the comparison algorithm in terms of the number of clustering results, the number of correct clusters, F-Measure, and SC indicators, it can be seen that the KMNN-DBSCAN algorithm proposed in this paper not only has strong identification efficiency. The ability of element points, and the accuracy and rationality of ultrasound information group division are better.

**3.3. Partition KMNN-DBSCAN Algorithm Applied to Ultrasound Information Group (UIG) Division.** The  $D_{037}$  dataset contains a total of 10 rail damage UIG. The KMNN-DBSCAN algorithm divides the  $D_{037}$  dataset into 7 clusters and fails to correctly divide all the ultrasound information clusters. This is because the densities of the UIG ⑧~⑩ and the UIG ①~⑦ in the  $D_{037}$  dataset are quite different. The KMNN-DBSCAN algorithm uses the same clustering parameters to process the datasets with large density differences, resulting in the UIG ⑧~⑩ is divided into noise. To solve this problem, this paper proposes a partition KMNN-DBSCAN algorithm. In order to verify the effectiveness of

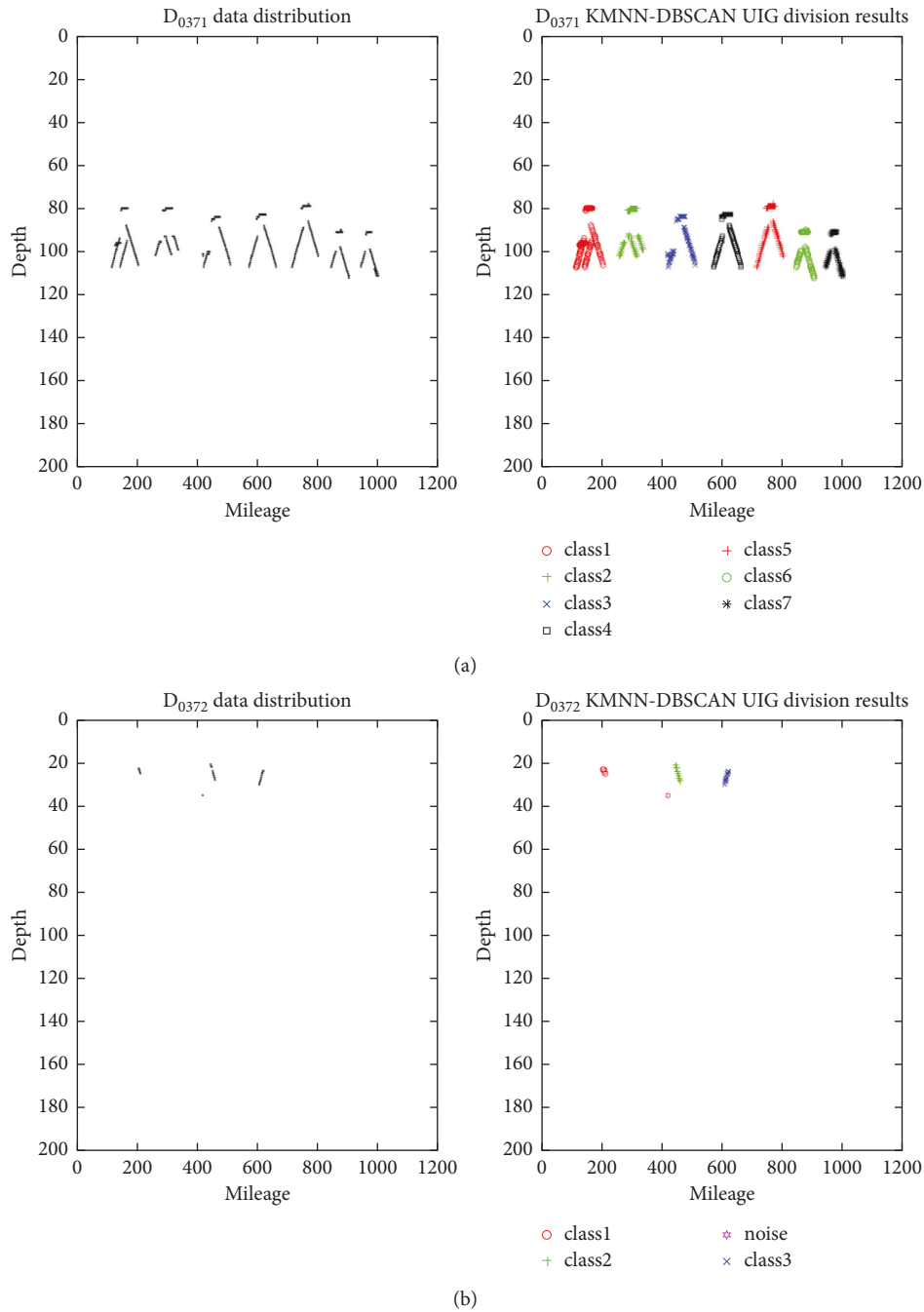


FIGURE 9: Sub-dataset KMNN-DBSCAN UIG division results. (a)  $D_{0371}$  dataset KMNN-DBSCAN UIG division results. (b)  $D_{0372}$  dataset KMNN-DBSCAN UIG division results.

the partition KMNN-DBSCAN algorithm, the algorithm was used in the  $D_{037}$  UIC dataset to divide ultrasound information groups, and the specific process is as follows:

- (1) Partitioning the  $D_{037}$  dataset based on the density distribution characteristics of the dataset.

The  $D_{037}$  data points are projected on the  $X$ -axis and the  $Y$ -axis, respectively, and the spatial distribution characteristics of the data element points of the  $D_{037}$  dataset on the  $X$ -axis and the  $Y$ -axis are shown in Figure 8.

As shown in Figure 8, the density of the  $D_{037}$  dataset is continuous on the  $X$ -axis, and there are two dense regions on the  $Y$ -axis, and the two densities are quite different. Where the density is the most sparse between the two density centers,  $D_{037}$  is divided into two sub-datasets of  $D_{0371}$  and  $D_{0372}$ .

- (2) The  $D_{0371}$  and  $D_{0372}$  datasets are divided into UIG by KMNN-DBSCAN.

Figure 9 shows the results of KMNN-DBSCAN dividing the  $D_{0371}$  and  $D_{0372}$  datasets by UIG.

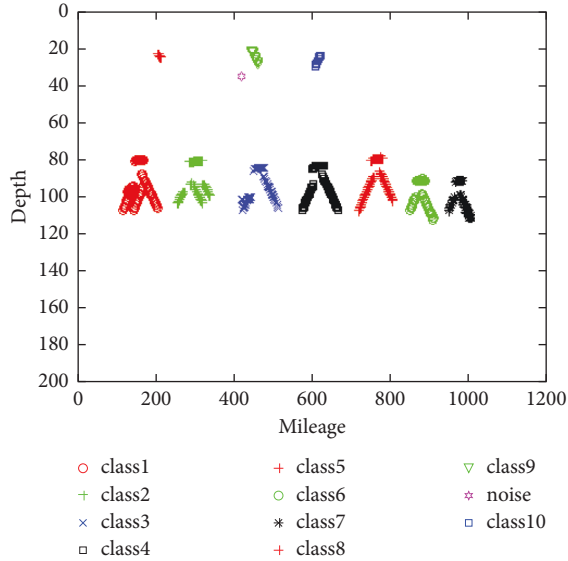


FIGURE 10: Partition KMNN-DBSCAN algorithm for UIG division of  $D_{037}$  dataset.

TABLE 3: Division results and evaluation indexes of UIG by partition KMNN-DBSCAN algorithm and KMNN-DBSCAN algorithm.

Algorithm name	Actual number of classes	Number of clustering results	Number of correct clusters	F-Measure	SC
KMNN-DBSCAN	10	7	7	0.957	0.855
Partition KMNN-DBSCAN	10	10	10	1	0.961

It can be seen from Figure 9 that the KMNN-DBSCAN algorithm is used to adapt the parameters of the two sub-datasets  $D_{0371}$  and  $D_{0372}$ , respectively, and the local optimal parameters are used for the division of the local UIG. Both sub-datasets realize the UIG accurate division.

(3) Merge the results of sub-dataset UIG division.

After completing the local clustering of each sub-dataset, all local clustering results should be merged to obtain the overall clustering result of the original dataset. By combining the division results of the UIG of the above two sub-datasets  $D_{0371}$  and  $D_{0372}$ , the result of dividing the UIG of the original input dataset  $D_{037}$  by partition KMNN-DBSCAN is shown in Figure 10.

Table 3 shows the comparison results of the UIG division of  $D_{037}$  by partition KMNN-DBSCAN and KMNN-DBSCAN. Judging from the number of clustering results and the number of correct clusters in Table 3, the partitioned KMNN-DBSCAN algorithm divides all UIGs of the  $D_{037}$  dataset correctly. From the F-Measure in Table 3, the F-Measure value of the partitioned KMNN-DBSCAN

algorithm is 1, which indicates that the partitioned KMNN-DBSCAN algorithm has identified all valid data points. From the SC index in Table 3, the SC value of the UIG division result of the partitioned KMNN-DBSCAN algorithm is significantly higher than that of the KMNN-DBSCAN algorithm. This shows that the clusters divided by the partitioned KMNN-DBSCAN have more compact elements in the cluster and more dispersed between different clusters; that is, the partitioned KMNN-DBSCAN algorithm UIG division is more reasonable.

## 4. Conclusion

In this paper, according to the characteristics of damage B-scan data, the density-based DBSCAN clustering algorithm is selected to study the extraction method of complete damage B-scan data. The main conclusions are as follows:

- (1) A parameter adaptive DBSCAN algorithm KMNN-DBSCAN is proposed, which solves the defect that the traditional DBSCAN algorithm needs to manually set the clustering parameters. KMNN-DBSCAN algorithm generates the list of Eps and Minpts parameter values according to the distance distribution characteristics of the input dataset, determines the optimal Eps and Minpts according to the relationship between the clustering results of the algorithm and the  $K$  under different parameter values, and realizes the complete adaptation of the two parameter values of Eps and Minpts.
- (2) The KMNN-DBSCAN algorithm is applied to the  $D_{037}$  dataset for UIG partitioning. The experimental results show that, compared with SA-DBSCAN algorithm and AF-DBSCAN algorithm, KMNN-DBSCAN algorithm has stronger ability to identify valid element points and performs better in the accuracy and rationality of UIG division.
- (3) The proposed partition KMNN-DBSCAN algorithm solves the problem that the traditional DBSCAN algorithm has errors in the clustering of datasets with uneven density distribution. The partition KMNN-DBSCAN algorithm is applied to the  $D_{037}$  dataset for UIG division. The experimental results show that the partition KMNN-DBSCAN algorithm can effectively realize the accurate UIG division, further improve the correctness and rationality of UIG division, and realize the effective extraction of complete damage data.

According to the principle of the partition KMNN-DBSCAN algorithm, the time complexity of the algorithm in this paper is relatively high. Therefore, how to effectively reduce the time complexity of the algorithm is the focus of our next work.

## 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 declared that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Key Scientific Research Program of Shaanxi Provincial Education Department (no. 20JY046), the Program of National Natural Science Foundation of China (no. 61871318), the Key Research Program of Shaanxi Province (no. 2021GY-259), and the Xi'an Science and Technology Project (21XJZZ0044).

## References

- [1] C. Sun, J. Liu, Y. Qin, and Y. Zhang, "Intelligent detection method for rail flaw based on deep learning," *Zhongguo Tiedao Kexue/China Railway Science*, vol. 39, no. 5, pp. 51–57, 2018.
- [2] C. Lu, Y. Wei, and W. Xu, "Study on B-scan imaging detection for rail tread tilted cracks using ultrasonic surface wave," *Yi Qi Yi Biao Xue Bao/Chinese Journal of Scientific Instrument*, vol. 31, no. 10, pp. 2272–2278, 2010.
- [3] X. Y. Huang, Y. S. Shi, and Y. H. Zhang, "BP neural network based on rail flaw classification of RFD car's B-scan data," *China Railway*, vol. 3, pp. 82–87, 2018.
- [4] C. Tastimur, M. Karakose, E. Akin, and I. Aydin, "Rail defect detection with real time image processing technique," in *Proceedings of the 2016 IEEE 14TH INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS (INDIN)*, pp. 411–415, IEEE, Poitiers, France, 19–21 July 2016.
- [5] M. Namratha, "A comprehensive overview of clustering algorithms in pattern recognition," *IOSR Journal of Computer Engineering*, vol. 4, no. 6, pp. 23–30, 2012.
- [6] Y. He, "Research and application of clustering algorithm based on partition," *Computer knowledge and technology*, vol. 13, no. 16, pp. 55–56, 2017.
- [7] Z. G. Chen and D. M. Zhu, "Hierarchic clustering algorithm used for anomaly detecting," *Procedia Engineering*, vol. 15, no. C, pp. 3401–3405, 2011.
- [8] B. Charles and B. S. Camille, "Model-based clustering of high-dimensional data: a review," *Computational Statistics & Data Analysis*, vol. 71, pp. 52–78, 2014.
- [9] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, AAAI Press, Germany, 1996.
- [10] J. Y. Song, Y. P. Guo, and B. Wang, "The parameter configuration method of DBSCAN clustering algorithm," in *Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI)*, no. 5, pp. 1062–1070, IEEE, Nanjing, China, 10–12 November 2018.
- [11] Z. P. Zhou, J. F. Wang, S. W. Zhu, and Z. Sun, "An improved adaptive and fast AF-DBSCAN clustering algorithm," *CAAI Transactions on Intelligent Systems*, vol. 11, no. 1, pp. 93–98, 2016.
- [12] L. N. Xia and J. W. Sa-DbSCAN, "A self-adaptive density-based clustering algorithm," *Journal of the Graduate School of the Chinese Academy of Sciences*, vol. 26, no. 4, pp. 530–538, 2009.
- [13] B. Thapana, A. Xiang, L. Yang, Z. Weihong, Z. Fuzhen, and H. Qing, "Grid-based DBSCAN: indexing and inference," *Pattern Recognition*, vol. 90, pp. 271–284, 2019.
- [14] S. Artur and C. Andrzej, "Grid-based approach to determining parameters of the DBSCAN algorithm," *Artificial Intelligence and Soft Computing*, vol. 24, pp. 555–565, 2020.
- [15] L. M. Zhang, Z. G. Xu, and F. Q. Si, "IEEE.GCMDDBSCAN: multi-density DBSCAN based on grid and contribution," in *Proceedings of the 2013 IEEE 11TH International Conference on Dependable, Autonomic and Secure Computing (DASC)*, pp. 502–507, IEEE, Chengdu, China, 21–22 December 2013.
- [16] J. B. Shen, X. P. Hao, Z. Y. Liang, Y. Liu, W. Wang, and L. Shao, "Real-time superpixel segmentation by DBSCAN clustering algorithm," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5933–5942, 2016.
- [17] S. G. Zhou, A. Y. Zhou, and J. Cao, "A data-partitioning-based DBSCAN algorithm," *Computer Research and Development*, vol. 37, no. 10, pp. 1153–1159, 2000.
- [18] W. Li and H. X. Zhang, "A FPGA based ultrasonic rail flaw detection system," in *Proceedings of the 2017 IEEE INTERNATIONAL SYMPOSIUM ON SIGNAL PROCESSING AND INFORMATION TECHNOLOGY (ISSPIT)*, pp. 150–155, IEEE, Bilbao, Spain, 18–20 December 2017.
- [19] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *Proceedings of the KDD Workshop on Text Mining*, p. 223, Boston, MA, USA, August 20–23, 2000.
- [20] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley, New York, 2009.

## Research Article

# Recognition of Persian/Arabic Handwritten Words Using a Combination of Convolutional Neural Networks and Autoencoder (AECNN)

Sara Khosravi  and Abdoloh Chalechale 

Department of Computer Engineering and Information Technology, Razi University, Kermanshah, Iran

Correspondence should be addressed to Abdoloh Chalechale; [chalechale@razi.ac.ir](mailto:chalechale@razi.ac.ir)

Received 24 January 2022; Revised 3 March 2022; Accepted 7 March 2022; Published 8 July 2022

Academic Editor: Ramin Ranjbarzadeh

Copyright © 2022 Sara Khosravi and Abdoloh Chalechale. 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.

Despite extensive research, recognition of Persian and Arabic manuscripts is still a challenging problem due to the complicated and irregular nature of writing, wide vocabulary, and diversity of handwritings. In Persian and Arabic words, letters are joined together, and signs such as dots are placed above or below letters. In the proposed approach, the words are first decomposed into their constituent subwords to enhance the recognition accuracy. Then the signs of subwords are extracted to develop a dictionary of main subwords and signs. The dictionary is then employed to train a classifier. Since the proposed recognition approach is based on unsigned subwords, the classifier may make a mistake in recognizing some subwords of a word. To overcome this, a new subword fusion algorithm is proposed based on the similarity of the main subwords and signs. Here, convolutional neural networks (CNNs) are utilized to train the classifier. An autoencoder (AE) network is employed to extract appropriate features. Thus, a hybrid network is developed and named AECNN. The known Iranshahr dataset, including nearly 17000 images of handwritten names of 503 cities of Iran, was employed to analyze and test the proposed approach. The resultant recognition accuracy is 91.09%. Therefore, the proposed approach is much more capable than the other methods known in the literature.

## 1. Introduction

Handwritten character recognition systems have recently been used in different areas such as recording and analyzing personal information and administrative forms, helping the blind to read, reading postal addresses of envelopes and sorting them out automatically, and processing bank checks [1]. A review of the literature indicates that recognition results of various printed texts of different languages such as Persian and Arabic have converged on an acceptable rate. However, the existing algorithms and applications are unable to achieve an acceptable accuracy on manuscripts. Thus, more studies should be conducted to design more efficient recognition algorithms.

Considering the available data, text recognition can be either online or offline. Offline recognition of manuscripts is more difficult than online recognition because the latter

benefits from different pieces of information such as the number of strokes, directions of strokes, writing speed in each stroke, and the pressure and duration of each stroke on a surface. These pieces of information are unavailable in offline recognition, which is based on the scanned image of a text [2, 3]. Diversity of writing styles and handwriting is another challenge in the recognition of manuscripts.

Generally, most of the conventional methods of text recognition include three major steps to design an efficient recognition system [4]: preprocessing the input words to enhance the quality of input raw data (through normalization, making images binary, noise elimination, dimension reduction, etc.), feature extraction (including 8-directional feature extraction [5], profiling [6], structural features [7], statistical and geometrical features, etc.), and finally classification through machine learning techniques such as support vector machine (SVM) [8, 9], fuzzy logic [10],

K-Means [11], neural networks (NNs) [12], integration of classifiers [13], and other evolutionary computing techniques. According to the machine vision approach, image classification is based on appearance and feature. In the feature-based method, classification is performed through the features extracted from images [14].

Most of the abovementioned methods are regarded as shallow learning techniques, which usually include a combination of extracted features and trainable classifiers. In such methods, the performance and efficiency of a recognition system depend on the researcher's creativity and innovation. Besides, finding an efficient method for a proper feature extraction has now become an essential issue in the image classification field. Unlike the automated feature extraction methods, most of the common feature extraction techniques are time-consuming and do not show satisfactory result.

On the contrary, there are deep learning (DL) methods [15–18], which have recently attracted many researchers conducting a plethora of studies on them. DL techniques are considered a subcategory of machine learning methods [19], which benefit from high-speed computer processors and a large amount of data to train large-scale NN and solve artificial intelligence problems. These algorithms differ from the abovementioned framework by providing an alternative solution. They have performed well in text recognition without using any feature extraction or preprocessing techniques. The distinct advantage of deep learning methods over shallow learning techniques is the automated extraction of features without considering the information. Therefore, DL methods can discover complicated structures and distinguish between large-scale datasets [20].

The first DL architectures for computer vision were implemented using artificial neural network (ANN) in the 1980s [21]. Recently, DL has been applied in many artificial intelligence (AI) applications; among its usages, we can mention emotion recognition [22, 23], coral classification [24], detecting robotic grasps [25, 26], natural language processing [27], biometric authentication based on finger knuckles and fingernails [28], brain tumor segmentation [29], and various usages in computer networks such as network traffic classification [30] and network intrusion detection systems [31]. The most important aspect of the DL is automatic feature extraction and using these features in the next layers of the network [32]. Some of exploited types are the autoencoders (AEs), the convolutional neural networks (CNNs), the recurrent neural networks (RNNs), the recursive neural networks, and the deep belief networks. This study focuses on the CNN and the AE.

The convolutional neural network is inspired by human learning strategy. The human brain recognizes objects visually whereby, examining similar images of an object, it obtains the ability to recognize objects it has not seen before. The ability to automatically extract important features of an object leads to high performance in CNN. Applications of CNNs include in-scene text recognition to obtain image encrypted information [33] and improving handwritten mathematical symbols classification [34]. Moreover, Steganalysis on JPEG images was introduced in [35]. This allows

identifying hidden embedding information without calculating the predefined image properties utilizing an experimental-based procedure.

An autoencoder is a special type of ANN which is used for optimal encoding in the learning process. Instead of training the network and predicting the target value for a particular input, an autoencoder rebuilds its inputs. Therefore, the output vector will have the same dimension as the input vector [36]. Among the usages of the autoencoders, we can mention deep feature learning for the medical image analysis [37], handwritten recognition [38, 39], and anomaly detection [40, 41].

In recent years, there has been research using a combination of CNNs and autoencoders. Dastider et al. [42] presented a framework to predict the disease severity of COVID-19 based upon the integration of CNNs and AEs. They considered both spatial and temporal features of the lung ultrasound (LUS) frames. Besides, deep convolutional autoencoder (CAE) has been proposed by Seyfioglu and Gurbuz [43]. This research is for radar-based activities recognition such as border security and control, pedestrian identification for automotive safety, and remote health monitoring.

The proposed recognition system is based on interconnected components. The words are first decomposed into their constituent subwords to enhance the recognition accuracy. Then the signs of subwords are separated to develop a dictionary of main subwords and signs. In the proposed system, a CNN is employed to categorize different classes. In such networks, the feature extraction of input images is performed automatically through the filter coefficient defined on the entire image. The filter coefficient of the entire image generates unnecessary and redundant features. Therefore, an autoencoder network is employed, prior to the CNN classifier, to extract effective features to improve the generated features. The CNNs are utilized to train the classifier. An autoencoder network is employed to extract appropriate and necessary features; thus, hybrid network was developed and named AECNN. After performing preprocessing and segmentation operations on a word, the recognition of subwords is done in the testing phase. Then the subwords are integrated. The integration of generated subwords should be based on how much the dictionary words resemble a corresponding word. A powerful algorithm is proposed for subword fusion based on the similarity to the error modification approach. Our contribution can be considered in three categories:

- (1) A combination of the CNNs and the AEs, which is called AECNN for the recognition of Persian and Arabic words.
- (2) Creating a dictionary of the main subwords and signs to increase the recognition accuracy.
- (3) Proposing a new subword fusion algorithm based upon the similarity of the main subwords and signs.

In this paper, Section 2 includes the writing features of Persian words. Section 3 addresses the proposed convolutional neural network and autoencoder based on deep

learning. Section 4 discusses the proposed feature extraction approach through an AE network, a CNN architecture, and an AECNN in full details. Section 5 deals with the research dataset. The results of simulating the proposed method are analyzed and compared with those of other studies in Section 6. Finally, Section 7 presents the conclusion and provides some strategies for future studies.

## 2. Writing Features of Persian/Arabic Words

Regarding the selection of appropriate methods for text recognition, it is necessary to learn Persian (or Arabic) writing rules. Hence, the following includes brief explanations of writing features in Persian/Arabic. For more clarity, in the following the English equivalents of Persian/Arabic words and characters are given in parentheses.

- (i) Unlike Latin texts, Persian/Arabic texts start from right to left and are written on the contour line, which is a horizontal line along the connected parts of the text having usually the largest number of word pixels.
- (ii) Persian consists of 32 characters, whereas Arabic contains 28 characters. Depending on the positions of characters, they appear in four different forms (at the beginning, in the middle, at the end, and separate). Table 1 indicates the different forms of Persian and Arabic alphabets with respect to their positions in words.
- (iii) In some writing styles of Persian texts, two or more letters are merged in a way where the resultant form looks nothing like the constituent letters. For instance, the combination of “ل” and “ا” is shown as “لا.”
- (iv) The only factor differentiating between the shapes of similar letters such as {“ث”, “ت”, “پ”, “ب”} (Se, Te, Pe, and Be), {“ح”, “خ”, “ج”, “چ”} (Khe, He, Che, and Jim), {“ذ”, “ز”} (Dal and Zal), and {“ر”, “ز”, “ز”} (Zhe, Ze, and Re) is the presence or absence of dots as well as the number of position of dots.
- (v) There are dots in 18 Persian characters and 15 Arabic characters. These dots may appear above or below the contour line. The number of these dots varies from one to three.
- (vi) In Persian/Arabic, words are divided into smaller units called subwords. In fact, these subwords are joined together to form the words. Figure 1 shows some samples of words with different subwords.
- (vii) Letters “ز”, “چ”, “پ”, and “گ” (Pe, Che, Zhe, and Gaf) exist in the Persian alphabet; however, the Arabic alphabets lack them.

## 3. Deep Networks

Deep learning is a subcategory of machine learning. The DL algorithm has a key role in object recognition. The following subsections address two deep learning techniques briefly.

**3.1. Convolutional Neural Networks.** Convolutional neural networks are known as one of the most famous deep

TABLE 1: Persian characters in four different positions.

Character	Characters positions			
	Beginning	Middle	End	Separate
1 (Alef)	ا	ا	ا	ا
2 (Be)	ب	ب	ب	ب
3 (Pe)	پ	پ	پ	پ
4 (Te)	ت	ت	ت	ت
5 (Se)	ث	ث	ث	ث
6 (Jim)	ج	چ	ج	ج
7 (Che)	چ	چ	چ	چ
8 (He)	ح	خ	ح	ح
9 (Khe)	خ	خ	خ	خ
10 (Dal)	د	د	د	د
11 (Zal)	ذ	ذ	ذ	ذ
12 (Re)	ر	ر	ر	ر
13 (Ze)	ز	ز	ز	ز
14 (Zhe)	ژ	ژ	ژ	ژ
15 (Sin)	س	س	س	س
16 (Shin)	ش	ش	ش	ش
17 (Sad)	ص	ص	ص	ص
18 (Zad)	ض	ض	ض	ض
19 (Ta)	ط	ط	ط	ط
20 (Za)	ظ	ظ	ظ	ظ
21 (Ayn)	ع	ع	ع	ع
22 (Ghayn)	غ	غ	غ	غ
23 (Fe)	ف	ف	ف	ف
24 (Ghaf)	ق	ق	ق	ق
25 (Kaf)	ک	ک	ک	ک
26 (Gaf)	گ	گ	گ	گ
27 (Lam)	ل	ل	ل	ل
28 (Mim)	م	م	م	م
29 (Noon)	ن	ن	ن	ن
30 (Waw)	و	و	و	و
31 (He)	ه	ه	ه	ه
32 (Ye)	ی	ی	ی	ی

learning methods. CNNs were developed by LeCun et al. earlier than two decades ago (1990) [44]. They were then developed by other researchers. A CNN is a multilayer neural network which can benefit simultaneously from an automated feature extractor and trainable classifier [45]. In recent years, CNNs have most widely been used in solving machine vision problems such as pattern recognition, object detection, or speech recognition [1]. They have also been widely used in Chinese handwritten text recognition [46]. Nevertheless, researchers are still trying to achieve a rapid and efficient training process and quick classification for CNNs.

CNNs outperform other deep learning methods in managing big data and sharing their weights [23]. These networks include a powerful learning technique by utilizing a multilayer structure and a specific type of supervised feedforward networks, in which layers are arranged differently for different applications. The back-propagation (BP) algorithm is usually employed to train and adjust the network parameters [47, 48].

Generally, a CNN consists of three major layers: convolutional layer, pooling layer, and fully connected layer. All of these layers are interconnected through weights. In this network, the convolutional layer and the pooling layer act as

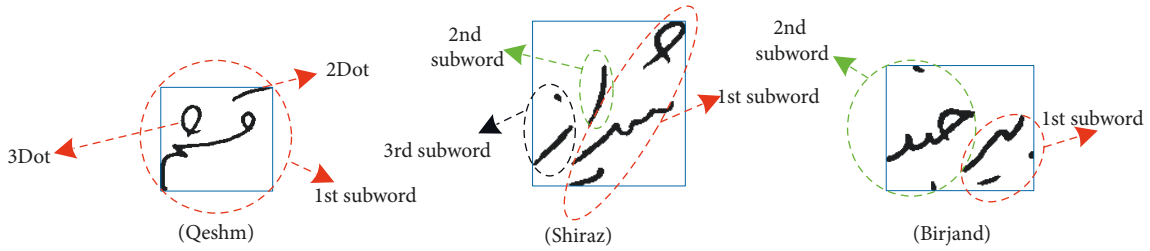


FIGURE 1: Some samples of words with different subwords.

feature extractors, whereas the fully connected layer acts as a classifier.

These layers consist of too many parameters, increasing the computing load and making the training process time-consuming. So far, some solutions have been provided for this problem. One solution is to reduce the number of connections in layers by employing basic CNN models such as GoogLeNet and AlexNet, in which the layer structure differs from the common type. The network depth can increase by adding more convolutional layers and benefiting from small convolutional filters [19].

There have also been some hybrid models, benefiting from the integration of shallow learning methods with deep learning techniques. In such models, features are usually extracted through a deep learning method. Then the differentiated features are given to a shallow learning technique such as an SVM for classification [19]. An application of such hybrid methods was discussed in the paper reviewed by [49]. In that paper, text detection was performed by integrating a CNN with RNN. In another study, a CNN was integrated with the Markov hidden model (HMM) for feature extraction in the recognition of handwritten words (based on two types of segmentation through letters and the sliding window) [50]. It was shown that features extracted through a CNN were more efficient than hand-crafted features in recognition. In the paper reviewed by [51], the particle swarm optimization (PSO) was integrated with a stochastic gradient descent (SGD), used as an alternative solution to optimize and enhance the network training process, based on the idea that the BP algorithm is prone to certain limitations on CNN training.

**3.2. Autoencoders.** An autoencoder is a type of neural network, in which the network input is the same as the network output. It also benefits from an unsupervised learning technique SGD for training [52, 53]. The autoencoder network consists of an encoder and a decoder. The encoder maps the input data  $x \in [0, 1]^{n+1}$  onto a latent coded space  $y \in [0, 1]^{m+1}$ , in which  $m$  is smaller than  $n$  most of the time. The encoder output is obtained through the following equation [53, 54]:

$$y = f(Wx + b). \quad (1)$$

In this equation,  $W$  and  $b$  are the weight matrix and mapping bias, respectively. Moreover,  $f$  is the activator function, which can be either linear or nonlinear. Then the decoder generates the reconstructed data  $\hat{x} \in [0, 1]^{n+1}$  by

mapping the latent coded space onto the main input space through the following equation:

$$\hat{x} = f(W'y + c). \quad (2)$$

In this equation,  $W'$  and  $c$  are named the weighted matrix and reverse mapping weight bias, respectively. The autoencoder tries to converge  $x$  on  $\hat{x}$  by adjusting weights and biases. The difference between  $x$  and  $\hat{x}$  is called the loss function, the value of which can be minimized to train the network [54].

For the retrieval of the input signal, an autoencoder should extract the important features of the input signal from the autoencoder output. Furthermore, if the number of autoencoder units is smaller than the input signal size, it is possible to present a compressed and low-dimensional display of the high-dimensional input signal [52].

#### 4. The Proposed Recognition System

Figure 2 shows a schematic view of the proposed recognition system, in which the training phase includes all of the images undergone through preprocessing steps including completion, noise elimination, and image size normalization.

The proposed recognition system is based on interconnected components. Since the dataset images are words, a segmentation step is required to divide words into interconnected components, which are hereinafter referred to as subwords. After this step, the subwords are labeled to generate a dictionary of subwords for network training (as shown in the following equation):

$$L = \{s_1, \dots, s_i, \dots, s_N\}. \quad (3)$$

Accordingly,  $s_i$  is the  $i$ th subword in the dictionary set  $L$ , and  $N$  indicates the total number of subwords.

In the proposed system, a CNN was employed to categorize different classes. In such networks, the feature extraction of input images is performed automatically through the filter coefficient defined on the entire image. The filter coefficient of the entire image generates unnecessary and redundant features. Therefore, an autoencoder network was employed, prior to the CNN classifier, to extract effective features to improve the generated features. The dimensions of features generated by the autoencoder are smaller than those of the input image. These features also contain pixel information.

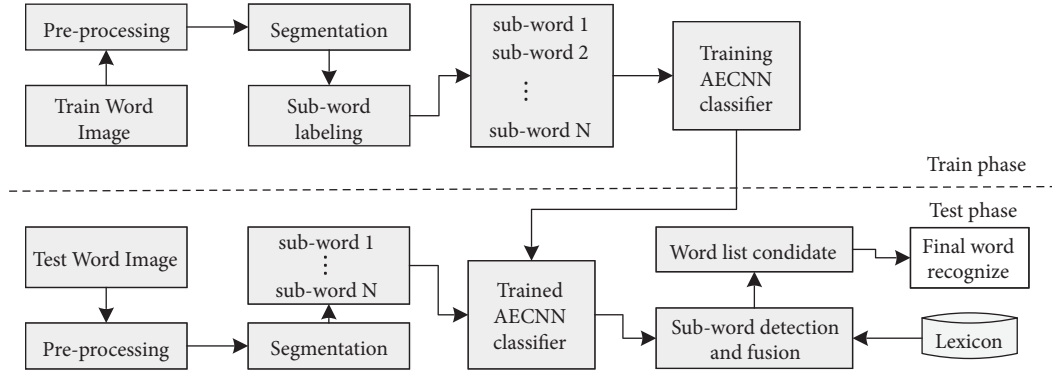


FIGURE 2: The block diagram of the proposed system.

$$\forall (0 < x < w, 0 < y < h) P(x, y) \in \{0, 1\}, \quad (4)$$

$$\forall (0 < x' < w', 0 < y' < h') F(x', y') \in [0, 1]. \quad (5)$$

In equation (4),  $P(x, y)$  indicates the value of pixel  $(x, y)$  of a binary image pertaining to a subword in  $w \times h$  dimensions. The pixel value is either one or zero. In fact, the pixel shows either the background (zero) or a subword outline (one). Equation (5) states the image output after being operated by the autoencoder in the  $w' \times h'$  dimensions. Furthermore,  $F(x', y')$  shows the value of pixel  $(x', y')$  ranging in the continuous span  $[0, 1]$  ( $h' < h$  and  $w' < w$ ).

After performing preprocessing and segmentation operations on a word, the recognition of subwords is done in the testing phase. Then the subwords are integrated. The integration of generated subwords should be based on how much the dictionary words resemble a corresponding word. In the proposed method, the recognized subwords are divided into main subwords and signs through the two following equations:

$$\text{SignSW} = \{\text{Dot}, \text{Zigbar}, \text{Hamze}, \text{Mad}\} \quad (6)$$

$$\text{MainSW} = U - \text{SignSW}. \quad (7)$$

In these equations,  $\text{SignSW}$  is the set of sign subwords including Dot, Zigbar, Hamza, and Mad. Moreover,  $\text{MainSW}$  is the set of main subwords including all of the subwords ( $U$ ) except for the sign subwords.

After dividing subwords into main and sign subwords based on the number of main subwords, it is necessary to determine how similar the combination of recognized subwords is to the dictionary words in order to introduce the most similar word/words. If several words have the highest rate of similarity, the similarity of secondary subwords is taken into account to reach the final decision, resulting in a unified output. All of the previous steps are discussed in detail here.

**4.1. Preprocessing.** Preprocessing is an important step in recognition systems. It prepares images for the next steps and puts them in the same conditions. The preprocessing step includes the following operations.

**4.1.1. Image Completion.** The dataset images have white backgrounds, although the background should be black in image processing operations when objects (subword outlines) are white. For this purpose, the supplementary image is determined through the following equation:

$$P'(x, y) = \begin{cases} 1 & \text{if } P(x, y) = 0 \\ 0 & \text{if } P(x, y) = 1. \end{cases} \quad (8)$$

In the above equation,  $P(x, y)$  and  $P'(x, y)$  show the main image and supplementary image, respectively.

**4.1.2. Noise Elimination.** There are usually tiny noise points on scanned images. They should be eliminated. The area filter of 10-pixel threshold was employed to eliminate these unwanted objects.

**4.1.3. Size Normalization.** The recognition system input images should be of the same size. Since the proposed system is based on subwords, each of them was normalized to the same size ( $50 \times 50$ ) after word segmentation and generation of subwords.

**4.1.4. Contour Line Detection and Gradient Modification.** In Persian, the contour line is a line on which letters and words are written and signed. Sometimes, the position of certain signs (such as dots) to the contour line can change the meaning of a word or even a sentence. For instance, different positions of a dot can completely change the meanings in “برو” meaning “go” and “ورن” meaning “do not go.” In Persian handwritten texts, contour line detection is usually done at the level of sentences because the word-level contour line detection is usually prone to errors.

This paper employed a method resembling the one used by [55] based on the horizontal histogram profile with a few changes to minimize the detection error. First, a moving-average filter of length  $L$  ( $L = 7$ ) was utilized to soften the horizontal histogram. Given the fact that Persian words are written on the contour line, the local maximums of the selected horizontal histogram are on the contour line. However, these local maximums, usually existing nearly above or below the image, are sometimes not on the contour line in Persian handwritten words. Figure 3 shows some

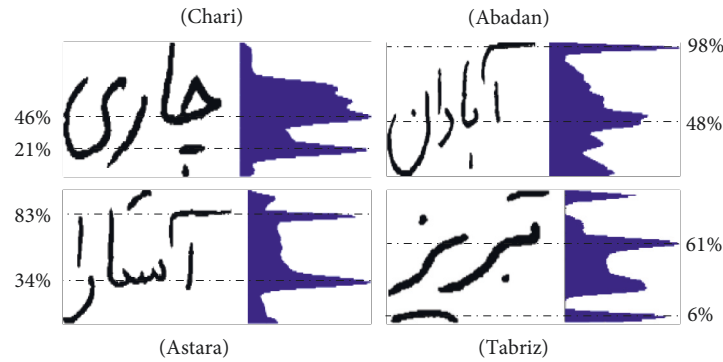


FIGURE 3: Incorrect and correct contour lines based on the maximum histogram.

samples of Persian handwritten words on the right and wrong contour lines based on the maximum histogram.

To avoid selecting an inappropriate contour line, it should be ensured that the maximum histogram is not placed 25% above or below the image. If the maximum value exists in these areas, the next maximum value is checked until the correct contour line is finally obtained. In other words, the maximum histogram is considered from 25% to 75% in the middle of the contour line image.

An unwanted gradient sometimes appears when words are written or images are scanned. Like the method used in [55], the images were rotated from  $-5$  to  $+5$  degrees with an angle of  $.5$  degrees to modify the unwanted gradient. Then the contour line histogram was determined. The largest histogram shows the best rate of rotation, which should be applied to images.

**4.2. Subword Segmentation and Labeling.** For subword segmentation and labeling, first the eight-connected neighboring method is used to identify and divide connected objects. Next, signs are removed from the connected components. After removing the signs, the similar ones are labeled by their representatives. For example, subwords such as “ب”, “پ”, “ت”, and “ث” (Be, Pe, Te, and Se) have the same outline and after removing of the dots are categorized in the group “ب.”

The rules of labeling the proposed interconnected components are as follows:

- (1) The beginning, middle, ending, and separate “ب”, “پ”, “ت”, and “ث” (Be, Pe, Te, and Se) are put in the “ب” (Be) group.
- (2) The beginning and middle “ن” and “ی” (Noon and Ye) are put in the “ب” (Be) group.
- (3) The beginning, middle, ending, and separate “ح”, “ج”, “خ”, and “گ” (He, Jim, Che, and Khe) are put in the “ح” (He) group.
- (4) The beginning, middle, ending, and separate “د” and “ذ” (Dal and Zal) are put in the “د” (Dal) group.
- (5) The beginning, middle, ending, and separate “ر”, “ز”, and “ژ” (Re, Ze, and Zhe) are put in the “ر” (Re) group.
- (6) The beginning, middle, ending, and separate “س” and “ش” (Sin and Shin) are put in the “س” (Sin) group.

- (7) The beginning, middle, ending, and separate “ص” and “ض” (Sad and Zad) are put in the “ص” (Sad) group.
- (8) The beginning, middle, ending, and separate “ط” and “ظ” (Ta and Za) are put in the “ط” (Ta) group.
- (9) The beginning, middle, ending, and separate “ع” and “غ” (Ayn and Ghayn) are put in the “ع” (Ayn) group.
- (10) The beginning and ending “ف” and “ق” (Fe and Ghaf) are put in the “ف” (Fe) group.
- (11) The beginning, middle, ending, and separate “ک” and “گ” (Kaf and Gaf) are put in the “ک” (Kaf) group.
- (12) The rest of Persian letters act as their beginning, middle, ending, and separate labels.
- (13) A dot, two dots, and three dots are put in the “1D”, “2D”, and “3D” groups, respectively.
- (14) The separated upper parts of “ک” and “گ” (Kaf and Gaf) are put in the “1ZB” and “2ZB” groups, respectively.
- (15) The Mad is put in the “Mad” group.
- (16) The Hamza is put in the “HZ” group.
- (17) The subword “ل” (La) is written like لا (La) by some people. Therefore, they are put in two groups of “ل” (La), one with “لا” (La) and the other one with two subwords “لا” (Lam) and “ا” (Alef).

According to the 17th rule, words with “ل” (La) such as “آق قلا” (Agh ghala) can be considered in two modes (“آق قلا” and “آق قلا”). Therefore, the number of words increased from 503 to 531 in the dataset. Table 2 shows the rules of labeling the interconnected components for the alphabet.

After applying the labeling rules to the interconnected components of 531 words in the dataset, 328 classes of subwords were obtained with different samples. These images were employed to train the classifier. The secondary classes included “1D”, “2D”, “3D”, “ZB”, “Mad”, and “HZ” known as the sign subwords. Table 3 shows the subwords of different written forms of “پولادشهر” (Poladshahr) along with some other sample images.

**4.3. The Dataset of Main and Sign Subwords.** After labeling the interconnected components and creating different classes of subwords, the dataset of main subwords and

TABLE 2: The labeling rules of the proposed interconnected components.

Characters	Characters positions				Label
	Beginning	Middle	End	Separate	
ا (Alef)	✓	✓	✓	✓	ا (Alef)
ب (Be)	ب (Pe)	✓	✓	✓	ب (Be)
ت (Te)	ث (Se)	✓	✓	✓	ب (Be)
ن	ي	✓	✓		ب (Be)
ح (He)	ج (Jim)	✓	✓	✓	ح (He)
چ (Che)	خ (Khe)	✓	✓	✓	ح (He)
د (Dal)	ذ (Zal)	✓	✓	✓	د (Dal)
ر (Re)	ز (Ze)	✓	✓	✓	ر (Re)
ژ (Zhe)		✓	✓	✓	ر (Re)
س (Sin)	ش (Shn)	✓	✓	✓	س (Sin)
ص (Sad)	ض (Zad)	✓	✓	✓	ص (Sad)
ط (Ta)	ظ (Za)	✓	✓	✓	ط (Ta)
ع (Ayn)	غ (Ghayn)	✓	✓	✓	ع (Ayn)
ف (Fe)	ق (Ghaf)	✓	✓		ف (Fe)
ف (Fe)				✓	ف (Fe)
ق (Ghaf)			✓	✓	ق (Ghaf)
ک (Kaf)	گ (Gaf)	✓	✓	✓	ک (Kaf)
م (Mim)		✓	✓	✓	م (Mim)
ن (Noon)			✓	✓	ن (Noon)
و (Waw)		✓	✓	✓	و (Waw)
ه (He)		✓	✓	✓	ه (He)
ی (Ye)			✓	✓	ی (Ye)

signs was generated. This dataset can be employed to integrate subwords and recognize the final word. For the generation of the main subword dataset, all of the words should be labeled by the rules of labeling the proposed interconnected components, discussed in the previous section. Regarding the sign subword dataset, it is necessary to determine the word under a sign and the position of the word to the contour line. The positions of “ZB,” “Mad,” and “HZ” are not important. The positions of sign subwords are checked to the contour line only for “1D,” “2D,” and “3D” classes. The signs placed below the contour line are labeled “D,” and those placed above the contour line are labeled “U.”

The rules of codifying the dataset of sign subwords are as follows:

- (1) The letters “ن,” “ط,” “خ,” “غ,” “ف,” “ض,” “ز,” and “ذ” (Noon, Zad, Fe, Ghayn, Khe, Za, Ze, and Zal) are labeled as “1D-U.”
- (2) The beginning, middle, and ending “ب” and “ج” (Be and Jim) are labeled as “1D-D.”
- (3) The beginning, middle, and ending “ق” and “ت” (Ghaf and Te) are labeled as “2D-U.”
- (4) The beginning and middle “ی” (Ye) are labeled as “2D-D.”
- (5) The beginning, middle, and ending “ث” and “ش” (Se and Shin) are labeled as “3D-U.”
- (6) The beginning, middle, and ending “چ” (Che) are labeled as “3D-D.”

Table 4 shows main subwords and signs of some samples in the dataset. Table 5 exhibits the number of main subwords of all the words existing in the dataset.

**4.4. The Autoencoder Network Used as the Feature Extractor.** As discussed earlier, autoencoder networks consist of an encoder and a decoder. They are employed to reconstruct the input image in the decoder output by adjusting weights and network biases. They are classified as unsupervised network training. The interesting feature of these networks is that they first create a compressed and coded vector of the image, based on which the input image is constructed. This vector can be used as the feature vector containing appropriate locational-pixel information in the CNN classifier. Since the CNN input is an image, the generated feature vector should be transformed into a 2-dimensional vector so that it can be injected into the CNN.

The proposed autoencoder network has 625 neurons in the encoder layer and an activator function in the rectified linear unit (ReLU). It also has 2500 neurons (the number of pixels on the input image) in the decoder layer and a sigmoid activator function. The generated feature vector has 625 components in the encoder output, which will be of size  $25 \times 25$  after being transformed into a 2-dimensional form. The SGD algorithm was employed along with the MSE function to train the autoencoder network. All of the training images of subwords, generated in the previous section, were utilized as the autoencoder network input and output. The training algorithm adjusts the network weights and biases to converge the network output on the network input. After training the network, the autoencoder output includes the feature vector and new training images used in the CNN. Figure 4 shows the proposed autoencoder network along with the input, output, and coded images.

**4.5. CNN Architecture.** Regarding the architecture of the network used in this study, it can be stated that the first layer is a convolutional layer including 32 filters of size  $4 \times 4$  and



TABLE 3: The resultant subwords of "پولادشهر."



















Subwords	Class	Sample		
Dots	1D			
	2D			
	3D			
پو (Po)	پو			
ال (la)	ال			
	ل			
لا (la)	ا			
د (d)	د			
شهر (shaher)	هرس			

TABLE 4: Main subwords and signs of some samples in the dataset.

Word	Main subword	Sign subword
آبیک (Abyek)	ب، پ، ک، ا	Mad, 1D-D, 2D-D, 1ZB
بیرجند (Birjand)	ب، پ، ر، ح، ب، د	1D-D, 2D-D, 1D-D, 1D-U
کرم‌ان‌شاه (Kermanshah)	ک، ر، م، ا، ب، س، ا، ه	1ZB, 1D-U, 3D-U
گرگان (Gorgan)	ن، ک، ر، ک، ا	2ZB, 2ZB, 1D-U
یزد (Yazd)	ب، ر، د	2D-D, 1D-U

TABLE 5: The number of main subwords in the dataset.

Number of main subwords	Number of words
1	36
2	156
3	169
4	121
5	38
6	8
7	3
Number of all words	531

images of size  $25 \times 25$ , which is the very autoencoder output. This layer employs ReLU to eliminate the negative values ( $C_1$ ). The pooling layer of  $2 \times 2$  window size and MaxPooling function is put after it ( $P_1$ ). Then another convolutional layer including 32 filters of size  $2 \times 2$  and ReLU ( $C_2$ ) are put along with a pooling layer in the same way as the previous step ( $P_2$ ). Softmax is an activator function employed to generate the output classes at the end of a fully connected layer with 328 neurons and ReLU (FC). Then the output layer of 328

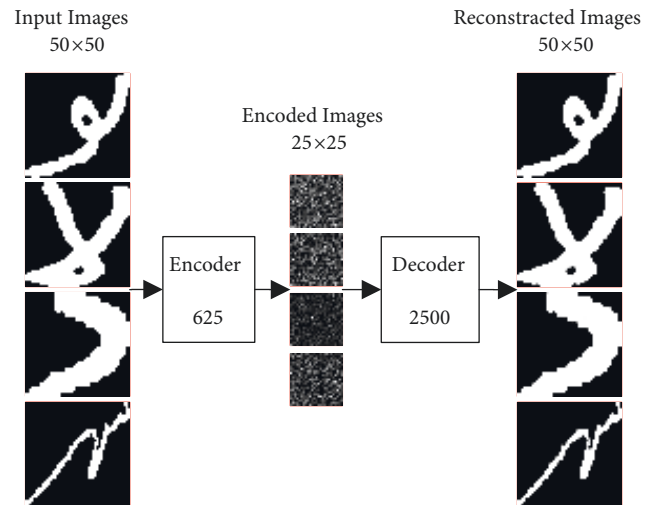


FIGURE 4: Feature extraction through the proposed autoencoder network.

neurons was used for different classes of subwords. Tables 6 and 7 show the local and global parameters of the proposed CNN, and Figure 5 indicates its architecture.

**4.6. Subword Classification Using the AECNN.** The proposed AECNN was employed to classify different subwords. This network consists of an AE network and a CNN in a subsequence. It performs feature extraction and classification simultaneously. First, all of the training images of subwords are put into the AE network. After training this network and reaching an appropriate MSE, the encoder output vector is

TABLE 6: The local parameters of the proposed CNN.

Layers	Filter no.	Filter size	Input layer size	Output layer size	Activation function	No. of parameters
Convolution ( $C_1$ )	32	4×4	25×25	22×22	ReLU	544
Pooling ( $P_1$ )	32	2×2	22×22	11×11	MaxPooling	0
Convolution ( $C_2$ )	32	2×2	11×11	10×10	ReLU	160
Pooling ( $P_2$ )	32	2×2	10×10	5×5	MaxPooling	0
Fully connected ( $FC$ )	328	1×1	5×5	328×1	ReLU	262728
Output layer	328	1×1	328×1	328×1	Softmax	107912

TABLE 7: The global parameters of the proposed CNN.

Parameters	Value
Image size	25*25
Optimizer	SGD
Batch size	200
Max epochs	40
Learning rate	0.001
Output classes	10

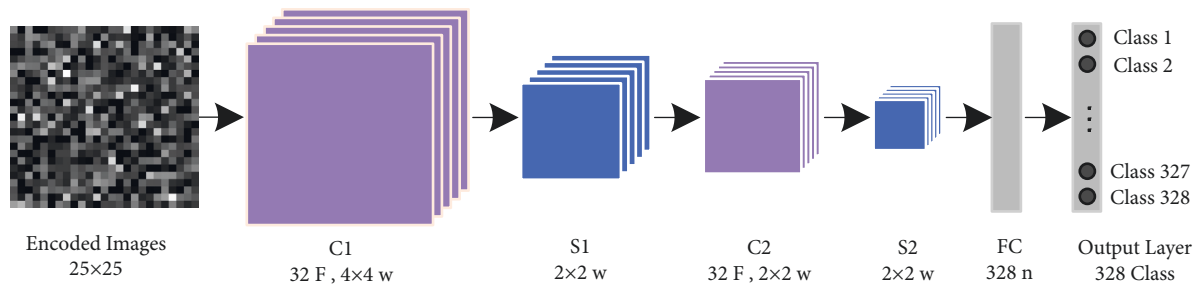


FIGURE 5: The architecture of the proposed CNN.

made two-dimensional and regarded as the CNN input images, including features affecting the recognition of subwords. The CNN is trained with these images and classifies different subwords. In fact, the process of training AECNN consists of two steps: training the AE network and training the CNN. Figure 6 shows the proposed AECNN.

#### 4.7. Integration of Subwords and Recognition of Final Words.

As discussed, the goal of the proposed recognition system is to detect an input word. As a result, all subwords of a word should be first recognized and then integrated correctly to form the input word of interest. Given labeling different letters of the same group, some words of the candidate dataset might belong to the input word. The signs of subwords might be needed to select them. It is also probable that the classifier may make mistakes in recognizing main subwords and signs and mistake main subwords for signs, and vice versa. As a result, the subword integration algorithm should predict these conditions and be resistant to them. The steps of the proposed integration algorithm are as follows:

- (1) Dividing the recognized subwords into main subwords and signs.
- (2) Integrating the main subwords in order of recognition and naming them MS according to the following equation:

$$MS = \{ms_1, \dots, ms_i, \dots, ms_n\}. \quad (9)$$

In the above equation,  $ms_i$  indicates the  $i$ th subword, and  $n$  shows the whole number of subwords.

- (3) Determining the similarity between MS and the number of datasets with three different assumptions:

- (3.1) The number of detected main subwords is correct: determining the similarity between MS and the words of dataset with  $n$  subwords (Table 5) and storing the highest rate of similarity as  $mss_1$  with corresponding words.
- (3.2) The number of detected main subwords is larger than the correct number of main subwords: According to equation (10), a new set is defined and named  $MS'$ , which is created by eliminating each  $ms_i$  subword:

$$MS' = \begin{cases} \{ms_2, ms_3, \dots, ms_n\} \\ \{ms_1, ms_3, \dots, ms_n\} \\ \vdots \\ \{ms_1, ms_2, \dots, ms_{n-1}\}. \end{cases} \quad (10)$$

Then the similarity of each row of  $MS'$  and words of the dataset with  $n - 1$  main subwords

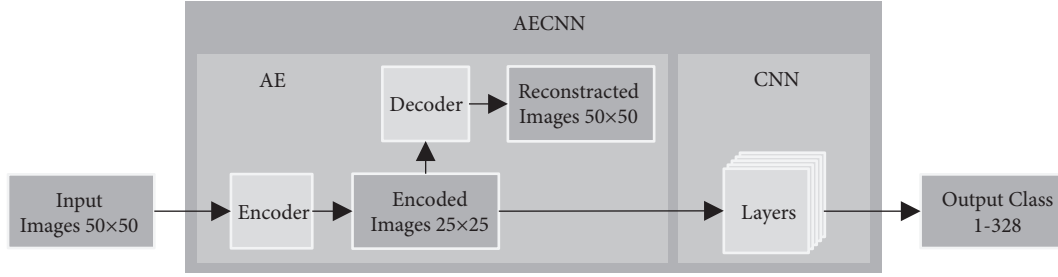


FIGURE 6: The proposed AECNN.

is determined, and the highest similarity rate is saved as  $mss_2$  with corresponding words.

- (3.3) The number of detected main subwords is smaller than the correct number of main subwords: According to equation (11), a new set is defined and named  $MS''$ , created by adding  $\emptyset$  to the position of each subword  $ms_i$ .

$$MS' = \begin{cases} \{\emptyset, ms_1, ms_2, \dots, ms_n\} \\ \{ms_1, \emptyset, ms_2, \dots, ms_n\} \\ \vdots \\ \{ms_1, ms_2, \dots, ms_n, \emptyset\}. \end{cases} \quad (11)$$

Then the similarity between each row of  $MS''$  and words of the dataset with  $n + 1$  main subwords is determined, and the highest rate of similarity is saved as  $mss_2$  with corresponding words.

After determining three maximum similarity rates ( $mss_1, mss_2, mss_3$ ), the similarity vector  $MSS$  is obtained from the following equation:

$$MSS = \{mss_1, mss_2, mss_3\}. \quad (12)$$

Equation (13) indicates the similarity percentage between two subword sets  $A$  and  $B$  of the same size:

$$\text{Similarity}(A, B) = \frac{\alpha}{n} \times 100. \quad (13)$$

In the above equation,  $\alpha$  and  $n$  show the number of matching subwords of the two sets and the total number of subwords, respectively.

- (4) Generating the sufficiency vector  $suf$ , indicating how fit a word is to be selected, by multiplying the similarity vector  $MSS$  into the sufficiency coefficient vector  $F$ , according to equation (15):

$$F = \{f_1, f_2, f_3\}. \quad (14)$$

$$Suf = MSS.F = \{mss_1.f_1, mss_2.f_2, mss_3.f_3\}. \quad (15)$$

In equation (14),  $f_1, f_2, f_3$  indicate the fitness coefficients corresponding to  $mss_1, mss_2, mss_3$ , respectively.

- (5) Detecting the word or words corresponding to the highest value of fitness. If there are several candidate

words, the next step is taken. Otherwise, the final word is printed, and the algorithm is terminated.

- (6) Detecting the contour line of the input word.  
 (7) Dividing the image into upper-contour and lower-contour sections.  
 (8) Labeling the upper-contour and lower-contour sign subwords as  $U$  and  $D$ , respectively.  
 (9) Integrating the sign subwords in order of detection and naming them as  $SS$  according to the following equation:

$$SS = \{ss_1, \dots, ss_i, \dots, ss_n\}. \quad (16)$$

In the above equation,  $ss_i$  is the  $i$ th sign subword, and  $n$  shows the total number of sign subwords.

- (10) Determining the similarity between  $SS$  and the candidate sign subwords, detecting and printing the final word based on the highest rate of similarity, and terminating the algorithm.

Considering the algorithm assumptions allows the correction of errors if a main subword is detected wrongly in the classification step. The values of fitness coefficients are also considered  $f_1, f_2, f_3$  controlling the fair selection of three assumptions. To determine their values, it should be taken into account that the first assumption is much stronger than the second ( $f_1 > f_2, f_3$ ), and the second and third assumptions are not superior to one another,  $f_2 = f_3$ . These coefficients should also be selected in a way to keep the value of  $suf$  between zero and 100. As a result, the fitness coefficient vector  $(1, f, f)$  can be determined. To select the value of  $f$ , the proposed recognition accuracy should be checked for different values of  $f$ , which can be seen in Figure 7. Accordingly, the highest accuracy was obtained in  $f = x$ ; therefore, the fitness coefficient vector is determined as  $\{1, .9, .9\}$ .

Figure 8 shows the steps of the proposed subword fusion algorithm for the word "تبریز" (Tabriz). Accordingly, the AECNN classifier output includes the following subwords: "Mad", "1", "ررD", "2", "ررD", and "1D." The first subword was detected as "Mad" instead of "2D." After dividing subwords into main subwords and signs, the set  $MS = \{"رر", "ررر"\}$  was obtained. For the first assumption, the similarity of  $MS$  and all words of the dataset should be determined to two main subwords, the number of which is 156 according to

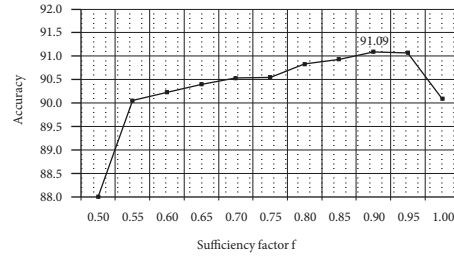


FIGURE 7: Classification accuracy based on different values of (f).

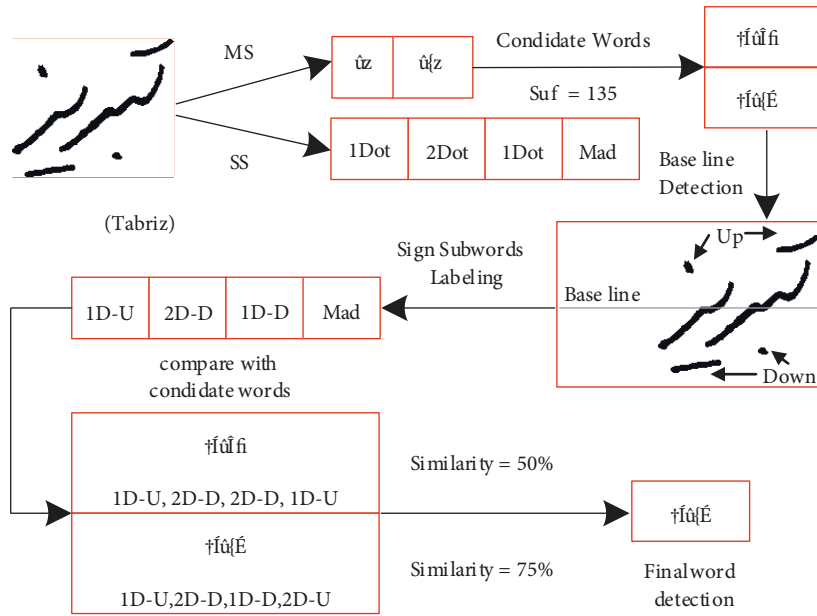


FIGURE 8: Subwords fusion steps and final word detection.

Table 6. Finally, “تبریز” (Tabriz) and “نریزن” (Neyriz) were the most similar words with  $mss_1 = 100$ . Regarding the second assumption,  $MS'$  was first developed as follows:

$$MS' = \begin{Bmatrix} \{ "رب" \} \\ \{ "ربب" \} \end{Bmatrix} \quad (17)$$

Then the similarity of each row of  $MS'$  with all words of the dataset with one main subword was obtained as 36 according to Table 6. After that, no candidate words were found for this assumption; thus,  $mss_2 = 0$ . Regarding the third assumption,  $MS''$  should first be developed as follows:

$$MS'' = \begin{Bmatrix} \{ "رب", "ربب", "" \} \\ \{ "رب", "", "ربب" \} \\ \{ "", "رب", "ربب" \} \end{Bmatrix} \quad (18)$$

Then the similarity of each line of  $MS''$  and all words of the dataset with three main subwords was obtained as 169 according to Table 6. Finally, “آبریز” (Abriz) was the most similar to the first row of  $MS''$  with  $mss_3 = 33$ . Thus, it was

selected as the candidate of the third assumption. Therefore, the similarity vector was  $MSS = \{100, 0, 33\}$ , which was obtained by multiplying the fitness coefficient functions into  $Suf = \{100, 0, 25\}$ . Based on the highest sufficiency coefficient, “تبریز” (Tabriz) and “نریزن” (Neyriz) were selected. The other steps of the algorithm should also be taken because the result was not unique. After detecting the contour line through the method introduced in the preprocessing section, the upper and lower signs were labeled as  $U$  and  $D$ ; then  $SS = \{ "1D\_U", "2D\_D", "1D\_D", "Mad" \}$  was created. Finally, the similarity of  $SS$  and candidate sings was determined. The signs of “تبریز” (Tabriz) were  $\{ "1D\_U", "2D\_D", "1D\_D", "2D\_U" \}$ , which were 75% similar, and those of “نریزن” (Neyriz) were  $\{ "1D\_U", "2D\_D", "2D\_D", "1D\_U" \}$ , which were 50% similar. Hence, “تبریز” (Tabriz) was selected as the final output.

### 5. Dataset

In this study, the widely used and known dataset Iranshahr was employed. It includes the names of 503 Iranian cities in

TABLE 8: Properties of Iranshahr dataset.

Dataset name	Classes	General			Samples	Dataset	No. of images	
		Dimension	Format	Resolution			Training set (%)	Testing set (%)
Iranshahr	503	Different	Binary	96 dpi	30–40	17000	85	15

different handwritings. All of the images were scanned with 96 dots per inch on a gray surface [54–57]. For the name of each city, there were nearly 30 to 40 different images. Totally, there were over 17000 images. In the training and testing section, the images were divided randomly into two groups: 85% of images for training and the other 15% for testing. Table 8 shows the research dataset in detail.

## 6. Evaluation of the Proposed Algorithm

A Hewlett-Packard (hp) computer was utilized to evaluate our approach with the 64-bit Windows 10 Home Operating System, installed memory (RAM) of 8.00 GB, and Intel (R) Core (TM) i7-6500U CPU. For data processing, the MATLAB and Statistics Toolbox Release 2021a were employed. The recognition results of the proposed method can be analyzed in two aspects: (1) what effects the autoencoder network have on the recognition accuracy and (2) what percentage of recognition accuracy pertains to the integration of the proposed subwords to generate the final words. The ability of any technique to recognize the word is measured by a parameter named “accuracy” and is demonstrated by

$$\text{Accuracy} = 100 \times \frac{TP + TN}{TP + TN + FP + FN}, \quad (19)$$

where TP, FP, TN, and FN are True Positive, False Positive, True Negative, and False Negative, respectively.

As discussed in the integration of subwords, the fitness coefficient vector was determined as  $\{1, f, f\}$ , controlling the fair selection of the three assumptions. Figure 7 shows the recognition accuracy of the proposed method based on different values of  $f$ . If  $f = .5$ , it means that it is probably as twice as the first assumption, and  $f = 1$  shows the sameness of the three assumptions. Accordingly, the highest accuracy was 91.09%, which was achieved in  $f = 0.9$ .

First, the CNN and the AECNN were compared in the same structure and parameters to analyze the effect of the autoencoder network on the recognition accuracy. Table 9 shows the comparison results, indicating the effect of the autoencoder network on the extraction of appropriate features and proper recognition of Persian handwritten words through the CNN. In fact, the accuracy was improved by 3%.

The proposed recognition system is based on subwords, which are integrated with each other in the final step to create the final words. Table 10 indicates the recognition accuracy of subwords without being integrated into other subwords to show the share of integration in the final accuracy.

Accordingly, the recognition accuracy of subwords was 85.93% in CNN; however, it was 86.68% in AECNN. According to the results of Tables 9 and 10, it is concluded that the

TABLE 9: A comparison between CNN and AECNN.

Method	Accuracy (%)
CNN	89.04
AECNN	91.09

TABLE 10: Subword recognition accuracy.

Method	Accuracy (%)
CNN without subwords fusion	85.93
AECNN without subwords fusion	86.68

proposed subword integration algorithm increased the recognition accuracy of CNN by 3.11%. It also increased the recognition accuracy of AECNN by 4.41%. As a result, although the Persian word recognition algorithm is based on subwords, the subword integration process is considered an error correction operation, which is an important part of the system.

If the problem is considered only from the prospective of the classifier, it is perceived that the recognition system is merely based on words, which were directly classified through CNN and AECNN with different parameters. In this case, the recognition accuracy would be similar to the results of Table 10 or even lower because there are 328 classes of subwords. However, there are 503 classes of words, and increasing the number of classes usually decreases the accuracy. This shows that dividing words into smaller components such as subwords or dividing words into main subwords and signs and classifying them into smaller components can significantly increase the recognition accuracy. However, such a classification can increase complexity and necessitates the existence of powerful algorithms in the component integration section in an effort to generate the final word. Table 11 shows some of the common mistakes in the classification step along with modifications by dividing subwords and detecting the final word.

Table 12 shows the recognition accuracy of the proposed method separately for each number of subwords. Accordingly, the lowest recognition accuracy (89.71%) came from words with one main subword, and the highest recognition accuracy (100%) came from words with seven subwords. Moreover, increasing the number of subwords relatively enhanced the recognition accuracy because a larger number of subwords in a word can increase the probability integrating subwords and correcting errors successfully. For instance, if the input word is only a subword with a recognition error, the integration of subwords will have no chance to correct the errors.

In Table 13, the recognition accuracy of the proposed method was compared with those of other methods on

TABLE 11: Classification errors and modifications by subword integration.

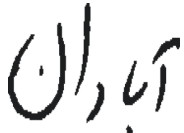
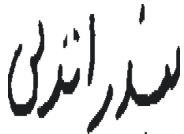
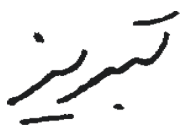
Input Image	Output Classifier	Detected Word
 (Abadan)	All: "1D", "ن", "ا", "د", "1D", "با", "ا", "2D" Main: "ن", "ا", "د", "با", "ا" Sign: "1D", "1D", "2D"	Max Main Similarity: (100, 75, 67) Candidate: ("آرادان", "باحکبران", "آبادان") Score: $(100, 75, 67) \times (1, 0.9, 0.9) = (100, 67, 60)$ Detected Word/Words : "آبادان"
 (Bandaranzali)	All: "سبید", "1D", "1D", "ر", "ا", "ر", "1D", "1D", "لی" Main: "سبید", "ر", "ا", "ر", "لی" Sign: "1D", "1D", "1D", "1D"	Max Main Similarity: (80, 50, 33) Candidate: ("واراباد", "زامهرمر", "بیدرابرلی") Score: $(80, 50, 33) \times (1, 0.9, 0.9) = (80, 45, 30)$ Detected Word /Words: "بندر انزلی"
 (Tabriz)	All: "1D", "2D", "بر", "1D", "بیر", "Mad" Main: "بیر", "بر" Sign: "1D", "2D", "1D", "Mad" Sign by baseline: "1D-U", "2D-D", "1D-D", "Mad"	Max Main Similarity: (100, 0, 33) Candidate: ("بربر", "-", "بیربر") Score: $(100, 0, 33) \times (1, 0.9, 0.9) = (100, 0, 30)$ Detected Word /Words: ("نیریز", "تبریز") Word Sign (Tabriz): ("1D-U", "2D-D", "1D-D", "2D-U") Word Sign (Neyriz): ("1D-U", "2D-D", "2D-D", "1D-U") Sign Similarity: (75, 50) Detected Word: "تبریز"

TABLE 12: Accuracies of the proposed method based on the number of subwords.

Number of main subwords	Accuracy (%)
1	89.71
2	90.14
3	90.77
4	90.79
5	92.34
6	94.59
7	100.00
Average accuracy	91.09

TABLE 13: Comparing the proposed method with the others.

Authors	Techniques	Accuracy (%)
Dehghan et al. [55]	HMM	63.00
Broumandnia et al. [56]	M-band packet wavelet	75.50
Younessy and Kabir [57]	RNN	83.90
Ghadikolaie et al. [54]	RNN	84.30
Proposed method	AECNN	91.09

Iranshahr dataset. Accordingly, the recognition accuracy of the proposed AECNN was 91.09%, indicating the efficiency of this method in comparison with other techniques.

### 7. Conclusion and Future Studies

Due to the complicated nature of the Persian and Arabic languages, in the proposed approach, words were divided into main subwords and sign subwords. Such an action increased the recognition accuracy. Regarding the feature extraction, the intrinsic characteristic of an autoencoder was employed to extract effective and useful features of subword images. After converting them into two-dimensional versions, they were classified through a CNN. The integration of an autoencoder and a CNN in an AECNN increased the recognition accuracy. Moreover, a powerful algorithm was proposed for subword fusion based on the similarity to the error modification approach. This algorithm increased the recognition accuracy by 4.41%. Due to the high similarity between Persian and Arabic, the proposed system can be used for Arabic handwritten word recognition. An important limitation of this research is that the proposed method is not applicable to other languages such as English, due to the intrinsic characteristics of Persian and Arabic languages which are very deferent from English-like languages.

In the future, we will focus on developing the proposed AECNN architecture by increasing the number of layers and deepening the network. Furthermore, it is possible to adopt a metaheuristic training algorithm for AECNN architecture in

order to reduce the training duration and enhance the network accuracy.

## Data Availability

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

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] k. younis and a. khateeb, "Arabic hand-written character recognition based on deep convolutional neural networks," *Jordanian Journal of Computers and Information Technology*, vol. 3, no. 3, p. 186, 2017.
- [2] M. Shafii, "Optical character recognition of printed Persian/Arabic documents," Doctoral Thesis, University of Windsor Canada, Canada, 2014.
- [3] H. Q. Ung, C. T. Nguyen, K. M. Phan, V. T. M. Khuong, and M. Nakagawa, "Clustering online handwritten mathematical expressions," *Pattern Recognition Letters*, vol. 146, pp. 267–275, 2021.
- [4] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: Investigation of normalization and feature extraction techniques," *Pattern Recognition*, vol. 37, no. 2, pp. 265–279, 2004.
- [5] X. Qu, W. Wang, K. Lu, and J. Zhou, "Data augmentation and directional feature maps extraction for in-air handwritten Chinese character recognition based on convolutional neural network," *Pattern Recognition Letters*, vol. 111, pp. 9–15, 2018.
- [6] H. Soltanzadeh and M. Rahmati, "Recognition of Persian handwritten digits using image profiles of multiple orientations," *Pattern Recognition Letters*, vol. 25, no. 14, pp. 1569–1576, 2004.
- [7] N. Shanthi and K. Duraiswamy, "A novel SVM-based handwritten Tamil character recognition system," *Pattern Analysis & Applications*, vol. 13, no. 2, pp. 173–180, 2009.
- [8] L. S. Bernardo, R. Damaševičius, V. H. C. De Albuquerque, and R. Maskeliūnas, "A hybrid two-stage SqueezeNet and support vector machine system for Parkinson's disease detection based on handwritten spiral patterns," *International Journal of Applied Mathematics and Computer Science*, vol. 31, no. 4, pp. 549–561, 2021.
- [9] A. H. Alkilani and M. I. Nusir, "Off-line handwritten verification model for processing bank checks based on truncated-SVD and support vector machine (SVM)," in *Proceedings of the 2021 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pp. 37–42, IEEE, Amman, Jordan, November 2021.
- [10] M. Zare, M. Jampour, A. S. Arezoomand, and M. Sabouri, "Handwritten recognition based on hand gesture recognition using deterministic finite automata and fuzzy logic," in *Proceedings of the 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pp. 93–99, IEEE, Tehran, Iran, March 2019.
- [11] N. Sakamat, *A guided hybrid k-means and genetic algorithm models for children handwriting legibility performance assessment*, PhD thesis, Universiti Teknologi MARA, Malaysia, 2021.
- [12] Y.-C. Wu, F. Yin, and C.-L. Liu, "Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models," *Pattern Recognition*, vol. 65, pp. 251–264, 2017.
- [13] Z. Tamen, H. Drias, and D. Boughaci, "An efficient multiple classifier system for Arabic handwritten words recognition," *Pattern Recognition Letters*, vol. 93, pp. 123–132, 2017.
- [14] S. Chen, G. Liu, C. Wu, Z. Jiang, and J. Chen, "Image classification with stacked restricted Boltzmann machines and evolutionary function array classification voter," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 4599–4606, IEEE, Vancouver, BC, Canada, July 2016.
- [15] J. Gan, W. Wang, and K. Lu, "Compressing the CNN architecture for in-air handwritten Chinese character recognition," *Pattern Recognition Letters*, vol. 129, pp. 190–197, 2020.
- [16] Y. Weng and C. Xia, "A new deep learning-based handwritten character recognition system on mobile computing devices," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 402–411, 2019.
- [17] M. Mhiri, C. Desrosiers, and M. Cheriet, "Convolutional pyramid of bidirectional character sequences for the recognition of handwritten words," *Pattern Recognition Letters*, vol. 111, pp. 87–93, 2018.
- [18] M. Taghizadeh and A. Chalechale, "A comprehensive and systematic review on classical and deep learning based region proposal algorithms," *Expert Systems with Applications*, vol. 189, Article ID 116105, 2022.
- [19] R. Sabzi, Z. Fotoohinya, A. Khalili et al., "Recognizing Persian handwritten words using deep convolutional networks," in *Proceedings of the 2017 Artificial Intelligence and Signal Processing Conference (AISP)*, pp. 85–90, IEEE, Shiraz, Iran, October 2017.
- [20] R. S. Alkhalwaldeh, M. Alawida, N. F. F. Alshdaifat, W. Z. a. Alma'aitah, and A. Almasri, "Ensemble deep transfer learning model for Arabic (Indian) handwritten digit recognition," *Neural Computing & Applications*, vol. 34, no. 1, pp. 705–719, 2022.
- [21] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and Cooperation in Neural Nets*, pp. 267–285, Springer, 1982.
- [22] H. Yang, J. Han, and K. Min, "Distinguishing emotional responses to photographs and artwork using a deep learning-based approach," *Sensors*, vol. 19, no. 24, p. 5533, 2019.
- [23] F. Wang, S. Wu, W. Zhang et al., "Emotion recognition with convolutional neural network and EEG-based EFDMs," *Neuropsychologia*, vol. 146, Article ID 107506, 2020.
- [24] A. Mahmood, M. Bennamoun, S. An et al., "Deep learning for coral classification," in *Handbook of Neural Computation*, pp. 383–401, Elsevier, 2017.
- [25] P. Shukla, N. Pramanik, D. Mehta, and G. Nandi, "Generative model based robotic grasp pose prediction with limited dataset," *Applied Intelligence*, vol. 52, pp. 1–15, 2022.
- [26] E. G. Ribeiro, R. de Queiroz Mendes, and V. Grassi Jr, "Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation," *Robotics and Autonomous Systems*, vol. 139, Article ID 103757, 2021.
- [27] I. Lauriola, A. Lavelli, and F. Aioli, "An introduction to deep learning in natural language processing: Models, techniques, and tools," *Neurocomputing*, vol. 470, pp. 443–456, 2022.

- [28] H. Heidari and A. Chalechale, "Biometric authentication using a deep learning approach based on different level fusion of finger knuckle print and fingernail," *Expert Systems with Applications*, vol. 191, Article ID 116278, 2022.
- [29] R. Ranjbarzadeh, A. B. Kasgari, S. J. Ghoushchi, S. Anari, M. Naseri, and M. Bendechache, "Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images," *Scientific Reports*, vol. 11, no. 1, pp. 1–17, 2021.
- [30] S. Izadi, M. Ahmadi, and A. Rajabzadeh, "Network traffic classification using deep learning networks and bayesian data fusion," *Journal of Network and Systems Management*, vol. 30, no. 2, pp. 1–21, 2022.
- [31] R. Atefinia and M. Ahmadi, "Network intrusion detection using multi-architectural modular deep neural network," *The Journal of Supercomputing*, vol. 77, no. 4, pp. 3571–3593, 2020.
- [32] S. Di Cataldo and E. Ficarra, "Mining textural knowledge in biological images: Applications, methods and trends," *Computational and Structural Biotechnology Journal*, vol. 15, pp. 56–67, 2017.
- [33] Z. Lei, S. Zhao, H. Song, and J. Shen, "Scene text recognition using residual convolutional recurrent neural network," *Machine Vision and Applications*, vol. 29, no. 5, pp. 861–871, 2018.
- [34] I. Ramadhan, B. Purnama, and S. Al Faraby, "Convolutional neural networks applied to handwritten mathematical symbols classification," in *Proceedings of the 2016 4th International Conference on Information and Communication Technology (ICoICT)*, pp. 1–4, IEEE, Bandung, Indonesia, May 2016.
- [35] D. Bashkirova, "Convolutional neural networks for image steganalysis," *BioNanoScience*, vol. 6, no. 3, pp. 246–248, 2016.
- [36] W. Wang, Y. Huang, Y. Wang, and L. Wang, "Generalized autoencoder: A neural network framework for dimensionality reduction," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 490–497, Columbus, OH, USA, June 2014.
- [37] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, "Deep feature learning for medical image analysis with convolutional autoencoder neural network," *IEEE Transactions on Big Data*, vol. 7, no. 4, pp. 750–758, 2017.
- [38] M. Shopon, N. Mohammed, and M. A. Abedin, "Bangla handwritten digit recognition using autoencoder and deep convolutional neural network," in *Proceedings of the 2016 International Workshop on Computational Intelligence (IWCI)*, pp. 64–68, IEEE, Dhaka, Bangladesh, December 2016.
- [39] J. Almotiri, K. Elleithy, and A. Elleithy, "Comparison of autoencoder and Principal Component Analysis followed by neural network for e-learning using handwritten recognition," in *Proceedings of the 2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1–5, IEEE, Farmingdale, NY, USA, May 2017.
- [40] H. D. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad, "Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management," *International Journal of Information Management*, vol. 57, Article ID 102282, 2021.
- [41] P. S. Vasafi, O. Paquet-Durand, K. Brettschneider, J. Hinrichs, and B. Hitzmann, "Anomaly detection during milk processing by autoencoder neural network based on near-infrared spectroscopy," *Journal of Food Engineering*, vol. 299, Article ID 110510, 2021.
- [42] A. G. Dastider, F. Sadik, and S. A. Fattah, "An integrated autoencoder-based hybrid CNN-LSTM model for COVID-19 severity prediction from lung ultrasound," *Computers in Biology and Medicine*, vol. 132, Article ID 104296, 2021.
- [43] M. Seyfioglu and S. Gurbuz, "Deep convolutional autoencoder for radar-based human activity recognition," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 99, 2018.
- [44] A. Alani, "Arabic handwritten digit recognition based on restricted Boltzmann machine and convolutional neural networks," *Information*, vol. 8, no. 4, p. 142, 2017.
- [45] G. Tong, Y. Li, H. Gao, H. Chen, H. Wang, and X. Yang, "MA-CRNN: A multi-scale attention CRNN for Chinese text line recognition in natural scenes," *International Journal on Document Analysis and Recognition*, vol. 23, no. 2, pp. 103–114, 2019.
- [46] G. Xie, K. Yang, and J. Lai, "Filter-in-Filter: Low cost CNN improvement by sub-filter parameter sharing," *Pattern Recognition*, vol. 91, pp. 391–403, 2019.
- [47] C. Wu, W. Fan, Y. He, J. Sun, and S. Naoi, "Handwritten character recognition by alternately trained relaxation convolutional neural network," in *Proceedings of the 2014 14th International Conference on Frontiers in Handwriting Recognition*, Hersonissos, Greece, September 2014.
- [48] S. J. Ghoushchi, R. Ranjbarzadeh, S. A. Najafabadi, E. Osgooei, and E. B. Tirkolaee, "An extended approach to the diagnosis of tumour location in breast cancer using deep learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1–11, 2021.
- [49] T. Bluche, H. Ney, and C. Kermorvant, "Feature extraction with convolutional neural networks for handwritten word recognition," in *Proceedings of the 2013 12th International Conference on Document Analysis and Recognition*, Washington, DC, USA, August 2013.
- [50] H. M. Albeahdili, T. Han, and N. E. Islam, "Hybrid algorithm for the optimization of training convolutional neural network," *International Journal of Advanced Computer Science and Applications*, vol. 1, no. 6, pp. 79–85, 2015.
- [51] T. Liu, Z. Li, C. Yu, and Y. Qin, "NIRS feature extraction based on deep auto-encoder neural network," *Infrared Physics & Technology*, vol. 87, pp. 124–128, 2017.
- [52] W. Luo, J. Li, J. Yang, W. Xu, and J. Zhang, "Convolutional sparse autoencoders for image classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 3289–3294, Jul 2018.
- [53] G. Li, S. Peng, C. Wang, J. Niu, and Y. Yuan, "An energy-efficient data collection scheme using denoising autoencoder in wireless sensor networks," *Tsinghua Science and Technology*, vol. 24, no. 1, pp. 86–96, 2019.
- [54] M. F. Y. Ghadikolaie, E. Kabir, and F. Razzazi, "Sub-word-based offline handwritten farsi word recognition using recurrent neural network," *ETRI Journal*, vol. 38, no. 4, pp. 703–713, 2016.
- [55] M. Dehghan, K. Faez, M. Ahmadi, and M. Shridhar, "Handwritten Farsi (Arabic) word recognition: A holistic approach using discrete HMM," *Pattern Recognition*, vol. 34, no. 5, pp. 1057–1065, 2001.
- [56] A. Broumandnia, J. Shanbehzadeh, and M. R. Varnoosfaderani, "Persian/Arabic handwritten word recognition using M-band packet wavelet transform," *Image and Vision Computing*, vol. 26, no. 6, pp. 829–842, 2008.
- [57] M. F. Younessy and E. Kabir, "A new classifier based on Recurrent neural network using multiple binary-output networks," *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 17, no. 3, pp. 63–69, 2015.



## Research Article

# Skin Lesion Segmentation Based on Edge Attention Vnet with Balanced Focal Tversky Loss

Majid Nour <sup>1</sup>, Hakan Öcal,<sup>2</sup> Adi Alhudhaif <sup>3</sup>, and Kemal Polat <sup>4</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

<sup>2</sup>Computer Engineering/School of Natural and Applied Sciences, Gazi University, Ankara, Turkey

<sup>3</sup>Department of Computer Science, College of Computer Engineering and Sciences in Al-Kharj, Prince Sattam Bin Abdulaziz University, P.O. Box 151, Al-Kharj 11942, Saudi Arabia

<sup>4</sup>Department of Electrical and Electronics Engineering, Faculty of Engineering, Bolu Abant İzzet Baysal University, 14280 Bolu, Turkey

Correspondence should be addressed to Kemal Polat; [kpolat@ibu.edu.tr](mailto:kpolat@ibu.edu.tr)

Received 29 April 2022; Accepted 25 May 2022; Published 14 June 2022

Academic Editor: Ramin Ranjbarzadeh

Copyright © 2022 Majid Nour 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.

Segmentation of skin lesions from dermoscopic images plays an essential role in the early detection of skin cancer. However, skin lesion segmentation is still challenging due to artifacts such as indistinguishability between skin lesion and normal skin, hair on the skin, and reflections in the obtained dermoscopy images. In this study, an edge attention network (ET-Net) combining edge guidance module (EGM) and weighted aggregation module is added to the 2D volumetric convolutional neural network (Vnet 2D) to maximize the performance of skin lesion segmentation. In addition, the proposed fusion model presents a new fusion loss function by combining balanced binary cross-entropy (BBCE) and focal Tversky loss (FTL). The proposed model has been tested on the ISIC 2018 Task 1 Lesion Boundary Segmentation Challenge dataset. The proposed model outperformed the state-of-the-art studies as a result of the tests.

## 1. Introduction

Melanoma, a deadly skin cancer, is predicted to be the fifth most frequently diagnosed cancer for men (57,180 cases) and women (42,600 cases) by 2022 [1]. In the United States, treating skin cancers' annual cost is \$ 8.1 billion: about \$ 4.8 billion for nonmelanoma skin cancers and \$ 3.3 billion for melanoma [2]. The number of diagnoses and treatments for nonmelanoma skin cancers in the USA between 1994 and 2014 reached 77% [3]. Approximately 90% of nonmelanoma skin cancers of nonmelanoma are associated with ultraviolet (UV) radiation from the sun [4]. Board-certified dermatologists very often use dermoscopy, a noninvasive technique that can be helpful in diagnosing skin lesions. Dermoscopy is a magnifying device that allows the application of liquid between dermoscopy and the patient's skin or the use of cross-polarized light to make the epidermis translucent, allowing the structures in the epidermis and

superficial dermis to be visualized. Seeing very small skin lesions that are not visible to the naked eye is essential in deciding for a physician. Dermoscopy provides better performance than traditional methods thanks to ABCD criteria [5]. However, diagnosing the lesioned area with a dermoscopy device is time-consuming and challenging due to artifacts such as skin hairs, blood vessels, and similarities and contrast of light between ordinary and lesioned skin. Therefore, deep convolutional neural networks (DCNN), which will automatically identify images, have been used to overcome this difficult task. Such systems try to determine lesion boundaries and make high-accuracy decisions based on skin lesion segmentation [6–11].

Convolutional neural networks (CNNs) performed higher for many segmentation tasks [12]. DCNNs require large amounts of data for high performance. In addition, there is no clear boundary between the lesion and the surrounding skin. In addition, since the sizes and shapes of

the lesions vary, it is not easy to have information about their characteristics. In addition, ink marks and air bubbles are other difficulties [13]. An edge attention Vnet (ET-Vnet) model is proposed to overcome the artefacts in this study. ET-Vnet combines two attention mechanisms to extract information about the lesion boundary better. Specifically, ET-Vnet contains two paths in its decoder, each embedded with an attention module. In addition, BBCE and FTL functions are combined as a novel approach to increase the performance of the proposed model. Czajkowska et al. [14] proposed a model consisting of two DeepLab v3+ models with a ResNet-50 backbone and a fuzzy connectivity analysis module for fine segmentation. Hu et al. [6] proposed a new attention synergy network (AS-Net) to improve the discriminative ability for skin lesion segmentation by combining spatial and channel attention mechanisms. Arora et al. [15] proposed a modified U-Net-based segmentation model for automatic skin lesion segmentation. The proposed model used group normalization (GN) instead of batch normalization (BN) in both encoder and decoder layers [16, 17]. Besides, attention gates (AG), which focuses on minute details in the skip connection, and Tversky loss (TL) function, which provides higher success in class imbalances, were used. Gu et al. [18] proposed an attention-based modified U-Net (CA-Net) that comprehensively presents based on U-Net architecture by adding multiple spatial attention between layers and awareness of the most critical spatial locations and channel scales. A new channel attention module was used to further focus the proposed model on the lesion area's feature map. A scaling module has been presented to scale the images, highlighting the most prominent feature maps. Goyal et al. [19] proposed an ensemble of R-CNN and Deeplab V3C methods, to achieve high sensitivity and specificity in lesion boundary segmentation methods. The proposed ensemble method ensemble-S achieved better performance than FrCN, FCNs, U-Net, and SegNet. Jiang et al. [20] proposed the CSARM-CNN (canal and spatial attention residue module) model for automatic skin lesion segmentation based on deep learning. Each convolutional layer of CSARM has created a new attention module by combining channel attention and spatial attention to make segmentation training more effective. The spatial pyramid pool acquires multidimensional input images. Finally, two different cross-entropy methods were fused for higher segmentation training in the model, and the final loss function was obtained. Lei et al. [21] proposed a general adversarial network (GAN) to overcome automated lesion segmentation challenges. This network includes a fully dense U-Net-based skip connection and double discrimination (DD) layers. The proposed method (U-Net-SCDC) uses lower resolution up-sampling convolutional layers that preserve fine-grained information. In contrast, the DD module increases training performance by controlling each other in opposite directions. Thus, the two different methods work as if they try to find each other's fault, focusing on their wrong points. Therefore, using a conditional discriminatory loss, it has been said that the model that checks each other simultaneously provided superior performance compared to other models.

Shan et al. [22] proposed a novel segmentation method named FC-DPN. The proposed method consists of the fully convolutional (FCN) and dual-path network (DPN). DPN is a model that enables the virtual feature maps of previous layers to be reused by using residual and densely connected ways. Sub-DPN projection blocks and sub-DPN processing blocks have been added instead of dense layers in the fully convolutional DenseNets (FC-DenseNets). It was stated that this method allows FC-DPN to acquire more representative and distinctive features to perform a more robust segmentation.

Dosovitskiy et al. removed the CNN-based encoder and replaced it with the vision transformer to improve the image recognition performance of the network [23]. Sarker et al. proposed a SLSNet network that reduces the computational cost by using 1-D core factor deep learning networks for sensitive skin lesion segmentation with minimal resources [24]. Wu et al. [25] proposed a new feature adaptive transformer network based on the classical encoder-decoder architecture called FAT-Net.

The organization of this manuscript is as follows. Section 2 provides information about image augmentation and preprocessing. It also includes information about the proposed method and evaluation metrics. In Section 3, details about the computational analysis of the application are given. In Section 4, we present the proposed model assessment. In this section, the results of the proposed method and other methods are presented comparatively. In Section 5, we analyzed and discussed our results and made various suggestions. Finally, the conclusion is given in Section 6.

## 2. Materials and Methods

*2.1. Preparing Dataset.* The proposed model was trained and tested in this study using the ISIC 2018 Lesion Boundary Segmentation dataset. The dataset used is provided by the International Skin Imaging Cooperation (ISIC) archive [12, 26]. The images in the dataset consist of 8-bit RGB dermoscopic images ranging in size from  $767 \times 576$  to  $6682 \times 4401$ . The dataset consists of 2594 training images obtained from different institutes, including various diagnostic challenges.

*2.2. Data Processing and Augmentation.* Deep learning models perform very poorly on datasets with low samples. A large amount of samples is needed for training these models. The ISIC 2018 dataset used for training the model consists of 2594 images. First, 2594 images in the ISIC 2018 dataset were divided into training (2075) and test (519) sets. Then, the amount of data was increased by applying boundary data augmentation with horizontal and vertical flips, random rotation, random distortion, elastic transformation, and scaling and clipping methods to the training samples obtained. Here, better detection of fine-grained features at the borders of the lesioned region was achieved with the border data augmentation method. In the random rotation method, augmented images are obtained from the original images by rotating the original image horizontally and vertically across

each row and column. With the applied data augmentation methods, 72000 training images were obtained. It is aimed to give more reliable and robust results by preprocessing the training images. Contrast stretching applied in data augmentation made the lesions more prominent. In addition, the sharpening algorithm was applied to the data during the contrast stretching. With sharpening and contrast stretching, the blurred edges of the lesion were made more prominent. All images are resized to  $512 \times 512$  as the images are of different sizes, and the suggested model is uniform. Figure 1 shows examples of preprocessed and enhanced images.

**2.3. Proposed Method.** Milletari et al. [27] proposed Vnet architecture for volumetric and fully convolutional 3D image segmentation. The ET-Vnet 2D model is shown in Table 1.

**2.3.1. The Decoder Stage.** Xavier weight initialization was used for the weight initialization of the model, and ReLU was used as the activation function for the layers [28]. In addition, ADAM was used as the optimizer [29]. In the ET-Vnet 2D model, each convolution layer consists of  $3 \times 3$  convolutions with GN to normalize the features in the channels [30]. The  $512 \times 512 \times 1$  input image is fed to the first block, as shown in Table 1. EGM highlighting low valence features in feature maps and ET-Net architecture with WAM emphasizing high valence features were proposed by Zhang et al. [31] for high-performance organ segmentation. The EGM and WAM modules of the proposed architecture have been applied to V-Net 2D. Here, the EGM module emphasizes low-value edge features in segmentation, while the WAM module emphasizes high-value features, allowing the proposed model to perform better segmentation. Also, the convolution layers doubled from 32 to 512 per block [32].

**2.3.2. The Encoder Stage.** The decoder stage takes only high-value features from the encoder and upsamples them to the input size in the decoder. As shown in Table 1, deconvolution, which upsamples feature maps from the down-sampling stage, is performed. A series of convolutional operations are applied at the encoder stage to enlarge the feature map from the encoder to the original low-resolution predicted input image. It is aimed to achieve higher performance segmentation of the network by synchronizing and comparing the decoder and encoder, thanks to the bypass connections from the encoder. In the decoder stages of the proposed model, the same convolution blocks used in the encoder are used in reverse. ReLU activation function is used in each layer, as in the encoder. In the last layer,  $1 \times 1$  convolution using the sigmoid activation function is used, since black and white image segmentation includes two classes. Figure 2.

**2.4. Evaluation Metrics.** The metrics used to score the performance of the models in the ISIC 2018 challenge competition were also used to test the performance of the

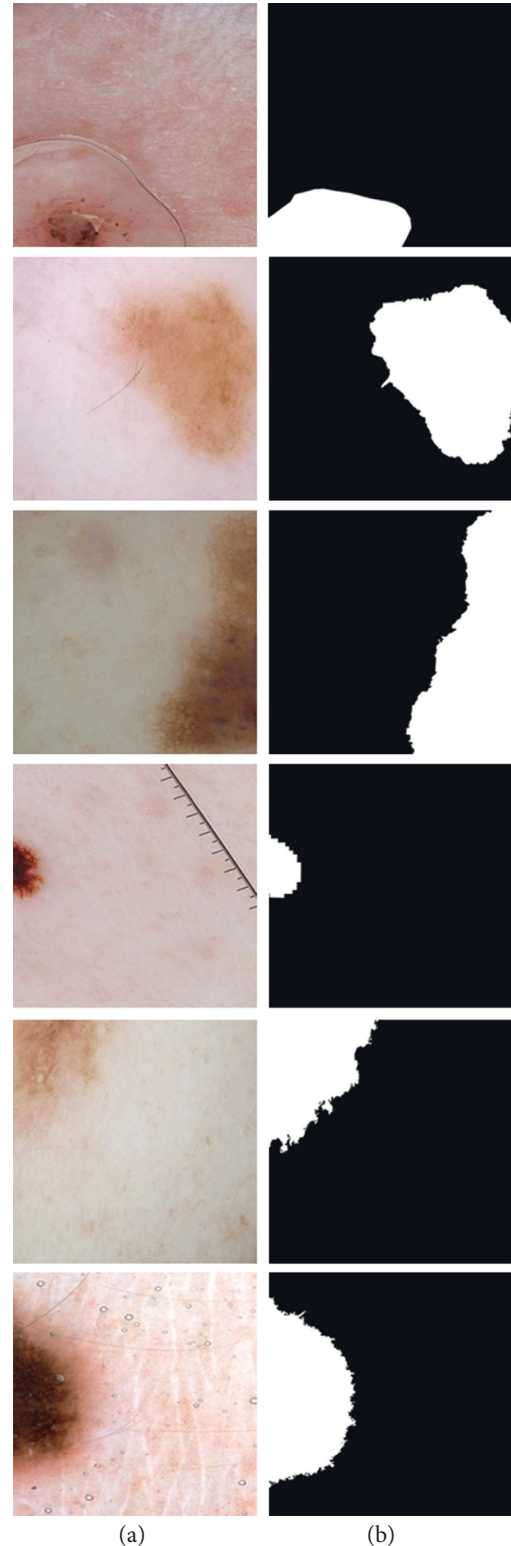


FIGURE 1: Image boundary augmentation. (a) Input image (b) Ground truth.

proposed model. The first of these metrics is the Sorensen-Dice coefficient (DSC), shown in equation (1). Dice is a simple measure that measures the similarity of two samples. The Jaccard index (Jaccard) is used to calculate the similarity

TABLE 1: ET-Vnet 2D models' details. Inputs (batch size, height, width, depth, Channels): (4,512,512,1,1).

Layer	Layers' details	Output size	Activate/Norm/Weight initializer
Layer 1	Conv $3 \times 3$ ResNet block	(4,512,512,1,32)	GN, ReLU, Xavier
Down 1	Filter $3 \times 3$ Filter $3 \times 3$ Filter $3 \times 3$	(4,256,256,1,64)	GN, ReLU, Xavier
Layer 2	ResNet block		GN, ReLU, Xavier
Down 2	Filter $3 \times 3$ Filter $3 \times 3$ Filter $3 \times 3$	(4,128,128,1,128)	GN, ReLU, Xavier
Layer 3	ResNet block		GN, ReLU, Xavier
Down 3	Filter $3 \times 3$ Filter $3 \times 3$ Filter $3 \times 3$	(4,64,64,1,256)	GN, ReLU, Xavier
Layer 4	ResNet block		GN, ReLU, Xavier
Down 4	Filter $3 \times 3$ Filter $3 \times 3$ Filter $3 \times 3$	(4,32,32,1,512)	GN, ReLU, Xavier
Layer 5	ResNet block		GN, ReLU, Xavier
Upsample 1	Filter $3 \times 3$ Filter $3 \times 3$ Filter $3 \times 3$	(4,64,64,1,256)	GN, ReLU, Xavier
Layer 6	ResNet block		GN, ReLU, Xavier
Unsample 2	Filter $3 \times 3$ Filter $3 \times 3$ Filter $3 \times 3$	(4,128,128,1,128)	GN, ReLU, Xavier
Layer 7	ResNet block		GN, ReLU, Xavier
Upsample 3	Filter $3 \times 3$ Filter $3 \times 3$ Filter $3 \times 3$	(4,256,256,1,64)	GN, ReLU, Xavier
Layer 8	ResNet block		GN, ReLU, Xavier
Upsample 4	Filter $3 \times 3$ Filter $3 \times 3$ Filter $3 \times 3$	(4,512,512,1,32)	GN, ReLU, Xavier
Layer 9	ResNet block		GN, ReLU, Xavier
Layer 10 3D single output	ET-Net2D Filter $1 \times 1$	(4,512,512,1,1) + ET-Net2D	GN, ReLU, Xavier

and diversity of sample sets, as defined in equation (2). We can also define Jaccard as the intersection of the union of two different sets. Accuracy (ACC), shown in equation (3), represents the percentage of correct predictions out of all predictions made by the model. Sensitivity (Sens), defined in equation (4), measures the proportion of samples predicted as true positive (TP) in the data set. Also, sensitivity can be defined as recall. As shown in equation (5), specificity (Spec) is a metric that calculates the proportion of correctly predicted nonlesional areas in an image. One drawback of the dice coefficient (Dice) is that false positive (FP) and false negative (FN) are equally weighted. While this increases the precision, it decreases the recall rate.

For this reason, Dice causes degradation in test performance in unbalanced datasets. The way to deal with this situation is to weigh FN more than FP. The Tversky similarity index is a generalization of the Dice coefficient that provides flexibility to its problems in balancing FPs and FNs as shown in equation (6).

$$\text{Dice} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}}, \quad (1)$$

$$\text{Jaccard} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}}, \quad (2)$$

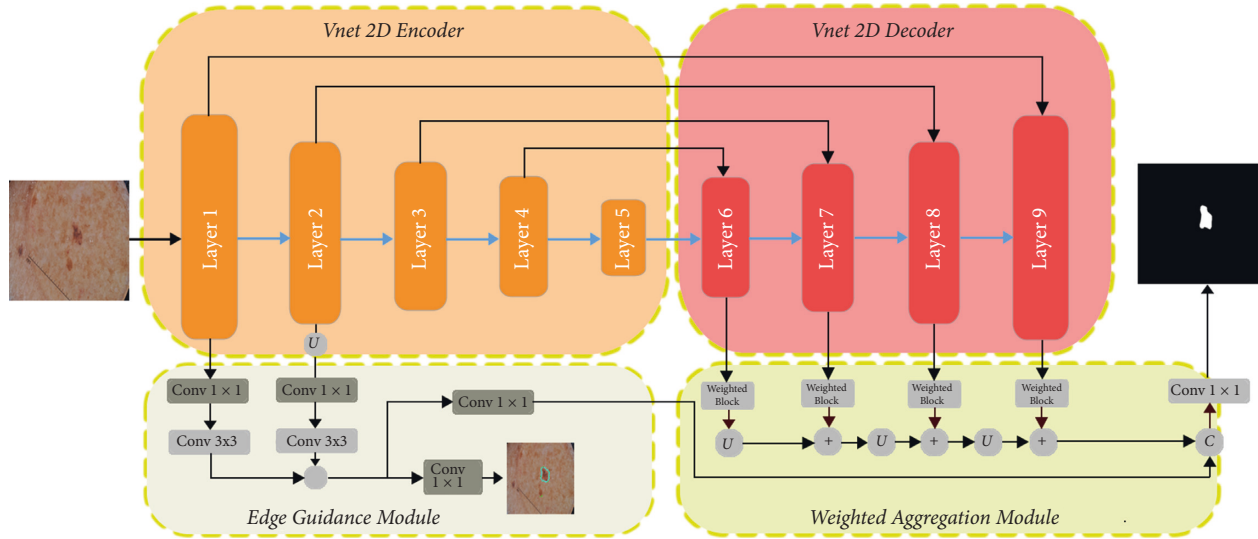


FIGURE 2: ET-Vnet 2D models' block diagram.

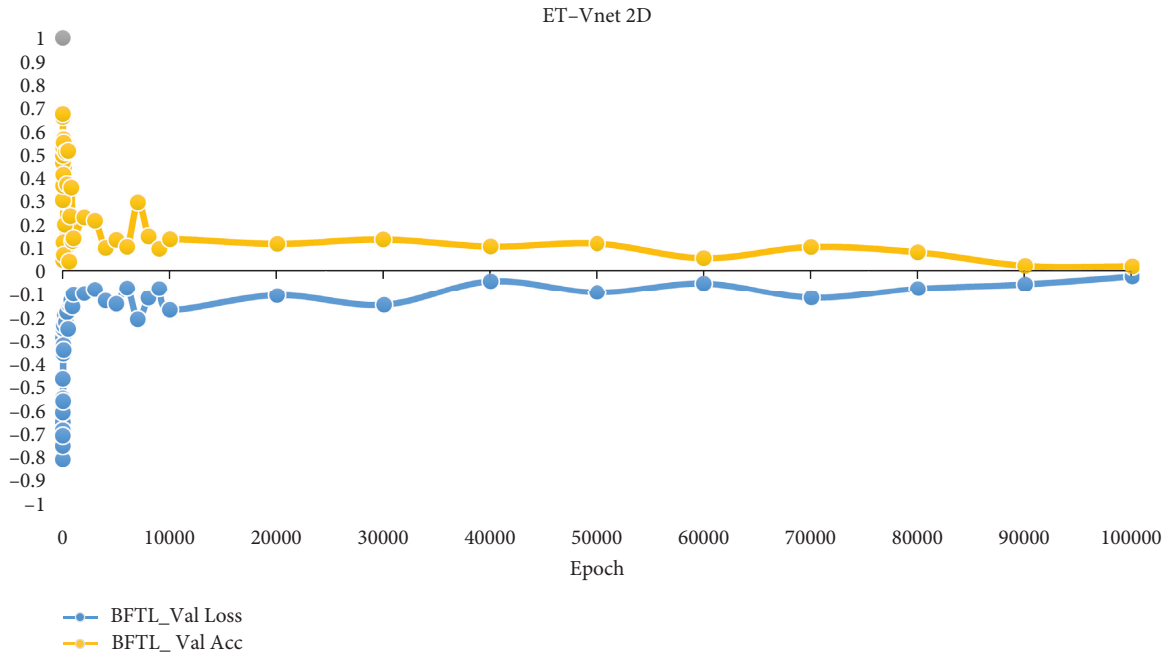


FIGURE 3: ET-Vnet 2D validation loss and validation accuracy results.

$$Acc = \frac{TN + TP}{TN + TP + FN + FP}, \quad (3)$$

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

$$Spec = \frac{TN}{TN + FP}, \quad (5)$$

$$TverskyIndex = \frac{TP}{TP + \alpha FN + \beta FP}, \quad (6)$$

where True Positive (TP) represents the correctly labeled lesion pixels, False Positive(FP) represents the incorrectly

predicted lesion pixels, True Negative(TN) represents the correctly predicted nonlesion pixels, and False Negative(FN) represents the incorrectly predicted lesion pixels. In equation (6),  $\alpha$  and  $\beta$  are adjustable parameters to increase the weight of the recall rate in unbalanced datasets.

**2.5. Balanced Focal Tversky Loss Function (BFTL).** The BFTL loss function proposed in this study is developed from the loss function proposed by Zhou et al. [33]. The FTL loss function is explained in equation (7) which effectively solved the problems of Dice loss in class imbalances [34, 35]. The FTL function shown in equation (8) is presented as a solution to data imbalances. In the proposed model, a loss

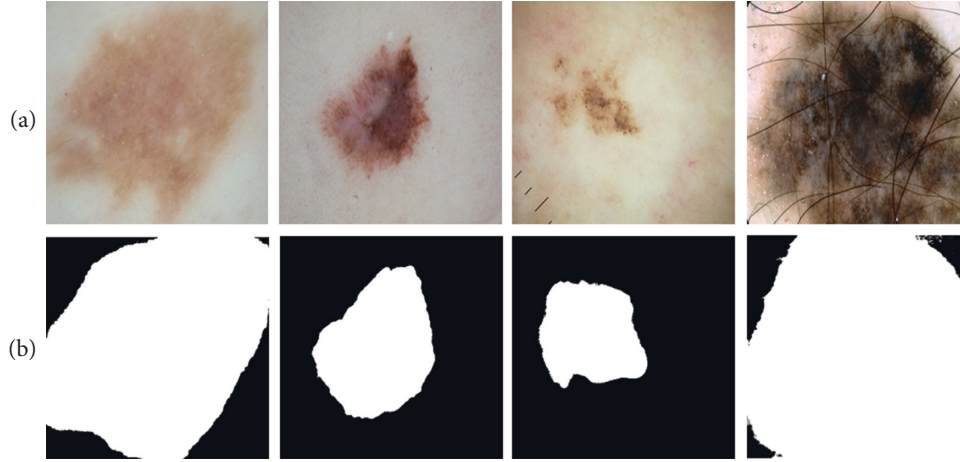


FIGURE 4: Test results for ISIC 2018 challenge 1000 test set.

TABLE 2: Performance scores of the proposed model with the latest models in the literature.

Methods	Performance metrics				
	Dice	Jaccard	Acc	Sens	Spec
DA-net	0.88	0.80	0.95	0.94	0.95
DRU-net [37]	0.86	0.76	—	0.88	0.92
Deeplabv3+	0.87	0.79	0.95	0.94	0.94
CKDNet [38]	0.88	—	0.95	0.91	<b>0.97</b>
DAGAN [24]	0.89	0.83	0.93	0.95	0.91
FAT-net [31]	0.89	—	<b>0.96</b>	0.91	<b>0.97</b>
Attn_U-net + GN [18]	0.91	0.83	0.95	0.94	0.95
SE_U-net [18]	0.91	0.83	0.95	0.89	0.96
ET-vnet (ours)	<b>0.91</b>	<b>0.83</b>	0.95	<b>0.95</b>	0.96

function is proposed that both take care of pixel type losses and solve data imbalances.

$$\text{FTL} = (1 - TI) \frac{1}{\gamma}, \quad (7)$$

$$\mathcal{L} \text{ Twersky} = \sum_{c=1}^{Nb} \left( 1 - \frac{\sum_{i=1}^N \text{pic}_{gic} + \epsilon}{\sum_{i=1}^N \text{pic}_{gic} + \alpha \sum_{i=1}^N \text{pic}_{gic} + \beta \sum_{i=1}^N \text{pic}_{gic} + \epsilon} \right)^{1/\gamma}. \quad (8)$$

The Tversky index is adapted to a loss function (TL) in [36] by minimizing  $\sum_c 1 - TI_c$ .

The BFTL function  $L$  is formulated in equation (9).

$$\mathcal{L} = \mathcal{L}bbce + \log(\mathcal{L} \text{ Twersky}), \quad (9)$$

where  $\mathcal{L}bbce$  is the BBCE [37], unlike binary cross-entropy,  $\beta$  weight is added to BBCE. Where  $\beta$  is the number of negative samples/total number of samples. In other words,  $\beta$  is the proportion of the dominant sample in a data set.  $1 - \beta$  denotes the fraction of the other class. In addition, the

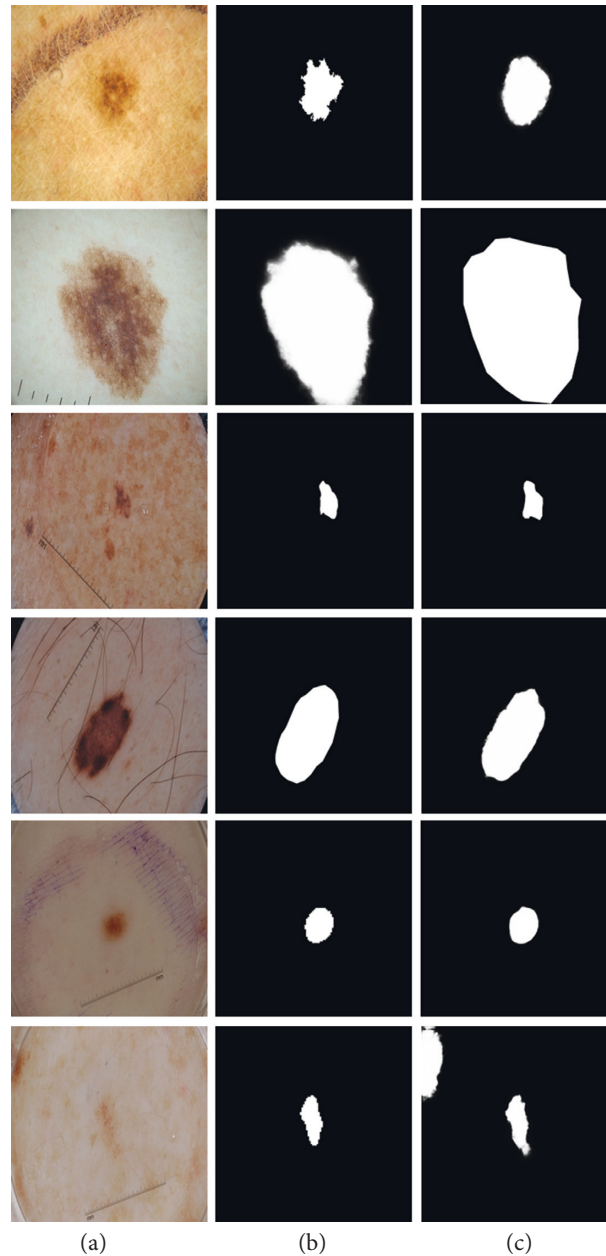


FIGURE 5: Test results are based on 519 test sets obtained from the ISIC 2018 training set. (a) Input image. (b) Image mask. (c) Predicted mask.

natural logarithm is applied to the FTL function to ensure consistency between the loss functions.

### 3. Calculation

**3.1. Application Details.** A computer with Intel I5–8300H processor and 32 GB of RAM and Nvidia GTX 1080 ti graphics card was used to train the proposed model. Besides, the computer's operating system is Windows 10–64 bit. The proposed deep learning model was created using the Python 3.6 programming language. In the proposed model, the learning rate was determined as  $1e-3$ , and the lot size = 4 due to the experiments. The model's training has been realized as 100,000 epochs, thanks to 72000 increased training images. Vnet 2D and ResUnet 2D models were separately trained in

the increased data set from the ISIC 2018 data set. The graphical analysis of the tests performed is shown in Figure 3.

The images of some samples from the proposed models' test results, their true lines, and the predicted segmentation result are shown in Figure 4. As can be seen from Picture 4, when the predicted results are compared with the actual lines, it is seen that the model exhibits very high performance.

**3.2. Qualitative Analysis.** The proposed model's segmentation and final results are shown in Figure 4. The figure demonstrates images obtained from the model with BBFTL loss function. Each column of the figure shows a unique introductory dermoscopic view. The proposed model efficiently analyses input images, mainly obtained through

boundary data augmentation, and predicts complex lesion boundaries with great precision.

**3.3. Hardware Analysis.** The proposed model has been compared to other cutting-edge models for many model parameters, storage requirements, and extraction rates using Nvidia Geforce GTX 1080 ti graphics. In addition, since the data set is stable, there is no need for pretraining scoring. The training of the model took approximately 8 hours. In addition, the model's estimation of the test data set took 52 seconds.

## 4. Prediction Results

The proposed segmentation model was tested on ISIC 2018 Task 1 dermoscopic lesion images. The BBFTL loss function combining BBCE and FTL function is used as a new approach. One of the main factors in applying the border data augmentation method is to enable the model to recognize better the lesion borders that separate the lesioned area from the skin.

**4.1. Comparative Analysis with the Latest Models.** Table 2 shows the comparative results of the proposed model with other models in the ISIC 2018 dataset. The proposed model surpassed the latest literature methods in the tests in the 2018 dataset, scoring 0.91 points in the Dice score, 0.83 points in the Jaccard score, and 0.95 points in the sensitivity, and the other models achieved superior performance.

Figure 4 shows the comparative results of the proposed model with other models in the ISIC 2018 dataset.

Figure 5 shows the visually estimated output of some of the complex samples in the 519 test images separated from training with the proposed approach. The figure compares the basic reality and predicted image based on the original input image. Each row in the figure shows the test results of a test input image in the data set. The third column shows the segmentation result predicted by the proposed fusion model of the test input image presented in the first column.

## 5. Discussion

In the proposed model, we get the ET-Vnet 2D network by fusing the EGM, WAM, and Vnet 2D networks from the ET-Net 2D network. In addition, the ET-Vnet 2D network training was completed in 8 hours. EGM and WAM from the ET-Net 2D network significantly improved the performance of the Vnet 2D network. Also, in the proposed model, balanced binary cross-entropy and focal Tversky loss functions are hybrids combined. The proposed hybrid loss function caused a slight prolongation of the training time. But it enabled the model to give more robust results.

## 6. Conclusion

In this study, EGM and WAM modules, which are two modules used in the ET-Net network, are combined with Vnet 2D to create the proposed model—tested on ET-Vnet

2D ISIC 2018 Lesion Boundary segmentation dataset. The proposed model consists of data set preprocessing and data augmentation, lesion identification, and prediction operations. In the dataset preprocessing stage, artifacts such as color inconsistency, exposure problem, and visibility of lesion borders are corrected. The visualization of a limited number of labeled skin lesions has been achieved sufficiently to train the proposed model, mainly thanks to the border and other data augmentation methods. In addition, a new loss function is proposed by combining BBCE, which calculates pixel type losses, and FTL, which is presented as a solution to class imbalances. The loss function proposed as a novel approach significantly improves the performance of the proposed model. The proposed hybrid loss function is thought to play a crucial role in challenging segmentation tasks. The data set's Dice and Jaccard similarity metrics were recorded as 0.95 and 0.83, surpassing the latest segmentation techniques. The proposed model will be tested in other organ segmentations to prove its robustness in future studies.

## Data Availability

We can send the datasets at the request of the authors.

## Ethical Approval

This article does not contain any studies with human participants. No animal studies were involved in this review.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

All authors contributed equally to this work. In addition, all authors have read and approved the final manuscript and given their consent to publish the article.

## References

- [1] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics," *CA: A Cancer Journal of Clinicians*, vol. 71, 2021.
- [2] G. P. Guy, S. R. Machlin, D. U. Ekwueme, and K. R. Yabroff, "Prevalence and costs of skin cancer treatment in the US, 2002-2006 and 2007-2011," *American Journal of Preventive Medicine*, vol. 48, no. 2, pp. 183-187, 2015.
- [3] S. V. Mohan and A. L. S. Chang, "Advanced basal cell carcinoma: epidemiology and therapeutic innovations," *Current Dermatology Reports*, vol. 3, no. 1, pp. 40-45, 2014.
- [4] H. K. Koh, A. C. Geller, D. R. Miller, T. A. Grossbart, and R. A. Lew, "Prevention and early detection strategies for melanoma and skin cancer. Current status," *Archives of Dermatology*, vol. 132, no. 4, pp. 436-443, Article ID 8629848, 1996.
- [5] M. E. Vestergaard, P. Macaskill, P. E. Holt, and S. W. Menzies, "Dermoscopy compared with naked eye examination for the diagnosis of primary melanoma: a meta-analysis of studies performed in a clinical setting," *British Journal of Dermatology*, vol. 159, no. 3, pp. 669-676, 2008.
- [6] K. Hu, J. Lu, D. Lee, D. Xiong, and Z. A. S.-N. Chen, "AS-Net: attention Synergy Network for skin lesion segmentation,"



- Expert Systems with Applications*, vol. 201, Article ID 117112, 2022.
- [7] N. Daldal, M. Nour, and K. Polat, "A novel demodulation structure for quadrature modulation signals using the segmentary neural network modelling," *Applied Acoustics*, vol. 164, Article ID 107251, 2020.
- [8] N. Daldal, A. Sengur, K. Polat, and Z. Cömert, "A novel demodulation system for base band digital modulation signals based on the deep long short-term memory model," *Applied Acoustics*, vol. 166, Article ID 107346, 2020.
- [9] N. Daldal, Z. Cömert, and K. Polat, "Automatic determination of digital modulation types with different noises using Convolutional Neural Network based on time - frequency information," *Applied Soft Computing*, vol. 86, Article ID 105834, 2020.
- [10] M. Nour, N. Daldal, M. F. Kahraman, H. Sindi, A. Alhudhaif, and K. Polat, "A novel tilt and acceleration measurement system based on Hall-effect sensors using neural networks," *Mathematical Problems in Engineering*, vol. 2022, Article ID 7000486, 13 pages, 2022.
- [11] M. F. Kahraman and S. Öztürk, "Experimental study of newly structural design grinding wheel considering response surface optimization and Monte Carlo simulation," *Measurement*, vol. 147, Article ID 106825, 2019.
- [12] N. C. F. Codella, D. Gutman, M. E. Celebi et al., "Skin lesion analysis toward melanoma detection: a challenge at the 2017 international symposium on biomedical imaging (ISBI) hosted by the international skin imaging collaboration (ISIC)," in *Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, Washington, DC, USA, April 2018.
- [13] S. A. Mahdiraji, Y. Baleghi, and S. M. Sakhaei, "Skin lesion images classification using new color pigmented boundary descriptors," in *Proceedings of the 2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pp. 102–107, Shahrekord, Iran, April 2017.
- [14] J. Czajkowska, P. Badura, S. Korzekwa, and A. Płatkowska-Szczerek, "Automated segmentation of epidermis in high-frequency ultrasound of pathological skin using a cascade of DeepLab v3+ networks and fuzzy connectedness," *Computerized Medical Imaging and Graphics*, vol. 95, Article ID 102023, 2023.
- [15] R. Arora, B. Raman, K. Nayyar, and R. Awasthi, "Automated Skin Lesion Segmentation Using an Attention-Based Deep Convolutional Neural Network," *Biomedical Signal Processing and Control*, vol. 65, 2020.
- [16] Y. Wu and K. He, "Group normalization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, Munich, Germany, March 2018.
- [17] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, <http://www.arxiv.org/abs/1502.03167>.
- [18] R. Gu, G. Wang, T. Song et al., "CA-net: comprehensive attention convolutional neural networks for explainable medical image segmentation," *IEEE Transactions on Medical Imaging*, vol. 40, no. 2, pp. 699–711, 2021.
- [19] M. Goyal, A. Oakley, P. Bansal, D. Dancey, and M. H. Yap, "Skin lesion segmentation in dermoscopic images with ensemble deep learning methods," *IEEE Access*, vol. 8, pp. 4171–4181, 2020.
- [20] Y. Jiang, S. Cao, S. Tao, and H. Zhang, "Skin lesion segmentation based on multi-scale Attention convolutional neural network," *Skin lesion segmentation based on multi-scale Attention convolutional neural network*, vol. 8, 2020.
- [21] B. Lei, Z. Xia, F. Jiang, X. Jiang, Z. Ge, and Y. Xu, "Skin Lesion Segmentation via Generative Adversarial Networks with Dual Discriminators," *Medical Image Analysis*, vol. 64, 2020.
- [22] P. Shan, Y. Wang, C. Fu, W. Song, and J. Chen, "Automatic skin lesion segmentation based on FC-DPN," *Computers in Biology and Medicine*, vol. 123, Article ID 103762, 2020.
- [23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, and T. Unterthiner, "An image is worth 16x16 words: transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations*, OpenReview.net, Canada, April 2021.
- [24] M. M. K. Sarker, H. A. Rashwan, F. Akram et al., "SLSNet: skin lesion segmentation using a lightweight generative adversarial network," *Expert Systems with Applications*, vol. 183, Article ID 115433, 2021.
- [25] H. Wu, S. Chen, G. Chen, W. Wang, B. Lei, and Z. Wen, "FAT-Net: feature adaptive transformers for automated skin lesion segmentation," *Medical Image Analysis*, vol. 76, Article ID 102327, 2022.
- [26] N. Codella, V. Rotemberg, P. Tschandl, M. E. Celebi, S. Dusza, and D. Gutman, "Skin Lesion Analysis toward Melanoma Detection 2018: A challenge Hosted by International Skin Imaging Collaboration (ISIC)," 2019, <https://arxiv.org/pdf/1902.03368>.
- [27] F. Milletari, N. Navab, and S. A. Ahmadi, "V-net: fully convolutional neural networks for volumetric medical image," in *Proceedings of the 2016 Fourth International Conference on 3D Vision*, California, CA, USA, October 2016.
- [28] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Aistats*, vol. 9, pp. 249–256, 2010.
- [29] J. Ba and D. P. Kingma, "Adam: a method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, pp. 1–11, ICLR, San Diego, CA, USA, May 2015.
- [30] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance Normalization: The Missing Ingredient for Fast Stylization," 2016, <https://arxiv.org/pdf/1607.08022>.
- [31] Y. Zhang, J. Wu, W. Chen, Y. Chen, and X. Tang, "Prostate segmentation using Z-net," in *Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pp. 11–14, Venice, Italy, April 2019.
- [32] Q. Jin, H. Cui, C. Sun, Z. Meng, and R. Su, "Cascade knowledge diffusion network for skin lesion diagnosis and segmentation," *Applied Soft Computing*, vol. 99, Article ID 106881, 2021.
- [33] W. Zhou, X. Tao, Z. Wei, and L. Lin, "Automatic segmentation of 3D prostate MR images with iterative localization refinement," *Digital Signal Processing*, vol. 98, Article ID 102649, 2020.
- [34] N. Abraham and N. M. Khan, "A novel focal Tversky loss function with improved attention U-net for lesion segmentation," in *Proceedings of the 2019 IEEE 16th International*

- Symposium on Biomedical Imaging (ISBI 2019)*, pp. 683–687, IEEE, Venice, Italy, April 2019.
- [35] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.
- [36] P. Wang and A. C. S. Chung, “Focal dice loss and image dilation for brain tumor segmentation,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Springer, Cham, Switzerland, pp. 119–127, 2018.
- [37] H. Jiang, T. Shi, Z. Bai, and L. Huang, “AHCNet: an application of attention mechanism and hybrid connection for liver tumor segmentation in CT volumes,” *IEEE Access*, vol. 7, Article ID 24898, 2019.
- [38] M. Jafari, D. Auer, S. Francis, J. Garibaldi, and X. Chen, “DruNet: An Efficient Deep Convolutional Neural Network for Medical Image Segmentation,” in *Proceedings of the 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, Iowa City, IA, USA, April 2020.

## Research Article

# A New Generation Communication System Based on Deep Learning Methods for the Process of Modulation and Demodulation from the Modulated Images

Nihat Daldal <sup>1</sup>, Zeynel Abidin Sezer <sup>1</sup>, Majid Nour,<sup>2</sup> Adi Alhudhaif <sup>3</sup>, and Kemal Polat <sup>1</sup>

<sup>1</sup>Department of Electrical and Electronics Engineering, Bolu Abant Izzet Baysal University, Bolu 14280, Turkey

<sup>2</sup>Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

<sup>3</sup>Department of Computer Science, College of Computer Engineering and Sciences in Al-kharj, Prince Sattam Bin Abdulaziz University, P.O. Box 151, Al-Kharj 11942, Saudi Arabia

Correspondence should be addressed to Kemal Polat; [kpolat@ibu.edu.tr](mailto:kpolat@ibu.edu.tr)

Received 17 April 2022; Revised 8 May 2022; Accepted 12 May 2022; Published 28 May 2022

Academic Editor: Saeid Jafarzadeh Ghouschi

Copyright © 2022 Nihat Daldal 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.

Demodulating the modulated signals used in digital communication on the receiver side is necessary in terms of communication. The currently used systems are systems with a variety of hardware. These systems are used separately for each type of communication signal. A single algorithm facilitates the classification and subsequent demodulation of signals without needing hardware instead of extra hardware cost and complex systems. This study, which aims to make modulation classification by using images of signals, provides this convenience. In this study, a classification and demodulation process is done by using images of digital modulation signals. Convolutional neural network (CNN), a deep learning algorithm, has been used for classification and recognition. Images of the signals of quadrature amplitude shift keying (QASK), quadrature frequency shift keying (QFSK), and quadrature phase shift keying (QPSK) digital modulation types at noise levels of 0 dB, 5 dB, 10 dB, and 15 dB were used. Thanks to this algorithm, which works without hardware, the success achieved is around 98%. Python programming language and libraries have been used in training and testing the algorithm. Demodulation processes of these signals have been performed for demodulation using the nonlinear autoregressive network with exogenous inputs (NARX) algorithm, an artificial neural network. As a result of using MATLAB, the NARX algorithm achieved approximately 94% success in obtaining the information signal. Thanks to the work done, it will be possible to classify and demodulate other communication signals without extra hardware.

## 1. Introduction

The communication systems used today mostly use digital communication methods. Additionally, the increase in digital devices has highlighted the need for fast data transmission. The receiver's distance and the transmitter, environmental reasons, and the devices used affect the communication quality [1]. The basic communication system given in Figure 1 includes the stages of transmitting an information signal. It is an important communication rule that this information can be transmitted quickly and without loss.

While the information signal is expressed as a baseband signal in digital communication, it is called passband

modulation with a high-frequency carrier [2]. The demodulation process must be performed following the modulation type of the transmitted signal [3]. However, in some cases, the modulation type of the signal obtained is unknown. In this case, the demodulation process is complicated, and extreme time loss occurs. In this study, it has been provided that the classification of the modulation signals is made by using the images, and thus, the demodulation processes will be easier. The convenience in this work, which is put forward for users such as security intelligence, where speed and time are valuable, will be a gain. Furthermore, developing a single software system that can receive and analyze different modulation signals will provide time and accuracy gains. Compared with the conventional

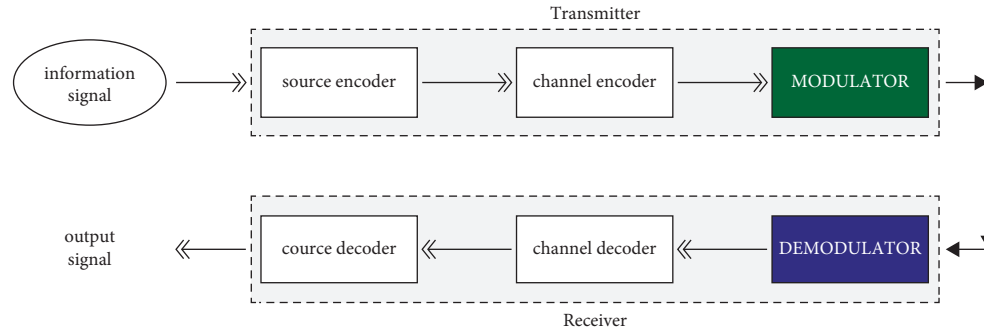


FIGURE 1: Basic communication system flowchart.

demodulator systems, a single software system that can also receive and analyze other modulation signals has been developed instead of systems that only receive a specific digital modulation signal.

Classical algorithms have a low digital signal modulation recognition rate at low signal-to-noise ratios [4]. Applications such as recognizing and classifying signals through machine learning are the important point. Machine learning, deep learning, and image processing methods achieve classification in a short time by eliminating extra hardware costs [5].

The automatic modulation classification and demodulation proposed in the studies in the literature show how critical this area is. The most important problem in modulation classification has been the noisy signals [6] because it is very difficult to recognize and classify a noisy signal. Thanks to the convolutional neural network (CNN) model developed to achieve swift and perfect accuracy results, the algorithm successfully has been performed for automatic modulation classification (AMC) [7]. Images of quadrature amplitude shift keying (QASK), quadrature frequency shift keying (QFSK), and quadrature phase-shift keying (QPSK) modulation signals are stored in three different folders. These images pertain to signals with 0 dB, 5 dB, 10 dB, and 15 dB noise ratios. Using images, folders belonging to individual signals were combined in a path, allowing them to be read with Python libraries. This model can understand whether any image loaded into the algorithm belongs to these three modulation types. If it is one of the QASK, QFSK, or QPSK, we can recognize it automatically. The created CNN model trained the images it reads from the folders according to the specified epoch number.

## 2. Related Work

Nandi et al. conducted experiments on modulation recognition using an artificial neural network in the literature. Using two modulation classification approaches, namely, the decision theory approach and the pattern recognition approach, they performed the recognition process with 100% accuracy for 0 dB, 95% for 3 dB, and 75% for 10 dB [6]. They used artificial neural networks in their studies by classifying radio communication signals to detect unknown communication signal types and achieved the success of over 85%. Wang et al. determined a 100% accuracy rate for images of

RZ-OOK, NRZ-OOK, RZ-DPSK, and PAM signals in the 10 dB to 25 dB noise range to select the modulation format and the estimation of the noise ratio using a convolutional neural network (CNN) [5]. In their study, Ali and Yangyu classified digital modulation using unsupervised pretraining and feed-forward deep neural network (DNN) for automatic modulation classification [8]. Ma et al., using CNN, DBN, and AdaBoost algorithms for demodulation of the signals, achieved a success of approximately 96% by demodulating these modulation types from the image set consisting of QAM QPSK, PPM, and OOK modulation signals [4]. Lee et al., in their studies, made a classification for BPSK (dual phase shift keying), QPSK, PSK, PAM, and QAM modulation types based on the images of these signals. They achieved 83.3% success by extracting features using the CNN algorithm and performing the classification process [9]. Daldal et al., in their study, used STFT and CNN-based hybrid models to classify the digital modulation type and achieved 99% success [10]. In Zhou et al.'s study, modulation recognition applications were carried out using deep learning. As they stated in the confusion matrix of the CNN algorithm, they found a success rate of 96.25% [11].

## 3. Material and Method

**3.1. Dataset and Structure of Quadrature Digital Modulation Signals.** Digital modulation means converting a baseband digital message signal into a bandpass signal at a carrier frequency [2]. Digital modulation is accomplished by changing the amplitude, frequency, or phase of the high-frequency sinusoidal analog carrier signal according to the incoming information carrying the digital baseband message signal [3, 12]. In quadrature or multilevel communications, multiple information with a single carrier is transmitted [10]. In multilevel modulation types, 2-bit can be transmitted simultaneously. These bit values consist of 00, 01, 10, and 11. Each symbol (00, 10, 01, and 11) corresponds to a phase state of the modulated carrier [12]. For QASK modulation, four amplitudes for each pair of bit values, four frequencies for each pair of bit values for QFSK modulation, and four phase values for each pair of bit values for QPSK modulation were determined [10].

The formulas of the quadrature modulation types are shown in Table 1. Modulated images were obtained by applying these formulas in MATLAB. Additive white

TABLE 1: Equations of QASK, QPSK, and QPSK modulation.

	00	01	10	11
QASK	$A_1 \times \text{Cos}W_c t$	$A_2 \times \text{Cos}W_c t$	$A_3 \times \text{Cos}W_c t$	$A_4 \times \text{Cos}W_c t$
QFSK	$A \times \text{Cos}(2.\text{pi}.f_1 t)$	$A \times \text{Cos}(2.\text{pi}.f_2 t)$	$A \times \text{Cos}(2.\text{pi}.f_3 t)$	$A \times \text{Cos}(2.\text{pi}.f_4 t)$
QPSK	$A \times \text{Cos}(W_c t + \theta_1)$	$A \times \text{Cos}(W_c t + \theta_2)$	$A \times \text{Cos}(W_c t + \theta_3)$	$A \times \text{Cos}(W_c t + \theta_4)$

Gaussian noise (AWGN) has been added as noise here. As a result of using the procedure, the images given in Figure 2 are obtained. Signals of data between 1 and 255 were coded according to the quadrature modulation type.

In addition, 5 dB, 10 dB, and 15 dB as Gaussian noise have been added to these images. As shown in Table 1, QASK has the same frequency and phase but different amplitude for each logic level, while QFSK has the same amplitude and phase but different frequency. For QPSK, the stages are different for each logic level [2]. The signals obtained from changing the amplitude, frequency, or phase of the carrier between more than two different values are called multilevel transmission [1]. Thuswise, it becomes possible to transmit more information with a single carrier. These signals obtained were saved as images. Images of each class were stored in separate folders. The images acquired are read from the paths they are in using Python libraries. Classes are class 0 for QASK, class 1 for QFSK, and class 2 for QPSK.

**3.2. Convolutional Neural Network (CNN).** Convolutional neural networks (CNN) that are trainable are made up of multiple stages [13]. CNN is a type of algorithm consisting of an input, an exit, and many hidden layers [14].

It includes the hidden, convolution, pooling, flattened linear unit, fully connected, and classification sections. *The convolution layer and pooling layer* have the task of editing feature maps [15]. Convolution for one-dimensional data is mathematically the name given to the process that a function with a real value that runs through another part to generate a new function [16]. The convolution layer enables the adjustment of the neurons in the image matrix, which is defined as the input called feature map and facilitates the learning of the properties [17]. After the convolution layer is detected locally, the same properties are combined thanks to the pooling layer [15]. Otherwise, the size of the feature matrix resulting from the convolution layer will increase and affect the duration of the training [18]. Or it can lead to overlearning. Each neuron in the feature map is connected to its previous neighbor thanks to filters, in other words, trainable weights [15]. The equation for the convolution layer used for the images is as follows:

$$(I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n), \quad (1)$$

used as [19]. The rectified linear unit (ReLU) layer performs the task of flattening the feature map that emerges after the convolution process [20]. Converting negative values to zero produces output between zero and positively infinite values [21]. There is no change in the size of the data in this layer. The ReLU activation function enables it to increase by affecting the nonlinear feature of the neural network. Other

activation functions, sigmoid, and tanh reduce the speed of the neural network algorithm and show lower performance than the ReLU activation function in terms of results [22] (Figure 3). Calculation of the ReLU function is simpler than other functions since it is not subjected to logarithmic operations [23].

Moreover, the gradient of the ReLU activation function is always 1. On the other hand, the different activation functions take the value 0. Therefore, if the input functions are not entered correctly, the gradient will always sub tend to zero in positive values. Such an undesirable situation prevents the training set from working effectively [19, 24]. The max-pooling layer performs the size reduction operation by performing the function operation defined as subsampling [25]. Additionally, thanks to this layer, excessive memorization is prevented. In this layer, the downsampling method is applied to the feature map created in the convolution layer, and the process of assigning the largest single value instead of large-sized values is performed [22].

In CNN architecture, the features produced by the final convolution layer correspond to a portion of the input image because the receiving area does not cover the entire spatial dimension of the image [24]. Therefore, the fully connected layer becomes mandatory [26]. Thanks to the fully connected layer, the properties that appear in the convolution and pooling layers become meaningful [17]. With each layer, the linear activation function in the previous layer is generated by a nonlinear activation function [25].

The neurons in the fully connected layer are interconnected with the neurons in the entire previous layer. Thus, the output of this layer consists of the labels belonging to the classes.

Regularization in the training phase in the convolutional neural network is an essential element for data augmentation, regularization of weights, and batch normalization [15]. Hereby, the method called dropout is used. Its main use is to prevent overfitting [27]. If the subutilization number is too high, its use in some cases causes learning problems since it dilutes the fully connected layer and filters in neural networks. The subutilization process used in the pooling layer instead of the convolution layers reduces the error rates in the test data [28].

**3.3. Nonlinear Autoregressive Network with Exogenous Inputs (NARX).** Nonlinear autoregressive network with exogenous inputs (NARX) is the name given to neural networks with feedback structure to reach target values [29]. It is used for nonlinear modeling data. Autoregression is defined as a concept that shows the relationship between the previous value and other values in the network [30].

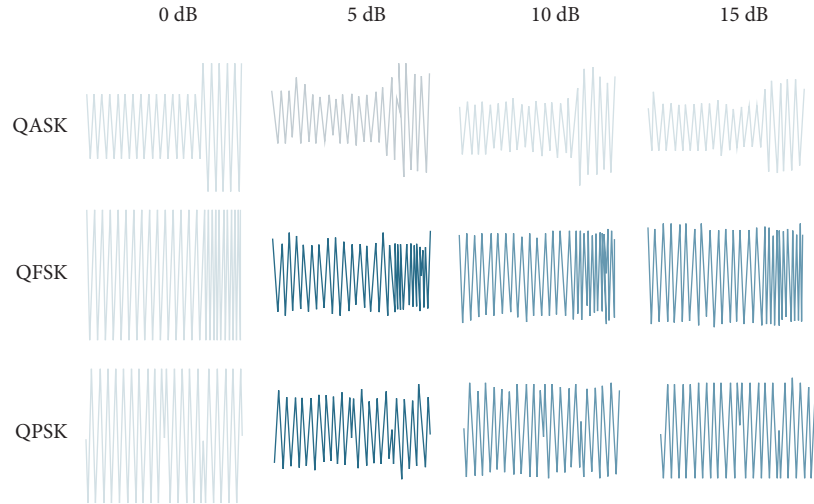


FIGURE 2: Images of QASK, QFSK, and QPSK modulation.

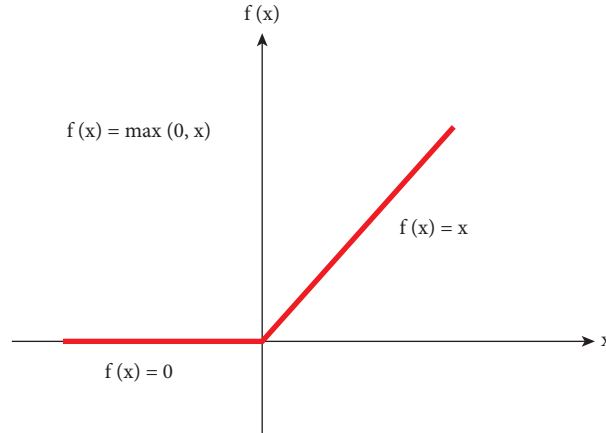


FIGURE 3: ReLU activation function.

Artificial neural networks are used in classification, clustering, object recognition, and prediction, which have a wide range of uses. Since it is a feedback neural network, it has a structure that learns errors and takes advantage of them. NARX model is a model that depends on nonlinear dynamic variables [31]. It has many layers due to the feedback it has. The weights and feedback of the NARX model can be randomly selected.

$$y(t) = f[y(t-1), y(t-2), \dots, y(t-n_y), u(t-1), u(t-2), \dots, u(t-n_u)]. \quad (2)$$

As shown in (1), the nonlinear function  $f$  represents the NARX output  $y(t)$ , the input values  $u(t)$ ,  $n_u$  and  $n_y$  the input-output layers [32]. The output value  $y(t)$  improves the duration of the network by converging to the previous values of independent or self-linked input signals. The NARX model is divided into two types. These are as follows.

*Series-Parallel Model.* In this structure, the next values of  $y(t-1)$  make predictions based on the current and past values of  $x(t)$ , as well as the actual past values of  $y(t)$ .

*Parallel Model.* It estimates by gaining experience from the values of  $x(t)$  and the predicted values of  $y(t)$  in the past.

The NARX neural network has a structure with feedback, that is, progressing according to the error. Since the NARX structure is a type of artificial neural network, it has many hidden layers, convolution layers, pooling, and full connection layers. The flowchart of the NARX model used for demodulation is shown in Figure 4. Figure 5 shows the proposed CNN structure in our study.

## 4. Experimental Results

*4.1. Classification Results.* The advantage of CNN is that they can process data with different spatial dimensions, resulting in less computational costs than traditional matrix multiplication neural networks [33]. Additionally, convolution is simple to implement. Convolution is applied to images of different widths and heights, a different number of times depending on the input size. The output of the convolution process is scaled accordingly [34].

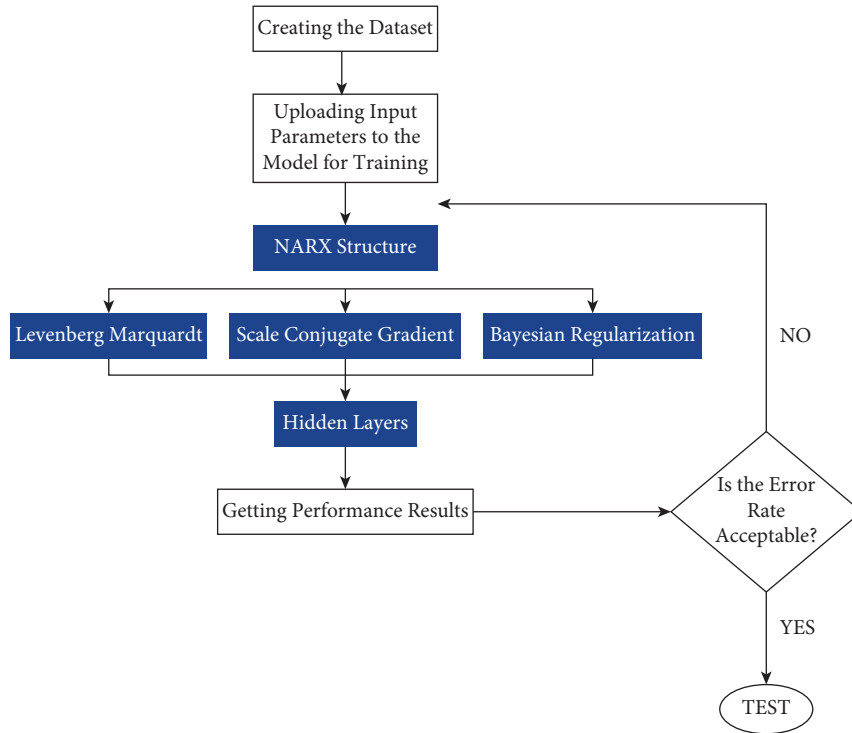


FIGURE 4: NARX algorithm flowchart.

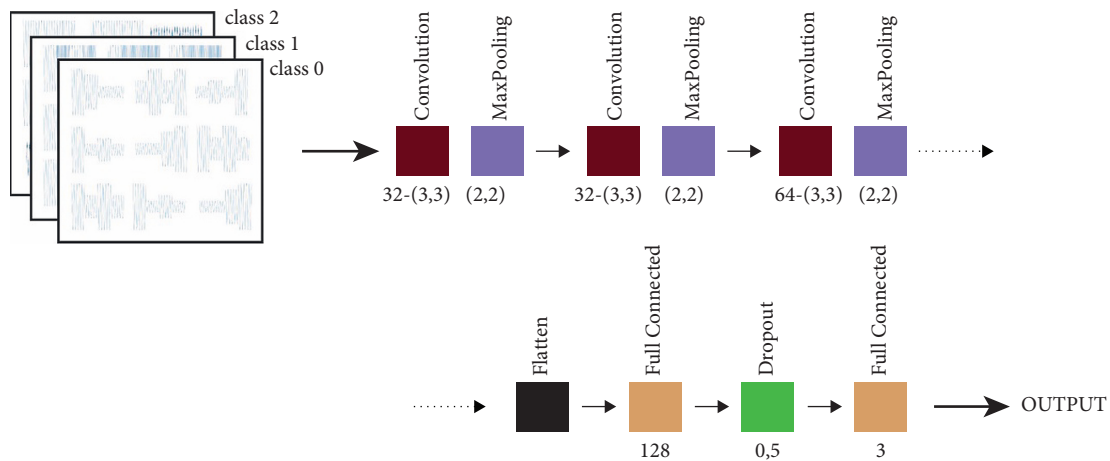


FIGURE 5: Proposed CNN structure in our study.

The dimensions are made the same thanks to the pooling layer that comes after the convolution layer, independent of the dimensions. In the classification made with CNN, images of the QASK, QFSK, and QPSK signals of 0 dB, 5 dB, 10 dB, and 15 dB noise ratios were classified into three different folders. Python libraries read these folders from their location and make them ready for the CNN algorithm. The operation system library has been used to read the images from the folders for processing. The images included in the algorithm were resized and brought to  $200 \times 200$  dimensions. This ensures that if the images are of different sizes, they are all the same format for CNN. Image matrices were saved, and classes were created by determining 0 for QASK, 1 for QFSK, and 2 for QPSK.

Then, the image matrices were assigned to a variable, and their labels were transferred to another variable and stored to be used for CNN.

CNN model was created using Keras. ReLU was chosen as the activation function in the CNN model. In the compiler of the model, the algorithm is run by selecting the loss function sparse categorical cross-entropy and Adam as the optimization function. Adam optimization is known as an algorithm that follows a probabilistic approach. Thus, it works fast for high-dimensional datasets [35]. Further, CNN has chosen as a good option for optimization problems for machine learning. As seen in the functional diagram of the layers belonging to the CNN model used, it is given. There are

many regularization methods in the CNN model. Besides dropout, data augmentation, normalization, and weight decay are some of them [36]. The sequential structure of the CNN model because of compiling the code in Python is given in Figure 6. While fitting the dataset to the model, 0.3 was determined as the validation split ratio, and 30% of the dataset was used for validation.

As can be seen in the CNN model used, there is a dropout layer, one of the regularization methods. In addition, data augmentation is also used. The rotation angle is 30 degrees, the zoom ratio is 0.2, the horizontal and vertical shift value is 0.1, and the horizontal and vertical random flip is chosen as true. The data augmentation stage is used to avoid overfitting and increase the model's predictive power.

Thanks to this stage, the quality of any input image or the angle of the objects in the image becomes insignificant. Since we have three classes and the targets are expressed in terms of the index (0, 1, 2), we used sparse categorical cross-entropy. The formula for it is

$$\text{CCE} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)], \quad (3)$$

where  $y_i$  represents true labels and  $\hat{y}_i$  represents predicted labels [37].

It was placed in two different folders to share the data as train and test. These two folders have different folders for every three classes. Each class comprises of 0 dB, 5 dB, 10 dB, and 15 dB signal images. These decibel values have 255 images. So, a class has 1020 images. Consequently, all classes have a total of 3060 images (Figure 7). Visualizing numbers of train and test data has a visualization of class numbers using the seaborn library. Learning rate, which is an important parameter in the training of the CNN model, is used as the update rate of the estimated weight errors. The weights in the hidden layers have updated the model. The purpose of this is to minimize the loss of function.

The magnitude of the learning rate also affects the speed of the training [38]. For example, a big ratio results in an unacceptable level of the loss function, while small level training is slow, and a very small amount of weight is updated [19]. In other words, the optimal learning rate to be selected for the model leads to approaching the best result.

**4.2. Demodulation Results Using Deep Learning and NARX Models.** The demodulation process using the NARX structure is carried out to obtain the information signal superimposed on the carrier. NARX consists of three parts: training, testing, and verification. Performance evaluation is made at the validation stage. At this stage, the success of the model is predicted according to the training and test results. 70% of the dataset was reserved for training, and 15% of the rest was divided for validation and 15% for testing. The dataset included in the algorithm from the MATLAB workspace and consisting of  $3060 \times 400$  dimensions was determined as target values, that is, information mark,

$3060 \times 1$  sized sequences from 1 to 255 for each signal type. Once the network is trained and used for prediction, the output is reported back to the network to obtain the forecast to generate the next prediction step. The serial-parallel model does the training by reducing the iteration time. The model is created for initial training by randomly assigning the weights to the initial training. Then, in each iteration, the model adjusts itself thanks to the feedback. The structure of the NARX model used is shown in Figure 8.

**4.3. The Obtained Results.** The model's results in training performed according to these three learning rates of 0.01 and 0.001 are shown in Figure 9. The epoch number is chosen as 20, and the process is performed according to the same epoch number. In consequence of compiling the model in Python, approximately 1.6 million trainable parameters were obtained. As can be seen in Figure 9, after the two learning rate trainings, the graphics are similar to each other. The training time was longer for the 0.0001 rates. The loss curve, which is one of the most used graphs for error detection in CNN models, gives us visual results to reveal the learning ability of the training process and the network. The lower the learning rate, the more delayed the convergence [19]. As can be seen in Figure 10, the convergence of the graph with a 0.001 learning rate was earlier. As it can be understood from the graphics in both Figures 9 and 10, there is no overfitting compatibility of the model in training. Although our dataset is not large enough, the result shows that the model is successful. According to the learning rates, the success of the model was 98.37% for 0.001, while it was 99.45% for 0.0001.

Figure 11 represents the confusion matrices of CNN models according to both learning rates. Using the CPU, the model's training took about 20 minutes for a learning rate of 0.001, compared with about 30 minutes at a rate of 0.0001. However, the training time for both operations performed when the GPU is used is less than one minute.

As it can be seen in the classification report in Figures 11 and 12, there has been a slight increase in the amount of data in the dataset as data increase is applied. Due to the classification report, individual precision, accuracy, and f-score values can be obtained for each class. These values have been revealed by using the sklearn library.

Figure 13 shows the closeness of demodulation estimation data obtained from modulated signals carrying 0-255 8-bit binary data in 0-5-10-15 dB noise of QASK, QFSK, and QPSK signals collectively. The first four signals belong to QASK 0-5-10-15 dB, the next four to QFSK 0-5-10-15 dB, and the last four to QPSK 0-5-10-15 dB. When looking at all modulated signals, it is understood that the estimation of demodulation estimation data of 5 dB noisy signals contains more errors than others.

Mean square error rate (MSE), one of the methods used for performance evaluation, provides information about the success of the neural network. Figure 14 shows the harmony between target values and estimated values. Therefore, the predictions here should be along the



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 120, 120, 32)	320
activation (Activation)	(None, 120, 120, 32)	0
max_pooling2d (MaxPooling2D)	(None, 60, 60, 32)	0
conv2d_1 (Conv2D)	(None, 60, 60, 32)	9248
activation_1 (Activation)	(None, 60, 60, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_2 (Conv2D)	(None, 28, 28, 64)	18496
activation_2 (Activation)	(None, 28, 28, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 64)	0
flatten (Flatten)	(None, 12544)	0
dense (Dense)	(None, 128)	1605760
activation_3 (Activation)	(None, 128)	0
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 3)	387
activation_4 (Activation)	(None, 3)	0

FIGURE 6: Sequential structure of our CNN Model.

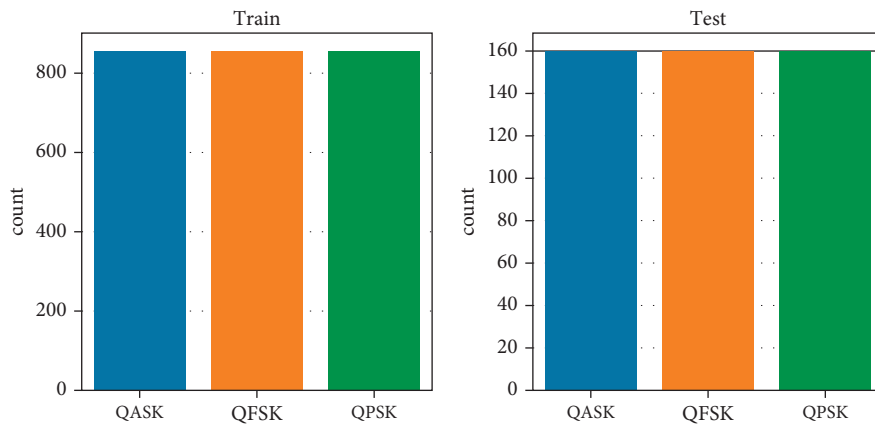


FIGURE 7: Visualizing numbers of train and test data.

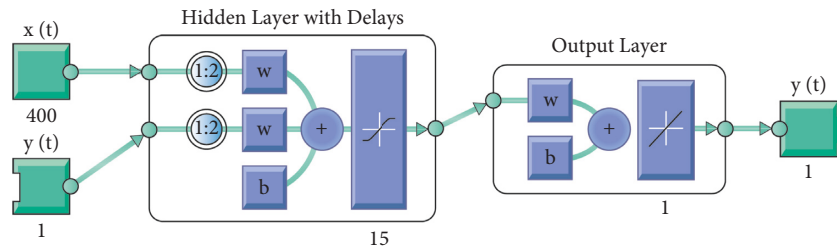


FIGURE 8: NARX neural network structure.

points expressed with dashed lines starting from the center point (0, 0).

As can be seen from Figure 14, the rate for validation is around 94%. Demodulation of noiseless signal images has been achieved with a high success rate. In addition, the epoch value is 107 for the lowest mean square error rate, as shown in Figure 15.

### 5. Discussion

As in the image given in Figure 16, individual test images for each of the three types of signals were loaded into the model and asked to classify. The model was assigned one for the images that made the class estimation correctly and zero for the others. Finally, the saved model was loaded and

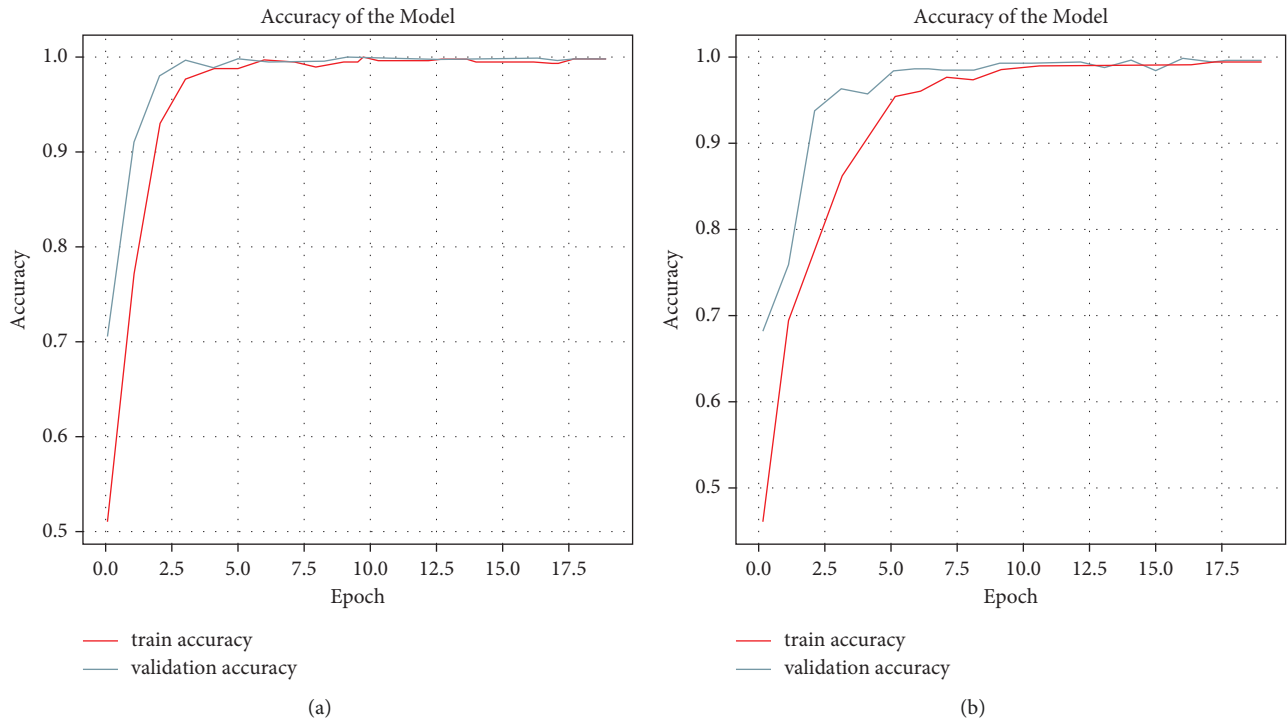


FIGURE 9: Training and test graphs according to the learning rates of the CNN model. (a) Learning rate graph selected as 0.001. (b) learning rate graph selected as 0.0001.

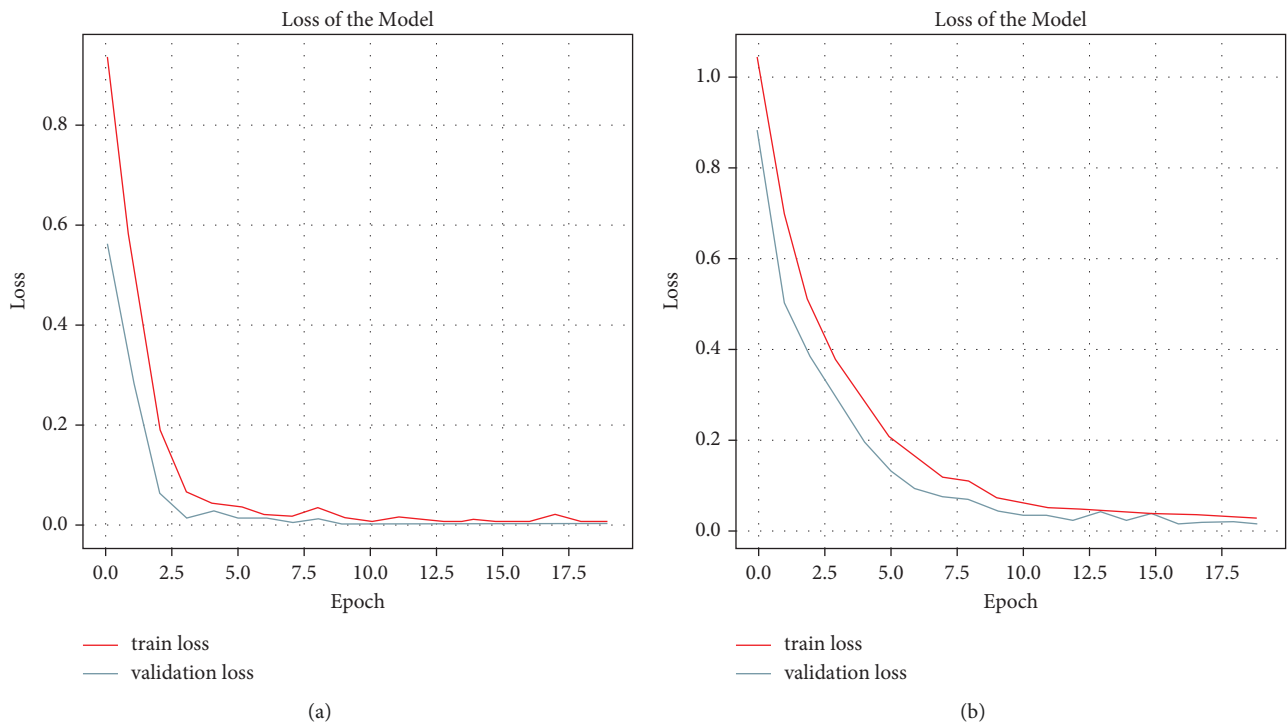


FIGURE 10: Loss plots according to train and test results of CNN model. (a) Learning rate graph selected as 0.001. (b) Learning rate graph selected as 0.0001.

run for test images. Test images tested anonymously are different from images produced using MATLAB. The purpose of this is to measure the model's reaction to other

images. Since the noiseless type of the QASK modulation signal is easy to detect, the noisy signal is used. The generated model is not affected by noise ratios, although the

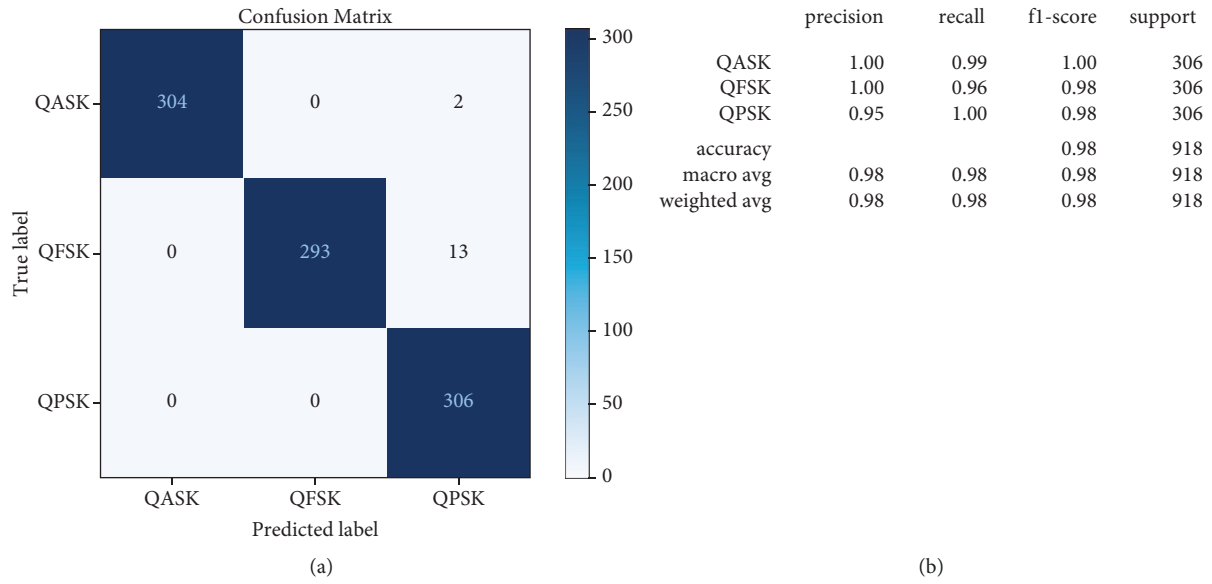


FIGURE 11: Model output for 0.001 learning rate. (a) Confusion matrices. (b) Classification report.

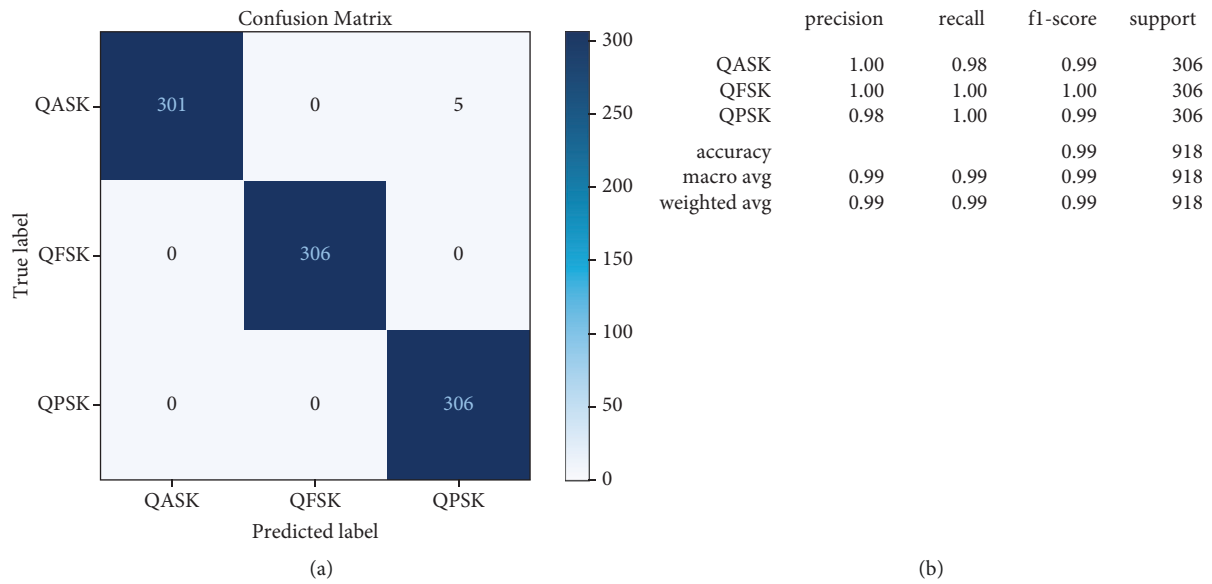


FIGURE 12: Model output for 0.0001 learning rate. (a) Confusion matrices. (b) Classification report.

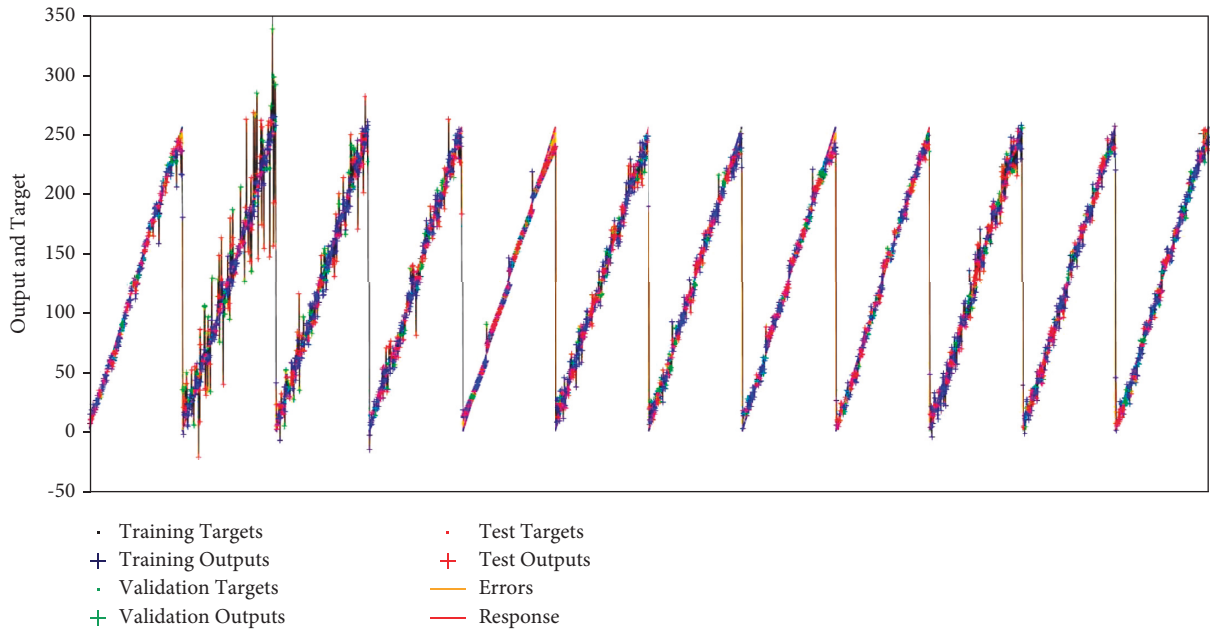


FIGURE 13: Response to output values.

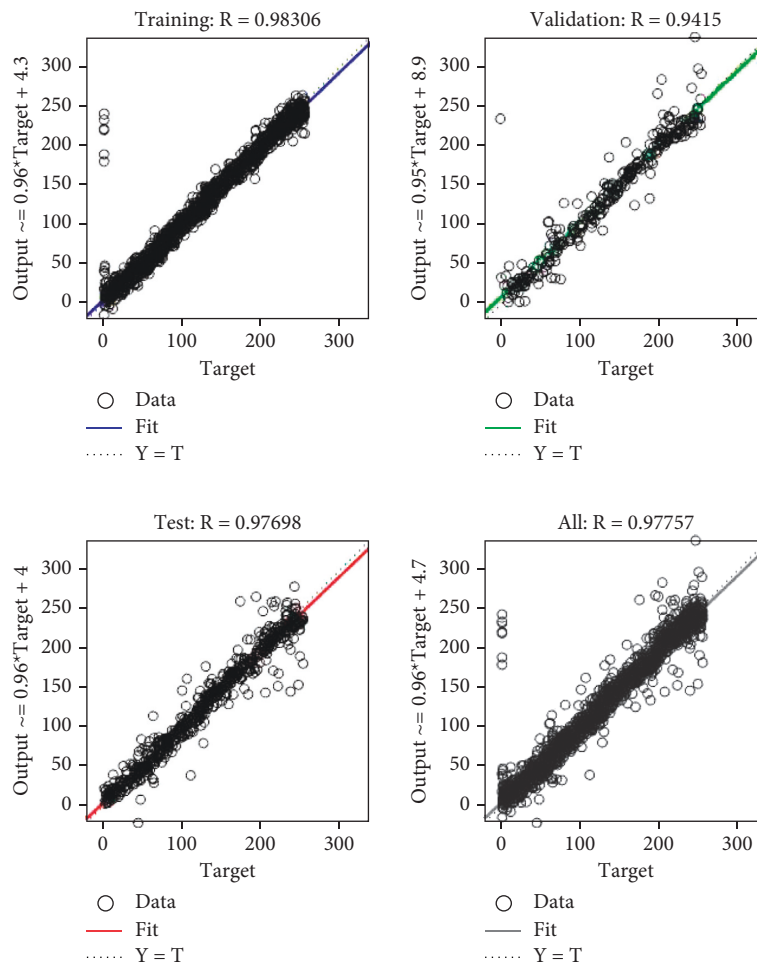


FIGURE 14: Error value graphics of the model's output.

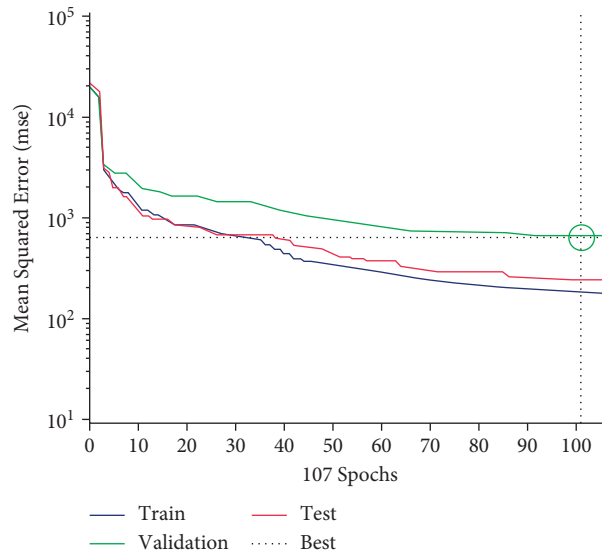
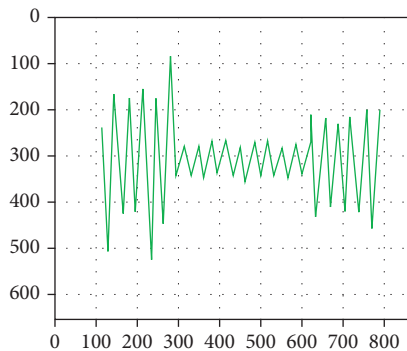


FIGURE 15: Best validation performance at epoch value. During these processes, Intel(R) Core i7-10750H, 16.0 GB RAM, 4 GB GDDR6 GTX1650, and 64-bit Windows 10 PC were used.

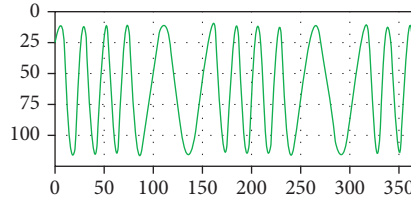
['QASK', 'QFSK', 'QPSK']

[[1. 0. 0.]]



['QASK', 'QFSK', 'QPSK']

[[0. 1. 0.]]



['QASK', 'QFSK', 'QPSK']

[[0. 0. 1.]]

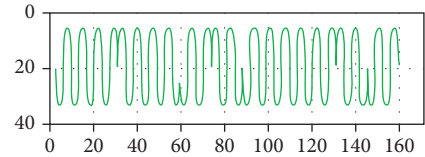


FIGURE 16: Test output of the model.

noise makes it difficult to distinguish the modulation signal type.

While demodulation of noiseless signals during the demodulation phase is easy, the algorithm has been challenged, especially for signals with 5 dB noise. One of the reasons for this is seen as the amount of data in the dataset. Therefore, increasing the amount of data or applying feature extraction can achieve higher success in obtaining the information signal from noisy signals.

## 6. Conclusions

The study used JPEG format images of the signals of digital modulation types QASK, QFSK, and QPSK with 0 dB, 5 dB, 10 dB, and 15 dB noise. The CNN model created using Python libraries has also been successful in 5 dB noisy signals. The model can be used for communication systems because it is challenging for the human eye to distinguish the images of signals with a 5 dB noise ratio, thanks to the

success rate of up to %99, the model can be used for communication systems. In other words, even if the amplitude, frequency, and phase values of the digital modulation signal are unknown, the signal's type can be easily understood.

Most of the studies for AMC are made from actual data of modulation signals. However, it may not always be possible to have all the data of a modulation signal. Modulation classification and demodulation, making it independent of a snapshot and its structure, size, shooting angle, or different contrasts, can provide great convenience. Therefore, modulation classification from images has yielded successful results thanks to this model without extra hardware or software. It has been demonstrated that the model has high success in terms of both training time and accuracy.

As a result of demodulation using a single algorithm, a beneficial effect has been obtained regarding time and cost. Furthermore, in areas where fast and secure communication

is essential, communication is also critical to security without different digital devices. Thus, the development of the algorithm will benefit users in areas where communication is vital.

## Data Availability

The datasets are available upon request from the authors.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Authors' Contributions

All authors contributed equally to this work. Additionally, all authors have read and approved the final manuscript and given their consent to publish the article.


## References

- [1] M. Zhang, Y. Zeng, Z. Han, and Y. Gong, "Automatic modulation recognition using deep learning architectures," *IEEE Work. Signal Process. Adv. Wirel. Commun. SPAWC*, IEEE, vol. 2018, pp. 1–5, 2018- June.
- [2] S. Ertürk, *Sayısal Haberleşme*, Birsen Yayınevi, İstanbul, 1st edition, 2005.
- [3] F. Xiong, *Digital Modulation Techniques*, First, Norwood, MA USA, 2000.
- [4] S. Ma, J. Dai, S. Lu et al., "Signal demodulation with machine learning methods for physical layer visible light communications: prototype platform, open dataset and algorithms," *IEEE Access*, vol. 7, pp. 30588–30598, 2019.
- [5] D. Wang, M. Zhang, Z. Li et al., "Modulation format recognition and OSNR estimation using CNN-based deep learning," *IEEE Photonics Technology Letters*, vol. 29, no. 19, pp. 1667–1670, 2017.
- [6] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals," *IEEE Transactions on Communications*, vol. 46, no. 4, pp. 431–436, 1998.
- [7] Y. Xu, D. Li, Z. Wang, Q. Guo, and W. Xiang, "A deep learning method based on convolutional neural network for automatic modulation classification of wireless signals," *Wireless Networks*, vol. 25, no. 7, pp. 3735–3746, 2019.
- [8] A. Ali and F. Yangyu, "Unsupervised feature learning and automatic modulation classification using deep learning model," *Physical Communication*, vol. 25, 2017.
- [9] J. H. Lee, K. Y. Kim, and Y. Shin, "Feature image-based automatic modulation classification method using CNN algorithm," *1st Int Conf Artif Intell Inf Commun ICAIIC*, vol. 2019, pp. 560–3, 2019.
- [10] N. Daldal, Z. Cömert, and K. Polat, "Automatic determination of digital modulation types with different noises using Convolutional Neural Network based on time-frequency information," *Applied Soft Computing*, vol. 86, Article ID 105834, 2020.
- [11] R. Zhou, F. Liu, and C. W. Gravelle, "Deep learning for modulation recognition: a survey with a demonstration," *IEEE Access*, vol. 8, pp. 67366–67376, 2020.
- [12] R. G. Gallager, *Principles of Digital Communication*, Cambridge University Press, Cambridge uk, 2007.
- [13] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [14] Y. Lecun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of the IEEE Int Symp circuits Syst*, pp. 253–256, IEEE, Paris, France, 30 June 2010.
- [15] Y. Bengio and Y. Lecun, *Convolutional Networks for Images Speech and Time-Series*, MIT Press, Cambridge, MA, USA, 1997.
- [16] H. C. Shin, H. R. Roth, M. Gao et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.
- [17] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: a comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017.
- [18] M. V. Valueva, N. N. Nagornov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Mathematics and Computers in Simulation*, vol. 177, pp. 232–243, 2020.
- [19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016.
- [20] B. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Proc 25th Int Conf Neural Inf Process Syst*, vol. 25, no. 2, 2012.
- [21] A. V. Joshi, *Machine Learning and Artificial Intelligence*, Springer, Salmon Tower NY USA, 2020.
- [22] Y. T. Zhou and R. Chellappa, "Computation of Optical Flow Using a Neural Network," in *Proceedings of the IEEE 1988 International Conference on Neural Networks*, 24 July 1988.
- [23] W. Ballard, *Hands-On Deep Learning for Images with TensorFlow Build Intelligent Computer Vision Applications Using TensorFlow and Keras*, Packt Publishing Ltd, Birmingham, UK, 2018.
- [24] A. Zhang, Z. C. Lipton, M. Li, and J. Smola, *Dive into Deep Learning*, 2021, <https://arxiv.org/abs/2106.11342>.
- [25] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *52nd Annu. Meet. Assoc. Comput. Linguist. ACL 2014 - Proc. Conf*, vol. 1, pp. 655–665, 2014.
- [26] S. H. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," *Neurocomputing*, vol. 378, pp. 112–119, 2019.
- [27] H. Wu and X. Gu, "Towards dropout training for convolutional neural networks," *Neural Networks*, vol. 71, pp. 1–10, 2015.
- [28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Lepton spectra as a measure of b quark polarization at LEP," *Journal of Machine Learning Research*, vol. 15, p. 1929, 2014.
- [29] T. Tsung-Nan Lin, C. L. Giles, B. G. Horne, and S. Y. Kung, "A delay damage model selection algorithm for NARX neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2719–2730, 1997.
- [30] L. Gonzaga, B. Ruiz, M. P. Cuéllar, M. D. Calvo-flores, M. Del, and C. Pegalajar, "An application of non-linear autoregressive neural public buildings," *Energies*, vol. 9, pp. 1–21, 2016.
- [31] M. Ardalani-Farsa and S. Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid Elman-NARX neural networks," *Neurocomputing*, vol. 73, no. 13-15, pp. 2540–2553, 2010.

- [32] Z. Boussaada, O. Curea, A. Remaci, H. Camblong, and N. M. Bellaaj, "A non-linear autoregressive exogenous (NARX) neural network model for the prediction of the daily direct solar radiation," *Energies*, vol. 11, 2018.
- [33] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA*, pp. 806–813, March 2014.
- [34] H. Wang and B. Raj, *On the Origin of Deep Learning*, pp. 1–72, 2017, <https://arxiv.org/abs/1702.07800>.
- [35] D. P. Kingma and J. L. Ba, "ADAM: a method for stochastic optimization," in *Proceedings of the 3rd Int Conf Learn Represent, DBLP, San Diego CA USA*, 7 May 2015.
- [36] C. Farabet, *Towards Real-Time Image Understanding with Convolutional Networks Clement*, Université Paris-Est, Marne, France, 2013.
- [37] P. Meletis and G. Dubbelman, "On boosting semantic street scene segmentation with weak supervision," *IEEE Intell Veh Symp Proc*, vol. 2019, pp. 1334–1339, 2019.
- [38] B. Schölkopf and A. J. Smola, *Learning with Kernels Support Vector Machines, Regularization, Optimization, and beyond*, The MIT Pres, Cambridge, MA UK, 2001.

## Research Article

# Microalgae Classification Using Improved Metaheuristic Algorithm

Liu Wei,<sup>1</sup> Su XiaoPan,<sup>2</sup> and Faezeh Heydari <sup>3</sup>

<sup>1</sup>Gannan University of Science & Technology, Ganzhou, Jiangxi 341000, China

<sup>2</sup>Ganzhou 851, Ganzhou, Jiangxi 341000, China

<sup>3</sup>Faculty of Engineering, Saveh Branch Islamic Azad University, Saveh, Iran

Correspondence should be addressed to Faezeh Heydari; faezeh.heydari@t-mco.ir

Received 16 March 2022; Revised 5 April 2022; Accepted 21 April 2022; Published 11 May 2022

Academic Editor: Ramin Ranjbarzadeh

Copyright © 2022 Liu Wei 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.

Microalgae are present at all levels of nutrients and food networks, so in aquatic environments they are an important part of the food chain of aquatic organisms, which also play an important role as biological purifiers of water resources and regulation. They also affect the pH of the environment; also plants are the only organisms capable of synthesizing long-chain fatty acids. Therefore, microalgae are the supplier and primary source of unsaturated fatty acids (PUFA) for all organisms present in the food chain of the aquatic environment. It should be noted that many microalgae are also biological indicators of water and reflect the ecological status of the environment. Precise classification of microalgae is related to the human observation capability. The present study proposes a new optimized classification technique with higher accuracy to provide a computer-aided classification of the microalgae. The method begins with an image segmentation to determine the region of interest. The segmentation part has been optimized by a new metaheuristic to provide higher accuracy. Then, the features have been extracted and fed to a Support Vector Machine (SVM) for final classification. The comparison results of the proposed method with some other methods show that the proposed method with 0.828 Kappa, and 0.342 and 0.855 min and max value of F1, provides the highest accuracy compared to the others.

## 1. Introduction

Long-chain polyunsaturated fatty acids (LC-PUFAs) such as eicosapentaenoic acid (EPA) and docosahexaenoic acid (DHA) are among the essential components of various metabolic processes in human life [1]. There are significant quantities of PUFAs in oily fish products that are used to prevent heart disease and cancer [2]. Some efforts have been made to decrease the expense of these products by replacing fish oil ingredients with microalgae products. In the *C. Cohn* microalgae life cycle, two distinct types are considered: motile and nonmotile cells (cysts). Furthermore, the varying sizes of the microalgae signify different phases of the cell life cycle. Small cells, for example, represent a young offspring and cysts represent an adult cell. Motile cells typically emerge in an oval form, while nonmotile cells are usually circular and appear larger than motile cells. Since microalgae are the main manufacturer in the aquatic food chain and

oxygen maker in the aquatic environment; it is a substantial microscopic aquatic life form [3]. Algae are considered as a biological index in water resource management to show water quality due to their sensitivity to environmental variations. This turns microalgae detection into a significant subject in water resource management. However, this purpose takes a lot of time due to the need for expert biologists to finish it. One simpler technique for reaching this aim is to use image processing [4]. This study presents a new method for automatic segmentation of microalgae in microscopic images. However, microalgae have a high significance; there are a small number of research works about them. For example, OSA et al. [5] proposed a method for the segmentation of green microalgae images based on region growth. The method was started by a morphological feature extraction followed by a taxonomical classification of the species. The method is based on the seeded region growing principle and a fine-tuned filtering preprocessing stage,



which are used for smoothing the input image. Simulation results indicated that the proposed method provides proper precision for green microalgae image segmentation compared with some state-of-the-art methods.

Giraldo-Zuluaga et al. [6] proposed an automatic detection system for *Scenedesmus* polymorphic microalgae in microscopic images. In the study, the image was first pre-processed by contrast equalization and then the image was segmented by the active contours. The statistical features evaluation with texture features was presented for algae description.

Wei et al. [7] proposed a method for improving the hyperspectral microscopic imaging system to provide hyperspectral images of microalgae samples over pretreated steps. Afterward, the Fisher algorithm was utilized for microalgae species identification. For analyzing the ability of the method, it is assessed in terms of sensitivity and specificity and the results showed high resolution for these two measurement indicators.

Final results indicated that using the proposed method provides fast and suitable results.

Liao et al. [8] presented a method based on deep learning to identify marine microalgae. The study used a convolutional neural network (CNN) to classify the marine microalgae. The study addressed two collections of classification experiments, where one group categorized the algae into 5 classes based on the family category, and the other one categorized the algae into 11 classes based on species category. Simulation results indicated that the polarization information developed using the high and robust classification precision of the proposed method for low-resolution microalgal images.

Xu et al. [9] proposed an identification methodology for microalgae images based on transmission hyperspectral microscopic imaging and transmission hyperspectral microscopic imager (THMI). Hyperspectral imaging (HSI) of three species of microalgae was proposed to confirm their absorption features. Principal Component Analysis (PCA) and peak ratio algorithms are used to analyze the transmission spectra for dimensionality reduction and feature extraction, and finally, the Support Vector Machine (SVM) model was employed for classification. Simulation results indicated that the proposed method has a good potential for the classification of microalgae.

As can be observed from the literature, lots of works were performed to provide a proper identifier for the microalgae. However, several works were done for this purpose; the recent works showed that using optimized techniques, especially metaheuristic algorithms, provides higher efficiency for microalgae classification. This motivates us to provide a new modified metaheuristic to use as an efficient tool for the classification. Therefore, in this research, we worked on a high-ability classifier for the identification of the microalgae in the images. The main contributions of this study can be highlighted as follows: (1) Providing a new technique for better segmentation of microalgae. (2) Designing a new improved version of the metaheuristic technique, called Quantum Thermal Exchange Optimization (QTEO) algorithm for this study. (3) Using the designed technique to

optimize Kapur's entropy thresholding method. (4) Validation of the proposed method with some state-of-the-art methods. In the next sections, the method for this purpose has been described and validated in detail.

## 2. Histogram Equalization

Image histogram equalization is one of the image processing methods that is used to better display the output image or for further analysis [10–16]. A simple example of histogram equalization for microalgae images is shown in Figure 1.

As can be observed from Figure 1 that the probability density value for the equalized image is more distributed than the original image. This can be so helpful for image segmentation and extracting helpful features from the images.

## 3. The Quantum-Based Thermal Exchange Optimization Algorithm

**3.1. Conception.** Optimization in engineering and science refers to selecting the best member from a set of achievable members [17, 18]. In the simplest form, an attempt is made to systematically select data from an achievable and computable set to obtain the optimal value (minimum or maximum) of a real function [19]. Many optimization problems in engineering are naturally more complex and difficult than can be solved by conventional optimization methods such as dynamic programming methods and the like [20]. Several optimization problems today, including nonpolynomial (NP-Hard) problems, are approximately solvable with existing computers [21]. One of the solutions to deal with such problems is the use of approximate and inexact algorithms [22]. These algorithms do not guarantee that the solution obtained is optimal and only with a lot of time can a relatively accurate solution be obtained, and, depending on the time spent, the accuracy of the solution changes [23]. One way to solve these kinds of problems is to consider all possible solutions and calculate the objective functions associated with them and, finally, select the best solution [24]. It is clear that the method of complete counting ultimately leads to an accurate solution to the problem, but in practice, it is impossible to use it due to a large number of possible solutions [25]. Due to the problems related to the complete counting method, there has always been an emphasis on creating more effective and efficient methods [26]. In this regard, various algorithms have been developed, for example, Genetic Algorithm (GA) [27], Ant Lion Optimizer (ALO) algorithm [28], equilibrium optimizer [29], Mayfly Optimization Algorithm (MOA) [30], and World Cup Optimization (WCO) algorithm [31].

Recently, a new metaheuristic, called Thermal Exchange Optimization (TEO) algorithm, has been proposed, which is derived by the temperature behavior of the objects and their positions and their variations between cold and warm parts [32]. In this study, a new improved version of the TEO algorithm, called the Quantum-based TEO (QTEO) algorithm, has been proposed to improve the precision and consistency of the original method. The main

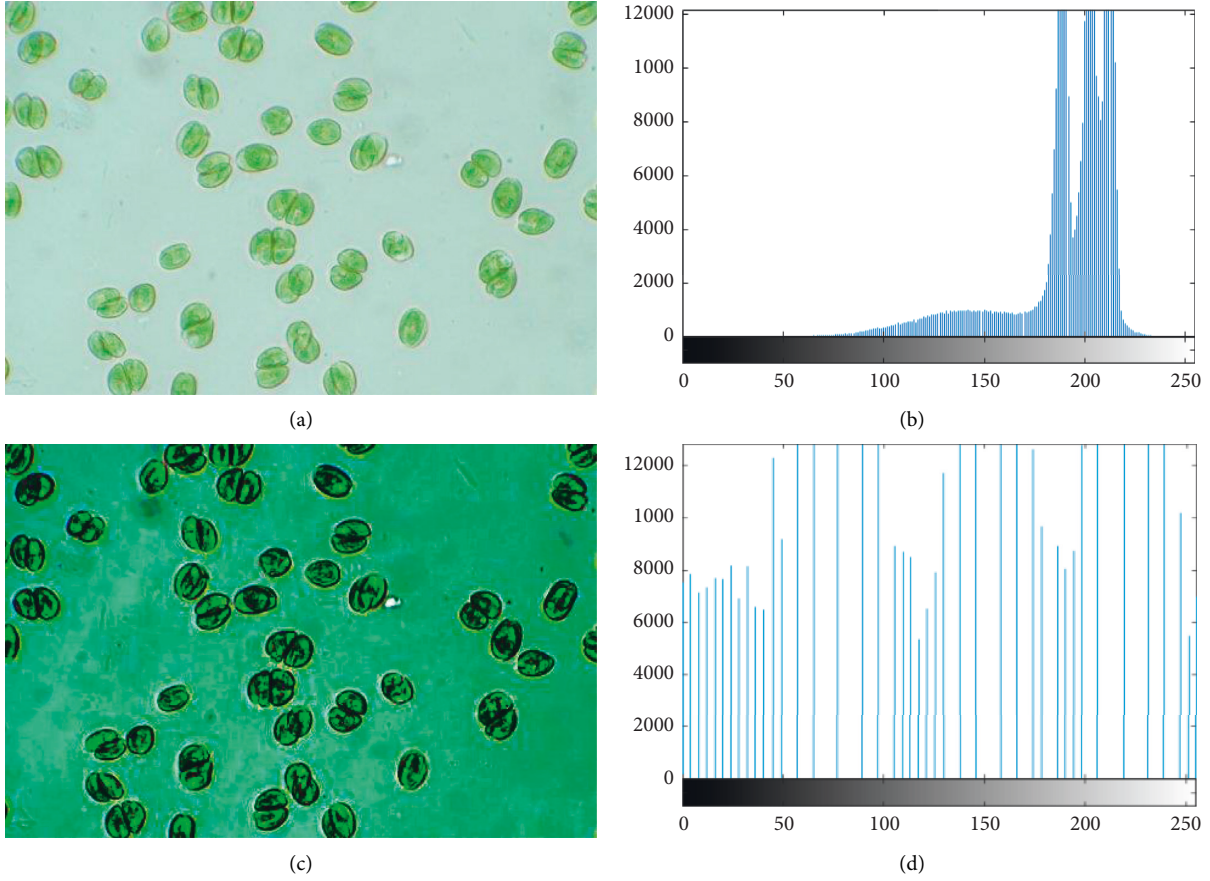


FIGURE 1: Image histogram equalization of a sample microalgae image for the original image (a) and its histogram (b) and the equalized image (c) and its histogram (d).

target is to utilize this new algorithm for microalgae segmentation. The main conception of the method is based on the Newton law of cooling. This law explains that the body temperature ratio varies proportionally around the body temperature. If the temperature variation in the body and its neighborhood is small, the mean value of the heat exchanged between the body and its neighborhood is almost proportional to the difference in temperature of the body and the environment which is due to the convection, infrared radiation, and conduction. The law of cooling is achieved by the following equation [33]:

$$\frac{dQ}{dt} = \alpha \times A \times (T_s - T_a), \quad (1)$$

where  $\alpha$  describes the coefficient of the heat transfer,  $Q$  signifies the heat,  $A$  states the area surface which transmits heat, object geometry, and surface state, and  $T_a$  and  $T_b$  represent the ambient and body temperatures.

The losing time is  $\alpha \times A \times (T_a - T)dt$  which determines the temperature variation in the reserved heat as the temperature falls  $dT$ , i.e.,

$$V \times c \times \rho \times dT = -\alpha \times A \times (T - T_b)dt, \quad (2)$$

where  $\rho$  signifies the density ( $kg/m^3$ ),  $V$  describes the volume ( $m^3$ ), and  $c$  defines the specific heat ( $J/kg/K$ ).

Henceforth [33],

$$\frac{T - T_b}{T_M - T_b} = \exp\left(\frac{-\alpha \times t \times A}{\rho \times V \times c}\right), \quad (3)$$

where  $T_M$  represents the early high temperature. The above equation is correct when  $\alpha \times A \times t / V \times \rho \times c$  does not depend on  $T$ :

$$\zeta = \frac{\alpha \times A}{V \times \rho \times c}. \quad (4)$$

Therefore, by considering  $\zeta$  as a constant,

$$\frac{T - T_b}{T_M - T_b} = \exp(-\zeta t). \quad (5)$$

Consequently,

$$T = (T_M - T_b) \times \exp(-\gamma t) + T_b. \quad (6)$$

**3.2. The Algorithm.** Based on the Thermal Exchange Optimization (TEO) algorithm, some solution candidates are assumed as cooling ingredients and the remaining candidates are considered as the environment. Afterward, this process is established reversely.

The TEO algorithm starts with initializing a certain number of randomly distributed candidates as the solution candidates. This is done by the following equation [34]:

$$T_i^0 = T_{\min} + \delta \times (T_{\max} - T_{\min}) \quad i = 1, 2, \dots, n, \quad (7)$$

where  $\delta$  describes a random value in the range  $[0, 1]$ ,  $T_i^0$  represents the algorithm primary population for the  $i^{\text{th}}$  object, and  $T_{\min}$  and  $T_{\max}$  represent the minimum and maximum limitations. The candidates are then evaluated on the cost function to verify their ability. Afterward,  $T$  number of best candidate vectors are saved in Thermal Memory (TM) to provide higher efficiency of the algorithm with lower complexity. Some of these candidates are then added up to the main candidates and the same number of them in the individuals has been eliminated. The candidates contain two equal types, including environment and the heat and cooling transfer objects. Figure 2 shows this state.

As can be seen from Figure 2, the candidates contain two equal types, including environment and the heat and cooling transfer objects. Assume  $T_1$  as environment object for  $T_{n/2+1}$  cooling object and vice versa. If  $\zeta$  is greater than the object, the temperature exchanges steadily.  $\zeta$  is mathematically achieved by the following:

$$\gamma = \frac{\text{Cos}(\text{object})}{\text{Cos}(\text{worst object})}. \quad (8)$$

To model the effect of time in the algorithm, the following formula has been considered:

$$t = \frac{\text{iteration}}{\text{Max.iteration}}. \quad (9)$$

For establishing the global searching, the environmental temperature variation is considered that can be assumed by the following equation:

$$T_i^e = (1 - (m_1 + m_2 \times (1 - t) \times \text{rand})) \times T_i^e, \quad (10)$$

where  $T_i^e$ , defines the preceding object temperature modified by  $T_i^e$ , and  $m_1$  and  $m_2$  describe the control variables, respectively.

By considering the previous models, the new object temperature is mathematically updated as follows:

$$T_i^+ = T_i^e + (T_i^{\text{old}} - T_i^e) \exp(-\zeta t). \quad (11)$$

Finally, the term  $Pr$  is a component to show the cooling of the objects.

The  $Pr$  candidates are compared with randomly distributed values,  $R(i)$  in the range  $[0, 1]$ .

In this situation, if  $R(i) < Pr$ , one dimension of the  $i^{\text{th}}$  the candidate is randomly chosen and the value is rephrased as follows:

$$T_{i,j} = T_j^{\min} + \text{rnd}(T_j^{\max} - T_j^{\min}) \exp(-\zeta t), \quad (12)$$

where  $T_{i,j}$  represents the  $j^{\text{th}}$  variable of the candidate number  $i$  and  $T_j^{\min}$  and  $T_j^{\max}$  describe the lower and the upper limitations of the  $j^{\text{th}}$  variable, respectively.

The algorithm is stopped if the termination condition has been reached.

3.3. *The Quantum Thermal Exchange Optimization (QTEO) Algorithm.* However, Thermal Exchange Optimization (TEO) algorithm is a well-organized optimization technique; it is sometimes stuck in the local optimum solution. This study uses a quantum theory-based modification to resolve this issue as is possible. Since the quantum position is not instantaneously definite, it should be determined by a wave function  $\phi(X, t)$ , where the candidates' position is known by  $X$ . This states that the square mode defines the density probability of the individual in space at position  $X$ . The equation for this case can be formulated as follows:

$$|\phi|^2 dx dy dz = Q dx dy dz, \quad (13)$$

where  $Q$  signifies the probability density function to describe the following conditions:

$$\int_{-\infty}^{\infty} |\phi|^2 dx dy dz = \int_{-\infty}^{\infty} Q dx dy dz = 1. \quad (14)$$

The formulation for the position of the candidates is achieved by the following:

$$P(t) = \delta \times P_b(t) + (1 - \delta) \times P_g(t),$$

$$m_b(t) = \frac{1}{M} \sum_{i=1}^M P_{b_i}(t),$$

$$D(t+1) = 2\beta \times |m_b(t) - X(t)|, \quad (15)$$

$$\beta = \underline{\beta} - (\underline{\beta} - \bar{\beta}) \times \left( \frac{t}{G_{\max}} \right),$$

$$X(t) = P(t) \pm \left( \frac{D}{2} \right) \times \ln \left( \frac{1}{\gamma} \right).$$

$M$  defines the population size,  $P(t)$  indicates the local attraction area for the  $t^{\text{th}}$  iteration,  $m_b(t)$  signifies mean value of the optimal situation,  $D$  determines the weighted distance between the individuals and the mean optimal situation of the population,  $G_{\max}$  describes the maximum number of iterations,  $P_g$  and  $P_b$  represent the global optimal positions and the candidate position, respectively,  $\delta$  and  $\gamma$  provide two stochastic values in the range  $[0, 1]$ , and  $\beta$  signifies the shrinkage-expansion coefficient in the range  $\beta$  to  $\bar{\beta}$ . Here,  $\underline{\beta}$  and  $\bar{\beta}$  are 1 and 0.5, respectively. Based on the above explanations, the initial position can be updated as follows:

$$T_i^0 = P(t) + \beta \times |m_b - x^k| \times \ln \left( \frac{1}{\gamma} \right). \quad (16)$$

## 4. Segmentation of Microalgae

4.1. *Concept of Entropy Criterion (Kapur's Method).* Image segmentation is one of the most important basic operations for image analysis. The purpose of segmenting images is to divide the image into unequal areas so that these areas are not shared. Image segmentation is used in many applications of image processing and computer vision, especially medical imaging. In recent years, many methods have been proposed for image segmentation. These methods can be divided into

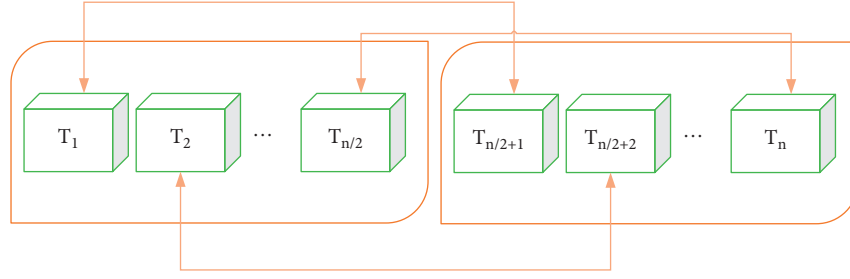


FIGURE 2: The structure of the candidates in the TEO algorithm.

two general categories of supervised and unsupervised methods. The unsupervised method is preferred in real-time applications because it does not require manual segmentation. Thresholding is one of the most important methods in image segmentation. The main goal in threshold-based methods is to find a threshold value for two-level thresholds and a few threshold values for multilevel thresholds. In two-level thresholding, only one threshold value is selected and the image pixels are divided into two groups. A popular method for thresholding is Kapur thresholding. The Kapur thresholding is a nonparametric technique to provide optimal threshold values based on image entropy and the histogram of the considered image. The key idea is to maximize the image entropy to get the optimal threshold value. To get this end, it assumes  $TH$  as a vector of the image threshold, i.e. [35],

$$TH = [th_1, th_2, \dots, th_{m-1}]. \quad (17)$$

In this situation, the Kapur entropy method is defined by the following equation:

$$J_{\max}(TH) = \sum_{i=1}^m H_i^c, \quad (18)$$

$$c = \begin{cases} 1, 2, 3, & \text{f RGB scale image,} \\ 1, & \text{if Gray scale image.} \end{cases}$$

$H_i^c$  describes the  $i^{th}$  entropy of the image and is obtained as follows:

$$H_1^c = \sum_{i=1}^{th} \frac{ph_i^c}{\omega_0^c} \ln\left(\frac{ph_i^c}{\omega_0^c}\right),$$

$$\vdots$$

$$H_m^c = \sum_{i=th+m}^L \frac{ph_i^c}{\omega_{m-1}^c} \ln\left(\frac{ph_i^c}{\omega_{m-1}^c}\right), \quad (19)$$

where  $\omega_i^c$  describes the  $i^{th}$  histogram for  $c_i$ , and  $ph_i^c$  signifies the histogram of the image intensity levels.  $\ln(\cdot)$  defines the natural logarithm.

This study uses the formerly explained new designed Quantum-based Thermal Exchange Optimization Algorithm to achieve the optimal threshold values by threshold points tuning until  $J_{\max}$  is met.

**4.2. Image Thresholding Using Quantum-Based Thermal Exchange Optimization Algorithm.** Each one of the candidates in the proposed algorithm shows a decision variable for the

threshold values and image segmentation. The population can be indicated as follows:

$$AT = [x_1^c, x_2^c, \dots, x_{\text{candidate}_m}^c]^T, \quad (20)$$

$$x_i^c = [th_1^c, th_2^c, \dots, th_m^c],$$

$$c = 1, 2, 3,$$

where  $AT$  describes the population size,  $x_i$  signifies the  $i^{th}$  element of  $AT$ ,  $T$  indicates the transpose sign, and  $c$  defines set for RGB images while  $c = 1$  is selected for grayscale images. The boundaries of the search space in the present study are between 0 and 255, which point to the image intensity levels.

**4.3. Image Segmentation Evaluation.** The proposed QTEO-based segmentation method is applied on MATLAB R2017b platform and based on a laptop with Intel®, CoreTM i7, and 16 GB RAM. The parameters of the proposed QTEO-based segmentation method are initiated with following values: the initial numbers of continents are selected as 5 and the number of individuals is set as 100. The value of the  $m_1$  and  $m_2$  is set to be 2, the dimension of the search space is  $Th$  (chosen threshold), and the maximum number of iterations is set 200. The algorithm will be terminated after maximizing the objective function ( $J_{\max}$ ). To obtain a fair analysis, the method has been performed 30 times iteratively and the mean value is considered as the optimal threshold. In the present study, several databases from the Internet are tested and assessed to determine the performance of the suggested segmentation methodology. Figure 3 and 4 show the image segmentation based on QTEO-Kapur.

As can be observed from figures, the method can provide useful results for the segmentation of the microalgae.

## 5. Feature Extraction

The next step is to use a feature extraction to extract the segmented image features for classification of the image with higher accuracy and less time. In this study, the Gray-Level Cooccurrence Matrix (GLCM) is utilized as the feature extraction method. Tissue characteristics were evaluated by tissue statistical analyses over the statistical distributions of grayscale compounds observed in a specific position about each other. The GLCM is a matrix with identical numbers of rows and columns that shows the grayscale levels of the image. In other words, if an image gray number is up to  $P$ ,

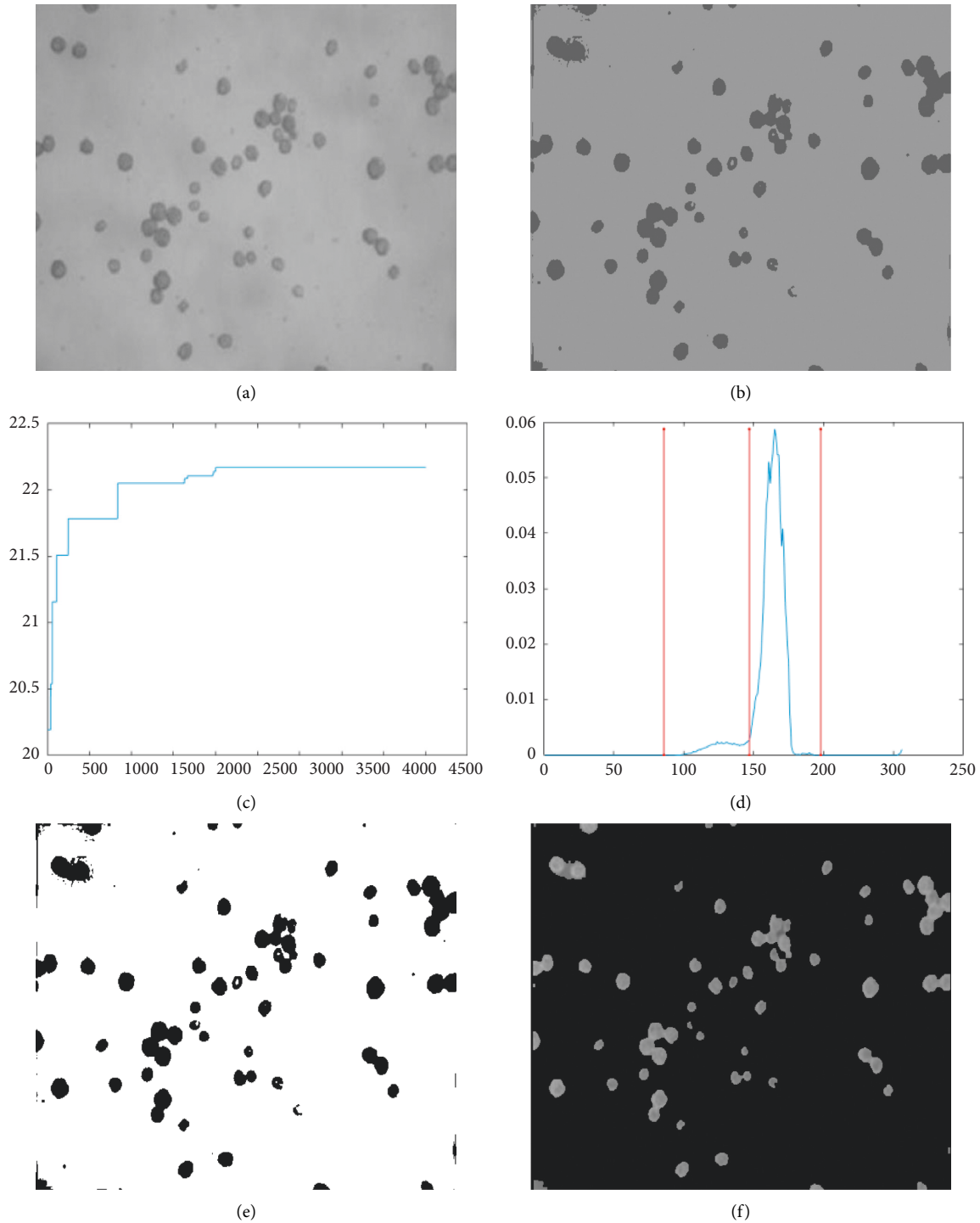


FIGURE 3: Image segmentation based on QTTEO-Kapur: (a) The original image, (b), the segmented image based on the proposed method with (number of thresholds = 3,) (c) the convergence profile of the fitness function, (d) the histogram of (B), (e) the binarized image, and (f) final segmented image.

the dimensions of the simultaneous event matrix will be  $P \times P$  matrix. The current study assessed the features including entropy, contrast, homogeneity, energy, and correlation. The formulation of these features is given below. The correlation feature is achieved as follows:

$$\text{corr} = \sum_{i=1}^M \sum_{j=1}^N \frac{p(i, j) - \mu_r \mu_c}{\sigma_r \sigma_c}. \quad (21)$$

The correlation feature is achieved as follows:

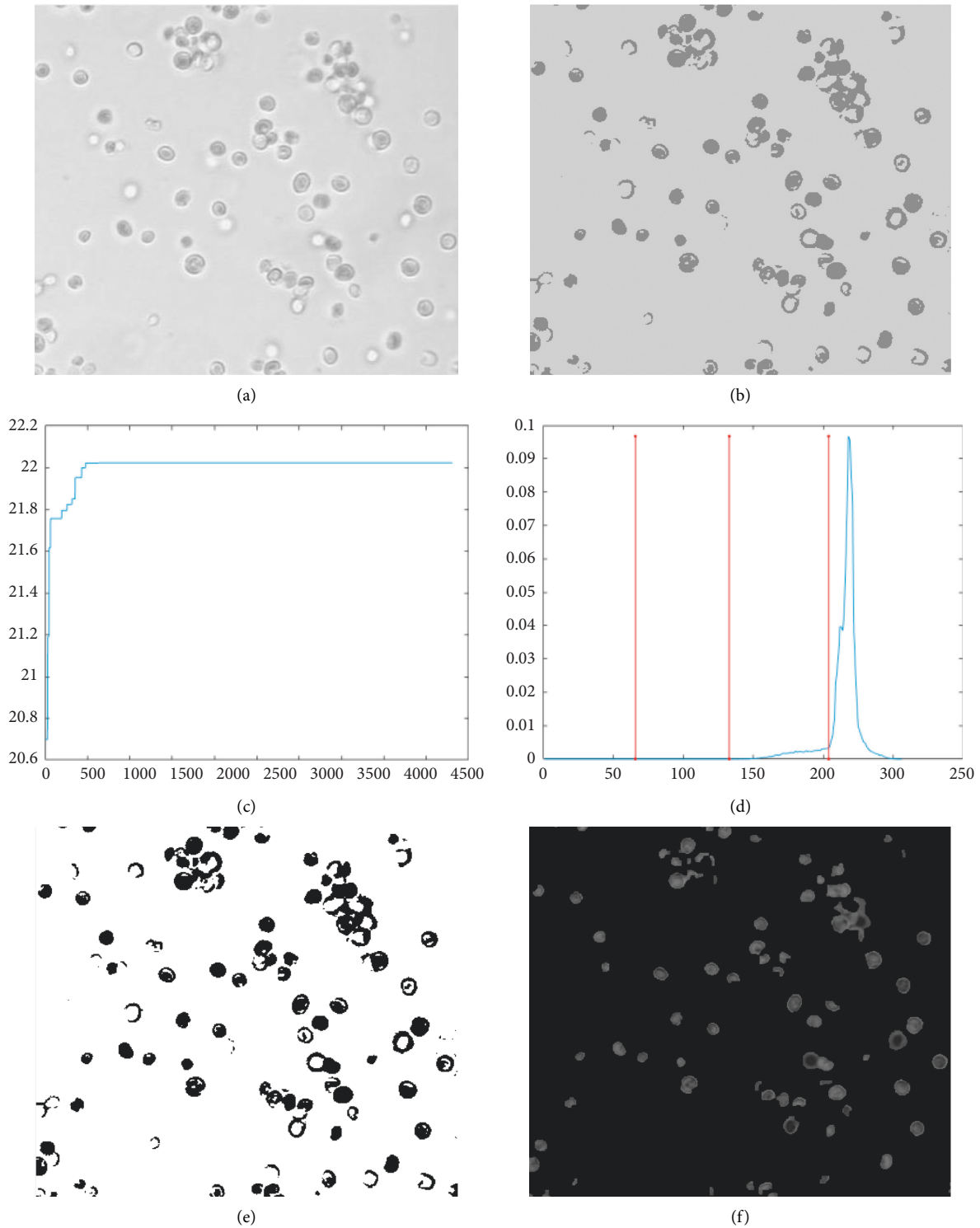


FIGURE 4: Image segmentation based on QTEO-Kapur: (a) The original image, (b), the segmented image based on the proposed method with (number of thresholds = 3), (c) the convergence profile of the fitness function, (d) the histogram of (B), (e) the binarized image, and (f) final segmented image.

$$\text{Cont} = \sum_{i=1}^M \sum_{j=1}^N p^2(i, j). \quad (22)$$

The entropy of the features is achieved by the following:

$$\text{Entropy} = - \sum_{i=1}^M \sum_{j=1}^N p(i, j) \log p(i, j). \quad (23)$$

The energy of the features is obtained as follows:

$$\text{Engy} = \sum_{i=1}^M \sum_{j=1}^N p^2(i, j), \quad (24)$$

and finally, the homogeneity is achieved by the following equations:

$$\text{Hgy} = \sum_{i=1}^M \sum_{j=1}^N \frac{p(i, j)}{1 + |i - j|}, \quad (25)$$

where  $\mu$  signifies the mean value,  $\sigma$  describes the standard deviation,  $p(i, j)$  states the intensity values for pixels at position  $(i, j)$ , and  $M$  and  $N$  represent the image size.

## 6. Simulation Results

Various databases have been proposed for assessing the microalgae diagnosis systems. In this study, we use the AlgaeVision database as a popular one for the analysis. AlgaeVision database contains a virtual image collection of freshwater and terrestrial algae that are collected from Britain and Ireland. AlgaeVision includes 2,300 images of nearly 250 genera and 680 species of British and Irish algae [36]. The data includes 19 different classes. The minimum number of the elements of the class is 10. The classes that are considered in this study are Bacillariophyceae-1, Spirotrichea, Peridinales, Nanoplankton <5  $\mu\text{m}$ , Nanoplankton 5–10  $\mu\text{m}$ , Gymnodinales, Prorocentrales, Cochlodinium, Ellipsoidal Microplankton, Ellipsoidal Nanoplankton 10–20  $\mu\text{m}$ , Spherical Nanoplankton 10–20  $\mu\text{m}$ , Spherical Microplankton, Pseudonitzschia, Chaetocerotales, Coscinodiscophyceae, Bacillariophyceae-2, Naviculales, Corsethron, and Rhizosoleniales.

For recognizing algae, the extracted features must be classified. In this study, the Support Vector Machine (SVM) algorithm is employed as a fast and simple classifier to classify positive and negative examples with a maximum margin. The SVM is an  $N$ -dimensional set of points that indicates the boundaries of the categories. The SVM algorithm finds the best limit in the data and makes the highest distance as much as possible from all categories. The idea is to recognize the best decision surface as follows:

$$y = \text{sgn} \left( \sum_{i=1}^N y_i \alpha_i K(x, x_i) + b \right), \quad (26)$$

where  $x$  defines a  $d$ -dimensional test set vector,  $x_i$  describes the  $i^{\text{th}}$  training set vector,  $y$  represents a class label in the range  $-1$  and  $1$ ,  $N$  is the training set numbers,  $\alpha = [\alpha_1 \dots \alpha_N]$  and  $b$  represent the model parameters, respectively, and  $K(x, x_i)$  states a kernel function. Here, the Sequential Minimal Optimization (SMO) is used for solving the quadratic programming (QP) problem of the SVM to remove the need for an extra storage matrix. To authenticate the efficiency of the suggested approach, different measurement indicators have been adopted. Due to the presence of multiple classes, we used specific features including Kappa coefficient, macro-, and microaverage. Kappa metric describes the agreement level of two or more observers in the same class.

TABLE 1: The classification results in the microalgae dataset.

Method/Indicator	Kappa	Max F1	Min F1
SMO	0.828	0.342	0.855
Region growing [37]	0.746	0.329	0.749
NAABB [38]	0.543	0.284	0.768
Label-free [39]	0.679	0.348	0.739
Neural network [40]	0.637	0.304	0.849

TABLE 2: The resampling results were performed with several sample sizes from 25% to 200%.

	Sample size	Kappa	Max F1	Min F1
SMO	25	0.703	0.329	0.851
	50	0.725	0.337	0.846
	100	0.732	0.343	0.831
	200	0.733	0.353	0.826
Neural network [40]	25	0.708	0.506	0.843
	50	0.725	0.627	0.915
	100	0.764	0.711	0.978
	200	0.777	0.724	0.988
Region growing [37]	25	0.620	0.296	0.843
	50	0.625	0.305	0.831
	100	0.613	0.325	0.827
	200	0.634	0.364	0.796
NAABB [38]	25	0.649	0.316	0.758
	50	0.658	0.328	0.746
	100	0.676	0.346	0.723
	200	0.681	0.358	0.698
Label-free [39]	25	0.538	0.324	0.712
	50	0.554	0.349	0.708
	100	0.562	0.382	0.692
	200	0.591	0.416	0.697

Kappa's value is between  $-1$  and  $1$ , where  $1$  describes the full agreement and  $0$  defines that the achieved results are too less than the expected value for agreement. F-Score metric indicates the harmonic mean value between the recall and precision. The MaxF1 and MinF1 describe the maximum value and the minimum value of the F-Score. The term MinF1 has more effect on the classifier efficiency on populations with more classes that represents the imbalanced dataset limitation. This metric is one of the widely used indicators to assess classifiers. Furthermore, the term MaxF1 has been affected by classes with small number of elements, which has an effective impact on the results evaluation. A 10-fold cross-validation method is used for evaluating the models. Table 1 illustrates the classification results in the microalgae dataset.

For analysis of the robustness of the algorithm and to achieve the oversampling strategy, a random resampling method has been used. This method can over/undersample the dataset based on a percentage through some original number of instances. This technique generates a random set of sample data by sampling with (or without) replacement. The uniformity of the resampled dataset in this technique is defined by a parameter called bias factor. It changes in the range  $[0, 1]$  where the values near  $1$  make a uniform dataset. The presented study evaluates different bias factors. This

method allows the classifier to retain smaller classes while optimizing accuracy. As can be observed from Table 1, the value of MinF1 is larger than the MaxF1 values, which are due to the high efficiency of the algorithm to correctly organize the majority classes. The term MaxF1 has been more affected by the smaller classes. Table 2 indicates the resampling results which are performed with several sample sizes from 25% to 200% to deal with the imbalance issue.

Table 2 illustrates the improvement of the results. The method has a satisfactory improvement in terms of MaxF1; therefore, it can be observed that better results are obtained by the SMO. Moreover, the Kappa value achieved by the proposed SMO algorithm provides a larger growth slope which is predicted due to its better efficiency for Gaussian distributed spaces.

As mentioned in this paper, the proposed method uses an optimal technique using the Kapur method for segmentation of the microalgae. The main advantage of this technique is that using metaheuristic-based techniques can help to run from the local minimization of the problem. However, this can make the problem about more complexity of the proposed method toward the classical one.

## 7. Conclusions

The present study proposed a new optimized and pipelined method for in-line detection of microorganisms. The method was first used as histogram equalization to improve the quality of the image quality. Then an image segmentation based on an optimized Kapur method was designed and used for separating the microalgae. The optimization of the Kapur in the study is based on a new improved version of the Quantum-based Thermal Exchange Optimization Algorithm. The new algorithm was designed for improving the segmentation accuracy. The Gray-Level Cooccurrence Matrix (GLCM) was then used for extracting the main features of the lesion area. A 10-fold classification was then performed to classify the Sequential Minimal Optimization- (SMO-) based Support Vector Machine (SVM). The suggested method was compared with some state-of-the-art methods by three different metrics. The final results showed that the suggested technique with 0.828 Kappa, and 0.342 and 0.855 min and max value of F1, provides the best efficiency against the other compared methods.

## Data Availability

The data are available at <https://algaevision.myspecies.info/>

## Conflicts of Interest

The authors announce that no conflicts of interest exist in this research.

## Acknowledgments

This research was supported by the Educational Science Foundation of Jiangxi Province (# 41562019).

## References

- [1] X. Qiu, X. Xie, and D. Meesapyodsuk, "Molecular mechanisms for biosynthesis and assembly of nutritionally important very long chain polyunsaturated fatty acids in microorganisms," *Progress in Lipid Research*, vol. 79, Article ID 101047, 2020.
- [2] A. Hogenkamp, A. Ehlers, J. Garssen, and L. E. M. Willemsen, "Allergy modulation by N-3 long chain polyunsaturated fatty acids and fat soluble nutrients of the mediterranean diet," *Frontiers in Pharmacology*, vol. 11, p. 1244, 2020.
- [3] P. Otálora, J. Guzman, F. Acien, M. Berenguel, and A. Reul, "Microalgae classification based on machine learning techniques," *Algal Research*, vol. 55, Article ID 102256, 2021.
- [4] S. Y. Teng, G. Y. Yew, K. Sukacova, P. L. Show, V. Masa, and J. S. Chang, "Microalgae with artificial intelligence: A digitalized perspective on genetics, systems and products," *Biotechnology Advances*, vol. 44, Article ID 107631, 2020.
- [5] V. R. P. Borges, M. C. F. Oliveira, T. C. da Silva, A. Vieira, and B. Hamann, "Region growing for segmenting green microalgae images," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 1, pp. 257–270, 2016.
- [6] J.-H. Giraldo-Zuluaga, A. Salazar, G. Diez et al., "Automatic identification of Scenedesmus polymorphic microalgae from microscopic images," *Pattern Analysis & Applications*, vol. 21, no. 2, pp. 601–612, 2018.
- [7] L. Wei, K. Su, S. Zhu et al., "Identification of microalgae by hyperspectral microscopic imaging system," *Spectroscopy Letters*, vol. 50, no. 1, pp. 59–63, 2017.
- [8] Z. Liu, R. Liao, H. Ma et al., "Classification of marine microalgae using low-resolution Mueller matrix images and convolutional neural network," *Applied Optics*, vol. 59, no. 31, pp. 9698–9709, 2020.
- [9] Z. Xu, Y. Jiang, J. Ji, E. Forsberg, Y. Li, and S. He, "Classification, identification, and growth stage estimation of microalgae based on transmission hyperspectral microscopic imaging and machine learning," *Optics Express*, vol. 28, no. 21, pp. 30686–30700, 2020.
- [10] A. Hu and N. Razmjoooy, "Brain tumor diagnosis based on metaheuristics and deep learning," *International Journal of Imaging Systems and Technology*, vol. 31, no. 2, pp. 657–669, 2020.
- [11] Q. Liu, Z. Liu, S. Yong, K. Jia, and N. Razmjoooy, "Computer-aided breast cancer diagnosis based on image segmentation and interval analysis," *Automatika*, vol. 61, no. 3, pp. 496–506, 2020.
- [12] R. Pugalenti, M. P. Rajakumar, J. Ramya, and V. Rajinikanth, "Evaluation and classification of the brain tumor MRI using machine learning technique," *Journal of Control Engineering and Applied Informatics*, vol. 21, no. 4, pp. 12–21, 2019.
- [13] S. C. Satapathy, V. Rajinikanth, A. S. Ashour, and N. Dey, "Multi-level image thresholding using Otsu and chaotic bat algorithm," *Neural Computing & Applications*, vol. 29, no. 12, pp. 1285–1307, 2018.
- [14] Z. Xu, F. R. Sheykhahmad, N. Ghadimi, and N. Razmjoooy, "Computer-aided diagnosis of skin cancer based on soft computing techniques," *Open Medicine*, vol. 15, no. 1, pp. 860–871, 2020.
- [15] N. Razmjoooy, M. Ashourian, M. Karimifard et al., "Computer-aided diagnosis of skin cancer: A review," *Current Medical Imaging*, vol. 16, no. 7, pp. 781–793, 2020.
- [16] V. Rajinikanth and S. C. Satapathy, "Segmentation of ischemic stroke lesion in brain MRI based on social group optimization



- and Fuzzy-Tsallis entropy,” *Arabian Journal for Science and Engineering*, vol. 43, no. 8, pp. 4365–4378, 2018.
- [17] N. Razmjooy, M. Ashourian, and Z. Foroozandeh, *Metaheuristics and Optimization in Computer and Electrical Engineering*, Springer, New York, NY, USA, 2020.
- [18] D. Yu, Y. Wang, H. Liu, K. Jermsittiparsert, and N. Razmjooy, “System identification of PEM fuel cells using an improved Elman neural network and a new hybrid optimization algorithm,” *Energy Reports*, vol. 5, pp. 1365–1374, 2019.
- [19] M. Ramezani, D. Bahmanyar, and N. Razmjooy, “A new optimal energy management strategy based on improved multi-objective antlion optimization algorithm: Applications in smart home,” *SN Applied Sciences*, vol. 2, no. 12, p. 2075, 2020.
- [20] N. Razmjooy, V. V. Estrela, and H. J. Loschi, “Entropy-based breast cancer detection in digital mammograms using World Cup optimization algorithm,” *International Journal of Swarm Intelligence Research*, vol. 11, no. 3, pp. 1–18, 2020.
- [21] Y. Guo, X. Dai, K. Jermsittiparsert, and N. Razmjooy, “An optimal configuration for a battery and PEM fuel cell-based hybrid energy system using developed Krill herd optimization algorithm for locomotive application,” *Energy Reports*, vol. 6, pp. 885–894, 2020.
- [22] U. R. Acharya, S. L. Fernandes, J. E. WeiKoh et al., “Automated detection of Alzheimer’s disease using brain MRI images—a study with various feature extraction techniques,” *Journal of Medical Systems*, vol. 43, no. 9, p. 302, 2019.
- [23] Y. Cao, Y. Li, G. Zhang, K. Jermsittiparsert, and N. Razmjooy, “Experimental modeling of PEM fuel cells using a new improved seagull optimization algorithm,” *Energy Reports*, vol. 5, pp. 1616–1625, 2019.
- [24] Y. Cao, Y. Wu, L. Fu, K. Jermsittiparsert, and N. Razmjooy, “Multi-objective optimization of a PEMFC based CCHP system by meta-heuristics,” *Energy Reports*, vol. 5, pp. 1551–1559, 2019.
- [25] X. Fan, H. Sun, Z. Yuan, Z. Li, R. Shi, and N. Razmjooy, “Multi-objective optimization for the proper selection of the best heat pump technology in a fuel cell-heat pump micro-CHP system,” *Energy Reports*, vol. 6, pp. 325–335, 2020.
- [26] Z. Yuan, W. Wang, H. Wang, and N. Razmjooy, “A new technique for optimal estimation of the circuit-based PEMFCs using developed Sunflower Optimization Algorithm,” *Energy Reports*, vol. 6, pp. 662–671, 2020.
- [27] O. Kramer, “Genetic algorithms,” in *Genetic Algorithm Essentials* Springer, New York, NY, USA, 2017.
- [28] M. Mani, O. Bozorg-Haddad, and X. Chu, “Ant lion optimizer (ALO) algorithm,” in *Advanced Optimization by Nature-Inspired Algorithms* Springer, New York, NY, USA, 2018.
- [29] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, “Equilibrium optimizer: A novel optimization algorithm,” *Knowledge-Based Systems*, vol. 191, Article ID 105190, 2020.
- [30] K. Zervoudakis and S. Tsafarakis, “A mayfly optimization algorithm,” *Computers & Industrial Engineering*, vol. 145, Article ID 106559, 2020.
- [31] N. Razmjooy, V. V. Estrela, R. Padilha, and A. C. B. Monteiro, “World Cup optimization algorithm: An application for optimal control of pitch angle in hybrid renewable PV/wind energy system,” in *Metaheuristics and Optimization in Computer and Electrical Engineering* Springer, New York, NY, USA, 2018.
- [32] N. S. M. Raja, S. L. Fernandes, N. Dey, S. C. Satapathy, and V. Rajinikanth, “Contrast enhanced medical MRI evaluation using Tsallis entropy and region growing segmentation,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2018.
- [33] A. Kaveh and A. Dadras, “A novel meta-heuristic optimization algorithm: Thermal exchange optimization,” *Advances in Engineering Software*, vol. 110, pp. 69–84, 2017.
- [34] A. Kaveh, A. Dadras, and T. Bakhshpoori, “Improved thermal exchange optimization algorithm for optimal design of skeletal structures,” *Smart Structures and Systems*, vol. 21, no. 3, pp. 263–278, 2018.
- [35] N. Dey, V. Rajinikanth, S. J. Fong, M. S. Kaiser, and M. Mahmud, “Social group optimization–assisted Kapur’s entropy and morphological segmentation for automated detection of COVID-19 infection from computed tomography images,” *Cognitive Computation*, vol. 12, no. 5, pp. 1011–1023, 2020.
- [36] P. York and D. John, “AlgaeVision-website of freshwater algal images’ A virtual collection and identification tool,” 2006.
- [37] V. R. P. Borges, M. C. Fd Oliveira, T. G. Silva, A. A. H. Vieira, and B. Hamann, “Region growing for segmenting green microalgae images,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 1, pp. 257–270, 2018.
- [38] P. Neofotis, A. Huang, K. Sury et al., “Characterization and classification of highly productive microalgae strains discovered for biofuel and bioproduct generation,” *Algal Research*, vol. 15, pp. 164–178, 2016.
- [39] C. Lei, T. Ito, M. Ugawa et al., “High-throughput label-free image cytometry and image-based classification of live *Euglena gracilis*,” *Biomedical Optics Express*, vol. 7, no. 7, pp. 2703–2708, 2016.
- [40] B. Franco, L. Navas, C. Gomez, C. Sepulveda, and F. Acien, “Monoalgal and mixed algal cultures discrimination by using an artificial neural network,” *Algal Research*, vol. 38, Article ID 101419, 2019.

## Research Article

# Novel Based Ensemble Machine Learning Classifiers for Detecting Breast Cancer

**Taarun Srinivas** <sup>1</sup>, **Aditya Krishna Karigiri Madhusudhan**,<sup>1</sup> **Joshuva Arockia Dhanraj** <sup>1</sup>,  
**Rajasekaran Chandra Sekaran**,<sup>1</sup> **Neda Mostafaeipour** <sup>2</sup>, **Negar Mostafaeipour** <sup>3</sup>,  
**and Ali Mostafaeipour** <sup>4</sup>

<sup>1</sup>Centre for Automation and Robotics (ANRO), Department of Mechatronics Engineering, Hindustan Institute of Technology and Science, Padur, Chennai 603103, India

<sup>2</sup>Tehran University of Medical Sciences School of Medicine, Tehran, Iran

<sup>3</sup>Medical School, Rafsanjan University of Medical Science, Rafsanjan, Kerman, Iran

<sup>4</sup>Industrial Engineering Department, Yazd University, Yazd, Iran

Correspondence should be addressed to Neda Mostafaeipour; mostafaei.neda@yahoo.com and Ali Mostafaeipour; mostafaei@yazd.ac.ir

Received 16 February 2022; Revised 8 April 2022; Accepted 11 April 2022; Published 10 May 2022

Academic Editor: Ramin Ranjbarzadeh

Copyright © 2022 Taarun Srinivas 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.

Nowadays, for many industries, innovation revolves around two technological improvements, Artificial Intelligence (AI) and machine learning (ML). ML, a subset of AI, is the science of designing and applying algorithms that can learn and work on any activity from past experiences. Of all the innovations in the field of ML models, the most significant ones have turned out to be in medicine and healthcare, since it has assisted doctors in the treatment of different types of diseases. Among them, early detection of breast cancer using ML algorithms has piqued the interest of researchers in this area. Hence, in this work, 20 ML classifiers are discussed and implemented in Wisconsin's Breast Cancer dataset to classify breast cancer as malignant or benign. Out of 20, 9 algorithms are coded using Python in Colab notebooks and the remaining are executed using the Waikato Environment for Knowledge Analysis (WEKA) software. Among all, the stochastic gradient descent algorithm was found to yield the highest accuracy of 98%. The algorithms that gave the best results have been considered in the development of a novel ensemble model and the same was implemented in both WEKA and Python. The performance of the ensemble model in both platforms is compared based on metrics like accuracy, precision, recall, and sensitivity and investigated in detail. From this experimental comparative study, it was found that the ensemble model developed using Python has yielded an accuracy of 98.5% and that developed in the WEKA has yielded 97% accuracy.

## 1. Introduction

Any environment can be made smart and intelligent with the help of technologies like AI and ML [1]. These technologies not only reduce the manpower and labor time but also improve the quality of life by automating society and creating more job opportunities [2, 3]. A system that can apply some logic, evaluate the different options available, and decide on its own can be considered to be an application of Artificial Intelligence. To provide this intelligence, the system has to be trained; i.e., the system has to learn. This is

done through machine learning, a branch of AI, where the system makes a decision based on the data. In ML, the performance of a task improves based on previous data or experience. Thus, it can be understood that performance, task, and experience are the three main characteristics of ML. Figure 1 represents the basic flow of machine learning pictorially.

The applications of ML can be found in many areas such as object recognition, spam detection, speech recognition, and healthcare industries for survival predictions, tumor detection, and cancer detection. The use of AI and ML

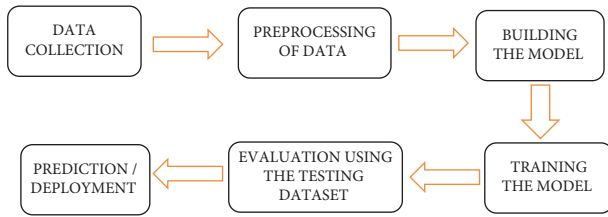


FIGURE 1: The basic flow of machine learning.

models assists doctors in the early detection and curing of cancer, tumors, and other deadly diseases. One such application in medicine is the detection of breast cancer using ML models. Breast cancer is the growth of unwanted cells in the breast and early detection of this cancer can lead to a better cure [4]. Many classification algorithms in machine learning can be used to detect breast cancer with high accuracy, but selecting a particular algorithm for breast cancer detection is a challenging task. There are many works done by researchers working in these areas and these kinds of works give rise to the establishment of smart hospitals and constructive exploitation of these technologies leads to the improvement in quality of life [5].

Chandrasekaran et al. [6] developed a fully integrated analog-based machine learning classifier that uses Artificial Neural Network (ANN) algorithms for breast cancer prediction. The developed ANN model mainly focused on nine attributes of breast cancer, such as (1) bland chromatin, (2) mitosis, (3) uniformity of cell shape, (4) marginal adhesion, (5) single epithelial cell size, (6) uniformity of cell size, (7) clump thickness, (8) normal nucleoli, and (9) bare nuclei. The authors achieved a mean accuracy of 96.9% with optimal energy consumption. Later, a Deep Neural Network with Support Value (DNNS) was developed by Vaka et al. [7] for predicting breast cancer. The model was developed in three stages. In the first stage, preprocessing was carried out by removing noise from the images in the dataset. In the second stage, the geometrical and textual features were detected from the preprocessed image. In the third stage, each image was segmented using the histosigmoid based fuzzy classifier. The authors achieved an accuracy of 97.21%. Mojriani et al. [8] developed a multilayer fuzzy expert system by integrating an Extreme Learning Machine (ELM) classification model with Radial Basis Function (RBF). The authors compared the proposed methodology with the linear Support Vector Machine (SVM) model. Accuracy, Precision, True Positive (TP rate), False Positive (FP rate), True Negative (TN rate), and False Negative (FN rate) were considered as the performance metrics to evaluate the proposed ML model. The authors achieved an accuracy of 98% and the proposed methodology was effective in detecting breast cancer when compared with the linear SVM model. Chaurasia and Pal [9] predicted and compared six machine learning algorithms. Classification and Regression Trees (CART), Linear Regression, Multilayer Perceptron (MLP), Naive Bayes (NB), K Nearest Neighbors (KNN), and SVM were the basic learners considered for prediction. Along with basic learners, a few ensemble algorithms like AdaBoost M1, Random Forest (RF), and Support Vector Clustering (SVC) were integrated

to yield high accuracies. The authors divided the training and test set in the ratio of 80 : 20, respectively. Accuracy was the metric considered by the authors and they achieved a good accuracy of 95.17%. Similar to [9], a comparative analysis using data visualization and ML applications for breast cancer detection and diagnosis was carried out by [10] and achieved an accuracy of 98% using the logistic regression model.

Having studied the recent works of these authors, it is understood that individual algorithms like either deep neural networks or ensemble algorithms have been used at a time. To further improve the accuracy of existing methodologies, a novel combination of ensemble and optimization-based approaches is required to detect breast cancer with high accuracy. In this paper, 20 different ML classification algorithms have been selected for predicting breast cancer as benign or malignant using Wisconsin's Breast Cancer dataset. Out of these 20, 9 were implemented in Google Colab notebooks using Python programming language and the remaining 11 were executed in WEKA data mining software. Along with the 20 algorithms, we propose a novel machine learning ensemble-based algorithm. The algorithms for ensemble were chosen based on the performance of the 20 algorithms. The ensemble model developed was made to run on both platforms and the results obtained were compared and investigated in detail. Thus, the work in this paper can assist doctors in accurately detecting and diagnosing breast cancer in the early stages. This is an attempt to improve and automate society, establish the concept of smart hospitals, and contribute to the development of these technologies.

The rest of this paper is organized as follows. Section 2 introduces the dataset and discusses the EDA performed to understand the importance and contribution of individual features. Section 3 describes the working of the selected 20 algorithms. Section 4 explains the evaluation parameters considered to investigate the performance of the algorithm and Section 5 discusses the results obtained and explains in detail the comparative study carried out in both platforms before concluding in Section 6.

## 2. Wisconsin's Breast Cancer Dataset

The Breast Cancer dataset from the University of Wisconsin Hospital was considered for carrying out the performance analysis of 20 different classification algorithms. This is a small dataset consisting of data from 569 patients with 31 different input features and a single output feature. The 31 different features are independent of each other and the output feature called "diagnosis" is dependent on the input features. The output feature "diagnosis" has two labels: malignant and benign.

*2.1. Exploratory Data Analysis (EDA).* The Breast Cancer dataset consists of data from 569 patients, out of which 359 patients were diagnosed as benign and 212 patients were diagnosed as malignant. The blue region from Figure 2 represents the percentage share of patients who were

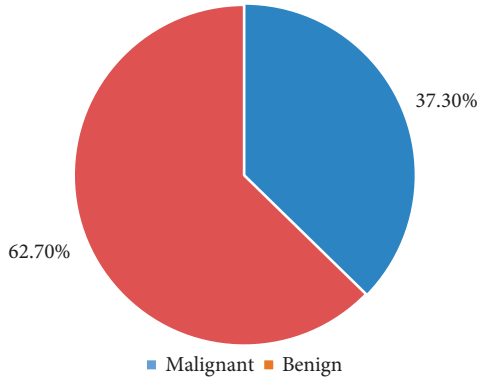


FIGURE 2: Benign and malignant cases.

diagnosed as malignant (37.3%) and the red region represents the percentage share of patients who were diagnosed as benign (62.7%).

As the percentage shares are not equal, there is a class imbalance and the predictions may not be generalized. To overcome this class imbalance problem and to have generalized results, the number of data samples must be equal for both labels. Inferring from Figure 3, if the mean radius is higher, then the patient is more likely to be diagnosed as malignant and if it is lower, then the patient is diagnosed as benign. So, the mean radius is directly proportional to the chances of a patient having breast cancer or not.

Similarly, from Figure 4 it is understood that people who were diagnosed as malignant have mean compactness in the range of 0.05 to 0.27 and those who have been diagnosed as benign have mean compactness in the range of 0 to 0.18. As the mean compactness increases, the chances of a person likely having cancer cells are more. When the cancer is detected as malignant, the mean area of spread of cancer is between 500 and 1800 mm and when the same has been detected as benign, the mean area of spread of cancer is between 0 and 700 mm and their respective sum is depicted in Figure 5. Early cancer detection helps to stop the spread of cancer when compared to malignant cases. Curing of malignant cases is very hard as the area of spread will be high.

### 3. Brief Description of the Classification Algorithms

The chart given in Figure 6 explains the types of ML. The algorithms listed in the chart below are the most common and popularly used classifiers. Instance-based algorithms such as KNN and statistical-based algorithms such as Naive Bayes, Bayes Net, Neural Network, Decision Trees, and more are experimentally verified in the next section.

**3.1. Logistic Regression.** It is a widely used machine learning approach for categorization. The logistic regression algorithm draws a decision boundary between the different input features. The model then maps the inputs to the outputs by computing the cost function. The sigmoid function, also called a logistic function, is used for predicting the target

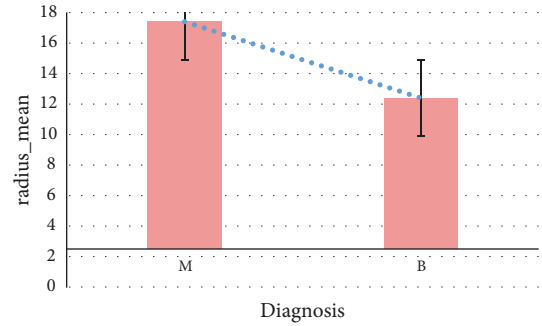


FIGURE 3: Diagnosis vs. radius mean.

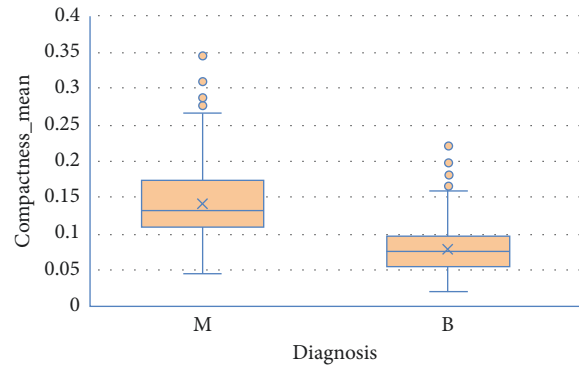


FIGURE 4: Diagnosis vs. compactness mean.

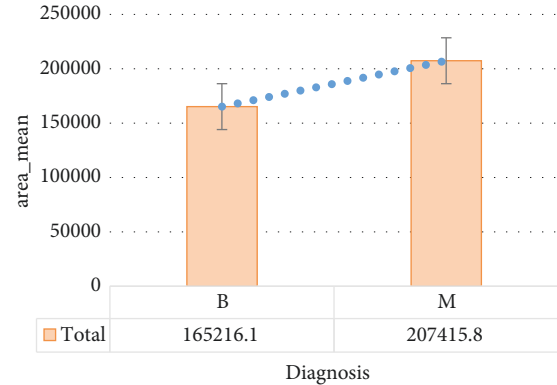


FIGURE 5: Diagnosis vs. area mean.

labels. Equations (1)–(3) mathematically describe the sigmoid function [11].

$$h(x) = g(\theta^T x), \tag{1}$$

$$g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}, \tag{2}$$

$$g(z) = \frac{1}{1 + e^{-z}}, \tag{3}$$

where  $h(x)$  is defined as the model and  $g(\theta^T x)$  is the sigmoid function.  $\theta^T$  is the model parameters and  $x$  represents the input features. After the sigmoid function is computed, the cost function is calculated and the results from the cost

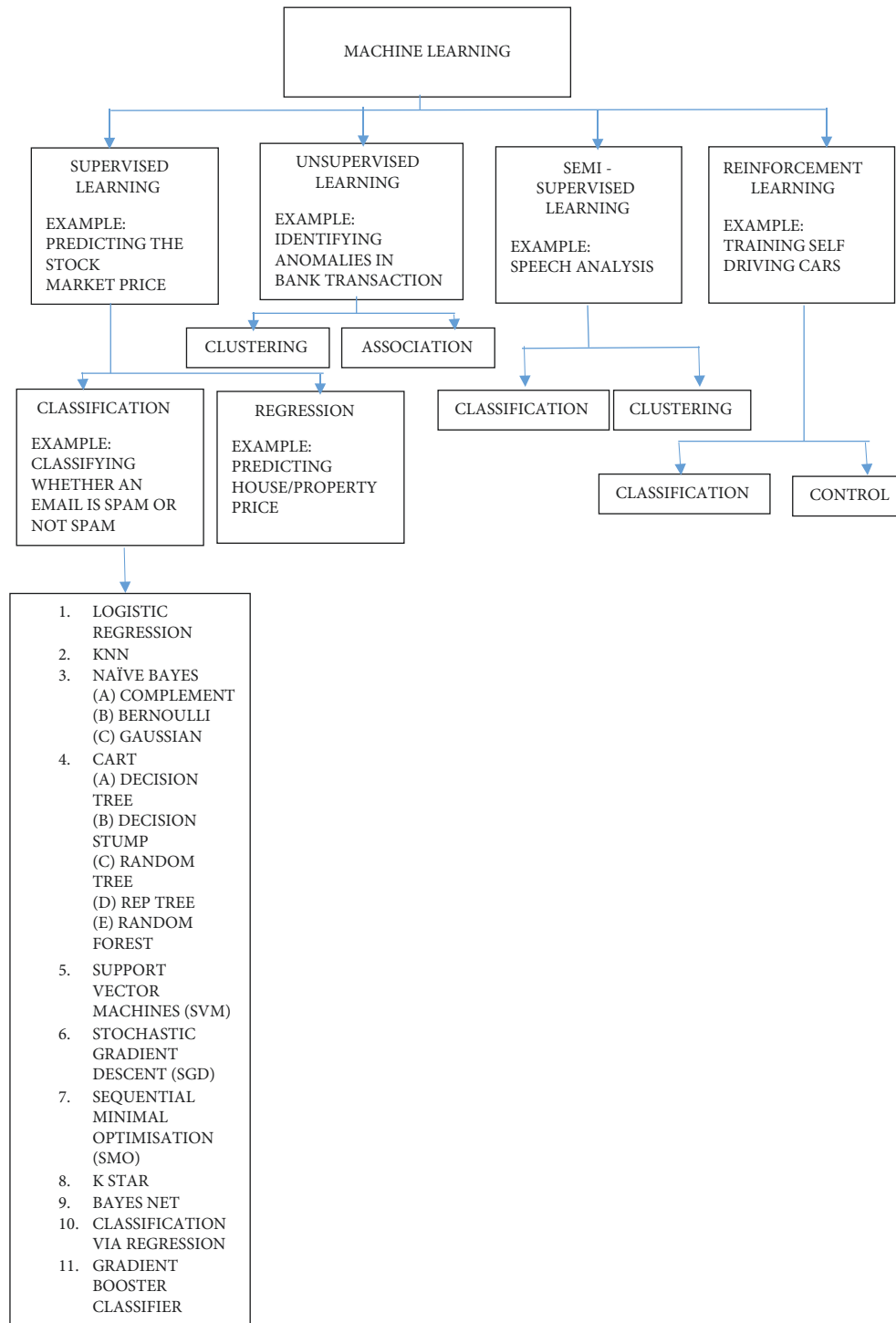


FIGURE 6: Types of machine learning and classification algorithms.

function are later optimized using a gradient descent algorithm.

**3.2. K Nearest Neighbors (KNN).** KNN is a widely used method for various real-time applications dealing with the classification of classes. The classification can be either binary in which only two output classes will be available or

multiclass classification, in which more than two output classes will be available. The KNN algorithm is a lazy learning and highly computational algorithm [12]. This algorithm uses the principle of matching the characteristic label of an unknown data sample with the characteristic label of a neighboring data sample based on Euclidean distance or, in some cases, the Manhattan distance [13]. Euclidean distance is the square root of the distance

between the two points in the sample space. The following equation (4) gives the Euclidean distance formula [14]:

$$d(x_i, x_j) = \sqrt{\sum_{n=1}^m (x_{in} - x_{jn})^2}, \quad (4)$$

where  $d(x_i, x_j)$  is the distance between a known data sample and an unknown data sample and  $x_i, x_j$  are the location of the data samples. The “ $K$ ” in KNN is the total number of neighbors that are selected for making predictions. It should be purely user-defined for fine prediction or else should be an odd square of the total number of data in the dataset chosen [15]. When a new, unknown data sample is introduced into the search space, the distance between the unknown data sample and the  $k$  nearest neighbors is determined. The least proximity value among the unfamiliar data sample to the nearest neighbor will determine the class for the unknown data sample.

**3.3. Naive Bayes Classifier.** It is a Bayes theorem-based conditional probabilistic machine learning model.

**3.3.1. Bayes Theorem.** The Bayes theorem in probability and statistics defines the likelihood of every occurrence based on prior knowledge of circumstances that may be relevant to the event. The following equation expresses the Bayes theorem mathematically [16]:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \quad (5)$$

where  $A$  and  $B$  are events and  $P(B) \neq 0$ .

Hence, utilizing this hypothesis, it is feasible to discover the likelihood of an incident, say  $A$ , given that  $B$  has happened. In this case,  $B$  represents the evidence and  $A$  represents the hypothesis. The assumption here is that the existence of one trait has no effect on the other. As a result, it is said to be naive. Saritas and Yasar used the Naive Bayes model and ANN in Breast Cancer Coimbra Dataset and found that the classes were classified with an accuracy of 86.95% with ANN and 83.54% with Naive Bayes algorithms, thus proving that Naive Bayes algorithms could be used for early breast cancer detection [17].

**3.3.2. Types of Naive Bayes Classifier**

**(1) Gaussian Naive Bayes Classifier (GNB).** When the predictors take on a continuous value in GNB, we assume that these values have a Gaussian or normal distribution. Consider training data with a continuous attribute “ $X$ .” After dividing the data by class, the mean and variance of “ $X$ ” in each class are computed. The probability distribution of some observation value “ $X_i$ ” given a class “ $Y$ ” can be computed by using the following equation [16]:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}}, \quad (6)$$

where  $\sigma_y$  is the variance of values associated with class  $y$  and  $\mu_y$  is the mean of values associated with the class  $y$ .

Piryonesi and El-Diraby explored the performance of different classification algorithms and used it for the examination of asphalt pavement degradation data. According to their findings, GNB can significantly improve the classifier’s accuracy [18].

**(2) Bernoulli’s Naive Bayes Classifier (BNB).** Features in the multivariate Bernoulli event model are independent Booleans (binary variables) that describe inputs. This model, like the multinomial model, is popular for document classification problems where binary term occurrence characteristics are utilized rather than term frequencies.

**(3) Complement Naive Bayes Classifier (CNB).** The complement Naive Bayes classifier, also called CNB, works well with imbalanced datasets. The main principle behind the CNB is that the probability of a feature or data sample is calculated for all the classes to which the data sample belongs and not just for a single particular class. The main disadvantage of CNB is that the model tends to overfit easily when there is a large number of data samples in the dataset.

**3.4. CART**

**3.4.1. Decision Tree.** The decision tree algorithm is used for many classification problems. The main advantage of the decision tree is that it finds hidden patterns from the data by pruning the dataset and delivers high accuracy [19]. Pruning is the process of removing unwanted nodes from the decision tree that does not provide meaningful insights into the classification process. The decision tree consists of three components, mainly the root, node, and leaves. Root and node classify the unknown data based on if-then criteria and deliver the output to the next layer of nodes. Leaves are the ends of a decision tree that provides the filtered result after pruning. There are different algorithms to implement a decision tree and one such simple way to implement the same is using an Iterative Dichotomiser 3 (ID3) algorithm. The ID3 algorithm uses information gain and entropy to determine the output labels. Entropy measures the impurity of data and information gain is used to determine the amount of information that the data contributes to the final outcome [20]. The information gain and the entropy are measured using equations (7) and (8) [21].

$$H(s) = \sum_{c \in C} -P(c) \log_2 P(c), \quad (7)$$

where  $H(s)$  is the entropy,  $s$  is the input data, and  $c$  is the number of output classes.

$$IG(A, S) = H(S) - \sum_{t \in T} P(t)H(t), \quad (8)$$

where  $IG(A, S)$  is the information gain from the data  $S$  depending on output  $A$ ,  $t$  is the subset after the data is split,  $P(t)$  is the number of elements in  $t$ , and  $H(t)$  is the entropy of the subset.

**3.4.2. Decision Stump.** The decision stump algorithm is similar to decision trees. In this algorithm, there is only one decision tree which consists of one root and a maximum of two leaf nodes for output predictions [22]. Because of this reason, decision stumps are weak learners and do not subclassify like decision trees. The algorithm terminates as soon as the first classification is executed. The overall prediction can be determined using the following equation [23]:

$$F(x) = \sum_{i=1}^n g^i(x_i), \quad (9)$$

where  $x$  is the input features,  $g^i$  is the predicted value of the  $i^{\text{th}}$  feature, and  $x_i$  is the  $i^{\text{th}}$  feature.

**3.4.3. Random Trees.** Random tree classifiers are a combination of decision trees and random forests. It is also called Random Forest Tree (RFT). It belongs to the category of ensemble classifiers. Random Trees improve the performance of weak learners as well as the accuracy of the machine learning model. A random tree is defined in three steps as (i) defining the method of splitting the leaves in the tree; (ii) selecting the predictor type for every leaf in the tree; (iii) providing the randomness into the leaves. One major disadvantage of random trees is that the classifier often tends to overfit the model [24].

**3.4.4. Reptree.** Reptree, also called representative tree algorithm, belongs to the family of decision trees and is one of the fast learner algorithms [25]. In this algorithm, multiple decision trees are created and the one with the highest probability is chosen as the representative, and based on this, other trees will be classified [26]. The reptree algorithm is a combination or ensemble of various individual trees or individual learners.

**3.4.5. Random Forests.** Random forest is the combination of many decision trees by randomly selecting the features and bootstrapping [27]. The random forest will first select a limited number of features from the entire set of input features and split those random features into nodes. The nodes, in turn, are split into subnodes or daughter nodes and the process is carried out until the required nodes are satisfied. The same process is repeated for “ $n$ ” number of times creating “ $D$ ” decision trees. The pseudocode for random forest is given in Table 1.

**3.5. Support Vector Machines (SVM).** SVMs provide highly accurate results with linear data as well as nonlinear data. In SVM, the input features are defined as input vectors, and multiple hyperplanes are drawn between these input features. The hyperplane acts as a decision boundary in SVM. The main objective of a hyperplane is to maximize the distance between two or more input features [28]. Because of this characteristic of the hyperplane, SVMs work well with nonlinear data. A hyperplane can be expressed using the following equation [29]:

$$W^T x + b = 0, \quad (10)$$

where  $x$  is the input vector containing the input features,  $b$  is the bias, and  $W$  is the weight that defines the distance separating the hyperplane. The best hyperplane is decided by finding different hyperplanes which classify the labels in the best way; then it will choose the one which is farthest from the data points or the one which has a maximum margin as depicted in Figure 7.

**3.6. Stochastic Gradient Descent (SGD).** SGD is a sophisticated version of gradient descent (GD). The only difference between GD and SGD is that the gradient descent is calculated for only a limited number of data samples. SGD selects data samples in a randomized manner and computes the gradients. Once the gradient is computed, the weights and the corresponding output labels are updated for all the data samples that were considered. This is an iterative process and continues until a minimum cost is reached [30]. The following equation is used to calculate the gradient descent [31]:

$$\nabla f(x) = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x), \quad (11)$$

where  $f(x)$  is the objective function,  $n$  is the total number of data samples,  $\nabla f(x)$  is the gradient descent of objective function, and  $\nabla f_i(x)$  is the gradient descent that is computed for limited data samples randomly. Once the gradients are computed for the selected data samples, the weights are updated using the following equation [32]:

$$x \leftarrow x - \alpha \nabla f_i(x), \quad (12)$$

where  $x$  is the data sample and  $\alpha$  is the learning rate.

**3.7. Sequential Minimal Optimization (SMO).** SMO is used for solving quadratic programming problems. It is used to train the SVM classifier using a Gaussian or polynomial kernel [32]. It converts the attributes or features into binary values and also replaces missing values in the input features. SMO breaks down a quadratic problem into linear complexity. Then this linear complexity is solved using Lagrange multipliers  $\alpha_1$  and  $\alpha_2$ . The constraints for the selection of Lagrange multipliers are given in equations (13) and (14) [33]:

$$0 \leq \alpha_1, \alpha_2 \leq M, \quad (13)$$

$$y_1 \alpha_1 + y_2 \alpha_2 = K, \quad (14)$$

where  $M$  is an SVM hyperparameter and  $K$  is the kernel function, both supplied by the user; and the variables  $\alpha_1, \alpha_2$  are Lagrange multipliers. SMO can be used when there is a class imbalance between the input features and works well with binary as well as multiclass classification [34].

**3.8.  $K$  Star.**  $K$  star or  $K^*$  is one of the lazy algorithms used for many complex classification problems. It is also called an instance-based classification algorithm [35]. Every data

TABLE 1: Pseudocode for random forest.

(1)	Start
(2)	Define the input features Training the data
(3)	(i) Select “ $i$ ” random features from the entire set of input features $i$ (ii) Determine the nodes from the “ $i$ ” features using the best split point
(4)	Repeat (i) The above three steps until a specific number of nodes are reached
(5)	Repeat (i) The above steps for $n$ times to create $m$ decision trees
(6)	Testing the data (i) Feed-in a new data sample and predict the output label (ii) The node which gives high prediction among the other output nodes will be taken as the final prediction
(7)	Stop

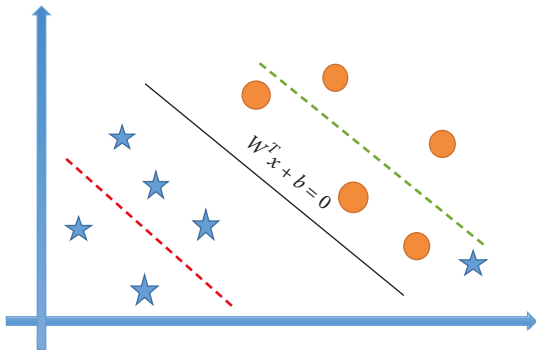


FIGURE 7: SVM hyperplane.

sample is an instance of this algorithm. The class of an instance of an unknown data sample depends on the entropic distance between that unknown data sample and the known data sample in the search space [36].

**3.9. Bayes Net.** A Bayesian Network, also called Bayes Net, is one of the classification algorithms used for solving machine learning problems. A Bayes Net is constructed using DAGs—Directed Acyclic Graphs. These DAGs consist of nodes and edges. The nodes represent a random variable, and the edges are used to establish the connection between two or multiple different nodes and carry the information regarding the probabilities between nodes. The only prerequisite of a DAG is that all the nodes must be connected in an open-loop and there must be no closed connection between the two nodes. The information sharing between the nodes depends on the Parental Markov Condition (PMC). PMC states that a variable represented by a node is independent of all variables in a nondescendant node and conditional on the parent node. The probabilities are calculated using the conditional probability formula as mentioned in the following equation [37]:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \quad (15)$$

**3.10. Classification via Regression.** In classification via regression algorithm, first regression is carried out. Classification is performed from the output features obtained from the regression model and finally, the output class is predicted using

logistic regression. The main disadvantage of this algorithm is that it provides fluctuating results. It is both a weak learner and a strong learner. The algorithm performs well when the number of input features is limited and if the input features are more, the accuracies may vary.

**3.11. Gradient Boosting Classifier.** Gradient boosting classifier is an ensemble of various weak learners which is used for both regressions and classification problems. A gradient boosting classifier performs gradient descent on a weak learner-like decision tree. This loss function can be defined as in the following equation[38]:

$$f(x) = px + q + e, \quad (16)$$

where  $f(x)$  is the loss function,  $e$  is the error,  $x$  is the vector containing input features, and  $p, q$  are weights to the loss function. The gradient boosting classifier can be applied to more complex datasets and it provides highly accurate results for nonlinear data.

**3.12. AdaBoost M1.** AdaBoost, also called Adaptive Boosting, is an ensemble-based algorithm that can be used for both binary and multiclass classification. In this boosting algorithm, classification is carried out by defining a threshold or weightage value for every classifier. The final output class is determined based on weighted votes from the individual classifiers. If there are any weak classifiers, then extra weights are given to the samples of these classifiers. This addition of extra weights to weak classifiers converts the weak learner into a strong learner [39, 40]. The weights for the individual classifier are determined using the following equation [41]:

$$\alpha_i = \frac{1}{2} \ln \left( \frac{1 - \varepsilon_i}{\varepsilon_i} \right), \quad (17)$$

where  $\alpha_i$  is the weight for sample  $i$  and  $\varepsilon_i$  is the weighted error for sample  $i$ . The final output prediction is determined using the following equation [42]:

$$H(s) = \sum_{i=1}^n \alpha_i h_i(s), \quad (18)$$



where  $H(s)$  is the final output prediction and  $h_i(s)$  is the output prediction for each sample in a classifier.

**3.13. LogitBoost.** LogitBoost is also called an additive logistic regression algorithm [43]. LogitBoost is a weak classifier. LogitBoost takes different data samples while training and produces a weak output prediction. Boosting is applied to this weak model to improve performance and produce high output predictions. LogitBoost works well with both binary and multiclass classification. LogitBoost is similar to the AdaBoost algorithm. The only difference between the two algorithms is that the logit boost algorithm produces a weak classifier during the initial period of training the model. The output predictions are determined by using the following equation [44]:

$$P(x) = \frac{e^{f(x)}}{e^{f(x)} + e^{-f(x)}}, \quad (19)$$

where  $P(x)$  are the probabilities of output class and  $f(x)$  is the fitness function

**3.14. Bagging.** Bagging is also called Bootstrap Aggregation. It is a powerful ensemble algorithm that works well with small datasets. Bagging is used to reduce the variance in a trained model and prevents overfitting of data while training [45]. In this algorithm, the dataset is split into clusters. Classification algorithms like decision trees and random forest are applied to this cluster, and predictions are computed. The predictions from the individual clusters are averaged and through a voting basis, the final output prediction is determined [46, 47]. The pseudocode for bagging is given in Table 2.

## 4. Evaluation Metrics

The performance of the above-discussed 20 classification algorithms is evaluated using the parameters discussed below. A confusion matrix with TP, TN, FP, and FN for actual and predicted data samples is formed as seen in Table 3. A confusion matrix is a table that allows you to examine how an algorithm performs. Each column of the matrix contains examples from an actual class, whereas each row represents instances from a projected class [11].

The letter TP denotes that the actual value was positive and that the model projected a positive result. The letter TN denotes that the actual value was negative and that the model expected a negative result. In both of these examples, the projected value corresponds to the actual value. Whereas FP denotes a negative actual value but a positive prediction from the model, FN denotes a positive actual value but a negative prediction from the model. In both cases, the predicted value was falsely predicted. Based on these 4 values, accuracy, precision, recall, F1 measure, sensitivity, and specificity can be calculated.

*Accuracy* gives the proportion of properly identified samples to total samples. It is obtained from the formula [48, 49],

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}. \quad (20)$$

*Precision* tells how many of the correctly anticipated samples were positive. It is calculated by dividing all positive cases by all positive examples projected to be positive [48].

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}. \quad (21)$$

Similarly, *recall* indicates how many of the actual positive samples were properly predicted by the model. In medical circumstances, recall is critical since positive cases should not go undiscovered. In the case of this dataset, recall would be a preferable statistic because we do not want to mistake a malignant individual for a benign person. Precision and recall would decide whether or not the model is dependable [48].

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}. \quad (22)$$

The *F1 metric* is a harmonic mean of recall and precision; hence it provides a comprehensive overview of these two measures. When we strive to improve our model's precision, the recall decreases, and vice versa. The F1-score incorporates both trends into a single number [48].

$$\text{F1 - measure} = \frac{2\text{TP}}{(2\text{TP} + \text{FP} + \text{FN})}. \quad (23)$$

*Sensitivity* is the ability of a test to correctly classify a sample as malignant/positive. It is defined as the rate of the predicted positive case to the total positive cases [48].

$$\text{Sensitivity} = \frac{\text{TP}}{(\text{TP} + \text{FN})}. \quad (24)$$

*Specificity*, on the other hand, is the ability of a test to correctly classify a sample as negative/benign. It gives the relationship of observed negative examples with all negative examples [48].

$$\text{Specificity} = \frac{\text{TN}}{(\text{TN} + \text{FP})}. \quad (25)$$

In addition to the parameters mentioned above, three model evaluation metrics in  $R$  are utilized to examine the performance of implementing algorithms (i.e., Kappa value, Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE)). The Kappa value, often known as Cohen's Kappa, is a measure of categorization accuracy. It is normalized to the dataset's random chance baseline. It is a statistic that compares the observed and expected accuracy. The Kappa statistic is used to evaluate not just a single classifier but also subclassifiers.

The MAE is the average of all absolute errors. It is also defined as the difference between the expected and actual values. References [50, 51] give the formula

TABLE 2: Pseudocode for bagging.

(1)	Start
(2)	Define the input features and the output label
(3)	Split data into training and testing sets
	Training the model
(4)	(i) Perform bootstrap process on the training data
	(ii) Choose the number of clusters to be formed based on the data
	(iii) Apply a classification algorithm like a decision tree
	Evaluate the model on the test set
(5)	(i) Compute the accuracy from each cluster in the model and average the accuracy
	(ii) Determine the output prediction by a voting process
(6)	Stop

TABLE 3: Confusion matrix representation.

	Benign	Malignant
Benign	TP	FP
Malignant	FN	TN

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n}, \quad (26)$$

where  $n$  is the total number of samples,  $y_i$  is the predicted value, and  $x_i$  is the actual value.

*Root mean square error* (RMSE). It is equal to the square root of the mean of the squares of all mistakes. RMSE is widely used and regarded as an excellent error metric for various sorts of predictions [51, 52].

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}}. \quad (27)$$

## 5. Results and Discussions

In this study, 20 various classification algorithms were discussed in the previous section. To analyze its performance, all the algorithms were applied to Wisconsin's Breast Cancer dataset to predict whether the cancer is benign or malignant. The analysis was justified based on various parameters such as precision, F1 measure, confusion matrix, recall, accuracy metrics, and run time discussed in Section 4. As discussed in Section 2, the dataset contains 569 samples, each with 33 attributes. The dataset is preprocessed by removing the unnecessary attributes which do not contribute to the output and checking for any incomplete data. After that, the dataset is divided into 426 training samples and 143 testing samples. All algorithms are trained on training samples before being verified on test data. The evaluation parameters given in this section are obtained from the validation of test samples. The current method employs a data driven approach. Based on historical data, the model will be able to detect breast cancer. Also, this method is faster in breast cancer detection when compared to traditional diagnostic methods on basis of performance and time. The novelty of this method is attributed to the fact that the ensemble model developed for the current study consists of combination of boosting and optimization algorithms which will dynamically reduce the errors within few iterations.

*5.1. Experimental Platform.* In this study, an Intel Core i5 powered 2.30 GHz/8.00 GB system running under the Windows operating system is used for processing purposes. Out of the 20 discussed algorithms, 9 of them (Logistic regression; KNN; Naive Bayes-Complement, Bernoulli, Gaussian; Decision tree; Random forest; Gradient Booster; Support Vector Machine) are run on Google Colab notebooks and coded using Python language. The Scikit-learn package is used to train and test the samples. The remaining 11 algorithms (SGD, SMO, K star, Decision Stump, random tree, REPTree, BayesNet, classification via regression, AdaBoost M1, LogitBoost, and bagging) are run using WEKA software which is a workbench for machine learning.

*5.2. Experimental Results.* Train and test data are split in the ratio of 80:20. The TP, TN, FP, and FN values of the implemented algorithms are given in Table 4. Based on these 4 values, accuracy, precision, recall, F1 measure, sensitivity, and specificity are calculated using equations (20)–(25) and tabulated in Table 5. Apart from the above-discussed parameters, three model evaluation metrics in “R” are also calculated using equations (26) and (27) and tabulated in Table 6. Also, the accuracy and run time of the algorithms are graphically depicted in Figures 8 and 9.

*5.3. Discussions.* The hyperparameters used to implement the algorithms are summarized in Table 7. From Table 5 it is clearly understood that SGD yields a higher accuracy of 98%, followed by random forest, SMO, and gradient booster with 97%. BNB yields the least accuracy of 63%. This is due to the fact that BNB is used for discrete data and operates on the Bernoulli distribution. The key characteristic of BNB is that it only takes binary values such as true or false. The Breast Cancer dataset has a continuous range of values for all the features. This is why BNB is not able to produce higher accuracy. All other algorithms yield accuracy ranging from 90% to 96%. KNN was able to provide an accuracy of 95% with its  $K$  value set to 4. The  $K$  value was decided using a search method where, for each  $K$  value from 1 to 10, the error obtained was plotted and visualized in Figure 10.

Smaller  $K$  values can be noisy and have a greater effect on the outcome, but bigger  $K$  values have smoother decision boundaries, which means reduced variance but increased bias and is computationally intensive. Hence an optimal  $K$  value of 4 was selected. The type of solver and learning rate

TABLE 4: Confusion matrix (TP, TN, FP, FN) of the implemented classifier.

Classifier	TP	TN	FP	FN
KNN	89	47	1	6
Logistic regression	85	51	5	2
CNB	89	44	1	9
BNB	93	0	0	53
GNB	88	48	2	5
Decision tree	86	50	4	3
Decision stump	80	47	6	9
Random tree	80	55	6	1
REPTree	77	51	9	5
Random forest	88	51	2	2
SVM	89	47	1	6
SGD	85	54	1	2
SMO	84	54	2	2
K star	84	50	2	6
Bayes net	81	52	5	4
Classification via regression	82	53	4	3
Gradient booster	88	51	2	2
AdaBoost M1	82	53	4	3
LogitBoost	84	52	2	4
Bagging	83	52	3	4

TABLE 5: Performance of implemented classifiers.

Classifier	Precision	Recall	F1 measure	Accuracy	Sensitivity	Specificity	Roc area
KNN	0.97	0.88	0.93	0.95	0.93	0.98	0.94
Logistic regression	0.91	0.96	0.93	0.95	0.97	0.91	0.95
CNB	0.97	0.83	0.89	0.93	0.90	0.97	0.90
BNB	0.00	0.00	0.00	0.63	0.63	0.00	0.50
GNB	0.96	0.90	0.93	0.95	0.94	0.96	0.94
Decision tree	0.92	0.94	0.93	0.95	0.96	0.93	0.95
Decision stump	0.88	0.83	0.86	0.89	0.89	0.89	0.85
Random tree	0.90	0.98	0.94	0.95	0.98	0.90	0.96
REPTree	0.85	0.91	0.87	0.90	0.93	0.85	0.93
Random forest	0.96	0.96	0.96	<b>0.97</b>	0.98	0.96	0.97
SVM	0.98	0.89	0.93	0.95	0.93	0.97	0.93
SGD	0.98	0.96	0.97	<b>0.98</b>	0.97	0.98	0.97
SMO	0.96	0.96	0.96	<b>0.97</b>	0.97	0.96	0.97
K star	0.96	0.89	0.92	0.94	0.93	0.96	0.98
Bayes net	0.91	0.93	0.92	0.94	0.95	0.91	0.99
Classification via regression	0.93	0.94	0.93	0.95	0.96	0.93	0.98
Gradient booster	0.96	0.96	0.96	<b>0.97</b>	0.98	0.96	0.97
AdaBoost M1	0.93	0.95	0.94	0.95	0.96	0.93	0.99
LogitBoost	0.96	0.92	0.95	0.96	0.95	0.96	0.99
Bagging	0.95	0.93	0.94	0.95	0.95	0.95	0.99

used for logistic regression was selected based on the grid search method. Three solvers “newton-cg,” “liblinear,” and “saga” were used and the learning rate was set in a range from 0.001 to 10. Logistic regression was able to yield an accuracy of 95% with a “Liblinear” solver and a learning rate of 10.

Figure 9 shows the run time of all implemented algorithms. The gradient booster algorithm was found to take more time than 7.74 seconds, followed by logistic regression with 4.3 seconds. This is because the gradient booster tends to overfit the model. Hence, setting a large number of boosting results increases the performance of the model. In this case, the boosting stages were set to

2000 which caused the algorithm to take more time. The same is in the case of a random forest. As discussed before, the grid search method was used in logistic regression to find the optimal solver and learning rate which resulted in more time. Decision tree was able to provide an accuracy of only 92%. The tree diagram for the dataset is given in Figure 11.

To increase the accuracy, pruning was executed. Pruning reduces the size of decision trees by removing nodes of the tree that do not contribute to model training. Pruning can also improve classification accuracy. The pruned decision was able to provide an accuracy of 95% and the same tree is shown in Figure 12.

TABLE 6: Kappa value, MAE, RMSE, and run time of implemented algorithms.

Classifier	Kappa value	MAE	RMSE
KNN	0.895	0.048	0.221
Logistic regression	0.892	0.048	0.221
CNB	0.849	0.069	0.264
BNB	0.006	0.370	0.608
GNB	0.897	0.048	0.221
Decision tree	0.891	0.048	0.221
Decision stump	0.773	0.168	0.310
Random tree	0.897	0.049	0.222
REPTree	0.799	0.128	0.287
Random forest	0.946	0.027	0.167
SVM	0.897	0.050	0.221
SGD	0.962	0.021	0.145
SMO	0.946	0.028	0.167
K star	0.885	0.057	0.228
Bayes net	0.878	0.059	0.236
Classification via regression	0.894	0.085	0.207
Gradient booster	0.942	0.030	0.170
AdaBoost M1	0.896	0.058	0.189
LogitBoost	0.917	0.054	0.174
Bagging	0.895	0.083	0.188

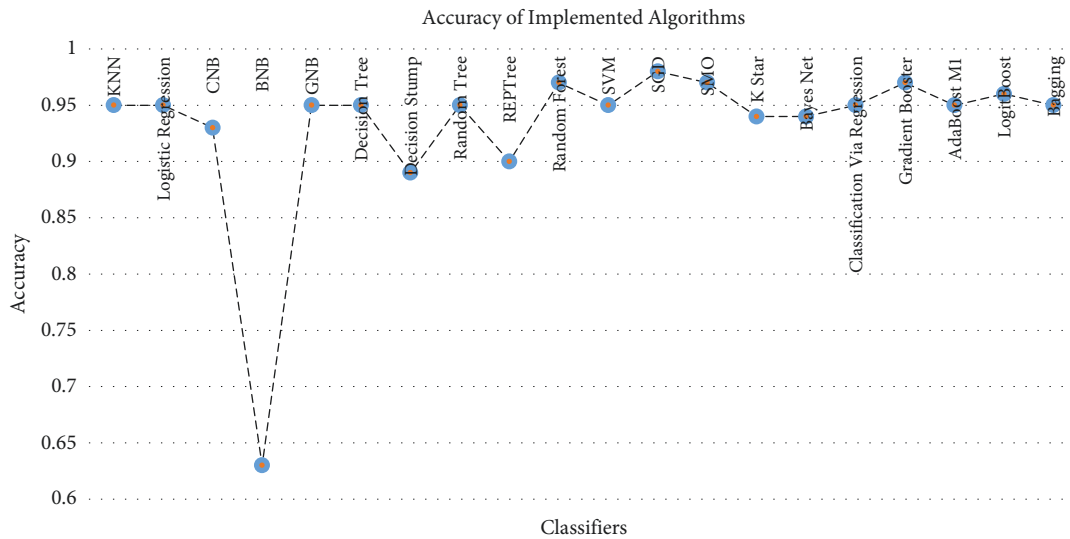


FIGURE 8: Accuracy of implemented algorithms.

5.4. *Development of Novel Ensemble Model.* The ultimate objective of every ML challenge is to find a model that can predict the desired outcome with high accuracy. Rather than creating a single model and hoping it is the best, we can use ensemble to take a variety of diverse models into consideration and average those models to get a single final model. In other words, the ensemble approach is a machine learning strategy that integrates numerous basic models to build a single highly accurate model. Common ensemble algorithms such as bagging, AdaBoost, LogitBoost, and gradient boosting are discussed above and implemented. In addition to that, a novel ensemble model has been developed combining logistic regression, stochastic gradient descent, and random forest. The hyperparameters of the developed

ensemble model are given in Table 8. The performance of this ensemble model is investigated by implementing it on both platforms (i.e., Colab notebook and WEKA software). The results obtained are compared and discussed further in the next section.

5.5. *Comparative Study between WEKA and Python Coded Results.* A comparative study has been performed between WEKA results and Python coded results. The ensemble model developed above has been used to perform this comparative study. Table 9 summarizes the results obtained through WEKA and Python by ensemble of the three algorithms.

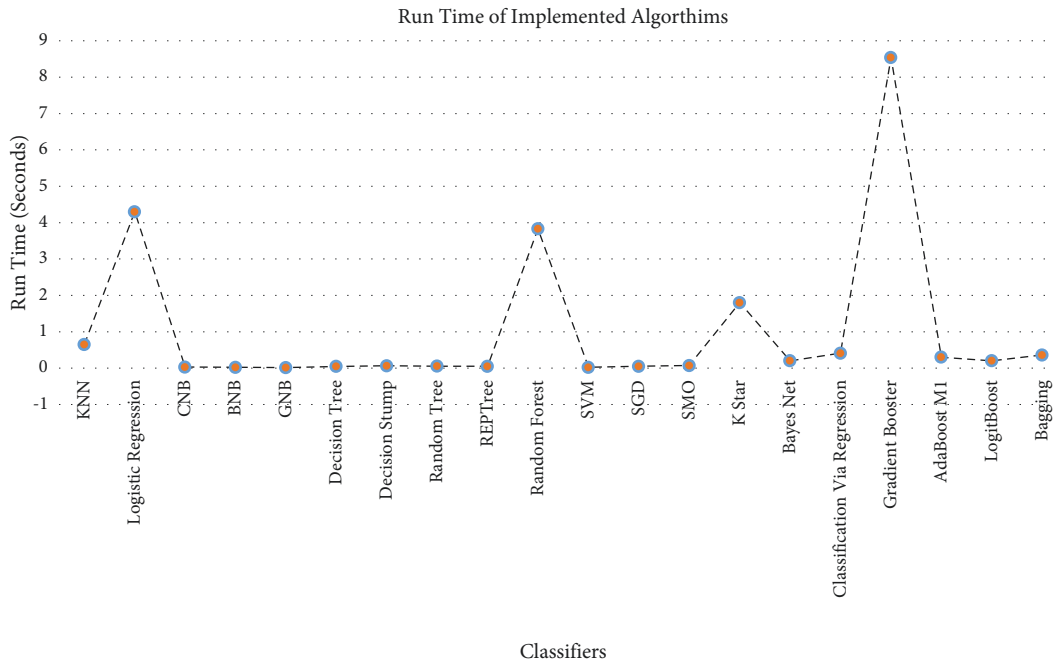


FIGURE 9: Run time of implemented algorithms.

TABLE 7: Hyperparameters of the implemented 20 classifiers.

Classifiers	Hyperparameters
KNN	Nearest neighbors—4; brute algorithm
Logistic regression	Solvers—Newton-cg, liblinear, saga; learning rate—0.001, 0.01, 0.1, 1; penalty—l2, max. iterations—2000
CNB, BNB, GNB	Alpha (smoothing parameter)—1.0 (default) var_smoothing—1.00e-09
Decision tree	Criterion—“gini”; ccp_alpha (minimal cost-complexity pruning)—0.1; splitter—“best”
Decision stump	Batch size—100
Random tree	K value—1; batch size—100; max. depth—2
REPTree	Min variance prop—0.001; batch size—100; num folds—3
Random forest	n_estimators—2000; criterion—“gini”; bootstrap—true; n_jobs—-1; ccp_alpha—0.1
SVM	C (regularization parameter)—1.0; kernel—“sigmoid”
SGD	Loss—“log”; penalty—l2; alpha—0.01
SMO	Calibrator—“logistic”; kernel—“poly Kernel”; C—0.1; epsilon—1.00e-12
K star	Batch size—100; global blend—20; entropic auto blend—true
Bayes net	Batch size—100; estimator—“simple”; search algorithm—“hill climber”
Classification via regression	Batch size—100; regression type—“additive”
Gradient boosting classifier	n_estimators—2000; loss —“deviance”; learning rate—0.01
AdaBoost M1	Batch size—100; weight threshold—100; seed—1; iterations—0
LogitBoost	Batch size—100; weight threshold—100; seed—1; iterations—50; z max—5; shrinkage—1; pool size—1
Bagging	Seed—1; iterations—50; batch size—100; bag size percent—50

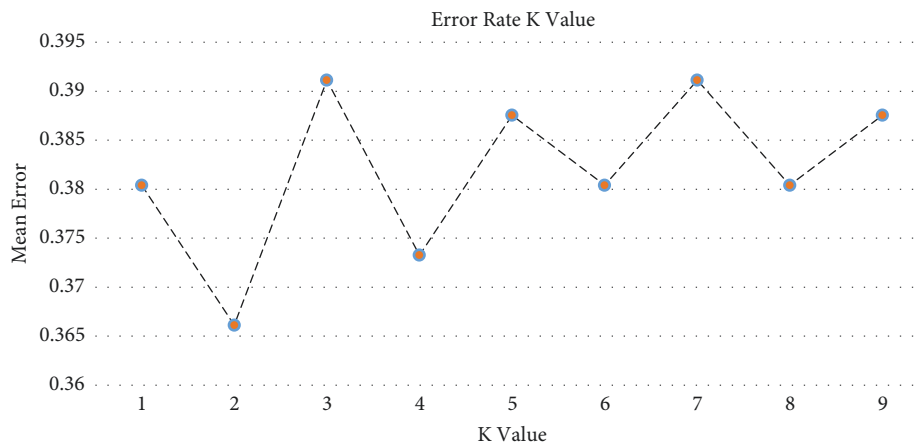


FIGURE 10: Error rate vs. K value for KNN.

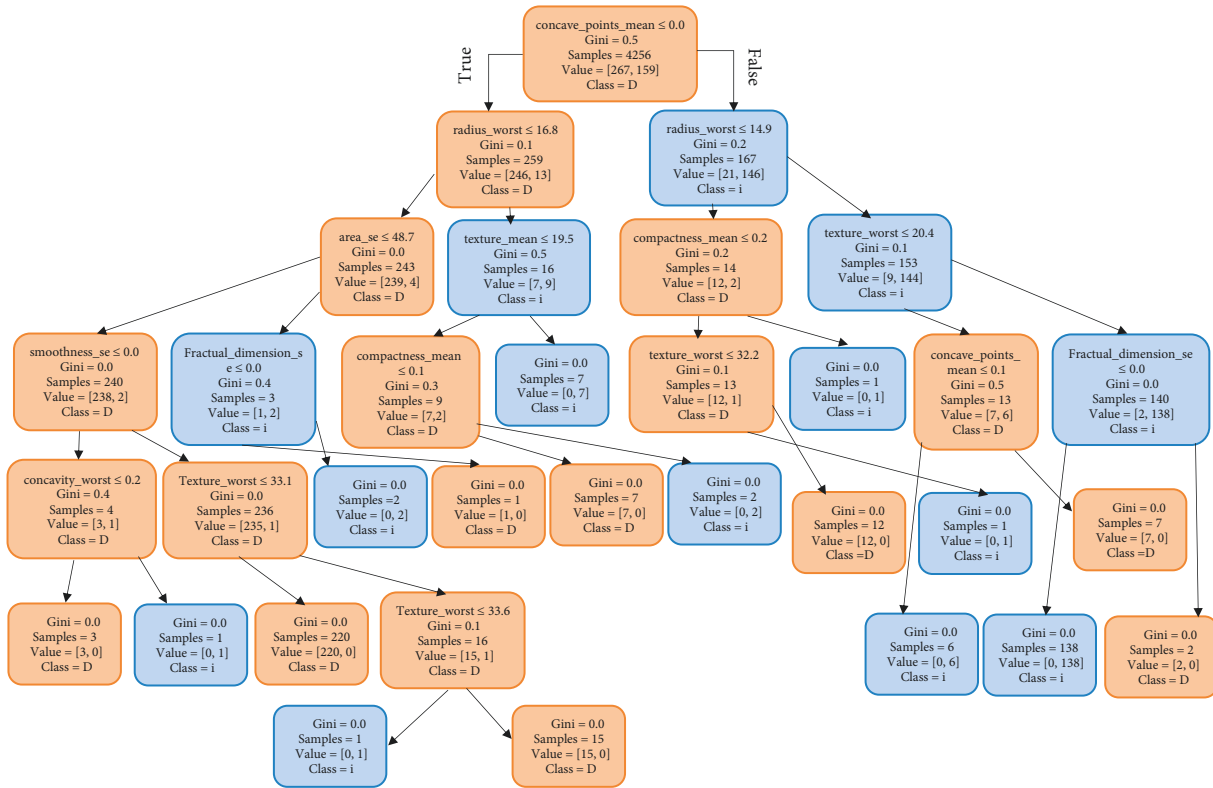


FIGURE 11: Tree diagram before pruning.

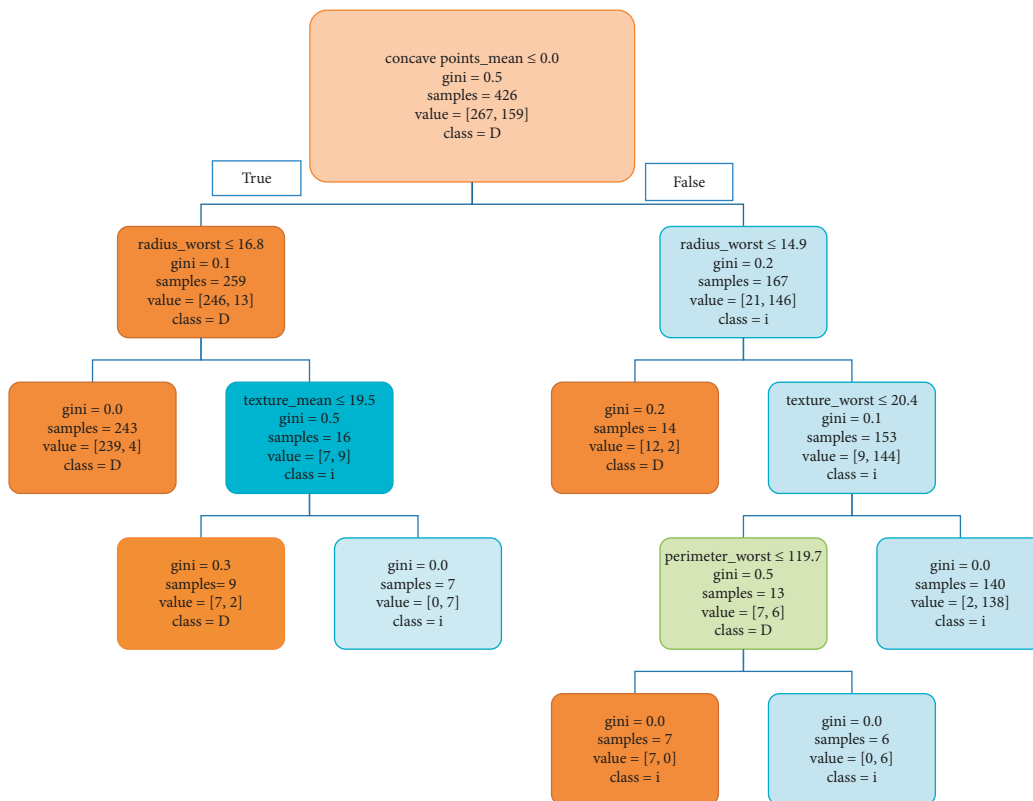


FIGURE 12: Tree diagram after pruning.

TABLE 8: Hyperparameters of the ensemble model.

Classifiers in ensemble model	Hyperparameters
Logistic regression	Solvers—Newton-cg, liblinear, saga; learning rate—0.001, 0.01, 0.1, 1; penalty—l2, max. iterations—2000 (grid search)
Random forests	$n_{\text{estimators}}$ —2000; criterion—“gini”; bootstrap—true; min. sample split—2
Stochastic gradient descent (SGD)	Loss—“log”; penalty—l2; alpha—0.01

TABLE 9: Comparison between WEKA and Python coded results.

Evaluation parameters	WEKA	Python coded
Accuracy	0.970	0.985
Precision	0.974	0.970
Recall	0.974	0.990
F1 score	0.974	0.980
True positive rate (TPR)	0.970	0.986
False positive rate (FPR)	0.030	0.015
Run time (seconds)	0.780	4.940
Kappa value	0.946	0.940
Mean absolute error (MAE)	0.045	0.026
Root mean square error (RMSE)	0.138	0.162

From Table 9, it is evident that the algorithm implemented in the Python programming language yielded an accuracy of 98.5%, while the model implemented in WEKA provided an accuracy of 97%. But the run time is more when executed in Python. This is because logistic regression uses a grid search technique to find the optimal solver and learning rate while the other 2 algorithms use a high value of boosting stages to yield better results which together accounts for the more computational time when compared to WEKA. Now, from Tables 5 and 9, it is evident that the ensemble model developed yields an accuracy of 98.5%, while the highest accuracy yielded by individual classifiers is 98% (SGD). Conventional ensembles like gradient booster, AdaBoost, LogitBoost, and bagging were able to achieve an accuracy of 97%, 95%, 96%, and 95% respectively. Hence, it is concluded that the performance of the developed ensemble model using Python is better than the traditional ML algorithms.

## 6. Conclusion

Breast cancer is a very common type of cancer and it is predominantly found in women. In this paper, 20 different ML classification algorithms were trained on a Breast Cancer dataset from the University of Wisconsin to detect the same. From the results obtained it was found that SGD has yielded the highest accuracy of 98%, while BNB provided the least accuracy of 63%. Apart from accuracy, various other evaluation parameters like confusion matrix, precision, recall, F1 measure, sensitivity, specificity, Kappa value, MAE, RMSE, and run time were calculated and tabulated. Further, a comparative study was carried out to analyze the performance of the algorithm when coded in Python and when run on WEKA. An ensemble model has been developed combining logistic regression, random forest, and SGD algorithms and deployed in the comparative study. The performance of the ensemble model in both platforms has

been compared and investigated in detail. It was found that the ensemble model developed using Python was able to achieve an accuracy of 98.5% and the model developed in the WEKA data mining tool yielded an accuracy of 97%. We believe that development and implementation of ensemble models can improve the quality of diagnosis and brings a great advantage compared to the recent works. Future directions of this work include the development of a real-time AI-based hardware prototype that can be deployed in hospitals around the world to assist doctors in detecting breast cancer. The use of AI and ML technology is limited not only to the detection of breast cancer, but also to the detection of other deadly diseases. The same technology can automate society and one such example is described in this paper. Thus, the work in this paper is an attempt to improve and automate society, establish the concept of smart hospitals, and contribute to the development of these technologies.

## Data Availability

Data will be available upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] X. Liang, X. Yu, and T. Gao, “Machine learning with magnetic resonance imaging for prediction of response to neoadjuvant chemotherapy in breast cancer: a systematic review and meta-analysis,” *European Journal of Radiology*, vol. 150, Article ID 110247, 2022.
- [2] W. Dong, W. P. Bensken, U. Kim, J. Rose, N. A. Berger, and S. M. Koroukian, “Phenotype discovery and geographic disparities of late-stage breast cancer diagnosis across U.S. Counties: a machine learning approach,” *Cancer Epidemiology Biomarkers & Prevention*, vol. 31, no. 1, pp. 66–76, 2022.
- [3] A. Pfob, B. J. Mehrara, J. A. Nelson, E. G. Wilkins, A. L. Pusic, and C. Sidey-Gibbons, “Towards patient-centered decision-making in breast cancer surgery: machine learning to predict individual patient-reported outcomes at 1-year follow-up,” *Annals of Surgery*, vol. 275, no. 4, 2022.
- [4] A. Rasool, C. Bunternghit, L. Tiejian, M. R. Islam, Q. Qu, and Q. Jiang, “Improved machine learning-based predictive models for breast cancer diagnosis,” *International Journal of Environmental Research and Public Health*, vol. 19, no. 6, p. 3211, 2022.
- [5] J. Y. Lee, K.-s. Lee, B. K. Seo et al., “Radiomic machine learning for predicting prognostic biomarkers and molecular subtypes of breast cancer using tumor heterogeneity and

- angiogenesis properties on MRI,” *European Radiology*, vol. 32, no. 1, pp. 650–660, 2022.
- [6] T. Chandrasekaran, S. Chandrasekaran, R. Hua, I. Banerjee, and A. Sanyal, “A fully-integrated analog machine learning classifier for breast cancer classification,” *Electronics*, vol. 9, no. 3, p. 515, 2020.
- [7] A. R. Vaka, B. Soni, and S. Reddy, “Breast Cancer Detection by Leveraging Machine Learning,” *ICT Express*, vol. 6, 2020.
- [8] S. Mojriani, G. Pinter, J. H. Joloudari et al., “Hybrid machine learning model of extreme learning machine radial basis function for breast cancer detection and diagnosis; A multilayer fuzzy expert system,” in *Proceedings of the RIVF International Conference on Computing and Communication Technologies (RIVF)*, pp. 1–7, Ho Chi Minh, Vietnam, Vietnam, July 2020.
- [9] V. Chaurasia and S. Pal, “Applications of machine learning techniques to predict diagnostic breast cancer,” *SN Computer Science*, vol. 1, no. 5, pp. 270–311, 2020.
- [10] M. F. Ak, “A comparative analysis of breast cancer detection and diagnosis using data visualization and machine learning applications Healthcare,” *Multidisciplinary Digital Publishing Institute*, vol. 8, no. 2, p. 111, 2020.
- [11] Y. Mahmood, N. Kama, A. Azmi, A. S. Khan, and M. Ali, “Software effort estimation accuracy prediction of machine learning techniques: a systematic performance evaluation,” *Software: Practice and Experience*, vol. 52, no. 1, pp. 39–65, 2022.
- [12] Q. Kuang and L. Zhao, “A practical GPU based KNN algorithm,” in *Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCSCST '09)*, pp. 151–155, P. R. China, December 2020.
- [13] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, “Efficient KNN Classification Algorithm for Big Data,” *Neurocomputing*, vol. 195, pp. 1–6, 2016.
- [14] B. O’Shea, “K-nearest Neighbors Algorithm (KNN) and Artificial Neural Network (ANN) Accurately Predicting Malignancy of Breast Cancer (BC) Tissue Solely Based of Features Acquired from Imaging Modalities,” 2020, <http://hdl.handle.net/1920/11759>.
- [15] X. Zhang, Y. Li, R. Kotagiri, L. Wu, Z. Tari, and M. Cheriet, “KRNN: k rare-class nearest neighbour classification,” *Pattern Recognition*, vol. 62, pp. 33–44, 2017.
- [16] S. Agarwal, B. Jha, T. Kumar, M. Kumar, and P. Ranjan, “Hybrid of naive bayes and Gaussian naive bayes for classification: a map reduce approach,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 6S3, pp. 266–268, 2019.
- [17] M. M. Saritas and A. Yasar, “Performance analysis of ANN and naive bayes classification algorithm for data classification,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 7, no. 2, pp. 88–91, 2019.
- [18] S. M. Piryonesi and T. E. El-Diraby, “Role of data analytics in infrastructure asset management: overcoming data size and quality problems,” *Journal of Transportation Engineering, Part B: Pavements*, vol. 146, no. 2, Article ID 04020022, 2020.
- [19] J. Tanha, H. Salarabadi, M. Aznab, A. Farahi, and M. Zoberi, “Relationship among prognostic indices of breast cancer using classification techniques,” *Informatics in Medicine Unlocked*, vol. 18, Article ID 100265, 2020.
- [20] R. R. Putra and H. W. Dhany, “Determination of accuracy value in id3 algorithm with gini index and gain ratio with minimum size for split, minimum leaf size, and minimum gain,” *IOP Conference Series: Materials Science and Engineering*, vol. 725, no. 1, Article ID 012088, 2020.
- [21] N. Yang, T. Li, and J. Song, “Construction of decision trees based entropy and rough sets under tolerance relation,” in *International Conference on Intelligent Systems and Knowledge Engineering*, pp. 1515–1519, Atlantis Press, Amsterdam, Netherlands, 2007.
- [22] T. N. Shah, M. Z. Khan, M. Ali, B. Khan, and N. Idress, “CART, J-48 Graft, J48, ID3, Decision Stump and Random Forest: A comparative study,” 2020.
- [23] Y. Wang, H. Zhang, H. Chen, D. Boning, and C.-J. Hsieh, “On lp-norm robustness of ensemble decision stumps and trees,” in *International Conference on Machine Learning* vol. 119, PMLR, Article ID 10104, 2020.
- [24] J. H. Joloudari, E. Hassannataj Joloudari, H. Saadatfar et al., “Coronary artery disease diagnosis; ranking the significant features using a random trees model,” *International Journal of Environmental Research and Public Health*, vol. 17, no. 3, p. 731, 2020.
- [25] J. S.K. and S. Sasikala, “S. S. Reptree classifier for identifying link SPAM in web search engines,” *ICTACT Journal on Soft Computing*, vol. 03, no. 02, pp. 498–505, 2013.
- [26] Y. Zhao and Y. Zhang, “Comparison of decision tree methods for finding active objects,” *Advances in Space Research*, vol. 41, no. 12, pp. 1955–1959, 2008.
- [27] C. Aroef, Y. Rivan, and Z. Rustam, “Comparing random forest and Support vector machines for breast cancer classification,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 2, p. 815, 2020.
- [28] R. S. Thakare, S. M. Deshmukh, and V. R. Raut, “Automatic breast segmentation and cancer detection using svm,” *Open Access International Journal of Science and Engineering*, vol. 5, no. 9, 2020.
- [29] S. V. M. Vishwanathan and M. N. Murty, “SSVM: a simple SVM algorithm,” in *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 3, pp. 2393–2398, IEEE, 2002.
- [30] A. S. Assiri, S. Nazir, and S. A. Velastin, “Breast tumor classification using an ensemble machine learning method,” *Journal of Imaging*, vol. 6, no. 6, p. 39, 2020.
- [31] M. Kiknadze and A. Gürhanlı, “Prediction of breast cancer using artificial neural networks,” *International Journal of Engineering Research in Africa*, vol. 10, no. 2, pp. 6–15, 2020.
- [32] Z. Sajjadnia, R. Khayami, and M. R. Moosavi, “Preprocessing Breast cancer data to improve the data quality, diagnosis procedure, and Medical Care Services,” *Cancer Informatics*, vol. 19, Article ID 117693512091795, 2020.
- [33] A. N. Oo and K. T. Win, “Feature selection based Sequential Minimal Optimization (SMO) for heart disease classification,” in *Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC, Seoul, Korea (South)*, August 2009.
- [34] S. B. Ud Din Tahir, A. Jalal, and M. Batool, “Wearable sensors for activity analysis using SMO-based random forest over smart home and sports datasets,” in *Proceedings of the 2020 3rd International Conference on Advancements in Computational Sciences (ICACS)*, pp. 1–6, IEEE, Lahore, Pakistan, February 2020.
- [35] D. Y. Mahmood and M. A. Hussein, “Intrusion detection system based on K-Star classifier and feature set detection,” *IOSR Journal of Computer Engineering*, vol. 15, no. 5, pp. 107–112, 2013.
- [36] T. Kavzoglu and I. Colkesen, “Entropic distance-based K-Star algorithm for remote sensing image classification,” *Fresenius Environmental Bulletin*, vol. 20, no. 5, pp. 1200–1207, 2011.
- [37] S. Hartmann, “Bayes Net and Rationality,” 2020, Preprint.



- [38] S. Dutta and S. K. Bandyopadhyay, "Early breast cancer prediction using artificial intelligence methods," *Journal of Engineering Research and Reports*, vol. 13, no. 2, pp. 48–54, 2020.
- [39] M. M. Rahman, Y. Ghasemi, E. Suley, Y. Zhou, S. Wang, and J. Rogers, "Machine Learning Based Computer Aided Diagnosis of Breast Cancer Utilizing Anthropometric and Clinical Features," *IRBM*, vol. 42, 2020.
- [40] E. Sevinç, "An empowered AdaBoost algorithm implementation: a COVID-19 dataset study," *Computers & Industrial Engineering*, vol. 165, Article ID 107912, 2022.
- [41] Y. Wang and L. Feng, "Improved Adaboost algorithm for classification based on noise confidence degree and weighted feature selection," *IEEE Access*, vol. 8, Article ID 153011, 2020.
- [42] D. G. Ganakwar and V. K. Kadam, "Face detection using logit boost algorithm with YCbCr color space," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. 1, pp. 184–189, 2020.
- [43] H. Jain, A. Khunteta, and S. Srivastava, "Churn prediction in telecommunication using logistic regression and logit boost," *Procedia Computer Science*, vol. 167, pp. 101–112, 2020.
- [44] A. M. Widodo, N. Anwar, B. Irawan, and L. Meria, "Data Mining Classification for Breast Cancer Prediction," *Procedia Computer Science*, vol. 169, 2019.
- [45] I. K. Omurlu, F. Cantas, M. Ture, and H. Ozturk, "An empirical study on performances of multilayer perceptron, logistic regression, ANFIS, KNN and bagging CART," *Journal of Statistics & Management Systems*, vol. 23, no. 4, pp. 827–841, 2020.
- [46] R. H. Saputra and B. Prasetyo, "Improve the accuracy of C4.5 algorithm using Particle Swarm Optimization (PSO) feature selection and bagging technique in breast cancer diagnosis," *Journal of Soft Computing Exploration*, vol. 1, no. 1, pp. 47–55, 2020.
- [47] R. Cristin, P. N. Jeipratha, T. Daniya, and N. Kanagavalli, "Logistic regression model with gradient descent parameter estimation," *Journal of Critical Reviews*, vol. 7, no. 12, pp. 2048–52, 2020.
- [48] A. F. Psaros, X. Meng, Z. Zou, L. Guo, and G. E. Karniadakis, "Uncertainty quantification in scientific machine learning: methods, metrics, and comparisons," 2022, <https://arxiv.org/abs/2201.07766>.
- [49] A. Suragala, P. Venkateswarlu, and M. China Raju, "A comparative study of performance metrics of data mining algorithms on medical data," in *ICCCE 2020*, pp. 1549–1556, Springer, Singapore, 2021.
- [50] M. E. Pérez-Pons, J. Parra-Dominguez, G. Hernández, E. Herrera-Viedma, and J. M. Corchado, "Evaluation metrics and dimensional reduction for binary classification algorithms: a case study on bankruptcy prediction," *The Knowledge Engineering Review*, vol. 37, p. e1, 2022.
- [51] M. Steurer, J. Hill, and P. Robert, "Norbert. Metrics for evaluating the performance of machine learning based automated valuation moders," *Journal of Property Research*, vol. 38, pp. 1–31, 2021.
- [52] C. Miller, Andrew, A. Gatys, Leon, J. Futoma, and B. Fox, "Emily. Model-Based metrics: sample-Efficient estimates of predictive model subpopulation performance," 2021, <https://arxiv.org/abs/2104.12231>.

## Research Article

# A Novel Faster RCNN with ODN-Based Rain Removal Technique

**Purnachand Kollapudi,<sup>1</sup> Mydhili K Nair,<sup>2</sup> S. Parthiban,<sup>3</sup> Abbas Mardani,<sup>4</sup> Sachin Upadhye,<sup>5</sup> Vinaykumar Nassa,<sup>6</sup> and Alhassan Alolo Abdul-Rasheed Akeji<sup>7</sup>**

<sup>1</sup>Department of CSE, B V Raju Institute of Technology, Narsapur, Telangana, India

<sup>2</sup>Department of Information Science & Engineering, MS Ramaiah Institute of Technology, Bengaluru, Karnataka, India

<sup>3</sup>Department of CSE, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, India

<sup>4</sup>Muma College of Business, University of South Florida, Tampa 33620, FL, USA

<sup>5</sup>Department of Computer Application, Shri Ramdeobaba College of Engineering and Management, Nagpur, India

<sup>6</sup>Department of AI/DS, Sharad Institute of Technology College of Engineering, Ichalkaranji, Kohlapur, India

<sup>7</sup>Department of Marketing and Corporate Strategy, Tamale Technical University, Tamale, Ghana

Correspondence should be addressed to Alhassan Alolo Abdul-Rasheed Akeji; [aaakeji@tatu.edu.gh](mailto:aaakeji@tatu.edu.gh)

Received 17 March 2022; Revised 9 April 2022; Accepted 16 April 2022; Published 4 May 2022

Academic Editor: Araz Darba

Copyright © 2022 Purnachand Kollapudi 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.

During rainy times, the impact of outdoor vision systems gets considerably decreased owing to the visibility barrier, distortion, and blurring instigated by raindrops. So, it is essential to eradicate it from the rainy images for ensuring the reliability of outdoor vision system. To achieve this, several rain removal studies have been performed in recent days. In this view, this paper presents a new Faster Region Convolutional Neural Network (Faster RCNN) with Optimal Densely Connected Networks (DenseNet)-based rain removal technique called FRCNN-ODN. The presented involves weighted mean filtering (WMF) is applied as a denoising technique, which helps to boost the quality of the input image. In addition, Faster RCNN technique is used for rain detection that comprises region proposal network (RPN) and Fast RCNN model. The RPN generates high quality region proposals that are exploited by the Faster RCNN to detect rain drops. Also, the DenseNet model is utilized as a baseline network to generate the feature map. Moreover, sparrow search optimization algorithm (SSOA) is applied to choose the hyperparameters of the DenseNet model namely learning rate, batch size, momentum, and weight decay. An extensive experimental validation process is performed to highlight the effectual outcome of the FRCNN-ODN model and investigated the results with respect to several dimensions. The FRCNN-ODN method produced a higher UIQI of 0.981 for the applied image 1. Furthermore, on the applied image 2, the FRCNN-ODN model achieved a maximum UIQI of 0.982. Furthermore, the FRCNN-ODN algorithm produced a higher UIQI of 0.998 on the applied image 3. The simulation outcome showcased the superior outcome of the FRCNN-ODN (Optimal Densely Connected Networks) model with existing methods in terms of distinct measures.

## 1. Introduction

Image acquisition in rainy times frequently encounter considerable reduction of scene perceptibility and unpleasantly disturb the efficiency of several image processing processes such as video surveillance and smart vehicles. Removal of raindrops from the rainy image becomes an essential process and finds it available in several application areas [1, 2]. Numerous single image rain removal techniques existed in the earlier times. Previous studies have utilized

distinct filtering techniques for the image decomposition process namely guided filtering and L0 smoothing filtering and afterward reconstructed the rain-free image. In recent times, several research works have discovered the physical characteristics of rain and backdrop layers and formulated them to distinct earlier terms for removing raindrops.

Gaussian mixture model (GMM) and discriminative sparse coding (DSC) are employed for modeling rain layers. In addition, joint convolution analysis and synthesis sparse

representation (JCAS) are the representative samples which explains the rain and backdrop layers with accurate mathematics-based approaches. They are found to be effective in particular cases. These techniques could not effectively adaptable to the rainy images with complex shapes of raindrops and backdrop scenarios. Presently, deep learning (DL) models are applied to remove rain drop process [3]. Many of the techniques are repeatedly necessitate recollecting massive training data with rain-free images for learning the nonlinear mapping among the rainy and rain-clean images in a dedicated way. Some of the important techniques involved are spatial attentive network (SPANet), DerainNet, multistream dense network (DID-MDN), depth attentional feature network (DAFNet), progressive image deraining network (PReNet), deep detail network (DDN), etc.

Although achievement of victory in specific context, several existing DL techniques to remove raindrop from single images poses several drawbacks. Particularly, the efficiency of the techniques is principally relying on the quality and quantity of recollected training data that comprises rain-free images pretending the network input and output. The training data includes probably a common range of raindrop shapes to cover the continually existing testing process. At the same time, the nonlinear mapping representation using the deep network model brought robust fitting ability to approximate massive structures of rain exist in the training data over the classical model-based approaches. At the same time, the complex network appearance inclines to perform repetitiveness in defining raindrops in the training data and unnecessary flexibility on adjusting with the new input rainy images. A DL technique accomplishing better training performance does not accurately filter the raindrops from real time images with complicated structures of raindrops which are not available in the training data.

The contribution of the study is given here. This paper develops a new Faster Region Convolutional Neural Network (Faster RCNN) with Optimal Densely Connected Networks (DenseNet)-based rain removal technique called FRCNN-ODN. A weighted mean filtering (WMF) technique is applied to perform the denoising process. The weighted mean filter varies from the average filter in that the mean value was calculated by repeating indicated pixels inside a neighborhood a provided number of times. A deep neural network is an ANN with numerous levels between hidden layer and output layer. Various types of neural networks occur, and they all start sharing the very same components: neural cells, synaptic, weight training, biases, and functional areas. Given that the median filter estimates the median value of a neighborhood rather than the mean filter, it has two significant advantages over the mean filter: because the median is a more robust average than the mean, a single pixel in a neighborhood that is exceptionally unrepresentative will have a little effect on the median value of the neighborhood. Besides, the DenseNet model is utilized as a baseline network to generate the feature map. Furthermore, sparrow search optimization algorithm (SSOA) is applied to choose the hyperparameters of the

DenseNet model namely learning rate, batch size, momentum, and weight decay. Using the Sparrow Search Optimization Algorithm (SSOA) requires few considerations. A number of optimization jobs outperform it. Problems with the algorithm are numerous such as inaccurate search and extended reach times. Following its introduction, the sparrow search algorithm was extensively tested. They changed it to make it better. In order to optimize search results, SSOA uses the same method as sparrows to find food. It has fewer adjustment parameters and requires less programming. It is also highly user-friendly. It used tent mapping based on random variables to improve each sparrow's sequence. So, it optimized the tents with tent perturbation and Gaussian mutation. The sparrow then discovered a way. The person on the opposite side is a fan because they are the first to learn about a sparrow. They hunt, follow, scan, find food, and inform your companion. Their "foraging area" is usually quite small. There are not many things to consider while using the Sparrow Search Optimization Algorithm (SSOA). On many optimization tasks, it outperforms other techniques. The algorithm is flawed in many complex optimization issues such as low search accuracy and long reach times. The sparrow search algorithm was intensively investigated and employed following its introduction. They improved it and made it more useful by changing it. A comprehensive simulation analysis is carried out to highlight the proficient results analysis of the FRCNN-ODN model and investigated the results with respect to several dimensions. In short, the paper contribution can be listed below:

- (i) Present a novel rain removing technique using the FRCNN-ODN model by incorporating the features of Faster RCNN and DenseNet models;
- (ii) Apply WMF technique as a denoising process and thereby enhance the image quality;
- (iii) Present the DenseNet model as a shared network to generate the feature map;
- (iv) Employ SSOA for hyperparameter optimization of the DenseNet model;
- (v) Validate the results in terms of MSE, PSNR, and SSIM.

## 2. Related Works

Depending upon the morphological component analysis, a raindrop removal process is presented as a signal decomposition issue [4]. Afterward, the researchers used a bilateral filter, dictionary learning, and sparse coding for acquiring low- and high-frequency components for obtaining the clear image outcome. Besides, Wang et al. [5] employed a screen blending technique and presented to make use of highly discriminative codes on a learnt dictionary for sparse approximation of rain and backdrop layers. For representing raindrop orientation and scaling of raindrops, Liu et al. [6] developed a GMM-oriented patch prior. Through the application of sparsity and gradient statistics of rain and backdrop layers, 3-regularization terms are defined in [7] for

progressive extraction of raindrops. Another sparse representation is presented for representing images in large-scale structure and synthesis sparse representation for describing image fine scale texture. In the meantime, a new method is developed to define the backdrop and rain layers, correspondingly. Although it performs well in particular cases, they are not adequate to properly operate in real time distinct shapes of raindrops owing to the simple and subjective considerations.

Fu et al. [8] initially developed the DerainNet for the prediction of rain-free images. To ease the training procedure, Fu et al. [9] presented the DDN technique to eliminate rain content in high frequency parts (HFP). Next, Paulraj [10] developed a conditional generative adversarial deraining network. By the consideration of the complicated nature of raindrops, the researchers in [11] additionally combined a remaining aware classification task and realized a rain density-aware multistream dense network. Moreover, the widely utilized rain method is redeveloped in [12] and created a multitask framework to cooperatively study the binary rain drop mapping, appearing raindrops, and rain-free backdrop. To obtain a high visual quality, the researchers have furthermore presented a detailed protection phase. In recent techniques, by means of repeated unfolding of the shallow ResNet with a recurring layer, Liu et al. [13] proposed a simplified baseline network, PReNet. Also, there are few studies generally focused on discovering the rain imaging techniques and generate few additional real-life rainy/background image sets.

The histogram of drop orientation and GMM is presented in [14] for detecting and extracting the rain layer, correspondingly. A deraining approach depends on error minimize among the frame and phase congruency is established in [15] for detecting and removing the raindrops. They perform better, although they need temporary video data. In [16], a general rank method was developed, where raindrops are considered as low rank. They executed the method in one image and the arrangement of images is characterized by the correlation of raindrops in a spatio-temporal way. For deraining, Wang et al. [17] applied a guided kernel on the texture element. This technique studied a mapping function among the rain images on detailed elements and raindrops.

Chen et al. [18] developed a deraining neural network which frequently erases the raindrops and growths. Even though it has obtained better outcomes in heavy rain, it led to oversmoothed in some cases. In [19], Zhang et al. has made a generative technique with an alternate update of generator and discriminator are utilized to produce the rain-free images. Multiscale discriminator is utilized to leverage characteristics from various scales; however, the introduced model included the undesired haze to the images. In addition, a 2-branch network supported by the rain technique cooperatively learned the method variables that were introduced for removing raindrops and haze like effects. It has tuned the rain-free results for controlling the degree of haze impact.

Deng et al. [20] presented the combination of the high frequency detailed layer content of the image and

degenerating the negative remaining data are efficient for the deraining method. It has reconstructed the structure effectively; however, it is ineffective in case of conceivable texture. Subbulakshmi and Prakash [21] projected a deep decomposition composition network (DDCN), a single-image deraining model. The decomposition section split the rain image to clean backdrop and rain layers. The composition network recreates the input by separating the rain-free image and rain transformation.

### 3. The Proposed FRCNN-ODN Model

The working process of the FRCNN-ODN model for rain removal is discussed here. Initially, the input image is fed into the WMF technique to remove the noise and enhance the image quality. Then, the Faster RCNN model is applied to detect the rain which contains the RPN and Fast RCNN model. Followed by, the DenseNet model is applied for the generation of feature map and the optimal hyperparameters are chosen by SSOA. The detailed working operation of these processes is discussed in the subsequent subsections.

*3.1. WMF-Based Denoising Technique.* The median filter is commonly employed to denoise an image owing to the effective smoothening outcome on noise with long-tailed probability distribution and effective preservation function on the image detail. However, the filtering window sizes include an essential impact to denoise the outcome of the classical median filter.

With smaller sizes of the filtering windows, the details are effectively stored with a low denoising outcome whereas the large size of the filtering window results in improved denoising outcome with insufficient details saved. For addressing these issues, the WMF technique is developed based on the idea of replacing the present pixel with the weighted median of adjacent pixels in the local window, while the present pixel undergoes replacement with the weighted median of its nearby pixels in a local window. It holds the unique features listed below.

- (1) The filtering kernel is nonseparable.
- (2) It could not be estimated through interpolation or down sampling.
- (3) None of the reiterative solutions exist.

So, the WMF is able to remove the noise proficiently from the noisy image whereas not robustly blur the edges of the image. This feature makes the WMF technique works well for rain removal process [22]. For determining the noisy images, a  $3 \times 3$  window is utilized for sliding over the images. Consider  $f(m, n)$  as the value of the intermediate pixel  $(m, n)$  of a local window  $R(m, n)$ , afterward, every pixel in the window is defined by

$$R(m, n) = |f(m+k, n+r)|k, \quad (1)$$

$$r = -1, 0, 1|.$$

For computing the average values of every pixel in the window  $R(m, n)$ ,

$$R(m, n)_{\text{averaged}} = \frac{1}{9} \sum_{k=-1}^1 \sum_{r=-1}^1 f(m+k, n+r). \quad (2)$$

Assume  $Z_{\max}$  and  $Z_{\min}$  are the maximum and minimum pixel values in the local window  $R(m, n)$ , at that time for the pixel value  $f(m, n)$  of the middle pixel  $(m, n)$ , if  $f(m, n) = Z_{\max}$ , or  $f(m, n) = Z_{\min}$ , or  $|f(m, n) - R(m, n)_{\text{averaged}}| > d_{m,n}$ , then, the pixel  $(m, n)$  is considered as the noise pixel. In addition, the  $d(m, n)$  denotes the threshold that can be computed as

$$d(m, n) = \frac{1}{3} \sqrt{\sum_{k=-1}^1 \sum_{r=-1}^1 [f(m+k, m+r) - R(m, n)_{\text{averaged}}]^2}. \quad (3)$$

**3.2. Rain Detection and Removal Process.** Faster RCNN is commonly used as a detector that utilizes RPN as a region proposal approach, and Fast RCNN as detector. It is an extended version of RCNN in such a way to achieve quick and accurate performance. It makes use of CNN to generate the object proposals known as RPN, as the traditional RCNN and Fast RCNN employ Selective Search at the beginning, called RPN. Alternatively, the RPN in Faster RCNN makes use of a baseline network (DenseNet) as a feature extractor for the generation of feature maps. Afterward, the feature map gets divided into several squared tiles and slides a small network across all the tiles. The small network will allocate a set of object confidence scores and coordinate points of bounding box in all the positions of the tiles. Here, DenseNet is applied to extract features and Fast RCNN to detect rain. It employs CNN to generate object proposals. The RPN applies the input feature map as input and generates a collection of object proposals, and object confidence score as the end output. The processes involved in the Faster RCNN are listed below:

- (i) Firstly, image is given as input to the DenseNet (CNN) to generate the feature map of the respective image.
- (ii) Next, RPN is utilized on feature map offering the object proposals with objectness score or value.
- (iii) Afterward, the RoI pooling layer is included to the proposals resulting in each proposal to an identical size.

Followed by, the proposals are fed into FC layer which comprises softmax and linear regression layers for classification process. Later, it generates the bounding box for the raindrops that exist in the input image.

**3.2.1. Region Proposal Networks (RPNs).** Here, the RPN is moved by offering an image as input that is considered as the backbone of CNN. At that time, the given input image is reformed, and therefore the short view is 600 px and the long one is <1000 px. Therefore, the resulted features of the backbone network ( $H \times W$ ) are low compared to the input image. The 2 backbone networks employ DenseNet for the

forthcoming rounds. It defines the two series of pixels that exist in the backbone output features corresponding to two points 16 pixels separately in the input image. In case of every point in the output feature map, the network gets aware of the position and size of the objects. It is performed through the fixation of a group of “anchors” on the input image to all the positions of output feature map from the backbone network. Every anchor represents the likely objects in distinct sizes and aspect ratios at respective positions. Next, totally 9 feasible anchors in three distinct aspect ratios and three various sizes are located on the input image for a point A on the outcome feature map.

Primarily, a  $3 \times 3$  convolution with 512 units is applied to the backbone feature map resulting in a 512-d feature map in case of all locations. Followed by, the 2 layers make use of identical process such that  $1 \times 1$  convolutional layer with 18 units are employed to classify objects, whereas  $1 \times 1$  convolutional with 36 units are utilized for bounding box regressor. Therefore, the 18 units in the classification model segment offer a resultant size of ( $H, W, 18$ ). Afterward, the result is applied to obtain the possibility of having object in each feature map within 9 anchors at varying points.

**3.2.2. Fast RCNN as Detection Network.** The Fast RCNN holds CNN with target final pooling layer which gets swapped using “ROI pooling” layer whereas the last FC layer undergoes replacement with 2 branches namely  $(K+1)$  category softmax layer and category specific bounding box regression branches.

- (i) As said earlier, the input image is fed into the CNN model to achieve the feature map of 60, 40, and 512. Also, the main reason for using RPN to generate proposals is to distribute the weights between the RBP and Fast RCNN.
- (ii) At that time, bounding box proposals gained from the RPN are given to extract features and are activated through ROI pooling layer.

**3.2.3. DenseNet Architecture.** It is developed from the classical version of ResNet that comprises of the building blocks Identity where additive is integrated to the earlier layer. Here, additive merge is applied to learn the residuals namely errors. It offers the integration of the outcome obtained from the earlier layers despite employing totaling process. Consider an image  $x_0$  which is fed as input to the CNN which has  $L$  layers. It performs a non-linear transformation  $H_l(\cdot)$  process in which the  $l$  represents layer index.  $H_l(\cdot)$  denotes the composite functions. Figure 1 shows the structure of DenseNet-169.

The output of the  $l$ th layer is denoted as  $x_l$ . The FFNN connects the outcome of the  $l$ th layer as input to  $(l+1)$ th layer which results to generate a layer transition:  $x_l = H_l(x_{l-1})$ . ResNet includes a skip connection that will bypass the nonlinear transformation by the use of an identity function:

$$x_l = H_l(x_{l-1}) + x_{l-1}. \quad (4)$$

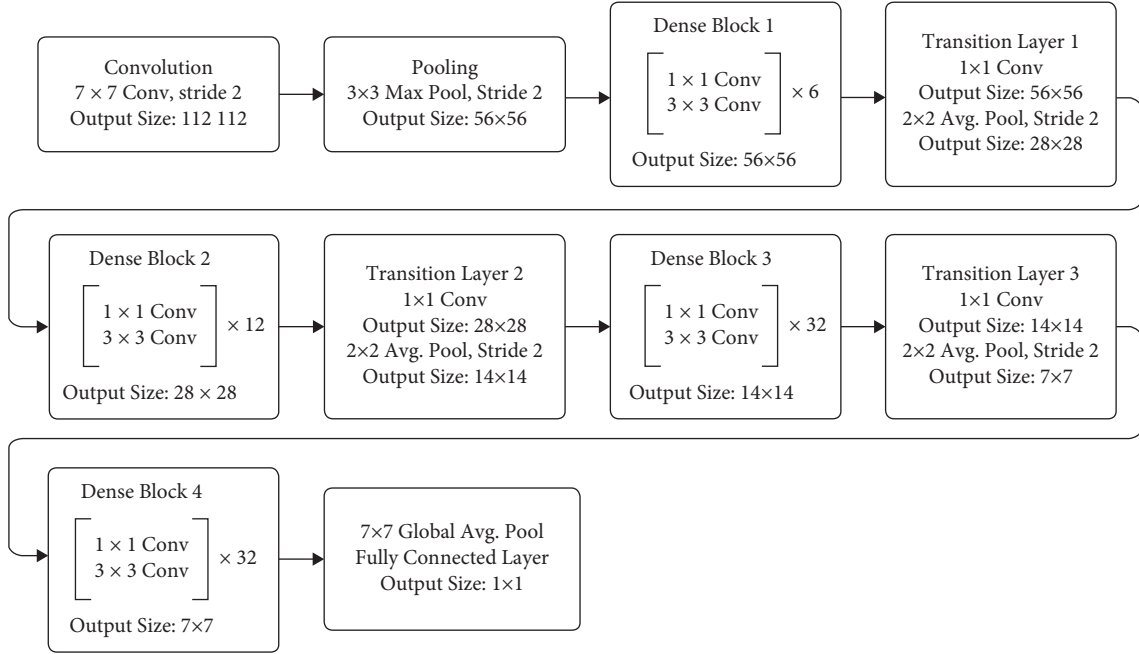


FIGURE 1: Structure of DenseNet-169.

A major benefit of ResNet is that the gradient control is straightforward from the present to earlier layers, and it is obtained by the use of an identity function. Therefore, the integration of the identity function and the end output  $H_l$  delays the data flow, which can be improved by the use of distinct connectivity patterns. The layers are linked to one another with no interruptions. At last, the  $l$ th layer attains the feature map of earlier layer,  $x_0, \dots, x_{l-1}$ , as input:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]), \quad (5)$$

where  $[x_0, x_1, \dots, x_{l-1}]$  defines the integration of the feature maps generated in layers 0,  $\dots, l-1$ .

**3.3. Parameter Optimization Using SSOA.** To choose the hyperparameters of the DenseNet model namely learning rate, batch size, momentum, and weight decay, the SSOA is applied with an intention of increasing the detection rate.

The SSOA is a population-oriented optimization algorithm which is inspired by the foraging and antipredative nature of sparrow colonies. At the foraging procedure of sparrows, the colony gets partitioned to finder and entrant. The finder holds optimal fitness value and offers foraging regions and direction for the whole colony of sparrows. At the same time, the entrant utilizes the location of the finder to consume food, if the colony of sparrows identifies a danger and the alarm. During the searching procedure, the finder with optimal fitness value is provided priority to get food. Since the finders hold the role of searching for food and directs the movements of the entire population, it identifies the food in a broad range over the other sparrows. At every round, the position of the finder gets updated as given as follows:

$$X_i^{t+1} = \begin{cases} X_i^t \times \exp\left(\frac{-i}{a \cdot T_{\max}}\right), & R_2 < ST, \\ X_i^t + Q \cdot L, & R_2 \geq ST, \end{cases} \quad (6)$$

with

$$\begin{aligned} X &= [X_1, X_2 \dots X_i \dots X_n]^T, \\ X_i &= [X_{i,1}, X_{i,2} \dots X_{i,d}], \end{aligned} \quad (7)$$

where  $t$  denotes the present round,  $n$  indicates the sparrow count,  $d$  is the variable dimensions,  $T_{\max}$  is the maximum round number,  $X_i^t$  denotes the location of the  $i$ th sparrow at round  $t$ ,  $\alpha \in (0, 1]$  is the arbitrary number,  $R_2 \in (0, 1]$  indicate the alarm value and  $ST \in [0.5, 1)$  is the safety threshold,  $L$  is a  $1 \times d$  matrix that every factor is 1, and  $Q$  is an arbitrary number. When  $R_2 < ST$ , it is represented that the foraging situation is harmless, whereas  $R_2 \geq ST$  implies that every individual has encountered predators and then, the sparrows are required to fly rapidly to other safer regions.

Based on equation (6), it is found that when  $R_2 < ST$ , the subsequent generation of finder moves over the presented position. Equation (7) revealed the difference in the finder location's value range.

$$y = \exp\left(\frac{-x}{a \cdot T_{\max}}\right), \quad (8)$$

where  $x$  indicates the number of rounds, and  $y$  denotes the range of value distinction of the finder location.

While  $x$  gets superior,  $y$  slowly narrows gradually in  $(0, 1)$  to around  $(0, 0.3)$ . If  $x$  is lesser, the possibility of  $y$  taking on a value closer to 1 is superior, and as  $x$  improves, the distribution of the values of  $y$  gets more even. So, if  $R_2 < ST$ ,

the value range difference of all dimensions of sparrow is obtaining lower. These search approaches create the SSOA widely able to local search, however, it is also causing a tendency to decrease as local optimum solutions in the last rounds.

The remaining of the sparrows are entrants that observe the finders regularly [23]. After they noticeable the finder has to initiate optimal food, it is left their present location as well as fly to optimal seeking areas. A position of the entrants is upgrading as under:

$$X_i^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}} - X_i^t}{\alpha \cdot T_{\text{max}}}\right), & i > \frac{n}{2}, \\ X_{\text{best}}^{t+1} + |X_i^t - X_{\text{best}}^{t+1}| \cdot A \cdot L, & i \leq \frac{n}{2}, \end{cases} \quad (9)$$

where  $X_{\text{best}}$  refers the select individual location, that is, the present best location;  $X_{\text{worst}}$  implies the present global worst location;  $A$  represents a  $d \times d$  matrix that all the factors inside are allocated 1 or  $-1$  in arbitrary. If  $i \leq n/2$ , it proposes that the  $i$ th entrant is seeking near the optimal position, when  $i > n/2$ , it implies that the  $i$ th entrant through the worst fitness is required to fly to other places for food. Figure 2 demonstrates the flowchart of SSA technique.

During the colony, every sparrow has the scouting as well as early warning method. In general, the sparrows which are conscious of hunter account to 15–30% of the colony. It can be defined as:

$$X_i^{t+1} = \begin{cases} X_{\text{best}}^{t+1} + \beta |X_i^t - X_{\text{best}}^t|, & f_i > f_b, \\ X_{\text{best}}^{t+1} + K \frac{|X_i^t - X_{\text{best}}^{t+1}|}{(f_i - f_w) + \varepsilon}, & f_i = f_b, \end{cases} \quad (10)$$

where  $\beta$  implies the arbitrary step length control coefficient. If  $f_i > f_g$ , the person is at the population's periphery and is vulnerable to natural enemies. If  $f_i = f_g$ , the individual is in the population's center. The sparrow needs to be close to other individuals at this time to minimize the likelihood of being taken.

#### 4. Experimental Validation

Extensive experimental validation of the FRCNN-ODN model was carried out to determine the effectual performance of the FRCNN-ODN model interms of distinct aspects. The presented FRCNN-ODN model is tested using a set of rainy images and some sample images are depicted in Figure 3.

Figure 4 visualizes the results analysis of the FRCNN-ODN model with respect to rainy and rain removed images. Figures 4(a), 4(c), and 4(e) illustrate the input rainy images and the equivalent rain-free images generated by the FRCNN-ODN model is displayed in Figures 4(b), 4(d), and 4(f).

Table 1 provides a detailed comparative result analysis of the FRCNN-ODN with existing methods with respect to distinct measures [24, 25].

Figure 5 demonstrates the PSNR analysis of the FRCNN-ODN with existing methods on the different sets of images 1–3. The figure depicted that the FRCNN-ODN model has obtained maximum PSNR value on all the applied images 1–3. For instance, on the applied image 1, the FRCNN-ODN model has resulted in an improved PSNR of 32.899 dB whereas the DID, DSC, LP, UGSM, TIP, CVPR, KGCNN, and DDCN models have demonstrated reduced PSNR of 22.409 dB, 22.302 dB, 22.227 dB, 22.763 dB, 18.045 dB, 23.325 dB, 27.851 dB, and 30.276 dB, respectively [26–29]. Similarly, on the applied image 2, the FRCNN-ODN model has obtained a maximum PSNR of 34.853 dB whereas the DID, DSC, LP, UGSM, TIP, CVPR, KGCNN, and DDCN models have demonstrated reduced PSNR of 25.294 dB, 22.219 dB, 23.034 dB, 20.626 dB, 19.215 dB, 23.781 dB, 31.231 dB, and 33.750 dB respectively. Simultaneously, on the applied image 3, the FRCNN-ODN model has accomplished a higher PSNR of 34.860 dB whereas the DID, DSC, LP, UGSM, TIP, CVPR, KGCNN, and DDCN models have demonstrated reduced PSNR of 22.359 dB, 22.519 dB, 22.393 dB, 22.382 dB, 18.793 dB, 24.013 dB, 28.031 dB, and 33.451 dB, respectively [30–32].

Figure 6 showcases the SSIM analysis of the FRCNN-ODN with existing techniques on the distinct set of images 1–3. The figure exhibited that the FRCNN-ODN model has reached superior SSIM value on all the applied images 1–3. For instance, on the applied image 1, the FRCNN-ODN model has resulted in an improved SSIM of 0.974 whereas the DID, DSC, LP, UGSM, TIP, CVPR, KGCNN, and DDCN models have outperformed minimum SSIM of 0.889, 0.863, 0.889, 0.909, 0.836, 0.902, 0.957, and 0.963, respectively [33–35]. Likewise, on the applied image 2, the FRCNN-ODN model has obtained a maximum SSIM of 0.971 whereas the DID, DSC, LP, UGSM, TIP, CVPR, KGCNN, and DDCN models have demonstrated reduced SSIM of 0.851, 0.598, 0.780, 0.641, 0.648, 0.674, 0.957, and 0.962, correspondingly. At the same time, on the applied image 3, the FRCNN-ODN model has accomplished a higher SSIM of 0.973 whereas the DID, DSC, LP, UGSM, TIP, CVPR, KGCNN, and DDCN models have showcased minimal SSIM of 0.865, 0.846, 0.894, 0.913, 0.872, 0.902, 0.966, and 0.964, correspondingly [36–38].

Figure 7 illustrates the FSIM analysis of the FRCNN-ODN with existing models on the different sets of images 1–3. The figure portrayed that the FRCNN-ODN approach has obtained higher FSIM value on all the applied images 1–3.

For instance, on the applied image 1, the FRCNN-ODN algorithm has resulted in an increased FSIM of 0.972 whereas the DID, DSC, LP, UGSM, TIP, CVPR, KGCNN, and DDCN models have demonstrated reduced FSIM of 0.927, 0.926, 0.919, 0.933, 0.911, 0.921, 0.957, and 0.963, respectively. In line with, on the applied image 2, the FRCNN-ODN model has attained a maximum FSIM of 0.975 whereas the DID, DSC, LP, UGSM, TIP, CVPR, KGCNN, and DDCN manners have demonstrated reduced FSIM of 0.924, 0.835, 0.872, 0.824, 0.851, 0.882, 0.970, and 0.962, correspondingly. Followed by, on the applied image 3, the FRCNN-ODN model has accomplished a higher FSIM of

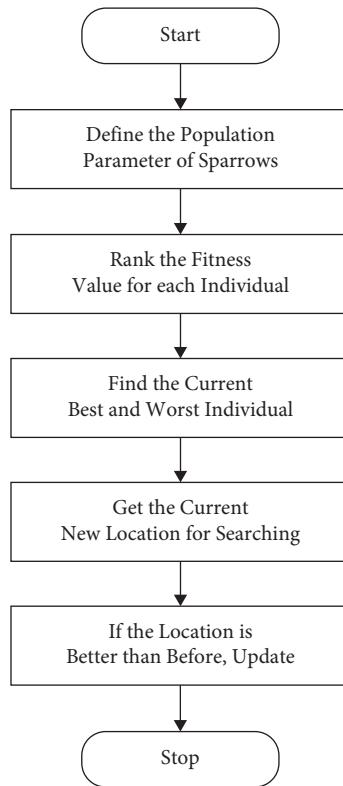


FIGURE 2: Flowchart of SSOA.



FIGURE 3: Sample test images.



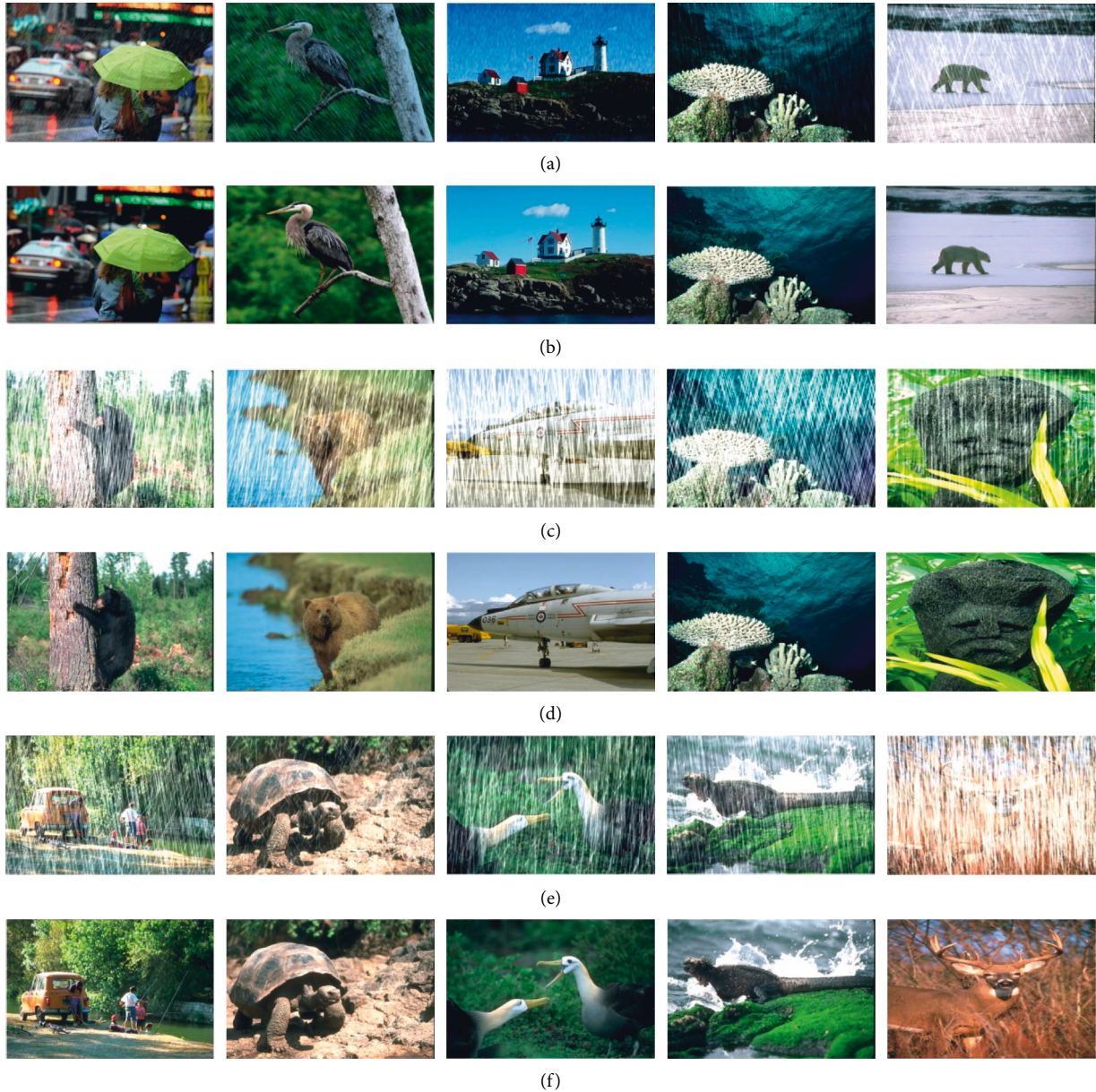


FIGURE 4: Result analysis of original images (a, c, e) and rain removal images (b, d, f).

TABLE 1: Quantitative result analysis of existing with proposed methods in terms of various measures.

Test images	Method	PSNR	SSIM	FSIM	UIQI
Image 1	DID	22.409	0.889	0.927	0.967
	DSC	22.302	0.863	0.926	0.953
	LP	22.227	0.889	0.919	0.911
	UGSM	22.763	0.909	0.933	0.911
	TIP	18.045	0.836	0.911	0.830
	CVPR	23.325	0.902	0.921	0.875
	KGCNN	27.851	0.957	0.957	0.978
	DDCN	30.276	0.963	0.963	0.979
	FRCNN-ODN	32.899	0.974	0.972	0.981

TABLE 1: Continued.

Test images	Method	PSNR	SSIM	FSIM	UIQI
Image 2	DID	25.294	0.851	0.924	0.813
	DSC	22.219	0.598	0.835	0.904
	LP	23.034	0.780	0.872	0.806
	UGSM	20.626	0.641	0.824	0.784
	TIP	19.215	0.648	0.851	0.757
	CVPR	23.781	0.674	0.882	0.846
	KGCNN	31.231	0.957	0.970	0.930
	DDCN	33.750	0.962	0.962	0.978
	FRCNN-ODN	34.853	0.971	0.975	0.982
Image 3	DID	22.359	0.865	0.902	0.975
	DSC	22.519	0.846	0.898	0.992
	LP	22.393	0.894	0.907	0.987
	UGSM	22.382	0.913	0.921	0.985
	TIP	18.793	0.872	0.903	0.969
	CVPR	24.013	0.902	0.908	0.994
	KGCNN	28.031	0.966	0.954	0.997
	DDCN	33.451	0.964	0.964	0.997
	FRCNN-ODN	34.860	0.973	0.973	0.998

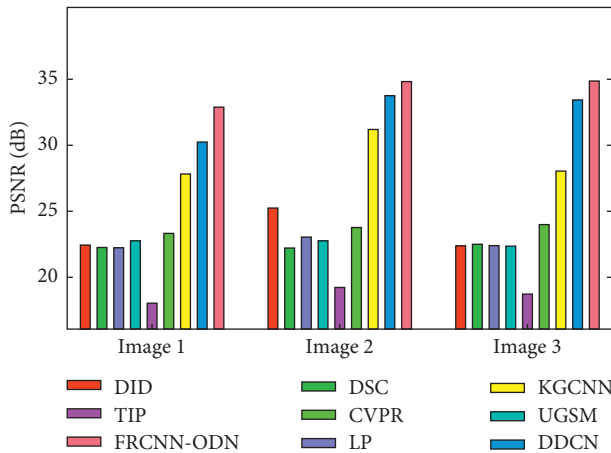


FIGURE 5: PSNR analysis of FRCNN-ODN model with existing methods.

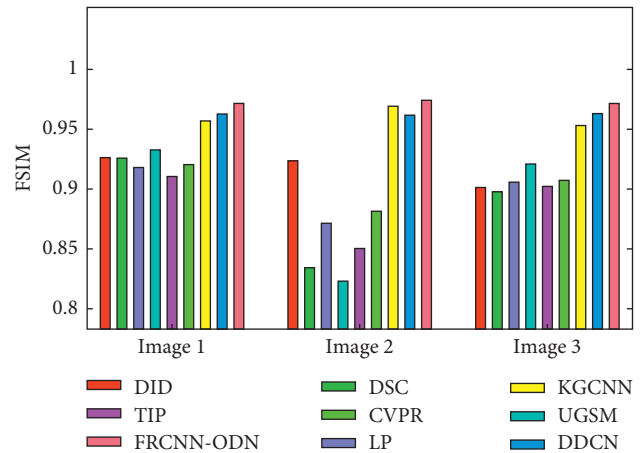


FIGURE 7: FSIM analysis of the FRCNN-ODN model with existing methods.

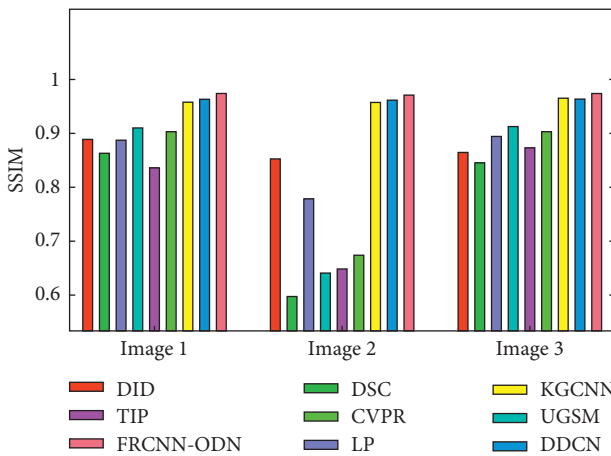


FIGURE 6: SSIM analysis of the FRCNN-ODN model with existing methods.

0.982 whereas the DID, DSC, LP, UGSM, TIP, CVPR, KGCNN, and DDCN models have outperformed lower FSIM of 0.813, 0.904, 0.806, 0.784, 0.757, 0.846, 0.930, and 0.978, respectively.

Figure 8 displays the UIQI analysis of the FRCNN-ODN with existing methods on the various set of images 1–3. The figure showcased that the FRCNN-ODN approach has achieved maximum UIQI value on all the applied images 1–3. For instance, on the applied image 1, the FRCNN-ODN method has resulted in a superior UIQI of 0.981 whereas the DID, DSC, LP, UGSM, TIP, CVPR, KGCNN, and DDCN methodologies have showcased reduced UIQI of 0.967, 0.953, 0.911, 0.911, 0.830, 0.875, 0.978, and 0.979, respectively. Also, on the applied image 2, the FRCNN-ODN model has achieved a maximum UIQI of 0.982 whereas the DID, DSC, LP, UGSM, TIP, CVPR, KGCNN, and DDCN techniques have demonstrated reduced UIQI of 0.813, 0.904, 0.806, 0.784, 0.757, 0.846, 0.930, and 0.978, correspondingly. Besides, on the applied image 3, the FRCNN-ODN

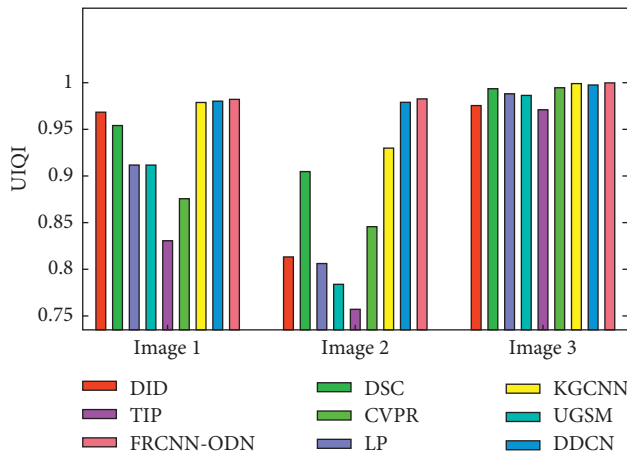


FIGURE 8: UIQI analysis of the FRCNN-ODN model with existing methods.

algorithm has accomplished a higher UIQI of 0.998 whereas the DID, DSC, LP, UGSM, TIP, CVPR, KGCNN, and DDCN models have demonstrated reduced UIQI of 0.975, 0.992, 0.987, 0.985, 0.969, 0.994, 0.997, and 0.997, correspondingly.

## 5. Conclusion

This paper has presented a novel rain removing technique using the FRCNN-ODN model by incorporating the features of Faster RCNN and DenseNet models. Initially, the input image is fed into the WMF technique to remove the noise and enhance the image quality. Then, the Faster RCNN model is applied to detect the rain which contains the RPN and Fast RCNN model. The RPN generates high quality region proposals that are exploited by the Faster RCNN model to detect raindrops. Followed by, the DenseNet model is applied for the generation of feature map. Finally, to choose the hyperparameters of the DenseNet model namely learning rate, batch size, momentum, and weight decay, the SSOA is applied with an intention of increasing the detection rate. A comprehensive simulation analysis is carried out to highlight the proficient results analysis of the FRCNN-ODN model and investigated the results with respect to several dimensions. The obtained simulation values ensured that the FRCNN-ODN model has surpassed the existing methods in terms of MSE, PSNR, and SSIM. The FRCNN-ODN approach produced an improved UIQI of 0.981 for the applied picture 1. Additionally, the FRCNN-ODN model achieved a maximum UIQI of 0.982 on the applied picture 2. Additionally, the FRCNN-ODN algorithm achieved a higher UIQI of 0.998 on the applied image 3. In future, the presented FRCNN-ODN model can be extended by the use of haze removal techniques. Further development of the FRCNN-ODN model with the aid of a haze removal algorithm and the application of haze removal approaches is anticipated in the future. The haze removal algorithms are that the light scattering through haze particles degrades the visual quality of an image, and haze removal algorithms are employed to improve this quality. This paper investigates a

variety of dehazing approaches that are commonly employed in the field of image processing. Comparing single image dehazing to other novel methods, the results are significantly superior.

## Data Availability

The manuscript contains all of the data.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.


## References

- [1] H. Wang, Y. Wu, M. Li, Q. Zhao, and D. Meng, "A Survey on Rain Removal from Video and Single Image," 2019, <https://arxiv.org/abs/1909.08326>.
- [2] R. Yasarla and V. M. Patel, "Confidence measure guided single image de-raining," *IEEE Transactions on Image Processing*, vol. 29, pp. 4544–4555, 2020.
- [3] G. Li, H. Xiang, Z. Wei, H. Chang, and L. Liang, "Non-locally enhanced encoder decoder network for single image de-raining," in *Proceedings of the 2018 ACM Multimedia Conference*, Seoul, Republic of Korea, October 2018.
- [4] M. A. Berlin, S. Tripathi, and M. A. Berlin, "IoT-based traffic prediction and traffic signal control system for smart city," *Soft Computing*, vol. 25, 2021.
- [5] Y. Wang, S. Liu, C. Chen, and B. Zeng, "A hierarchical approach for rain or snow removing in a single color image," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3936–3950, 2017.
- [6] D. Liu, B. Wen, X. Liu, and T. S. Huang, "When Image Denoising Meets High-Level Vision Tasks: A Deep Learning Approach," 2017, <https://arxiv.org/abs/1706.04284>.
- [7] L. Zhu, C. W. Fu, D. Lischinski, and P. A. Heng, "Joint bi-layer optimization for single-image rain streak removal," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2526–2534, Venice, Italy, October 2017.
- [8] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: a deep network architecture for single-image rain removal," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, 2017.
- [9] X. Fu, J. Huang, D. Zeng, H. Yue, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3855–3863, Honolulu, HI, USA, July 2017.
- [10] D. Paulraj, "An automated exploring and learning model for data prediction using balanced CA-svm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1–12, 2020.
- [11] S. Neelakandan, "A gradient boosted decision tree-based sentiment classification of twitter data," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 18, no. 4, Article ID 205027, 21 pages, 2020.
- [12] Z. Tufail, K. Khurshid, A. Salman, I. F. Nizami, K. Khurshid, and B. Jeon, "Improved dark channel prior for image defogging using RGB and YCbCr color space," *IEEE Access*, vol. 6, Article ID 32587, 2018.
- [13] Y. Liu, H. Li, and M. Wang, "Single image dehazing via large sky region segmentation and multiscale opening dark channel model," *IEEE Access*, vol. 5, pp. 8890–8903, 2017.

- [14] J. Bossu, N. Hautière, and J. P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *International Journal of Computer Vision*, vol. 93, no. 3, pp. 348–367, 2011.
- [15] R. Kamalraj, M. K. Ranjith, V. S. R. Chandra, R. Anand, and H. Singh, "Interpretable filter based convolutional neural network (IF-CNN) for glucose prediction and classification using PD-SS algorithm," *Measurement*, vol. 183, 2021.
- [16] X. Min, G. Zhai, K. Gu, X. Yang, and X. Guan, "Objective quality evaluation of dehazed images," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 2879–2892, 2018.
- [17] Y. T. Wang, X. L. Zhao, T. X. Jiang, L. J. Deng, Y. Chang, and T. Z. Huang, "Rain Streak Removal for Single Image via Kernel Guided CNN," 2018, <https://arxiv.org/abs/1808.08545>.
- [18] B.-H. Chen, S.-C. Huang, and S.-Y. Kuo, "Error-optimized sparse representation for single image rain removal," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6573–6581, 2017.
- [19] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, 2019.
- [20] L.-J. Deng, T. Z. Huang, X. L. Zhao, and T.-X. Jiang, "A directional global sparse model for single image rain removal," *Applied Mathematical Modelling*, vol. 59, pp. 662–679, 2018.
- [21] P. Subbulakshmi and M. Prakash, "Mitigating eavesdropping by using fuzzy based MDPOP-Q learning approach and multilevel Stackelberg game theoretic approach in wireless CRN," *Cognitive Systems Research*, vol. 52, pp. 853–861, 2018.
- [22] Z. Shi, Y. Li, C. Zhang, M. Zhao, Y. Feng, and B. Jiang, "Weighted median guided filtering method for single image rain removal," *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, pp. 1–8, 2018.
- [23] G. Liu, C. Shu, Z. Liang, B. Peng, and L. Cheng, "A modified sparrow search algorithm with application in 3d route planning for UAV," *Sensors*, vol. 21, no. 4, Article ID 1224, 2021.
- [24] S. Neelakandan, R. Annamalai, S. J. Rayen, and J. Arunajsmine, "Social media networks owing to disruptions for effective learning," *Procedia Computer Science*, vol. 172, pp. 145–151, 2020.
- [25] P. P. Mathai, C. Karthikeyan, M. Ashok, R. Kohar, J. Avanija, and S. Neelakandan, "Deep learning based capsule neural network model for breast cancer diagnosis using mammogram images," *Interdisciplinary Sciences: Computational Life Sciences*, vol. 14, 2021.
- [26] C. D. C. Pretty, J. R. Beulah, M. Prakash, A. Harshavardhan, and D. Sivabalaselvamani, "An automated learning model for sentiment analysis and data classification of Twitter data using balanced CA-SVM," *Concurrent Engineering Research and Applications*, vol. 29, no. 4, pp. 386–395, 2021.
- [27] R. R. Bhukya, B. M. Hardas, T. C. Anil, M. Ashok, A. Arun, and S. Neelakandan, "An automated word embedding with parameter tuned model for web crawling," *Intelligent Automation & Soft Computing*, vol. 32, no. 3, pp. 1617–1632, 2022.
- [28] P. Asha, L. Natrayan, B. T. Geetha et al., "IoT enabled environmental toxicology for air pollution monitoring using AI techniques," *Environmental Research*, vol. 205, Article ID 112574, 2022.
- [29] Y. Alotaibi, S. Alghamdi, and I. Osamah, "An efficient metaheuristic-based clustering with routing protocol for underwater wireless sensor networks," *Sensors*, vol. 22, no. 2, 415 pages, 2022.
- [30] 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. 8, 2020.
- [31] C. Al-Atroshi, V. K. Nassa, B. Geetha, S. Neelakandan, G. Sunitha, and M. G. Galety, "Deep learning-based skin lesion diagnosis model using dermoscopic images," *Intelligent Automation & Soft Computing*, vol. 31, no. 1, pp. 621–634, 2022.
- [32] A. V. R. Mayuri, N. Arulkumar, and N. Shelke, "An efficient low complexity compression based optimal homomorphic encryption for secure fiber optic communication," *Optik*, vol. 252, Article ID 168545, 2022.
- [33] D. K. Jain, S. K. S. Tyagi, and L. Natrayan, "Metaheuristic optimization-based resource allocation technique for cybertwin-driven 6G on IoE environment," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4884–4892, 2022.
- [34] S. Li, W. Ren, J. Zhang, J. Yu, and X. Guo, "Single image rain removal via a deep decomposition-composition network," *Computer Vision and Image Understanding*, vol. 186, pp. 48–57, 2019.
- [35] G. Sunitha, A. K. Singh Pundir, S. Hemalatha, and V. Kumar, "Intelligent deep learning based ethnicity recognition and classification using facial images," *Image and Vision Computing*, vol. 121, 2022.
- [36] X. Chen, X. Huang, D. Zhu, and Y. Qiu, "Research on chaotic flying sparrow search algorithm," *Journal of Physics: Conference Series*, vol. 1848, Article ID 012044, 2021.
- [37] Y. Lei, G. De, and L. Fei, "Improved sparrow search algorithm based DV-hop localization in WSN," in *Proceedings of the 2020 Chinese Automation Congress (CAC)*, Shanghai, China, November 2020.
- [38] C. Zhang and S. Ding, "A stochastic configuration network based on chaotic sparrow search algorithm," *Knowledge-Based Systems*, vol. 220, Article ID 106924, 2021.

## Research Article

# PSO-Based ANN for Studying on Carrying Capacity of Iron Ore Resources-Economy-Environment in Southern Shaanxi, China

Li YiCan,<sup>1</sup> Wei JunHao,<sup>1</sup> Pan Na,<sup>2</sup> and Hakimeh Morabbi Heravi<sup>3</sup> 

<sup>1</sup>School of Earth Resources, China University of Geosciences, Wuhan 430074, Hubei, China

<sup>2</sup>Science and Technology Development Institute, China University of Geosciences, Wuhan 430074, China

<sup>3</sup>Department of Statistics, University of Bojnord, Bojnord, Iran

Correspondence should be addressed to Hakimeh Morabbi Heravi; h.morabi@ub.ac.ir

Received 23 February 2022; Revised 13 March 2022; Accepted 24 March 2022; Published 30 April 2022

Academic Editor: Ramin Ranjbarzadeh

Copyright © 2022 Li YiCan 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.

The southern Shaanxi Province, China, has abundant iron ore resources due to local ideal metallogenic conditions. The development and utilization of iron ore resources and other related industries serve as important pillars of the regional economy. In this paper, principals and methodology of system dynamics are adopted to develop a system model of interactions of development and utilization of mineral resources, social economy, and ecological environment, using the iron ore reserves in the southern Shaanxi region as a resource base, the regional GDP as a benchmark of regional economic development, and the environmental treatment expenditure as the measurement of the regional environmental protection efforts. The model is performed based on an optimized neural network by the particle swarm optimization algorithm. Through setting variables and parameters for the system and model simulation as well as comparative research, this paper analyzes the interconnectivity and interaction of the three subsystems and explores associated relationships among mineral resources, social economy, and environmental carrying capacity in the southern Shaanxi region. In addition, the present study also proposes advice on the development and utilization of iron ore resources, ecological protection, and high-quality development of the economy in the southern Shaanxi region based on the research results.

## 1. Introduction

Mineral resources are an important material basis for the survival of mankind and sustainable economic and social development. Since the reform and opening-up, China's economy has grown rapidly and its economic scale has expanded at a fast pace. Therefore, the consumption of bulk and highly needed mineral resources (mainly iron) has been at a high level. In recent years, the development and utilization of iron ore and other mineral resources, high-quality economic development, and the sustainable development of the ecological environment have become important hotspot issues. In order to scientifically study the relationship among mineral resource development, social and economic growth, and ecological environmental protection and to promote the development of green mining, high-quality economic development, and ecological

conservation in southern Shaanxi Province, China, this paper uses this area as an example and adopts system dynamics approach to study the internal correlation among the development and utilization of iron ore resources, social and economic development, and ecological and environmental protection system in southern Shaanxi region. Models are established to carry out mineral resources–social economy–ecological environment carrying capacity research and analyze the correlation between variables, as well as the contribution and impact of mineral resources to the regional GDP and the ecological environment. Based on research results, this paper puts forward advice on scientific and rational development and utilization of mineral resources on the premise of protecting and optimizing the ecological environment, to provide scientific and technological support for promoting the high-quality development of Shaanxi's mining industry.

## 2. Exploitation Status and Iron Ore Resources Utilization in Shaanxi Region

*2.1. Exploitation Status of Iron Ore Resources Utilization in Southern Shaanxi Region.* The southern Shaanxi region is mainly located in the southern part of the Qinling metallogenic belt, with favorable geological conditions for metallogenesis and abundant mineral resources. By the end of 2017, more than 100 mineral resources have been discovered in this region, among which reserves of more than 60 metal resources such as iron, copper, lead, zinc, gold, silver, molybdenum, mercury, antimony, and others have been identified economically [1–4]. The iron ore resource reserves in the southern Shaanxi region account for 96.55% of the total iron ore resource reserves of the Shaanxi Province. The iron deposits of industrial significance are mainly distributed in the Mian-Lue-Yang area of Hanzhong City, the Bijigou area of Yangxian County, the Ziyang-Zhenping area of Ankang City, and the Zhashui-Shanyang area of Shangluo City. These areas belong to the administrative region of Hanzhong City, Ankang City, and Shangluo City. There are 62 iron mines above designated size in the southern Shaanxi region, of which 4 are large deposits, 9 are medium deposits, 49 are small deposits, and 116 are ore occurrences. A cumulative iron resource reserve of 1.16 billion tons has been identified, of which 1.050 billion tons are from large- and medium-sized deposits, accounting for 90.50% of the total iron ore resource reserves in the southern Shaanxi region. The recoverable reserve of iron ore resources is 1.059 billion tons. Of the 62 iron ore deposits above the designated size, 13 are currently being mined, accounting for 20.97% of the total number of iron ore deposits above the designated size, 12 are suspended, and 37 are unexploited. The total designated production capacity of iron ore of the Shaanxi Province is 15,752,000 tons/year, which is 137,000 tons/year higher than that of 2016. The output of iron ore is 3,509,300 tons, and the capacity utilization rate is 22.28%, of which the production capacity of large- and medium-sized mines accounts for 93.76% [2].

The development and utilization of iron ore resources in the southern Shaanxi region and its related industries play an important role in the regional economy and are an important pillar of the economy in the southern Shaanxi region and the whole province. According to statistics data, in 2017, Shaanxi Province's industries above designated size (enterprises of which main business annual income reaches 20 million Yuans) achieved a total industrial output of 2,485.443 billion Yuans, and mining and related processing and manufacturing industries yielded 1,419.827 billion Yuans, accounting for 57.1% of province's total industrial output above designated size, all indicating that the development of iron ore resources and its related industries is essential to the local economy [3]. The evaluation results of the iron ore resources potential of Shaanxi Province show that reserve of iron ore resources above 1,000 meters in the southern Shaanxi region reaches up to 1.548 billion tons, indicating huge potential. Therefore, the scientific and rational development and utilization of these iron ore

resources will play an important role in the high-quality development of the regional economy.

*2.2. Analysis of the Main Current Problems.* The environmental carrying capacity of iron ore resources is the basis for the operation of the entire regional ecological and economic system; therefore, rational development and utilization of the iron ore resources is an important basis for the sustainable development of the economy in the southern Shaanxi region. Mineral resources are nonrenewable resources, the development of which cannot be carried out in an overexploited manner; instead, it must be on a scientific and rational basis to achieve extended utilization of the mineral resources and sustainable development of regional economy and ecological environment.

The Qinling Mountains are the central water tower of China and serve as an important ecological barrier. The important metal minerals deposits in Shaanxi Province are mainly located in the Qinling metallogenic belt. If the environmental protection of this region is largely overlooked, the disorderly mining development will cause serious damage to the Qinling ecological environment. Mining development must align with the concept of green development, adhere to the principle of giving priority to environmental protection, and promote the development of ecological mining. The transformation of mineral resource utilization and management methods must be advanced to build a new pattern of mining development and utilization in Shaanxi Province.

The iron ore resource industries in the southern Shaanxi region have played a very important role in supporting the economic development of the entire region. In recent years, due to environmental protection and other reasons, some mining production has been stopped, and the mining industries experienced a serious decline, affecting regional economic development to a certain extent. Shaanxi Province's natural resources authority has carried out the campaign of "Three Protections and Three Governances" to rectify the mining development activities, which has played a positive role in the environmental protection of the Qinling Ecological Zone and the rational development and utilization of mineral resources. However, there are still many problems, and the solution needs to take into consideration of the actual situation of the southern Shaanxi region and establish a scientific governance system and scheme to ensure the coordinated development of resource utilization, environmental protection, and economic development.

To sum up, iron ore resources, economy, and environment cannot exist in isolation, but the three form a complex systematic project in which the three factors interact and restrict each other. It is necessary to carry out systematic research on resources, economy, and environment carrying capacity to explore the interactions among mineral resources, economy, and environmental carrying capacity in this region and provide reasonable advice to support decision-making in achieving sustainable development of the regional economy.

### 3. System Dynamics Construction of Mineral Resources Model, Social Economy, and Ecological Environment

At present, most research on the carrying capacity of iron ore resources–social economy–ecological environment is carried out under specific environmental or economic conditions, such as unilateral research on environmental carrying capacity with mining area as the main target. In fact, instead of limiting the research object of carrying capacity of the mineral resources–social economy–ecological environment to a certain mining area, a larger region should be included. At the same time, the mineral resources, economy, and environment in the region do not exist independently, but the three are interrelated and restrict each other and form a complex system. This paper uses the system dynamics model to establish a complex system that can simulate the changes of regional mineral resources, social economy, and ecological environment and to explore the carrying capacity of regional mineral resources, social economy, and ecological environment.

#### 3.1. Boundary Definition and Overall Structure of the Model

**3.1.1. Model System Boundary Definition.** The theory of system dynamics mainly obeys the concept of system science that “Every system must have a structure, and the system structure determines the system function.” Based on primarily the mutual feedback between the various components within the system, causal relationships and flow relationships between the systems are formed. Different results are then simulated according to different scenarios. At present, system dynamics mainly focus on the supply and demand relationship of a certain mineral resource in the field of mineral resource development and utilization. For example, in the study by Naderi et al. qualitative representation and quantitative system dynamics simulation of the water resources system in the Qazvin Plain, Iran, are presented taking into account the energy intensity of water supply and interconnected water use sectors [5]. In the study by Li et al., a system dynamics model was established to clarify the causal feedback and dynamic interaction mechanism between the components of urban resilience [6]. In the study by Song et al., the land green supply chain is divided into subsystems of technology (T), energy (E), environment (E), and economy (E), which construct the TEEE dynamic model [7].

This paper focuses on the iron ore resource reserves in the southern Shaanxi region and selects Shangluo City, Ankang City, and Hanzhong City in Shaanxi Province as the boundaries of the research area and years 2010 to 2030 as the time boundary, of which the actual statistics of years 2010 to 2016 are used for model simulation and years 2017 to 2030 for the forecast. The time interval is set to 1 year. The content boundary includes subsystems related to mineral resources, economy, and “the Three Waste Discharge” (exhaust gas, wastewater, and solid waste). The key variables are determined based on the influencing factors among the three

major systems of mineral resource development and utilization, regional economic development, and regional ecological environment from which the related variables are derived until the final variables affecting the entire system are determined and then the boundary of the entire system is set.

**3.1.2. Model Master Structure.** The development and utilization process of mineral resources is complicated and closely linked to social economy and ecological environment, involving many elements. In line with the hypothesis of rational man in economics, three major subsystems of iron ore resources, social economy, and ecological environment are established and form an organic whole system. And according to the influence and interaction among various elements in the whole system, state variables, rate variables, auxiliary variables, and constants are introduced, respectively, dividing the whole system into 3 parts. Research is carried out at multiple levels and from multiple aspects.

In the model system, the iron ore resource subsystem uses resource reserves as the core which is also the basic assurance for regional economic development. The mining, beneficiation and metallurgy, and comprehensive utilization of iron ore resources can promote regional economic development on the one hand, but on the other hand, such processes are liable to produce the Three Waste and hence damaging the ecological environment and introducing negative effects. To cope with these issues requires mining enterprises to attach importance to environmental protection, upgrade and transform their development and utilization technology, and rationally develop and utilize iron ore resources on the premise of protecting the environment.

The economy subsystem in the system model is based on regional GDP, which provides economic support for the balanced development of other subsystems, and is a benchmark of regional economic development. On the one hand, regional economic development can provide financial support for the technological transformation and upgrading of mineral resource exploration, development, and utilization and improve the resource insurance capability and technical level for sustainable development; on the other hand, the sustainable development of the regional economy can increase investment in environmental governance and restoration, improve and enhance environmental governance and restoration technologies, and address the negative impact on the ecological environment due to resource development and utilization.

The ecological environment subsystem in the model relies on the investment of environmental protection and governance expenditure and is an important guarantee for the development and utilization of regional mineral resources and the coordinated development of ecological environment and economy. On the one hand, prioritizing environmental protection in the process of mineral resource development and increasing investment in comprehensive environmental management and restoration can continuously improve the environment and improve quality and

level of people's life; on the other hand, a beautiful and harmonious environment can feedback the development of mining and promote the development and utilization of mineral resources. The organic combination of these two can jointly promote the sustainable development of regional resources, economy, and environment.

The system dynamics model constructed in this paper includes the iron ore resources subsystem, economy subsystem, and ecological environment subsystem, namely, the iron ore resources–social economy–ecological environment system (Figure 1).

**3.2. Cause-and-Effect Diagram, Flow Chart, and Main Parameters of the System Model.** Basic data of iron ore resources in the southern Shaanxi region included in this study were collected from the followings materials: Shaanxi Statistical Yearbook 2010–2017, Shaanxi Mining Annual Report 2010–2017, China Environment Database, China Economic and Social Development Statistical Database, China Regional Economic Database, National Mineral Resources Potential Evaluation Project-Report on Evaluation Results of Iron Ore Resource Potential in Shaanxi Province, and National Mineral Resources Potential Evaluation-Report on Prediction Results of Important Minerals in Shaanxi Province and China Mineral Geology [8–12]. According to the system structure diagram shown in Figure 1 and the analysis and research on the representativeness and importance of the basic data of iron ore resources in each subsystem, the representative variables are selected [13]. According to the interactions of each variable and the relationship among them, the vensim6.4 platform is used to establish the system dynamics cause-and-effect diagram and flow chart of the iron ore resources–social economy–ecological environment cycle, aiming to analyze the future development and utilization of mineral resources and regional economic development trends and to provide corresponding countermeasures [14].

Because the iron ore resources–economy–environment system model in the southern Shaanxi region is relatively complex, the parameters in this model are numerous and almost inaccessible, mainly including constants, initial values, and so on [15–17]. The constants were determined by using correlation analysis, regression analysis, expert opinion, a weighted average of historical data, and so on.

**3.2.1. Iron Ore Resource Subsystem.** The main variables of the iron ore resource subsystem include the total regional GDP, investment in iron ore resource exploration, resource reserves (new and known resource reserves), development scale, production, output value, and benefits of development and utilization of iron ore resources (see Table 1). As the basis, the iron ore resource reserves in the subsystem determine the development scale, production, output value, and development and utilization benefits of iron ore resources, while the development output value and benefits of iron ore resources determine the contribution of iron ore resources development and utilization to regional GDP, as well as further investment in exploration and the increase in

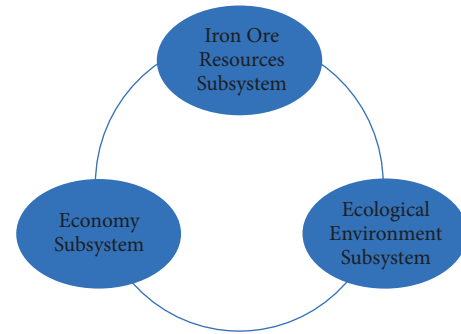


FIGURE 1: Iron ore resources–social economy–ecological environment system.

iron ore reserves [18]. These main variables constrain and promote each other and form mutual cause-and-effect relationships. The cause-effect relationship diagram among the main variables of the iron ore resource subsystem is designed with the vensim6.4 platform (Figure 2).

The specific cause-effect relationship of the iron ore resource development subsystem is deduced from Figure 2:

Regional GDP  $\rightarrow$  investment in iron ore resources  $\rightarrow$  new reserves  $\rightarrow$  iron ore resource reserves  $\rightarrow$  development scale of iron ore resource  $\rightarrow$  iron ore resource production  $\rightarrow$  iron ore output value  $\rightarrow$  regional GDP (positive feedback).

The concept presented above depicts a positive feedback loop. The cause-effect diagram reflects that with the increase in regional GDP, more investment in the exploration and development of mineral resources will be spent, and therefore the mineral reserves, development scale, and production will increase accordingly and the mining output value and benefits will also show an incremental trend, finally further promoting the continuous rise of regional GDP.

According to the cause-effect diagram of the development and utilization of iron ore resources and considering the mutual influence and restriction among various elements, the regional GDP and iron ore resource reserves are selected as state variables, with annual GDP increment and new iron ore resource reserves as the rate variables and investment in iron ore resource exploration, mineral investment ratio, development and utilization efficiency of iron ore resource, recovery rate of dressing and smelting, comprehensive utilization rate, mining recovery rate, annual mineral growth rate, iron ore resource production, iron ore output value as auxiliary variables or constants. The flow chart of the iron ore resource development subsystem designed by software is shown in Figure 3.

The main model parameter equations of the iron ore resource development subsystem are as follows:

- (1) Investment in iron ore resource exploration = the ratio of iron ore investment regional GDP
- (2) Iron ore resource reserves = INTEG (new-added iron ore resource reserves; 404)
- (3) Development and utilization efficiency of iron ore resource = LN (iron ore resource reserves \* (extract



TABLE 1: Variables of iron ore resource subsystem.

Variables	Definition
Regional GDP	GDP of southern Shaanxi region (Shangluo city, Ankang city, Hanzhong city)
Exploration and investment in iron ore resources	Total investment in geological projects in southern Shaanxi, basic geological survey projects, geological scientific research and technological innovation, and mineral resources exploration projects
Iron ore resource reserves	New resource reserves and known reserves of iron ore in southern Shaanxi
Development scale and production of iron ore resource	Iron ore resource production and development scale in southern Shaanxi, with iron ore production selected
Iron ore output value	The gross value of iron ore production in southern Shaanxi
Development and utilization benefits of iron ore resources	Comprehensive utilization efficiency of iron ore resources in southern Shaanxi

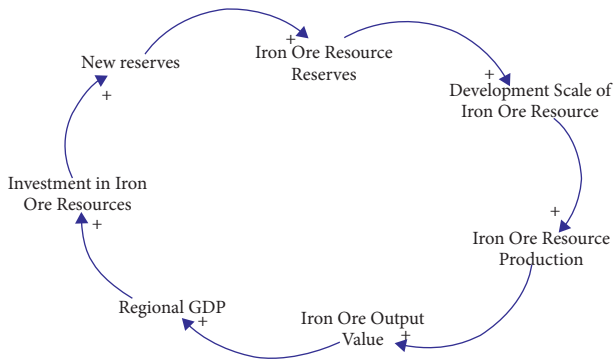


FIGURE 2: Cause-effect relationship diagram of the iron ore resource subsystem.

recovery rate + comprehensive utilization rate + recovery rate of Beneficiation and Metallurgy))

- (4) Iron ore resource production = total demand for iron ore resource \* annual iron ore growth rate \* development and utilization efficiency of iron ore resource
- (5) Iron ore output value = 0.4 \* iron ore resource production \* the ratio of mineral investment

Among them, 404 in the iron ore resource reserve is the initial value of the iron ore mineral resource reserve, and 0.4 in the iron ore output value is the incremental regression coefficient of the iron ore output value and GDP.

3.2.2. *Economy Subsystem.* The economy subsystem is the economic base for the coordinated development of the regional resource-economy-environment. This subsystem mainly studies the benefits of regional economic development on the development and utilization of regional iron ore resources, regional environmental governance, and the future development trend of regional GDP. The economy subsystem mainly includes factors such as regional GDP, primary industry output, secondary industry output, tertiary industry output, total demand for iron ore resource, production of iron ore resource, iron ore output value, and environmental governance and restoration. The main variable, regional GDP, is the sum of that in southern Shaanxi cities, including Shangluo City, Ankang City, and Hanzhong City. The output value of the primary, secondary, and

tertiary industries corresponds to the added value of them in Shaanxi over the years. The demands for iron ore resources in the primary, secondary, and tertiary industries are the consumption of iron ore in the three major industries. The total iron ore demand is manifested as the total iron ore consumption of the three major industries.

Analysis results shown in Figure 4 demonstrate the cause-effect relationship diagram of the economy subsystem designed by software.

The specific cause-effect relationship of the economy subsystem is deduced from Figure 4:

- (1) Regional GDP → total demand for iron ore resource → development scale of iron ore resource → iron ore resource production → iron ore output value → regional GDP (positive feedback).
- (2) Regional GDP → primary industry output value → demand for iron ore resource of the primary industry → total demand for iron ore resource → development scale of iron ore resource → iron ore resource production → iron ore output value → regional GDP (positive feedback).
- (3) Regional GDP → secondary industry output → demand for iron ore resource of the secondary industry → total demand for iron ore resource → development scale of iron ore resource → iron ore resource production → iron ore output value → regional GDP (positive feedback).
- (4) Regional GDP → tertiary industry output → demand for iron ore resources of the tertiary industry → total demand for iron ore resource → development scale of iron ore resource → iron ore resource production → iron ore output value → regional GDP (positive feedback).

The abovementioned feedback loops are all positive. The cause-effect relationship diagram mainly reflects that the increase in regional GDP brings an increase in the demand for iron ore resources for the primary, secondary, and tertiary industries, respectively, and provides investment for the development and utilization of iron ore resources. The increase in iron ore resource production leads to the increase in iron ore output value, which in turn affects the growth of regional GDP and forms a sustainable development of regional economy.

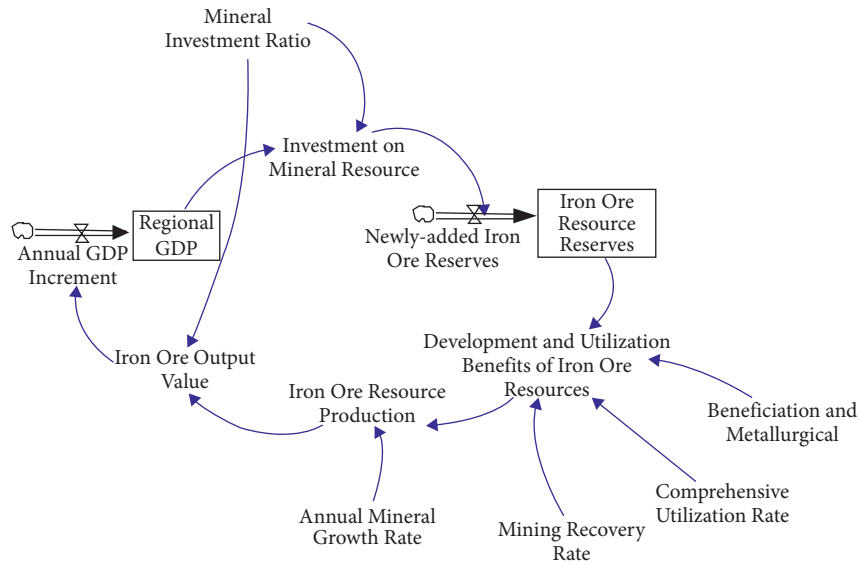


FIGURE 3: Flow diagram of iron ore resource development subsystem.

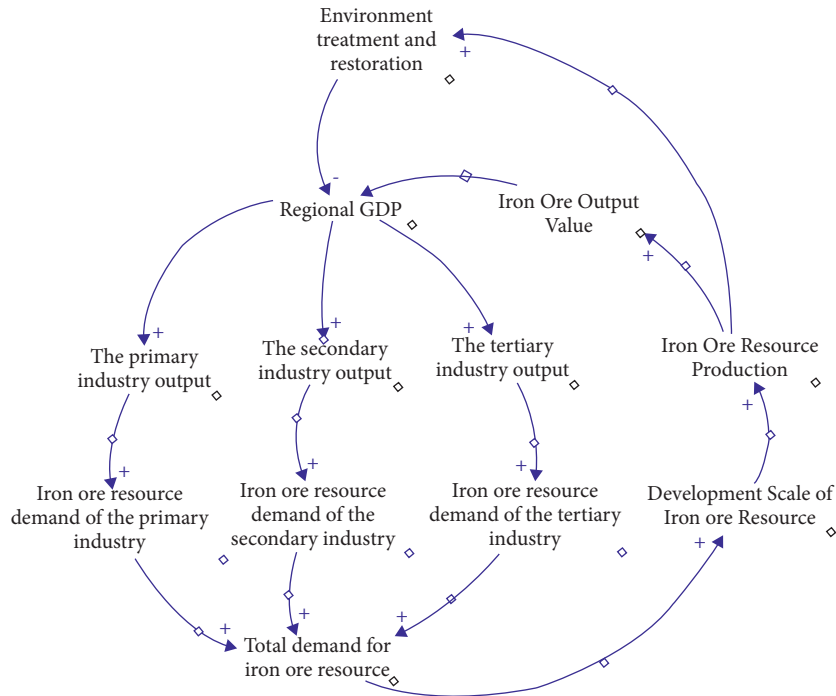


FIGURE 4: The cause-effect relationship diagram of the economy subsystem.

- (5) Regional GDP  $\rightarrow$  total demand for iron ore resource  $\rightarrow$  development scale of iron ore resource  $\rightarrow$  iron ore production  $\rightarrow$  environmental governance and restoration  $\rightarrow$  regional GDP (negative feedback).
- (6) Regional GDP  $\rightarrow$  primary industry output value  $\rightarrow$  demand for iron ore resource of primary industry  $\rightarrow$  total demand for iron ore resource  $\rightarrow$  development scale of iron ore resource  $\rightarrow$  iron ore resource production  $\rightarrow$  environmental governance and restoration  $\rightarrow$  regional GDP (negative feedback).

- (7) Regional GDP  $\rightarrow$  secondary industry output value  $\rightarrow$  demand for iron ore resource of the secondary industry  $\rightarrow$  total demand for iron ore resource  $\rightarrow$  development scale of iron ore resource  $\rightarrow$  iron ore resource production  $\rightarrow$  environmental governance and restoration  $\rightarrow$  regional GDP (negative feedback).
- (8) Regional GDP  $\rightarrow$  tertiary industry output value  $\rightarrow$  demand for iron ore resource of the tertiary industry  $\rightarrow$  total demand for iron ore resource  $\rightarrow$  development scale of iron ore resource  $\rightarrow$  iron ore resource production  $\rightarrow$  environmental

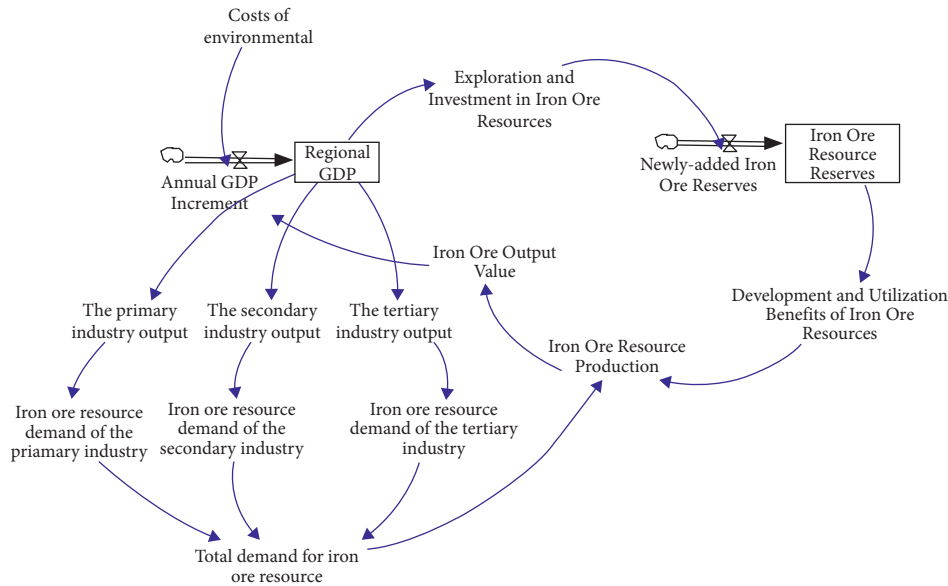


FIGURE 5: Economy subsystem flow diagram.

governance and restoration  $\rightarrow$  regional GDP (negative feedback).

The abovementioned feedback loops are all negative. The cause-effect diagram mainly indicates that the development of the three major industries worsened the regional ecological environment pollution, and therefore it is imperative to increase investment in environmental governance and restoration. However, the consequence of which is that the regional economic growth, the expansion of economic scale, and the sustainable development of the regional economy will be curbed.

According to the cause-effect relationship diagram of the economy subsystem, combining the economy subsystem with the iron ore resource development subsystem and considering the characteristics of various elements, regional GDP and iron ore resource reserves are selected as state variables, with annual GDP increment and newly added iron ore resource reserves as rate variables. The output value of three main industries, the demand for iron ore resource of the three industries, the total demand for iron ore resource, and the cost of environmental treatment and restoration are set as auxiliary variables or constants. The flow diagram of the economy subsystem designed by software is shown in Figure 5.

The main model parameter equations of the economy subsystem are as follows:

- (1) The primary industry output value = regional GDP \* 0.1
- (2) The secondary industry output value = regional GDP \* 0.6
- (3) The tertiary industry output value = regional GDP \* 0.3
- (4) Total demand for iron ore resources = iron ore resources demand of the primary industry + iron ore resources demand of the tertiary industry + iron ore resources demand of the secondary industry

Among them, the coefficients of the primary industry output value, the secondary industry output value, and the tertiary industry output value are averaged value based on the proportion of the three major industries output value in southern Shaanxi in 2010.

**3.2.3. Ecological Environment Subsystem.** The ecological environment subsystem is an important assurance for the development and utilization of regional mineral resources and the coordinated development of ecological environment and economy. The ecological environment restricts regional economic development and the scale of mineral resources development and utilization. The system mainly studies the environmental pollution caused by the development of iron ore resources and the investment in environmental governance and restoration and the impact on the entire regional economy and the redevelopment and utilization of iron ore resources. The main variables of the environmental subsystem include regional GDP, solid waste discharge, and wastewater discharge and exhaust gas emissions due to iron ore development, environmental pollution governance, and restoration costs, and iron ore resource development scale. Environment subsystem cause-effects relationship is shown in Figure 6 as designed by software.

Based on Figure 6, the main cause-effect diagram of the ecological environment subsystem can be drawn as follows:

- (1) Regional GDP  $\rightarrow$  iron ore resource development scale  $\rightarrow$  solid waste discharge  $\rightarrow$  environmental pollution  $\rightarrow$  environmental pollution treatment costs  $\rightarrow$  regional GDP increment  $\rightarrow$  regional GDP (positive feedback)
- (2) Regional GDP  $\rightarrow$  iron ore resource development scale  $\rightarrow$  wastewater discharge  $\rightarrow$  environmental pollution  $\rightarrow$  environmental pollution treatment costs  $\rightarrow$  regional GDP increment  $\rightarrow$  regional GDP (positive feedback)

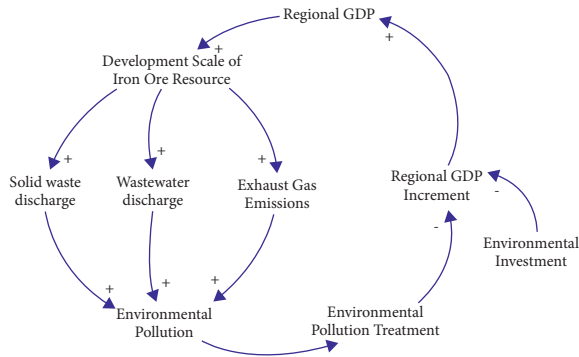


FIGURE 6: The causal diagram of the ecological environment subsystem.

- (3) Regional GDP  $\rightarrow$  iron ore resource development scale  $\rightarrow$  exhaust gas emissions  $\rightarrow$  environmental pollution  $\rightarrow$  environmental pollution treatment costs  $\rightarrow$  regional GDP increment  $\rightarrow$  regional GDP (positive feedback)

The feedback presented above is a positive feedback path. This cause-effect diagram mainly reflects that with the expansion of the development scale of iron ore resources, the discharge of wastewater, exhaust gas, and solid waste will increase proportionally, so it is necessary to increase the cost of environmental pollution treatment and restoration, which will lead to the slowdown of regional GDP and influence the regional economic development.

Now, the iron ore resource development subsystem, economy subsystem, and ecological environment subsystem are combined into an organic whole, with the full consideration of the mutual influence and interaction among these subsystems. Regional GDP, iron ore resource reserves, and the Three Waste pollution are selected as state variables; annual GDP increment, newly added iron ore resource reserves, and generation and reduction of the Three Waste pollution are set as rate variables; unit treatment cost of three wastes pollution, environmental treatment and restoration cost, and the Three Waste pollution changes per unit of output are taken as auxiliary variables or constants. The resulted flow diagram of the ecological environment subsystem designed by software is shown in Figure 7.

The main model parameter equations of the ecological environment subsystem are as follows:

- (1) Wastewater pollution = INTEG (wastewater production-wastewater reduction; 158)
- (2) Solid waste pollution = INTEG (solid waste generation-solid waste reduction; 143.5)
- (3) Exhaust gas pollution = INTEG (exhaust gas generation-exhaust gas reduction; 323.5)
- (4) Wastewater production = per unit wastewater generation/the benefit of iron ore resources development and utilization
- (5) Wastewater reduction = change in per unit wastewater reduction

- (6) Solid waste production = change in per unit solid waste generation/the benefit of development and utilization of iron ore resources
- (7) Wastewater reduction = change in per unit wastewater reduction
- (8) Solid waste reduction = per unit solid waste reduction \* comprehensive utilization rate of solid waste
- (9) Exhaust gas generation = per unit exhaust gas production/the benefit of development and utilization of iron ore resources
- (10) Exhaust gas reduction = change in per unit exhaust gas reduction
- (11) Environmental treatment cost = (per unit treatment cost of solid waste + per unit treatment cost of exhaust gas + per unit treatment cost of wastewater + change in per unit solid waste reduction + change in per unit waste gas reduction + change in per unit wastewater reduction)

For these parameters, 158 is the initial value of wastewater pollution, 143.5 is the initial value of solid waste pollution, and 323.5 is the initial value of exhaust gas pollution.

The flow diagram of the iron ore resource-economy-environment model system is shown in Figure 8.

The data simulation in this study is based on optimized neural networks which are explained in the following.

#### 4. Optimized Multilayer Perceptron Neural Networks

Today, the use of intelligent systems, especially artificial neural networks, has become so widespread that these tools can be classified as basic and common tools in basic mathematical operations. One of the most basic neural models available is the multilayer perceptron (MLP) model, which simulates the translational function of the human brain. In this type of neural network, most of the behavior of human brain networks and signal propagation has been considered, and hence they are sometimes referred to as feedforward networks. Each neuron in the human brain receives input (from another neuron or non-neuron) and processes it, transmitting the result to another cell (neuronal or non-neural) [19]. This behavior continues until a definite result is reached, which is likely to eventually lead to a decision, process, thought, or move.

An artificial neural network is made up of inputs, outputs, weights, biases, and an activation function. Weights and biases are randomly assigned. The inputs are multiplied by the weights, and the values obtained are added together and then biased. The result passes through the activator function and forms the output of neurons [20]. However, by randomly assigning weights, the result is usually not appropriate. So, the weights need to change. Weight changes should be made in such a way that the neuron outputs are close to the actual outputs.

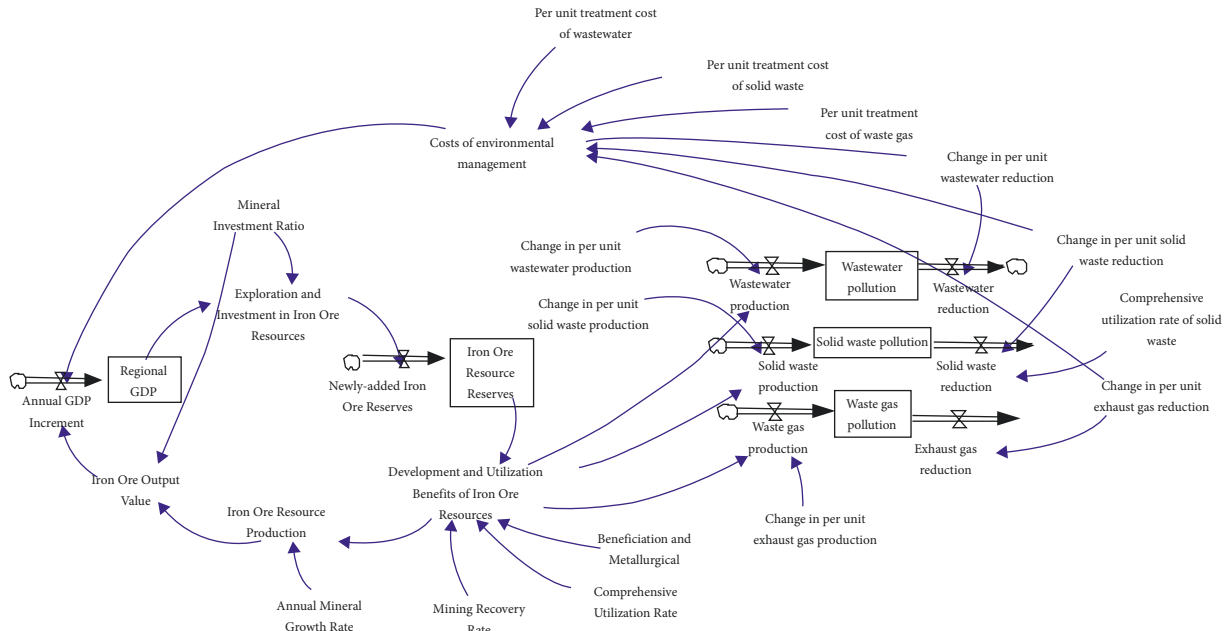


FIGURE 7: Flow diagram of ecological environment subsystem.

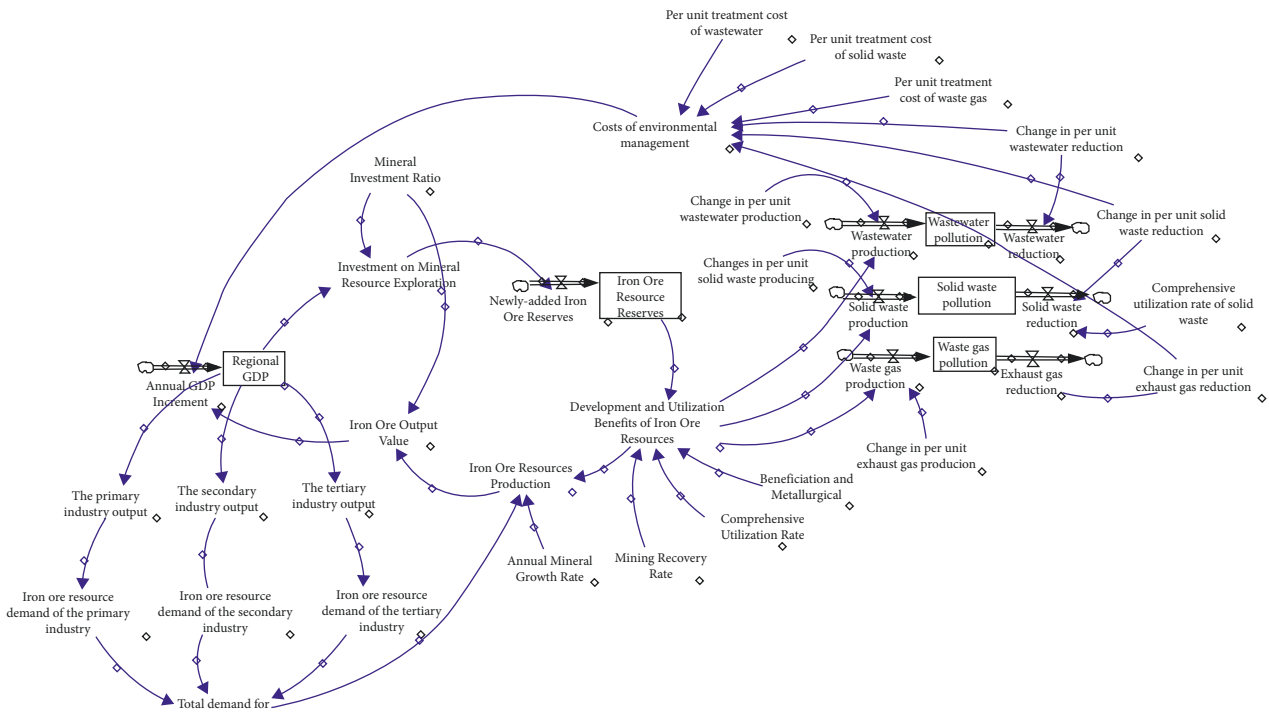


FIGURE 8: Flow diagram of the iron ore resource-economic-environmental model system.

The process of changing the weights of neurons to achieve the desired output is called neuron learning. We use a dissipation function to check the function of neurons. The goal of neuron learning is to reduce the amount of loss to zero or at least close to zero. Here, an optimization algorithm is used to do this. The task of the optimization algorithm is to find the weights that minimize the mean square error (MSE) function between the network and the desired values for all the training samples. The formula for the MSE is described as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^n (z_i(k) - z_i^*(k))^2, \quad (1)$$

where  $z^*$  describes the desired value,  $n$  describes the number of steps in the training dataset, and  $z$  is the network output and is achieved as follows:

$$z_i = \sum_{j=1}^n w_{ij} \alpha_j + \beta_j, \quad (2)$$

where  $\alpha_i$  and  $\beta_j$  represent the input and the bias variables and  $w_{ij}$  is the connection weight between  $\alpha_i$  and the hidden neuron  $j$ .

The next stage is to utilize the activation function and to trigger the output of the neurons. Here, the sigmoid function has been employed. The mathematical model of this function is given as follows:

$$f_j(x) = \frac{1}{1 + e^{-z_j}}. \quad (3)$$

Now, as aforementioned, for optimizing the network based on weights and biases, (1) should be optimized. In this study, metaheuristic technique has been used for minimizing the MSE.

The purpose of metaheuristic algorithms is to find an acceptable solution, given the limitations and needs of the problem. In determining the solution to a problem, there may be different solutions to it. In this study, the particle swarm optimization (PSO) algorithm as the most popular optimization algorithm has been utilized. The PSO algorithm simulates bird swarming behaviors. Imagine the following scenario: a group of birds is accidentally exposed to food in an area. There is only one food item in the search area. Not all birds know where food is. However, they know how much food there is in each iteration. The solution is to look for the bird that is closest to the food. PSO designers adapted this scenario and used it to solve optimization problems. In PSO, each solution is a "bird" in the search space that is called "particle." All particles have proportion values that are evaluated by the proportion function for optimization and have velocities that guide the particle flight. Particles flow through the problem space with the optimal particles.

The PSO algorithm starts with a group of random particles (solution) and then searches with the generation update. In each iteration, each particle is updated with two "best" values. The first is the best solution ever obtained. This value is called  $p_{best}$ . Another "best" is the value that has been achieved so far by every particle in the population. This is the best universal value and is called  $g_{best}$ . When a particle takes a part of the population as its topological neighbors, the best value is the best local and is called  $l_{best}$ . After finding the best values of  $p_{best}$  and  $g_{best}$ , the particle updates its speed and position with the following equations:

$$\begin{aligned} X_{new} &= X_{current} + V_{new}, \\ V_{new} &= V_{current} + c_1 * r \\ &* (X_{p_{best}} - X_{current}) + c_2 * r \\ &* (X_{g_{best}} - X_{current}), \end{aligned} \quad (4)$$

where  $V$  describes the particle velocity,  $X_{current}$  defines the current particle (solution),  $p_{best}$  and  $g_{best}$  are already defined,  $r$  specifies a random number between 0 and 1, and  $C_1$  and  $C_2$  represent learning factors. Here,  $c_1 = c_2 = 2$ .

The MLP network here uses BP technique for the network training. This BP technique is gradient descent-based technique which forms some disadvantages, such that trapping into local criteria is the most important

Initialize the particles of  $N$  weight and the algorithm other parameters.  
 Calculate the fitness value for each PSO-MLP.  
 Position updating of the parameters based on the algorithm mechanisms.  
 Check termination condition.  
 If termination criteria are not reached, go to (3).  
 If the criteria condition is reached, go to (7).  
 End.

ALGORITHM 1: The hybrid PSO-MLP.

shortcoming. Here, we used the PSO algorithm to resolve this issue. The algorithm of the hybrid PSO-MLP is given as follows:

## 5. Simulation and Result Analysis

**5.1. Model Checking.** No theoretical model could be perfectly consistent with a real system. However, an objective real system can be described and simulated by a theoretical model. Whether the structure of the theoretical model is reasonable requires comparing the simulation results of the model with the actual data to determine the authenticity and validity of the theoretical model, which also affects the formulation and implementation of relevant policies. Therefore, in order to use the system dynamics model to study the carrying capacity of iron ore resources-economic-environment in the southern Shaanxi region, it is essential to test the validity and authenticity of the model, to provide a scientific basis for local government to formulate relevant policies.

**5.1.1. Historical Test.** The historical test mainly analyzes whether the simulation results of the model are consistent with the actual results, which is one of the indicators of model validity. In this study, the rationality of the model is judged by comparing the variation trend of the model simulation results with the practical data. It is generally believed that the relative error between historical data and simulation data is within the range of 10%. And the model is considered to be better for realistic simulation.

We select statistical data from 2010 to 2016 to test the authenticity of regional GDP and iron ore resource reserves in the southern Shaanxi area, to compare and analyze the running results of the model, as shown in Tables 2 and 3, among which iron ore resources in southern Shaanxi use 30% of the mineral investment as the base.

Based on the historical test results, it can be seen that the relative numerical errors between the output data and the real data from 2010 to 2016 are between  $-0.1$  and  $0.1$ . Although the errors in 2011, 2015, and 2016 are a little large, these data fall within the normal range. Therefore, the model is consistent with the development trend of the actual system and can reflect the development of iron ore resources and economic development in the region to a certain extent. Therefore, the model is considered to be effective.

TABLE 2: Comparison of historical and simulated values of regional GDP.

Year	Real GDP (100 million yuans)	Simulated GDP value (100 million yuans)	Error (%)
2010	1122.66	1122.66	0.00
2011	1417.60	1350.50	-4.97
2012	1674.79	1624.25	-3.11
2013	1997.16	1953.20	-2.25
2014	2267.26	2270.52	0.14
2015	2433.18	2548.68	4.53
2016	2691.48	2825.30	4.74

Data source: Shaanxi statistical yearbook.

TABLE 3: Comparison of historical value and simulated value of iron ore reserves.

Year	Real value of iron ore reserves (100 million tons)	Simulated value of iron ore reserves (100 million tons)	Error (%)
2010	4.04	4.04	0
2011	3.87	4.01	3
2012	3.85	3.98	3
2013	3.99	3.95	-1
2014	3.98	3.85	-3
2015	4.00	3.80	-5
2016	3.97	3.82	-4

Data source: Shaanxi mining annual report.

*5.2. Model Simulation.* Through the system simulation, we can better understand the development of iron ore resources in southern Shaanxi by increasing the investment ratio of mineral resources in southern Shaanxi by 10% and 20%, respectively, to simulate and compare the regional GDP and iron ore resource demand in southern Shaanxi area. The software can simulate the GDP and demand for iron ore resources in southern Shaanxi. The simulated results are displayed in Figures 9 and 10. Curve 1 represents a 20% increase in the mineral investment ratio; curve 2 represents a 10% increase in the mineral investment ratio; and curve 3 represents the initial state.

Based on Figures 9 and 10, the following results can be obtained:

- (1) The change of the mineral investment ratio parameter has a significant impact on the regional GDP, showing a positive correlation. Due to the increase in the investment ratio of iron ore resources, the regional GDP growth in southern Shaanxi has been promoted. In addition, the increasing scale of iron ore exploration and mining activity will lead to aggravation of environmental pollution. It is necessary to balance the scale of mining and regional economic development so that the entire circulatory system will be developed better.
- (2) There is a positive correlation between the change of the mineral investment ratio parameter and the total demand for iron ore resources. When the ratio of investment in minerals increases, the demand for iron ore resources in the region also continues to increase. At the same time, the increasing market demand for iron ore resources will inevitably lead to an increase in the investment in iron ore exploration and development, and the over-exploitation of iron ore resources will cause serious environmental pollution.

Therefore, the balance between the mining scale of iron ore resources and the ecological environment affects the sustainable economic development of southern Shaanxi.

To sum up, the sustainable development of the regional economy requires moderate development and utilization of iron ore resources. At present, the consumption of iron ore resources in China has been running at a high level. And the demand for iron ore is booming, which the excessive exploitation will exert a huge impact on the ecological environment. Appropriate resource development and utilization can not only keep stable regional economic growth but also ensure the protection of the ecological environment. Therefore, it is necessary to evaluate the carrying capacity of regional resources, economy, and environment to ensure the sustainable development of the regional economy.

*5.3. Changes in Regional Carrying Capacity.* The resource carrying capacity is reflected among the difference between the reserves of mineral resources and the production of mineral resources. The larger the difference, the smaller the pressure, and the stronger the carrying capacity, vice versa.

The economic carrying capacity is reflected among the difference between the iron ore output value and the regional GDP. The larger the difference, the smaller the pressure, and the stronger the carrying capacity, vice versa.

The environmental carrying capacity is reflected among the difference between the cost of environmental governance and restoration and the regional GDP. The smaller the difference, the greater the pressure, and the weaker the carrying capacity, vice versa.

According to the actual data of iron ore resource reserves and iron ore production in the southern Shaanxi region, the results of iron ore resource carrying capacity in the southern

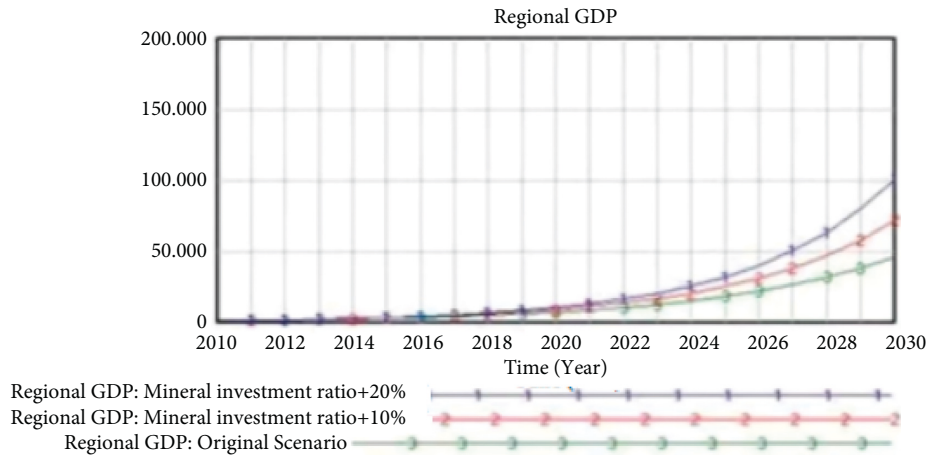


FIGURE 9: Regional GDP comparison curve.

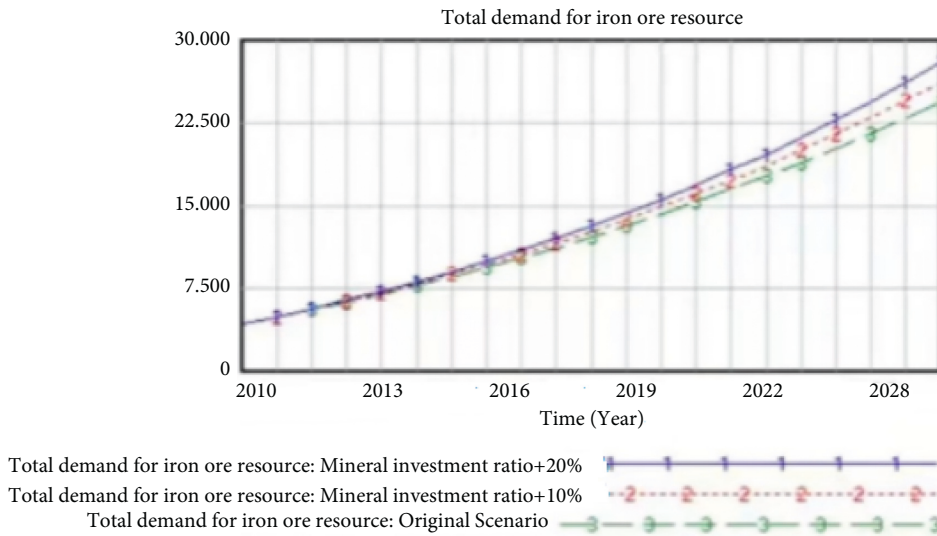


FIGURE 10: Comparison curve of total demand for iron ore resources.

Shaanxi region are obtained through system model simulation (Figure 11). The iron ore resource reserves in the southern Shaanxi region have been decreasing since 2010. Meanwhile, iron ore production has become higher and higher. The discrepancy between these two parameters began to shrink after 2011 and the resource carrying pressure was getting stronger. After 2015, the production of iron ore gradually decreased, and the difference between the two fluctuated in the range between 34,000 tons and 36,000 tons, and the situation tended to be stable. This suggests that the resource carrying capacity has become stable since 2015. Before 2015, only the development of iron ore resources was emphasized and investment in iron ore exploration was largely neglected, leading to relying on retained reserves. Thus, the carrying capacity of iron ore resources became weaker. Therefore, with the development and consumption of iron ore reserves, the exploration of iron ore needs to be strengthened to continuously increase the reserves of iron ore resources and enhance the carrying capacity of regional iron ore resources.

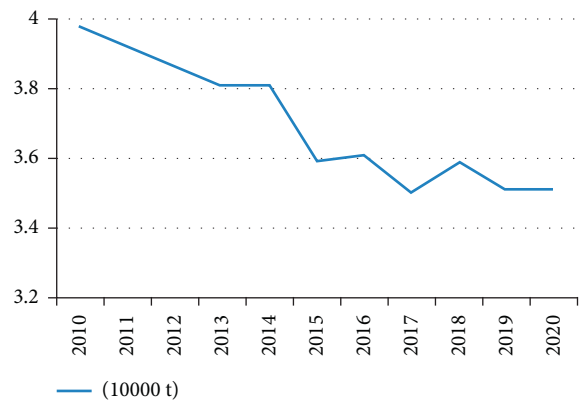


FIGURE 11: The carrying capacity of iron ore resources in southern Shaanxi region.

According to the actual data of the GDP in the southern Shaanxi region and the system model simulation, the relation between the mineral output value and regional GDP in the southern Shaanxi region in recent years is calculated



to obtain the economic carrying capacity (Figure 12). The relation between the cost of environmental governance and restoration and regional GDP in the southern Shaanxi region is calculated to obtain the environmental carrying capacity (Figure 13).

The result analysis of economic carrying capacity (Figure 12) and environmental carrying capacity (Figure 13) is as follows:

- (1) Since 2010, the difference between the iron ore output value and the regional GDP in the southern Shaanxi region is quite large, and it increased year by year from 2010 to 2015. After 2015, the growth rate of the difference is relatively slower, but the difference is still increasing. The reason is that the development speed of regional GDP is significantly higher than that of the regional iron ore output value. Therefore, it shows that the regional economic carrying capacity is relatively strong. The contribution and impact of development and utilization of iron ore in the southern Shaanxi region on regional economic development show decreasing or stabilizing trend. The regional economy depends less on the development and utilization of iron ore, and thus the regional economic carrying capacity is relatively strong.
- (2) From 2010 to 2016, the difference between the growth rate of environmental governance and restoration cost and the growth rate of regional GDP is small, indicating that the environmental carrying capacity is relatively weak. In recent years, on the one hand, thanks to the environmental protection requirements, the closure of some mines, and reduction and restriction of mine production result in the reduction of cost of the environmental governance and restoration; on the other hand, China has issued strict regulations and policies on environmental governance and restoration to build up green mines, and natural resources and environmental protection departments have strengthened supervision and management, so the awareness of environmental protection has been generally enhanced, and the investment in environmental governance and restoration has continued to increase. As a result, the difference between the growth rate of environmental governance and restoration costs and the growth rate of regional GDP has increased, and the regional environmental carrying capacity has increased and continuously enhanced.

To sum up, iron ore resources, economy, and environment do not exist independently, but the three are interrelated and restrict each other and cannot be neglected. Under the current situation of strong demand for iron ore in China, sufficient iron ore reserves can ensure that the regional demand for iron ore can be met. The scientific and rational development of iron ore resources is beneficial to the economic and social development of the southern Shaanxi region. Economic

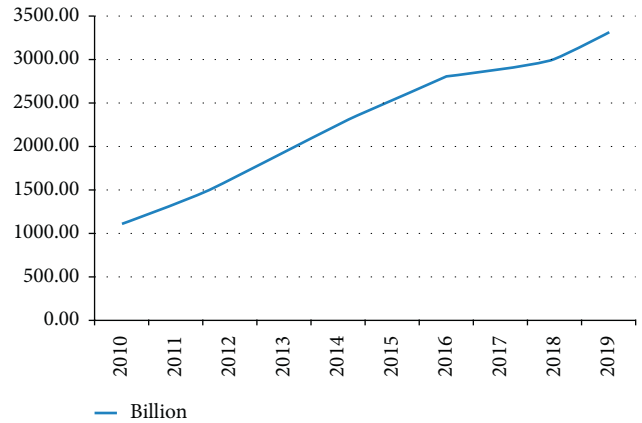


FIGURE 12: Economic carrying capacity in southern Shaanxi region.

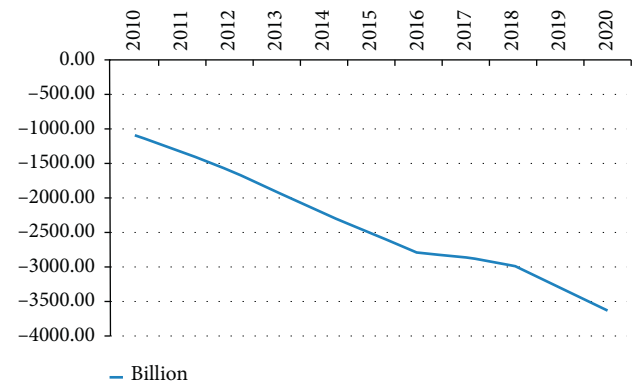


FIGURE 13: Environmental carrying capacity in southern Shaanxi region.

development can provide more funds for environmental governance and restoration, which can better protect the environment and continuously enhance the regional environment and economic carrying capacity. The high-quality economic development in the southern Shaanxi region and the beautiful mining environment will improve people’s ideological awareness of practicing green development, the utilization rate of mineral resources, and the life quality of mine workers and will promote the high-quality development of enterprises.

## 6. Conclusions

Based on theoretical knowledge and quantitative analysis, this paper takes the iron ore resources in the southern Shaanxi region as the research object. By establishing the iron ore resources-socioeconomic-ecological environment cycle system in the southern Shaanxi region and using the system dynamics model to construct subsystems of the development and utilization of mineral resources in the southern Shaanxi region, economy subsystems, and environmental subsystems, the present study analyzes the carrying capacity of mineral resources, economic, and environment of the region. The main conclusions are as follows:

- (1) Using historical test analysis, comparing the actual data of regional GDP and mineral resources reserves in the southern Shaanxi region from 2010 to 2016 with the simulation results of the model, it is verified that the model design is relatively reasonable.
- (2) The model sensitivity test shows that the change of the mineral investment ratio parameter has a significant impact on the regional GDP and the total demand for mineral resources, both of which are positively correlated. The increase in the mineral investment ratio will promote the continuous increase in regional GDP and demand for mineral resources.
- (3) The establishment of the system dynamics model and the simulation results show that iron ore resources, economy, and environment do not exist independently, but the three are interrelated and restrict each other in a complex system engineering, and the impact of any one of which cannot be neglected.
- (4) In the process of developing and utilizing iron ore resources, if the investment in iron ore exploration is not increased while relying on retained reserves, the carrying capacity of iron ore resources will become weaker over time. Therefore, while developing and consuming iron ore reserves, the exploration of iron ore needs to be strengthened, to continuously increase the reserves of iron ore resources and enhance the carrying capacity of regional iron ore resources.
- (5) The contribution and influence of iron ore development and utilization to regional economic development in the southern Shaanxi region decrease and tend to be stable, the regional economy's dependence on iron ore development and utilization decreases, and the regional economic carrying capacity is strong.
- (6) From 2010 to 2016, the environmental carrying pressure in the southern Shaanxi region was relatively high, and the environmental carrying capacity was relatively weak. In recent years, people's awareness of environmental protection is increasing, and new laws, regulations, policies, and measurements are implemented in environmental protection governance and restoration, while the investment in environmental governance and restoration has continued to increase. At the same time, supervision and management have been strengthened, the difference between the growth rate of environmental governance and restoration costs and the growth rate of regional GDP has become larger, and the regional environmental carrying capacity has been continuously enhanced.
- (7) According to the analysis, the southern Shaanxi region is mainly a resource-based area with a relatively unbalanced economic structure. The secondary industry is the main economic support industry in the southern Shaanxi region, which accounts for a high proportion of the regional economy.
- (8) In order to ensure that the development and utilization of iron ore resources, environmental protection, and economy develop in an associated manner in the southern Shaanxi region, according to this research, the following advice is proposed:
  - (1) Adjust the structure of the three major economic industries in the region, take advantage of natural resources and environment, and actively develop alternative industries, such as developing tourism and related service industries.
  - (2) Local governments should vigorously introduce high-tech enterprises, increase investment or introduction of technical talents, and explore new industrial forms, such as biotechnology, high-tech and other emerging cutting-edge industries, and so on, to realize the transformation of economic growth mode and economic development transformation upgrade.
  - (3) Practice the concept "lucid waters and lush mountains are invaluable assets," strengthen supervision and management, and develop iron ore resources scientifically and rationally on the premise of strengthening the ecological environment protection of mines. Develop the mining economy to protect the environment, improve the safety of mine workers and the quality of their lives, and promote high-quality social and economic development in the southern Shaanxi region.

Through the system simulation, we can better understand the development of iron ore resources in southern Shaanxi by increasing the investment ratio of mineral resources in southern Shaanxi by 10% and 20%, respectively, to simulate and compare the regional GDP and iron ore resource demand in southern Shaanxi area.

### Data Availability

The data used to support the findings of this study are collected from the Shaanxi Province Yearbook 2010–2017, Shaanxi Local Chronicle 2010–2017, and Internal data from Northwest Nonferrous Geology and Mining Group Co., Ltd.

### Conflicts of Interest

The authors declare that there are no conflicts of interest in this research.

### References

- [1] G. Zhao, S. Chen, and R. Zuo, "Identifying geochemical anomalies associated with Au–Cu mineralization using multifractal and artificial neural network models in the Ningqiang district, Shaanxi, China," *Journal of Geochemical Exploration*, vol. 164, no. 4, pp. 54–64, 2016.
- [2] G. Minhuan, "A study on exploitation and utilization of mineral resources in Shanxi Province," *China's Manganese Industry*, vol. 34, no. 2, pp. 76–78, 2016.
- [3] X. Zhang, B. Su, and J. Yang, "Jianhui Cong. Analysis of Shanxi Province's energy consumption and intensity using input-

- output framework (2002–2017),” *Energy*, vol. 250, no. 1, pp. 12–25, 2022.
- [4] X. Maoxi, H. Weizhong, and L. Yonghong, “An analysis of geological environment problems of mines in Shaanxi,” *Journal of Environmental Management*, vol. 36, no. 3, pp. 89–91, 2014.
- [5] M. M. Naderi, A. Mirchi, A. R. M. Bavani, E. Goharian, and K. Madani, “System dynamics simulation of regional water supply and demand using a food-energy-water nexus approach: application to Qazvin Plain, Iran,” *Journal of Environmental Management*, vol. 280, Article ID 111843, 2021.
- [6] G. Li, C. Kou, Y. Wang, and H. Yang, “System dynamics modelling for improving urban resilience in Beijing, China,” *Resources, Conservation and Recycling*, vol. 161, Article ID 104954, 2020.
- [7] M. Song, X. Cui, and S. Wang, “Simulation of land green supply chain based on system dynamics and policy optimization,” *International Journal of Production Economics*, vol. 217, no. 217, pp. 317–327, 2019.
- [8] Z. Feng and Li Peng, “The origin and development of the concept of carrying capacity: a discussion from the perspective of resources and environment,” *Journal of Natural Resources*, vol. 33, no. 9, pp. 1475–1489, 2018.
- [9] F. Niu, Z. Feng, and H. Liu, “Review and prospect of evaluation methods of resources and environment carrying capacity,” *Resources Science*, vol. 40, no. 4, pp. 655–663, 2018.
- [10] Y. Jing and Y. Chen, “Research review and thinking on resource carrying capacity,” *China’s population, resources and environment*, vol. 5, pp. 11–14, 2006.
- [11] Q. An and L. Wang, “System dynamics simulation of resource and environmental carrying capacity of mining cities,” *Resources and Industry*, vol. 18, no. 6, pp. 1–7, 2016.
- [12] W. Lu, J. Liu, and Na Wei, “Simulation analysis of industrial transformation of resource-exhausted mining cities based on system dynamics - taking Liaoyuan City, Jilin Province as an example,” *Geographical Sciences*, vol. 3, no. 5, pp. 577–583, 2012.
- [13] T. Lv, M. Zhang, and L. Qiang, “Scenario forecast of Indian energy and coal demand based on system dynamics,” *Resources Science*, vol. 27, no. 6, pp. 1199–1206, 2015.
- [14] L. Cai, *Application of System Dynamics in Sustainable Development Research*, pp. 4–10, China Environmental Science Press, Beijing, China, 2008.
- [15] G. Lu and H. Zhang, “Research progress of ecological carrying capacity at home and abroad,” *China Population, Resources and Environment*, vol. 17, no. 02, pp. 19–26, 2007.
- [16] H. Zeng, *Research on the Spatial and Temporal Differences of Resource and Environmental Carrying Capacity in Wuhan City circle*, China University of Geosciences, Wuhan, China, 2013.
- [17] S. Chen, “Carrying capacity: the transition from static to dynamic China’s Population,” *Resources and Environment*, vol. 19, no. 2, pp. 117–121, 2003.
- [18] D. Zheng, W. Xu, J. Jiang, and L. Lv, “Analysis on the evolution trend and coordinated development of China’s water resources carrying capacity and urbanization quality,” *Economic Geography*, vol. 02, no. 2, pp. 72–81, 2021.
- [19] S. Zhang, P. Shi, and Z. Wang, “Coupling analysis of urbanization and water resources environment system in inland river basin in arid areas: taking shiyang river basin as an example,” *Economic Geography*, vol. 32, no. 8, pp. 142–148, 2012.
- [20] Z. Yang and C. Wen, “Evaluation of China’s green development efficiency and regional differences,” *Economic Geography*, vol. 03, no. 3, pp. 10–18, 2017.

## Research Article

# Image Semantic Segmentation Method Based on Deep Learning in UAV Aerial Remote Sensing Image

Min Ling <sup>1</sup>, Qun Cheng <sup>1</sup>, Jun Peng <sup>2</sup>, Chenyi Zhao <sup>3</sup> and Ling Jiang<sup>2</sup>

<sup>1</sup>Shanghai Urban Construction Engineering School (Shanghai Gardening School), Shanghai 200232, China

<sup>2</sup>School of Geographic Information and Tourism, Chuzhou University, Chuzhou 239000, China

<sup>3</sup>Hongya Education and Technology (Shanghai) Co., Ltd., Shanghai 200241, China

Correspondence should be addressed to Jun Peng; [ipengjun2020@163.com](mailto:ipengjun2020@163.com)

Received 16 March 2022; Revised 8 April 2022; Accepted 12 April 2022; Published 26 April 2022

Academic Editor: Ramin Ranjbarzadeh

Copyright © 2022 Min Ling 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.

The existing semantic segmentation methods have some shortcomings in feature extraction of remote sensing images. Therefore, an image semantic segmentation method based on deep learning in UAV aerial remote sensing images is proposed. First, original remote sensing images obtained by S185 multirotor UAV are divided into smaller image blocks through sliding window and normalized to provide high-quality image set for subsequent operations. Then, the symmetric encoding-decoding network structure is improved. Bottleneck layer with  $1 \times 1$  convolution is introduced to build ISegNet network model, and pooling index and convolution are used to fuse semantic information and image features. The improved encoding-decoding network gradually strengthens the extraction of details and reduces the number of parameters. Finally, based on ISegNet network, five-classification problem is transformed into five binary classification problems for network training, so as to obtain high-precision image semantic segmentation results. The experimental analysis of the proposed method based on TensorFlow framework shows that the accuracy value reaches 0.901, and the F1 value is not less than 0.83. The overall segmentation effect is better than those of other comparison methods.

## 1. Introduction

With the rapid development of remote sensing technology in recent years, especially the rise of high-resolution remote sensing images, remote sensing technology has become a necessary method for timely regional Earth observation [1]. The emergence of various high-resolution remote sensing satellites has led to the increase of remote sensing image collection sources and the expansion of the scale of multimodal datasets [2]. In the face of massive, multimodal remote sensing image data, the traditional image processing and analysis methods do not perform well in the big data environment [3, 4]. Deep learning has the ability to extract main features from massive data and can achieve real-time data processing. Therefore, remote sensing image semantic

segmentation based on deep learning has gradually become the focus of research [5].

Semantic segmentation of remote sensing image is a transition link and key step to realize object-oriented extraction from data to information [6]. As a bridge to advanced tasks, semantic segmentation is widely used in the field of computer vision and remote sensing, such as automatic driving, attitude estimation, and remote sensing image interpretation. Previous image segmentation was usually based on the image pixel itself and the representation of image low-order visual information, such as pixel clustering segmentation and graph segmentation [7–9]. Although these methods are with low computational complexity because of lacking model training based on dataset, due to the lack of sufficient manual annotation

information, the segmentation results are not ideal in image segmentation for complex tasks [10]. The rise of deep learning makes image segmentation enter a new stage, such as image block segmentation using convolutional neural network [11].

The existing semantic segmentation methods have the problem of insufficient extraction depth in remote sensing image feature extraction, and the massive small targets in the image cause new difficulties in using deep learning to obtain robust feature representation [12]. Therefore, in order to effectively solve the above problems, an image semantic segmentation method based on deep learning is proposed to realize semantic segmentation and target recognition in UAV aerial remote sensing images.

## 2. Related Work

The basic process of semantic segmentation of remote sensing image is to preprocess the data of remote sensing image and then extract tensor features of image according to the preprocessed data. The input of the model is the tensor features. The model is initialized with a given training network model and the segmentation and prediction of remote sensing image targets are completed [13].

Early semantic segmentation is in the stage of traditional image segmentation, which requires participants to annotate the features manually, but the accuracy of annotation greatly affects the segmentation results [14]. The human and material resources consumed cannot meet the high-efficiency requirements of large-scale applications. With the emergence of full convolution neural network, semantic segmentation has officially entered the era of deep learning. The goal of image semantic segmentation is to mark each pixel of the image with the corresponding class. In order to understand the research status, existing problems, and development prospects of image semantic segmentation, [15] introduced the mainstream image semantic segmentation methods on the basis of extensive investigation. The segmentation results of common image semantic segmentation algorithms were summarized and compared. Based on the summary of common image semantic segmentation datasets and evaluation standards, the development trend of image semantic segmentation in the future was prospected. Reference [16] proposed Generative Adversarial Networks for image semantic segmentation, including edge adversarial network and semantic segmentation adversarial network, which effectively improved the segmentation accuracy in fog, but the segmentation efficiency needs to be improved. Reference [17] studied image segmentation in lane departure system and compared traditional processing methods with deep learning semantic segmentation methods, and the results showed that deep neural network image semantic segmentation had better robustness and efficiency, but it is still slightly insufficient in the processing performance of high-order task set. Reference [18] designed a fusion network for small target image segmentation, that is, extracting the feature information of RGB image and depth image to complement each other, but it has high requirements for image acquisition.

At the same time, in order to enhance the segmentation performance of the deep learning network, some improvements are made. For example, [19] proposed a superpixel enhanced depth neural model to solve the problems of distinguishing surface image features and classifying ground objects in the process of image segmentation. It adopted an end-to-end method combined with the depth convolution neural network to achieve better classification accuracy. However, for a large number of small targets in remote sensing images, the effect of segmentation and recognition needs to be improved. In order to solve the challenge of learning spatial context of deep convolution neural network in high-resolution image semantic segmentation, [20] proposed a new segmentation model, which deduced a symbolic distance map for each semantic class from the real label map to improve the segmentation accuracy. However, the training process is complex, and the processing efficiency of a large number of remote sensing images needs to be improved. Reference [21] proposed a new pixel-level feature extraction model with convolutional encoder and decoder for medical image recognition, which improved the global and average accuracy through dataset training, and the improvement of feature extraction enhanced the accuracy of semantic segmentation, but it cannot consider the processing efficiency. Reference [22] proposed a depth convolution neural network based on U-Net to realize end-to-end semantic segmentation. The model fusion strategy effectively improved the segmentation accuracy, but the segmentation effect is not good for partial occlusion or fog. Aiming at the problem that traditional segmentation methods find it difficult to deal with high-resolution remote sensing images containing complex ground objects, an image semantic segmentation method based on deep learning in unmanned aerial vehicle (UAV) aerial remote sensing images is proposed. Its contributions are summarized as follows:

- (1) Considering that the traditional SegNet method has low segmentation accuracy and long training time, the proposed method expands the convolution layer and adds the Bottleneck layer to obtain the Improved SegNet (ISegNet) network, so that it can express more complex features.
- (2) In order to avoid the problem that classification results of a single classifier are greatly affected by misclassification, the proposed method converts five-classification problem into five binary classification problems for network training based on ISegNet network, so as to improve the segmentation accuracy of the model.

## 3. Proposed Research Method

*3.1. Classification Model Establishment.* The main flow of the proposed semantic segmentation method is shown in Figure 1. The whole process is more concise and effective than the traditional image semantic segmentation based on regional features. There is no need to search the region containing the target image in the early stage, and there is no need to merge similar regions in the later stage.

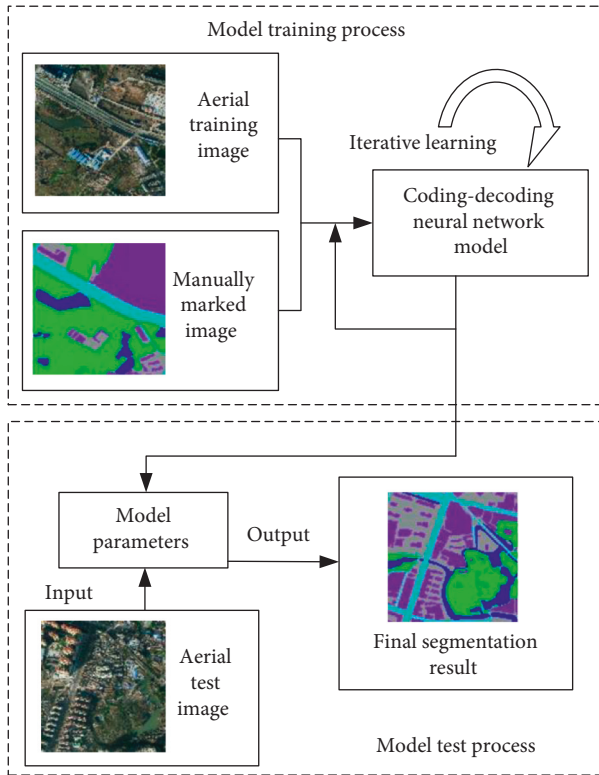


FIGURE 1: Semantic segmentation process based on deep learning.

The aerial image and the corresponding manually marked image are input into the encoding-decoding network, the optimal model parameters are obtained through multiple iterative learning, and the model and corresponding parameters are saved. In the inference stage, the final segmentation result can be obtained by directly inputting the aerial test image into the saved model. However, directly convoluting the image will make the image smaller, and the contribution of the information at the edge of the original image to the image content is small. Therefore, padding is used to solve this problem; that is, 0 is filled in the image edge to expand the image size in the process of convolution, so that the image size remains unchanged after multiple convolution operations [23].

**3.2. UAV Hyperspectral Image Acquisition.** In order to fully understand the differences of the research objects in hyperspectral images, a spectral information database was established, and the hyperspectral images of UAVs in two small areas in the field of Shanghai suburb were obtained. In one area, discontinuous UAV images of four sorties were obtained to fully cover the object types in the study area. In the other area, relatively continuous UAV images of five sorties were obtained to provide a data source for the classification and identification of typical object types. The S185 UAV system was used in the test, which mainly included Cubert S185 hyperspectral data acquisition system, six-rotor electric UAV system (maximum load is about 6 kg; flying time is 15 min–30 min), triaxial stabilized camera, and data processing system, as shown in Figure 2.

The acquisition of UAV hyperspectral data should be carried out in sunny days to avoid cloud shadow on the image and affecting the image quality, and the solar deflection angle should not be too large during acquisition to avoid too large shadow area in the image. The experimental data collection time is between 10:00 and 13:00 on December 18 and December 20, 2020. The weather is sunny and cloudy. Hyperspectral images of 9 UAVs in two research areas are obtained. In the experiment, it is ensured that the cloud amount and sunlight intensity in the air have little difference in the collection process of each sortie.

**3.3. Data Preprocessing.** High-resolution remote sensing images are usually large in size and cannot be processed by convolution [24]. For example, the average size of ISPRS images from the Vaihingen dataset is  $2493 \times 2063$  pixels, while most convolution operations support resolution of  $256 \times 256$ . In view of the memory limitation of the current graphics processing unit (GPU), the proposed method uses a sliding window to segment the original remote sensing image into smaller image blocks. If the convolution step is smaller than the image block size, in the case of overlapping continuous image blocks, multiple predictions are averaged to obtain the final classification of the overlapping pixels [25]. This operation can smooth the prediction of each image block boundary and eliminate possible discontinuities.

The goal of the proposed method is to apply the typical artificial neural network structure to earth observation data. Therefore, using the artificial neural network originally designed for RGB data, the processed image must comply with 3-channel format. The three channels in the ISPRS dataset will be processed into RGB images. The dataset contains the data of digital surface model (DSM) obtained from the aerial laser sensor. Normalized digital surface model (NDSM) will also be used, and then normalized difference vegetation index (NDVI) will be calculated from near-infrared and infrared channels. Finally, DSM, NDSM, and NDVI information will be used to build a corresponding composite image for each IRRG image.

**3.4. Network Structure Design.** Because the traditional SegNet method has low segmentation accuracy and long training time, an ISegNet network is proposed, and its structure is shown in Figure 3. The original SegNet convolution layers are expanded from 26 layers to 27 layers, and Bottleneck is added. Rectified linear unit (ReLU) and exponential linear unit (ELU) are introduced, respectively, into activation functions to increase the nonlinearity of the network, so that the network can express more complex characteristics.

ISegNet includes 5-layer encoding structure and 5-layer decoding structure. The 5-layer encoding structure is composed of 13 convolution layers with kernel size  $3 \times 3$ , 5 batch normalization (BN) layers, and 4 Maxpooling layers with kernel size  $2 \times 2$  and step size 2; the 5-layer decoding structure consists of 13 convolution layers with kernel size  $3 \times 3$ , 5 BN layers, and 4 upsampling layers, which is completely symmetrical with the encoding structure.



FIGURE 2: S185 multirotor UAV system.

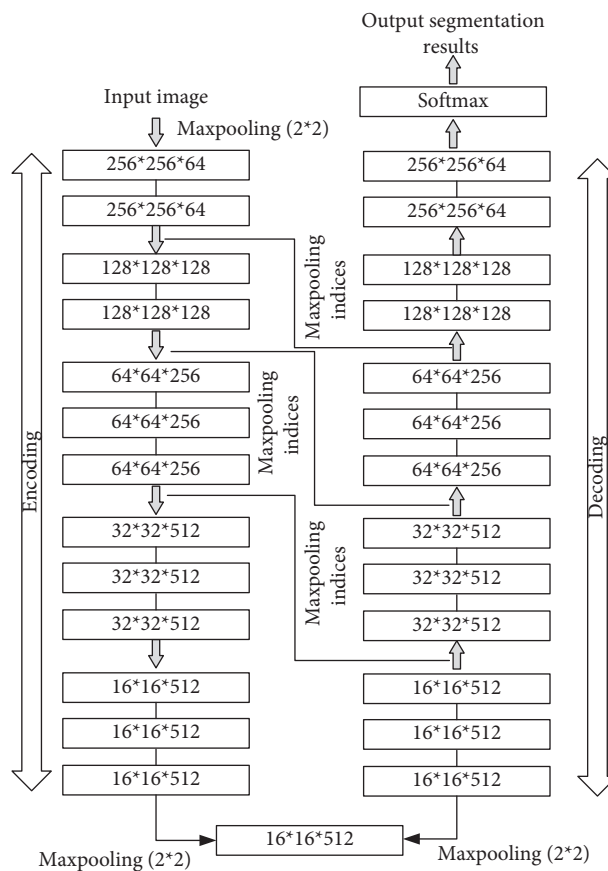


FIGURE 3: Structure of ISegNet remote sensing image semantic segmentation network.

Convolution layer with kernel size  $1 \times 1$  is set at the connection between the encoding structure and the decoding structure to further extract nonlinear features of the encoded image data, which deepens the network depth and reduces the amount of parameters.

3.4.1. *Encoder Structure.* The images normalized to  $256 \times 256$  pixels are used as the input of the network, and the image features are extracted by the encoder composed of convolution layer and pooling layer. ISegNet uses the convolution of same mode to ensure that the size of image

remains unchanged. Image features are transmitted to the decoder through the Maxpooling indices for nonlinear upsampling to obtain the lost image spatial semantic information in the coding process [26]. The BN layer is added after each Maxpooling layer, and the value range of the features after nonlinear function approaches the saturation region, so as to standardize the distribution of output features. BN mainly includes normalization and transformation reconstruction.

(1) *Normalization.* The normalization expression of relevant parameters of input sample data in the network is

$$\begin{aligned}\mu &= \frac{1}{k} \sum_{i=1}^k x_i, \\ \sigma^2 &= \frac{1}{k} \sum_{i=1}^k (x_i - \mu)^2, \\ \hat{x}_i &= \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}},\end{aligned}\quad (1)$$

where  $k$  is the input batch size;  $\mu$  is the average value of the input;  $\sigma^2$  is variance;  $\varepsilon$  is a constant set to maintain parameter stability;  $x_i$  is the  $i$ th input sample value;  $\hat{x}_i$  is the normalized value corresponding to the  $i$ th input sample value. After this step, the feature parameters learned by the network change, and the feature distribution needs to be reconstructed.

(2) *Feature Distribution Reconstruction.* The expression of feature distribution reconstruction is

$$z_i = \omega \hat{x}_i + b, \quad (2)$$

where  $\omega$  and  $b$  are the weight and bias, respectively. BN layer can effectively prevent overfitting in model training and speed up learning.

In addition, ISegNet uses ReLU and ELU as the activation function of the network, respectively. ReLU is a commonly used activation function, which changes all negative values to 0 and positive values retain the output, so that neurons are sparsely activated. ReLU has stronger feature mining ability. ReLU is used to fit the training data and to further prevent the gradient from disappearing. The ReLU function is

$$y = \begin{cases} 0, & x \leq 0, \\ x, & x > 0, \end{cases} \quad (3)$$

where  $x$  is neuron input and  $y$  is neuron output.

ELU is also a kind of correction activation function, which adds a nonzero output for negative input. Unlike ReLU, ELU activation function includes a negative exponential term. The function expression of ELU is

$$y = \begin{cases} a[\exp(x) - 1], & x < 0, \\ x, & x \geq 0, \end{cases} \quad (4)$$

where  $a$  is a constant value.

3.4.2. *Decoder Structure.* The decoder is composed of upsampling layer, convolution layer, and pooling layer. The upsampling layer can recover part of the lost information in Maxpooling layer and recover the pixel position information according to the pooling index. The convolution of same mode is also used in the decoder. The upsampling process is shown in Figure 4.

The decoder uses the index stored in each Maxpooling layer to upsample the corresponding feature map. In order to combine coarse and fine textures and prevent parameter redundancy, a Bottleneck layer is set at the connection of encoding-decoding structure. By introducing convolution layer with kernel size  $1 \times 1$ , the encoder output unit achieves feature dimension reduction. The quantity of encoder output parameters can be expressed as

$$\begin{aligned}D &= \sum_{i=1}^n R_{l_i}^2 \cdot R_{c_i} \cdot h, \\ D' &= h \cdot R + \sum_{i=1}^n R_{l_i}^2 \cdot R_{c_i} \cdot R,\end{aligned}\quad (5)$$

where  $D$  is the output parameter quantity of the encoder;  $D'$  is the output parameter quantity after adding  $1 \times 1$  convolution kernel;  $R_{l_i}$  is the kernel size of  $i$ th convolution layer;  $R_{c_i}$  is the kernel number of  $i$ th convolution layer (filter depth);  $l_i$  is the neuron at  $i$ th layer;  $c_i$  is the  $i$ th convolution kernel;  $h$  is the depth of input feature;  $R$  is the number of  $1 \times 1$  convolution kernels. In general,  $R < h$ . Therefore, after adding  $1 \times 1$  convolution kernel, the quantity of calculated parameters is reduced, and each pixel after encoding is linearly combined on different channels, which not only retains the original structure of the encoded image but also expands and widens the network.

Fully connected layer, previous layer of output layer, is replaced with a convolution layer to improve the efficiency of network forward propagation. The features output from the last convolution layer are input into the Softmax classifier, and the classifier finally outputs 5 probabilities, which, respectively, predict the probability that the sample belongs to each category; that is, the number of classification labels is 5. The predicted segmentation corresponds to the category with the maximum probability at each pixel.

3.5. *Classifier Design.* For the semantic segmentation of remote sensing images, the ensemble learning method is further used to improve the segmentation accuracy. The classification results obtained by Softmax classifier are combined through some strategies to achieve better performance than a single classifier. This model fusion method is essentially a relearning process. Even if one classifier makes an erroneous prediction, other classifiers can correct the error. Generally speaking, this ensemble learning method can improve the segmentation effect to a certain extent, but the process of multimodel learning and relearning will increase the cost of calculation [27]. For the combination strategy of ensemble learning, the proposed method adopts the Plurality Voting method. Assuming that the prediction category is  $\{g_1, g_2, \dots, g_m\}$ , for any



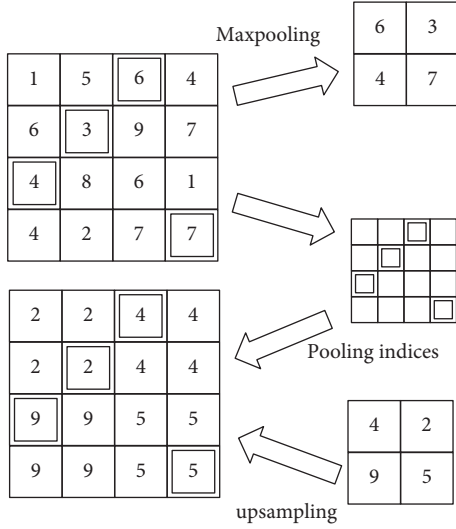


FIGURE 4: Maxpooling index and upsampling.

prediction sample  $x$ , the prediction results of  $T$  classification models are  $(y_1(x), y_2(x), \dots, y_T(x))$ , respectively. Select the category  $g_i$  with the largest number of prediction results of  $T$  classification models for sample  $x$  as the final classification category. If more than one category gets the highest vote at the same time, select one at random as the final category. Using this ensemble idea, by finding a compromise between multiple classifiers, the worst classifier can be avoided, and some pixels with obvious classification errors can be effectively improved, so as to improve the prediction ability of the model. The ensemble learning process based on multiple classifiers is shown in Figure 5.

First, SegNet is used to complete the semantic segmentation of aerial images, and the prediction results are obtained. Second, the network is improved on the basis of SegNet. The feature maps of different scales in the encoding module are copied to the corresponding upsampling part by skip connection. The improved network model (ISegNet) is trained again, and the prediction results based on this model are obtained. Then, based on ISegNet, five-classification problem is transformed into five binary classification problems for training, and the prediction results of each category are combined to obtain the prediction result. Finally, the segmentation results obtained from the above three different models are combined through ensemble learning, and the ensemble prediction results are obtained by using the Plurality Voting method.

#### 4. Experiments and Analysis

In the experiment, the training equipment of deep learning network is 4-core 8-thread Intel i7-7700K CPU, 32 GB memory, NVIDIA GTX1080 video card, and 8G video memory. The software environment is Ubuntu 16.04.01 operating system. The development platform is Anaconda 4.3.1. The built-in Python version is 3.6.1. The deep learning software framework is TensorFlow 1.2. At the same time,

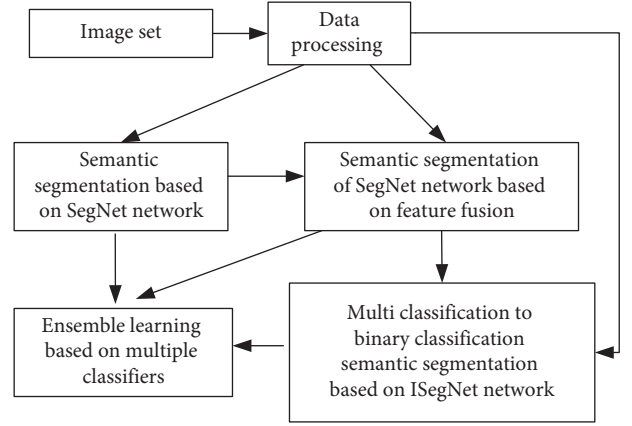


FIGURE 5: Ensemble learning process based on multiple classifiers.

the experimental dataset is divided according to 8:1, and the learning rate is set to 0.01 and the number of iterations is 60.

**4.1. Evaluation Index.** Kappa coefficient has an important application in the accuracy evaluation of remote sensing classification images. Value range of Kappa coefficient is  $(-1, 1)$ . A value greater than 0.8 means good classification, and a value of 0 or lower means poor classification. Kappa coefficient is calculated as follows:

$$\varphi = \frac{p_0 - p_e}{1 - p_e},$$

$$p_0 = \frac{\sum_{i=1}^r y_{ii}}{N}, \quad (6)$$

$$p_e = \frac{\sum_{i=1}^r (y_{i+} \cdot y_{+i})}{N^2},$$

where  $p_0$  is the observation precision ratio, which reflects the proportion of correctly segmented cells;  $p_e$  is the contingency consistency ratio, which indicates the proportion of wrong segmentation caused by accidental factors.  $N$  is the total number of samples;  $y_i$  is the segmented sample.

In addition, the F1 value is used to evaluate the experimental results, and the calculation is as follows:

$$F_{1t} = 2 \times \frac{\text{precision}_t \times \text{recall}_t}{\text{precision}_t + \text{recall}_t},$$

$$\text{recall}_t = \frac{p_{t0}}{Q_t}, \quad (7)$$

$$\text{precision}_t = \frac{p_{t+}}{P_t},$$

where  $p_{t+}$  and  $p_{t0}$  are the numbers of pixels correctly recognized and recalled in category  $t$ , respectively.  $Q_t$  is the number of pixels belonging to category  $t$ ;  $P_t$  is the number of pixels in category  $t$  identified by the model. In addition, the boundary of the object in the test label image is eroded by a

circular area with a radius of 3 pixels. These eroded areas are ignored in the evaluation process to reduce the impact of uncertain boundary definition. Therefore, the performance of the test set is slightly better than that of the validation set.

**4.2. Network Loss and Accuracy Curve.** When training the proposed image semantic segmentation network model, the experimental loss and accuracy curve are shown in Figure 6. Generally speaking, the lower the loss and the higher the accuracy, the better the segmentation performance of the network.

As can be seen from Figure 6, when the number of iterations reaches 35, the loss value and accuracy value of the proposed network begin to stabilize. When the iteration is completed, the two values are 0.08 and 0.901, respectively. Combined with the training oscillation degree, network segmentation effect, convergence speed, and other factors, the comprehensive performance of the proposed network is ideal. The deep image features are extracted through ISegNet to reduce the loss of detail information. At the same time, the ensemble idea is introduced to aggregate the classification results to further ensure the classification accuracy. The ensemble idea makes the network model more stable, reduces the training oscillation frequency, and makes the remote sensing image segmentation more accurate.

**4.3. Remote Sensing Image Semantic Segmentation Results.** Before the detailed quantitative evaluation of remote sensing image semantic segmentation results, it is necessary to qualitatively show the results of remote sensing image semantic segmentation by the proposed method and the methods in [17, 22] to demonstrate the performance of the proposed method. The comparison results of the three methods are shown in Figure 7.

As can be seen from Figure 7, compared with other comparison methods, the segmentation result of the proposed method is closest to the artificial mark, and the overall visual perception is the best. Because the ISegNet includes ReLU and ELU activation functions, the nonlinearity of the network can be increased, more complex features can be expressed, and the fused classifier design further reduces the segmentation error, especially the image details. Reference [17] used deep neural network for image semantic segmentation, but the segmentation effect of complex images was not ideal. For example, it is difficult to distinguish vegetation and water, and the results contain misclassification. Reference [22] realized pixel level end-to-end semantic segmentation based on the improved U-Net deep convolution neural network. The segmentation result is clear and the loss of targets is less. However, compared with the proposed method, it lacks the reclassification of classifier. Therefore, for the segmentation of small targets, its edge effect needs to be improved.

**4.4. Semantic Segmentation Quality Evaluation of Different Categories.** In order to better evaluate the performance of the proposed method, three methods are quantitatively analyzed, and the results are shown in Figure 8.

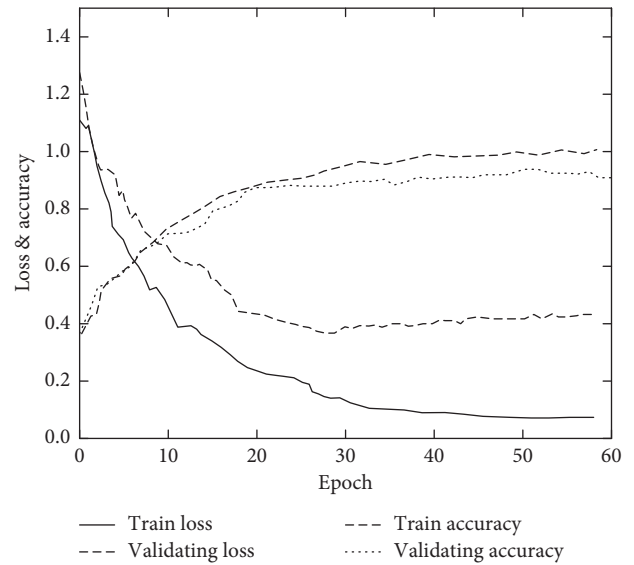


FIGURE 6: Network loss and accuracy curve.

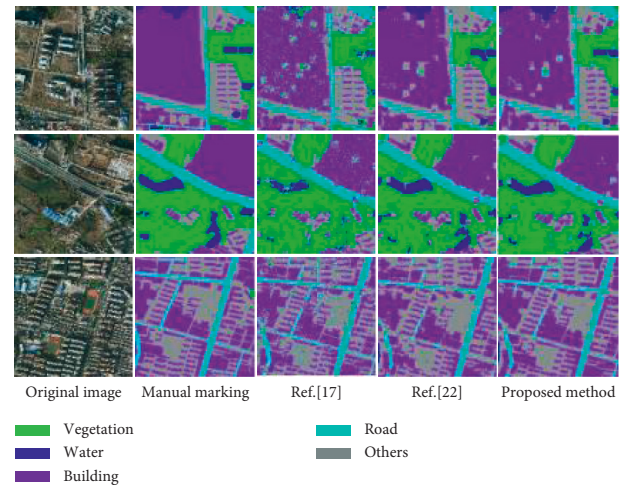


FIGURE 7: Semantic segmentation results of remote sensing images.

As can be seen from Figure 8, the proposed method is more ideal for the segmentation results of buildings and water bodies, while it is weak for the image segmentation of roads and other categories. Because the roads are narrow and long and easy to be blocked by trees, there may be missing points, resulting in low segmentation result values. However, the proposed method performs relearning classification based on the segmentation results obtained by ISegNet. Compared with other comparison methods, its segmentation result is more ideal. Taking the F1 value as an example, the segmentation results of the proposed method for vegetation, buildings, and water bodies exceed 0.85 and are generally not less than 0.83. Reference [17] used deep neural network for image segmentation, but the segmentation model lacked the ability to extract and learn complex features, so the segmentation result of road and other types of images was lower than 0.73. Reference [22] realized image segmentation based on improved U-Net depth convolution neural network. The segmentation results of some image

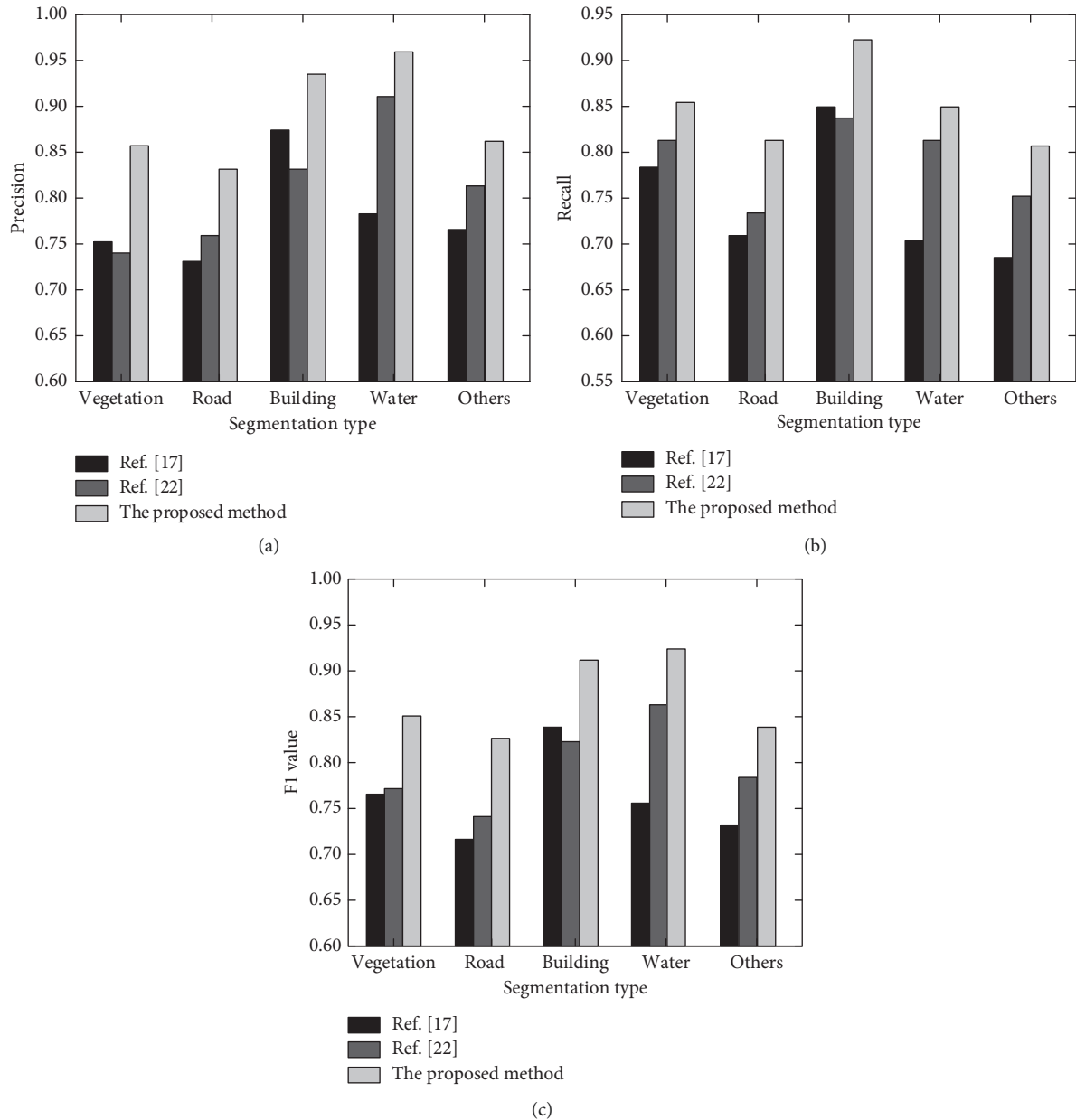


FIGURE 8: Evaluation index values of different semantic segmentation methods of remote sensing images. (a) Precision. (b) Recall. (c) F1 value.

categories such as water body were similar to those of the proposed method, but the segmentation result of some small buildings was not ideal. Taking precision value as an example, it is lower than 0.85. Overall, the proposed method has the best segmentation result for five categories of images.

## 5. Conclusions

Image segmentation is an important basic part of remote sensing interpretation. UAV remote sensing image contains complex ground object information, and the application of traditional segmentation methods is greatly limited. Therefore, an image semantic segmentation method based on deep learning in UAV aerial remote sensing image is

proposed. The preprocessed image is input into ISegNet for learning and analysis, and five-classification problem is transformed into 5 binary classification problems for training in the classifier, so as to output high-precision image semantic segmentation results. The experimental results show the following:

- (1) ISegNet uses pooling index and  $1 \times 1$  Bottleneck layer to further extract image details, so the segmented image is closest to the artificial standard, and the accuracy value reaches 0.901.
- (2) The proposed method improves the classifier and relearns the classification problem to ensure the segmentation result. Its F1 value is not less than 0.83,

and the overall segmentation effect is better than those of other comparison methods.

Although the proposed method can obtain high segmentation precision, the model used is more complex and has more parameters. When the extracted features are more abstract, the model complexity is higher and the training time is longer. In practical application, efficiency needs to be further considered, such as parallel processing. Further optimization will be carried out from the model itself or distributed computing using deep learning framework.

## Data Availability

The data included in this paper are available without any restriction.

## Conflicts of Interest

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

## Acknowledgments

This work was supported by Natural Science Research Projects in Colleges and Universities of Anhui Province under Contract no. KJ2020A0719; Chuzhou Science and Technology Planning Program under Contract no. 2021ZD010; key project of research and development in Chuzhou Science And Technology Program under Contract no. 2020ZG016; Open Fund of Hunan Provincial Key Laboratory of Geo-Information Engineering in Surveying, Mapping and Remote Sensing, Hunan University of Science and Technology under Contract no. E22136; and Innovation program for Returned Overseas Chinese Scholars of Anhui Province under Contract no. 2021LCX014.

## References

- [1] Q. Zhou, Y. Wang, J. Liu, X. Jin, and L. J. Latecki, "An open-source project for real-time image semantic segmentation," *Science China (Information Sciences)*, vol. 62, no. 12, pp. 246-247, 2019.
- [2] L. Jing, Y. Chen, and Y. Tian, "Coarse-to-Fine semantic segmentation from image-level labels," *IEEE Transactions on Image Processing*, vol. 29, no. 2, pp. 225-236, 2019.
- [3] B. Yan, X. Niu, B. Bare, and W. Tan, "Semantic segmentation guided pixel fusion for image retargeting," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 676-687, 2020.
- [4] Y. Li, T. Shi, Y. Zhang, W. Chen, Z. Wang, and H. Li, "Learning deep semantic segmentation network under multiple weakly-supervised constraints for cross-domain remote sensing image semantic segmentation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 175, no. 3, pp. 20-33, 2021.
- [5] S. Y. Wang, S. C. Wang, and D. Wang, "A road extraction method for remote sensing image based on encoder-decoder network," *Journal of Surveying and Mapping: English Edition*, vol. 2020, no. 2, pp. 16-25, 2020.
- [6] H. Liu, H. Du, D. Zeng, and Q. Tian, "Cloud detection using super pixel classification and semantic segmentation," *Journal of Computer Science and Technology*, vol. 34, no. 3, pp. 622-633, 2019.
- [7] X. Zhang, Z. Xiao, D. Li, M. Fan, and L. Zhao, "Semantic segmentation of remote sensing images using multiscale decoding network," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 9, pp. 1492-1496, 2019.
- [8] K. Zhang, Y. Han, J. Chen, Z. Zhang, and S. Wang, "Semantic segmentation for remote sensing based on RGB images and lidar data using model-agnostic meta-learning and partial swarm optimization," *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 397-402, 2020.
- [9] C. Peng, Y. Li, L. Jiao, Y. Chen, and R. Shang, "Densely based multi-scale and multi-modal fully convolutional networks for high-resolution remote-sensing image semantic segmentation," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 8, pp. 2612-2626, 2019.
- [10] X. Sun, B. Wang, Z. Wang, H. Li, H. Li, and K. Fu, "Research progress on few-shot learning for remote sensing image interpretation," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, no. 7, pp. 2387-2402, 2021.
- [11] Y. Li, B. Peng, L. He, K. Fan, and L. Tong, "Road segmentation of unmanned aerial vehicle remote sensing images using adversarial network with multiscale context aggregation," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2279-2287, 2019.
- [12] X. Li, F. Xu, L. Xin et al., "Dual attention deep fusion semantic segmentation networks of large-scale satellite remote-sensing images," *International Journal of Remote Sensing*, vol. 42, no. 9, pp. 3583-3610, 2021.
- [13] L. Zabawa, A. Kicherer, L. Klingbeil, R. Töpfer, H. Kuhlmann, and R. Roscher, "Counting of grapevine berries in images via semantic segmentation using convolutional neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 164, no. 5, pp. 73-83, 2020.
- [14] M. Wurm, T. Stark, X. X. Zhu, M. Weigand, and H. Taubenböck, "Semantic Segmentation of slums in satellite images using transfer learning on fully convolutional neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, no. 4, pp. 59-69, 2019.
- [15] W. Song, N. Zheng, R. Zheng, X.-B. Zhao, and A. Wang, "Digital image semantic segmentation algorithms: a survey," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 10, no. 1, pp. 196-211, 2019.
- [16] K. Liu, Z. Ye, H. Guo, D. Cao, L. Chen, and F.-Y. Wang, "FISS gan: a generative adversarial network for foggy image semantic segmentation," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 8, pp. 1428-1439, 2021.
- [17] W. Chen, W. Wang, K. Wang, Z. Li, H. Li, and S. Liu, "Lane departure warning systems and lane line detection methods based on image processing and semantic segmentation: a review," *Journal of Traffic and Transportation Engineering*, vol. 42, no. 6, pp. 25-51, 2020.
- [18] W. Liu, "Real-time obstacle detection based on image semantic segmentation and fusion network," *Traitement du Signal*, vol. 38, no. 2, pp. 443-449, 2021.
- [19] L. Mi and Z. Chen, "Superpixel-enhanced deep neural forest for remote sensing image semantic segmentation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, no. 8, pp. 140-152, 2020.
- [20] A. Dc, B. Sn, and D. Jhc, "Aerial image semantic segmentation using DCNN predicted distance maps," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 161, no. 2, pp. 309-322, 2020.
- [21] M. Shahzad, A. I. Umar, M. A. Khan, S. H. Shirazi, Z. Khan, and W. Yousaf, "Robust method for semantic segmentation of whole-slide blood cell microscopic image," *Computational*

- and Mathematical Methods in Medicine*, vol. 2020, no. 3, Article ID 4015323, 13 pages, 2020.
- [22] M. Pai, V. Mehrotra, U. Verma, and R. M. Pai, "Improved semantic segmentation of water bodies and land in SAR images using generative adversarial networks," *International Journal of Semantic Computing*, vol. 14, no. 1, pp. 55–69, 2020.
- [23] F. Mohammadimanesh, B. Salehi, M. Mandianpari, E. Gill, and M. Molinierd, "A new fully convolutional neural network for semantic segmentation of polarimetric SAR imagery in complex land cover ecosystem," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 151, no. 5, pp. 223–236, 2019.
- [24] E. Collier, S. Mukhopadhyay, K. Duffy et al., "Semantic segmentation of high resolution satellite imagery using generative adversarial networks with progressive growing," *Remote Sensing Letters*, vol. 12, no. 5, pp. 439–448, 2021.
- [25] A. Kumthekar and G. R. Reddy, "An integrated deep learning framework of U-Net and inception module for cloud detection of remote sensing images," *Arabian Journal of Geosciences*, vol. 14, no. 18, pp. 1–13, 2021.
- [26] H. Shao, Y. Li, Y. Ding, Q. Zhuang, and Y. Chen, "Land use classification using high-resolution remote sensing images based on structural topic model," *IEEE Access*, vol. 8, no. 6, Article ID 215943, 2020.
- [27] I. Kotaridis and M. Lazaridou, "Remote sensing image segmentation advances: a meta-analysis," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, no. 3, pp. 309–322, 2021.

## Research Article

# The Offset Azimuth Prediction of Light Buoy Based on Markov Chain Optimization Multiplicative Seasonal Model

Zhizheng Wu <sup>1</sup>, Lüzhèn Ren <sup>1</sup>, Shibo Zhou,<sup>1</sup> Yuqi Zhang,<sup>2</sup> Wenpeng Xu,<sup>1</sup>  
and Heyang Zhang<sup>1</sup>

<sup>1</sup>Navigation College, Jimei University, Xiamen 361021, China

<sup>2</sup>Yonsei University, Wonju 03722, Republic of Korea

Correspondence should be addressed to Lüzhèn Ren; lzren@jmu.edu.cn

Received 14 February 2022; Revised 21 March 2022; Accepted 24 March 2022; Published 14 April 2022

Academic Editor: Ramin Ranjbarzadeh

Copyright © 2022 Zhizheng Wu 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.

Aimed at the problem of the large error caused by uncertain factors in the fitting process of the traditional multiplicative seasonal model, the advantages of the Markov chain in this study are applied to the multiplicative seasonal model to optimize the prediction results. Based on the residual value between the theoretical and actual values, the values of different intervals are divided into states. The transition probability matrix is established through different probabilities; then, the weighted sum of different prediction probabilities is carried out to select the optimal prediction state. The real number of Meizhou Bay portlight buoys is used to verify the prediction effect of the model, and MAE, MAPE, RMSE, RRMSE, SSE,  $R^2$  are used to calculate the error between the predicted value and the actual value. The results show that compared to the traditional multiplicative seasonal model and other prediction models, the prediction MAE of the MC-SARIMA model is decreased by 2.19003794, the MAPE is decreased by 0.66%, the RMSE is decreased by 2.092671823, the RRMSE is decreased by 0.006221352, the SSE is decreased by 404.0231931, and the  $R^2$  is increased by 0.224686247. It shows that the multiplicative seasonal model optimized by the Markov chain can predict the azimuth data of the light buoy more effectively than the traditional multiplicative seasonal model and other prediction models.

## 1. Introduction

As an important time series analysis method, the multiplicative seasonal model (SARIMA) [1] can establish a specific mathematical model according to the correlation between the data, integrate the seasonal, trend, random interference, and other characteristics of the time series data to predict its future data, and then compare the actual data with the real data, and the prediction results are better. Since the study of the offset azimuth of the light buoy is still in the preliminary stage, this study will review the literature from the perspectives of the SARIMA optimization model and time series prediction model.

The prediction model based on optimization of the multiplicative seasonal model is widely used in the fields of medicine, economy, meteorology, hydrology, and transportation throughout the world [2–6]. For example, Parviz [7] used the time series decomposition hybrid model based

on SARIMA and used cluster analysis to analyze the time series. The results show that the hybrid model is a valuable decision-making process tool, which effectively improves the level of precipitation forecast. Peirano [3] et al. combined LSTM with SARIMA to predict the inflation rate of five emerging economies in Latin America. The results show that the LSTM-SARIMA model has high precision in predicting the expansion of associates. To effectively predict the incidence of the cholera virus, Daisy et al. [8] proposed an SVM-SARIMA, which combines the two main factors influencing rainfall and maximum temperature, to evaluate its relationship with the incidence of the cholera virus. The results show that the model can effectively predict the incidence of the cholera virus and make better preparations for future public health management. Based on the demographic and distribution characteristics of AHC, Liu [9] et al. constructed the ETS-SARIMA model to predict the trend of incidence of AHC in mainland China. The results showed that AHC

mainly occurred in farmers and children under 9 years old in southern and eastern China. The prediction results were good and the error was low. Wang and Zhou [10] proposed a new runoff prediction method that combines FPCA and time series analysis. The results show that, compared to traditional one-dimensional FPCA and SARIMA, the combination of FPCA and time series analysis is more suitable for runoff prediction. Berke [11] et al. used the ANN-SARIMA model to predict the monthly incidence of cryptosporidiosis and used MAPE and RMSE to check the error. The results show that the ANN-SARIMA model can effectively predict the public health time series of the monitoring system. To reduce the dependence on fossil fuels and optimize the integration of renewable resources, Blazquez-Garcia [1] et al. proposed a SARIMA model to predict the energy consumption of green elevators integrated with photovoltaic and batteries in different short periods and used a genetic algorithm to optimize the combination of the selected parameters of SARIMA. The results show that the model has good effect in short-term prediction. Qian [6] et al. proposed a SARIMA-GARCH model based on the periodicity and trend of the time series data of the monthly passenger traffic. This model can effectively improve the characterization accuracy and eliminate the influence of heteroscedasticity. The ARCH test is used. The results show that, compared to the SARIMA model, the short-term prediction performance of the SARIMA-GARCH model is better.

The improved methods above improve the prediction accuracy of SARIMA to a certain extent and take into account the different special circumstances of time series, but there are also some shortcomings. For example, ANN, as a widely used algorithm, can fully approximate any linear relationship with complex structures and adopt the parallel distributed processing method. However, the human and material resources' loss of the algorithm is too large, which requires a lot of time to control the details of the algorithm. The famous "black box" problem of the algorithm is also a major problem that needs to be solved by the algorithm. The SVM model has the advantages of nonlinearity, high precision, and good generalizability, but it is difficult for the algorithm to train time series with large amounts of data. Therefore, the operation of the algorithm requires a lot of machine memory and operation time, and SVM has a certain difficulty in solving the classification problem. The ETS algorithm is flexible in use, has wide application, and can obtain the prediction results required by the model based on fewer training data. However, due to the large proportion of the short term given by the ETS algorithm and the small proportion of the long term given by the ETS algorithm, it is unable to predict the corresponding time series for a long time. Moreover, the algorithm has certain randomness in selecting the smoothing index, which cannot withstand scrutiny.

Research on the prediction model is based on time series [12]. A PCA prediction method for sports event model evaluation is proposed to solve the problems of poor average fitness and low-risk prediction accuracy of traditional sports event model evaluation prediction methods. The results show that the

method has good average fitness. Tian [13] proposed a new prediction method for short-term wind speed based on local mean decomposition (LMD) and combined kernel function least squares support vector machine (LSSVM), and the results show that the proposed prediction method has higher prediction accuracy and can reflect the laws of wind speed correctly. Li et al. [14] constructed a WebGIS scheme of forest information system and open-source network geographic information system oriented to spatial information field and supported OGC and used the grey model to predict the development trend. The results showed that the method realized the double growth of forest area and forest volume, and the forest coverage rate reached a new level. Liu et al. [15] used data from TCGA to create multigene features and evaluated the predictive significance of each lncRNA related to cell apoptosis for survival. Tian et al. [16] proposed a prediction approach for short-term wind speed using ensemble empirical mode decomposition-permutation entropy and regularized extreme learning machine, and the results show that the prediction approach in this study has higher reliability under the same confidence level. Zeng et al. [17] proposed a time series prediction method based on pattern analysis and used the probability relaxation method to classify the probability vectors of the basic pattern. Yuan et al. [18] proposed a kernel-HFCM model based on kernel mapping and HFFCM to predict time series inspired by the kernel method and SVR. Sebastian et al. [19] proposed a fractal interpolation method, which can generate finer-grained time series from insufficient data sets, and applied the grey theory model GM (1, 1) to the prediction of the price of Panama-type two-wheeled ships, to deal with bulk carriers with different periods and different sample sizes. Her et al. [20] applied the grey theory model GM (1, 1) to the prediction of the price of Panama-type two-wheeled ships, to deal with bulk carriers with different periods and different sample sizes. Mei et al. [21] used the BP-ANN model to predict COD removal efficiency and TEC and used particle swarm optimization to optimize BP-ANN, thereby improving the weights and thresholds of prediction accuracy optimization. Duan et al. [22] used the Bayesian model to predict the time and space of suspicious criminals to predict the driving traces of their social activities.

The Markov chain (MC) [23] refers to the stochastic process that contains Markov property in mathematical statistics and probability theory and exists in discrete state space and index set. The Markov chain applied to the continuous exponential set can also be called a Markov process. As an important method to study the state transition law of things, the Markov process can determine the best state of the development of things according to the initial probability and different state distribution of the research object and the transition probability matrix, and make the corresponding prediction.

Therefore, based on the analysis of the latest research on the optimization of the SARIMA model, a multiplicative seasonal model based on Markov chain optimization is proposed in this study. Based on the prediction of the multiplicative seasonal model, the prediction results of the model are optimized to further improve the prediction accuracy of the model.

The angle between the drift position of the sediment marine light buoy and the position of the sediment in Meizhou Bay Port is used as data to verify the model. The results show that the prediction error of the multiplicative seasonal model optimized by the Markov chain is lower than that of the traditional multiplicative seasonal model, which can be used on a scientific basis.

## 2. Multiplicative Seasonal Model

The multiplicative seasonal model refers to the multiplicative of the differential autoregressive moving average model and the random seasonal model. It is an important method for processing time series with seasonality and trend. The general form is  $SARIMA(p, d, q) \times (P, D, Q)_S$ , among them, the parameters  $p, d, q, P, D, Q$ , and  $S$  represent the non-seasonal regression order, moving average order, nonseasonal moving average order, seasonal autoregressive order, seasonal difference order, and seasonal average moving order, and the general form is shown as follows:

$$\nabla^d \nabla_S^D x_t = \frac{[V(\beta^S) \Theta(\beta) \varepsilon_t]}{[\phi(\beta) U(\beta^S)]}, \quad (1)$$

where  $\nabla_S^D = (1 - \beta^S)^D$ .

**2.1. ARIMA.** The autoregressive integrated moving average model (ARIMA) [24] is a time series prediction method proposed by Jenkins and Box. It is mainly used to study time series with periodicity, trend, and seasonality. ARIMA  $(p, d, q)$  is a combination of the autoregressive model  $AR(p)$  and the moving average model  $MA(q)$ ; its general form is shown in the following formula:

$$\begin{cases} AR(p) = \phi(\beta)x_t = (1 - \varphi_1\beta^1 - \varphi_2\beta^2 - \dots - \varphi_p\beta^p)x_t, \\ MA(q) = \Theta(\beta)\varepsilon_t = (1 - \theta_1\beta^1 - \theta_2\beta^2 - \dots - \theta_q\beta^q)\varepsilon_t, \end{cases} \quad (2)$$

where  $\beta$  is a delay operator,  $\varepsilon_t$  is a zero-mean white noise sequence,  $x_t$  is the time series,  $\phi(\beta)$  is an autoregressive coefficient polynomial,  $\Theta(\beta)$  is a moving average coefficient polynomial, and  $\varphi_1, \varphi_2 \dots \varphi_p$  and  $\theta_1, \theta_2 \dots \theta_p$  are expressed as its corresponding coefficients, respectively.

After integrating  $AR(p)$  and  $MA(q)$ , the  $D$ -order difference of its seasonal part with the ARIMA  $(p, d, q)$  model is obtained, and its general form is shown in the following formula:

$$\phi(\beta)x_t \nabla^d = \Theta(\beta)\varepsilon_t, \quad (3)$$

where  $\nabla$  is difference operator,  $\nabla^d$  is trend difference, and  $\nabla^d = (1 - \beta)^d$ .

**2.2. Random Seasonal Model.** The random seasonal model [25] is a time series with only periodicity and seasonality obtained by the integration of seasonal autoregressive model and seasonal moving average model through seasonal periodic difference, and the general forms of  $AR(P)$  and  $MA(Q)$  are shown in the following equation:

$$\begin{cases} AR(P) = U(\beta^S)x_t = (1 - u_1\beta^S - u_2\beta^{2S} - \dots - u_p\beta^{pS})x_t, \\ MA(Q) = V(\beta^S)\varepsilon_t = (1 - v_1\beta^S - v_2\beta^{2S} - \dots - v_q\beta^{qS})\varepsilon_t, \end{cases} \quad (4)$$

where  $U(\beta^S)$  is a seasonal autoregressive coefficient polynomial,  $V(\beta^S)$  is a seasonal moving average coefficient polynomial, and  $u_1, u_2 \dots u_p$  and  $v_1, v_2 \dots v_q$  are corresponding coefficients, respectively.

After integrating  $AR(P)$  and  $MA(Q)$ , the  $S$  and  $D$ -order difference processing are performed for seasonal and periodic parts, respectively, and the general expression of the obtained ARIMA  $(P, D, Q)$  model is shown in the following equation:

$$\nabla_S \nabla^D U(\beta^S)x_t = V(\beta^S)\varepsilon_t, \quad (5)$$

where  $\nabla^D$  is a periodic difference and  $\nabla_S$  is a seasonal difference.

## 3. Markov Chain

The Markov process refers to the division of a system into several small systems with different states, and each state can be transferred to another corresponding state according to its inherent transition probability. It is concluded that the Markov chain represents a process between one state and another, and that the probability of this process depends only on the corresponding preceding and following states [26]; its basic definition is as follows: let the random variable in discrete space  $I = (X_n, n \geq 0)$ , if any random variable  $n$  and any state  $i_0, i_1, i_2, i_3, \dots, i_{n-1}, i, j$  all exist  $P(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_{n+1} = j | X_n = i)$ , then the random variable  $I$  is called Markov chain.

**3.1. Transition Probability Matrix.** Under the condition that  $I = (X_n, n \geq 0)$  is Markov chain, then  $P_{ij}(n) = P(X_{n+1} = j | X_n = i)$  represents the one-step transition probability matrix of random variable  $n$ ; if the one-step transition probability matrix  $P_{ij}(n)$  is independent of  $n$ , then the Markov chain is called a homogeneous Markov chain, and the specific expression form is as follows:



$$P_n = P_{ij}(n) = \begin{pmatrix} P_{00}^n & P_{01}^n & \cdots & P_{0m}^n \\ P_{10}^n & P_{11}^n & \cdots & P_{1m}^n \\ \vdots & \vdots & & \vdots \\ P_{m0}^n & P_{m1}^n & \cdots & P_{mm}^n \end{pmatrix} P_1 = P_{ij}(1) = \begin{pmatrix} P_{00}^1 & P_{01}^1 & \cdots & P_{0m}^1 \\ P_{10}^1 & P_{11}^1 & \cdots & P_{1m}^1 \\ \vdots & \vdots & & \vdots \\ P_{m0}^1 & P_{m1}^1 & \cdots & P_{mm}^1 \end{pmatrix}. \quad (6)$$

**3.2. Autocorrelation Coefficient.** The correlation coefficient [27] is the degree of interaction between the time series of two different things; autocorrelation coefficient is the degree of correlation between different time series of the same thing. Therefore, according to the initial time-series data, the optimal multiplicative seasonal model is established. After the theoretical value is obtained, the actual value is subtracted from the theoretical value and the residual sequence is obtained. According to the residual value, the autocorrelation coefficient  $r_k$  between the theoretical value and the actual value is calculated, and its general form is shown in the following formula:

$$r_k = \frac{\sum_{i=1}^{n-k} (x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad (7)$$

where  $\bar{x}$  is residual mean,  $x_i$  is the  $i$ -th data in the residual sequence, and  $x_{i+k}$  is the  $i+k$ -th data in the residual sequence.

**3.3. Markov Chain Weight and Probability.** Weight refers to the importance of a factor relative to a thing, and its meaning and proportion are different, but the meaning expressed is not only the percentage of a factor to a thing, but it also focuses more on the importance of a factor to a thing. The weight of the Markov chain reflects the importance of the target training data to the time series of future prediction. The higher the weight, the greater the possibility. Therefore, the model weight  $w_k$  is obtained by standardizing the autocorrelation coefficient of each order, and its general form is shown in the following formula:

$$w_k = \frac{|r_k|}{\sum_{k=1}^m |r_k|}, \quad (8)$$

where  $w_k$  is the weight of the  $k$ -th step and  $m$  is the largest order of the time series.

After the weight of the Markov chain and the transition probability matrix of different steps are obtained, the corresponding state probability of each order is multiplied by the weight, and then the probability of the same state is added to obtain different prediction probabilities. Finally, the final probability  $p_i$  is obtained by weighting the probability of each order, and its general form is shown in the following formula:

$$p_i = \sum_{k=1}^m w_k P_i^{(k)}, i \in E. \quad (9)$$

## 4. MC-SARIMA

The multiplicative seasonal model refers to the use of a specific mathematical model to describe the trend between a set of random variables related to time. It can comprehensively consider components such as season, trend, and random interference and has a good prediction effect on time series. However, future data prediction often appears high and low or even seriously deviates from the actual situation, and the model has certain requirements for data stability [28]. The Markov chain refers to the random process as a theoretical basis to discuss the development law and prediction direction of the selected data. It is a time series itself, and the obtained value has one with the present, but it has nothing to do with the historical situation. However, in terms of data fitting, it often needs to cooperate with other models to play its maximum prediction effect [29]. Based on considering the advantages of the two and applying the multiplicative seasonal model to the selected time series, this study comprehensively considers the relationship between the existing data and further optimizes the prediction results by using the Markov chain model, to improve the prediction accuracy.

The MC-SARIMA model is based on the SARIMA model and uses the Markov chain to further optimize the error. The predicted value obtained by the SARIMA model is taken as the theoretical value and then compared with the actual value to obtain the residual value of the original data. According to the size relationship between the residual data, each data is divided into states, Markov test, probability calculation, etc., to optimize the latter prediction value. Then, the data are pushed back to a time node as a whole, and the data of the later time node are predicted and optimized by the same method to predict and optimize the value of the future time node; the steps are detailed as follows:

**Stage 1. Data Detection.** The stability of the data is an important prerequisite for the application of the multiplicative seasonal model. The ADF test can be used to test the stability of the data and white noise. If the data are stationary, go to step 3; if the data are not stationary, go to step 2.

**Stage 2. Differential Processing.** The difference is an important method to deal with the stability of time series. The results reflect a change between discrete variables and are a tool for studying discrete mathematics.

*Stage 3. Parameter Selection.* Using the AIC best criterion method to select the optimal parameters and bring them into the multiplicative seasonal model to obtain the theoretical value data.

*Stage 4. Acquisition of Residuals.* The residual value is obtained by the direct difference between the theoretical value and actual value of the multiplicative seasonal model, which is used as the training data of the Markov chain model.

*Stage 5. State Division.* Divide the residual value into different state intervals according to its size and distribution, and calculate the MAE, MAPE, RMSE, RRMSE, R2, and SSE [30].

*Stage 6. Markov Chain Test.* The Markov chain test [31] is an important prerequisite for testing whether trained time series conform to the one-dimensional Markov chain distribution and also testing the independence of transitions between different states, and the statistical formula is shown in equation (9).

$$\chi^2 = 2 \sum_{i=1}^{\beta} \sum_{j=1}^{\beta} f_{ij} \left| \log \left( \frac{P_{ij}}{P_{.j}} \right) \right|. \quad (10)$$

*Stage 7. Transfer Probability Matrix Calculation.* According to different state intervals, the transition probability from the target state to different states is an important process condition to obtain the prediction results.

*Stage 8. Predictive Optimal.* After calculating the optimal probability according to the autocorrelation coefficient, weight, and probability calculated in the Markov chain model, the predicted value is optimized according to the state interval and optimization formula corresponding to the optimal probability, and the optimization formula is shown in formula (10).

$$\alpha^* = \alpha + \frac{(\bar{\alpha} + \alpha^+)}{2\beta p_i}, \quad (11)$$

where  $\alpha^*$  is the predicted value after optimization,  $\alpha$  is the theoretical value,  $\alpha^+$  is the upper interval of the corresponding state interval,  $\bar{\alpha}$  is the lower interval of the corresponding state interval, and  $\beta$  is the number of states divided.

## 5. Case Analysis

In this study, the light buoy azimuth data for 96 consecutive hours from August 11 to August 14, 2018, at Meizhou Bay Port are taken as the actual value data and are incorporated into the MC-SARIMA model to predict the light buoy azimuth data for 12 hours on August 15, 2018. Firstly, the training data through the stationarity test and the white noise test are brought into the SARIMA model, and the corresponding parameters are selected to obtain the theoretical value data. Then, the residual data from 1:00 on August 11 to 24:00 on August 14 are brought into the MC-SARIMA model to obtain the residual prediction

data at 1:00 on August 15, and then the residual data from 2:00 on August 11 to 1:00 on August 15 are brought into the MC-SARIMA model to obtain the residual prediction data at 2:00 on August 15. By analogy, the residual prediction data from 1:00 to 12:00 on August 15 are obtained respectively. Then, the predicted azimuth data were obtained according to the relevant formulas, and the specific steps are shown as follows:

*5.1. Stationarity and White Noise Test.* To meet the requirements of the multiplicative seasonal model for the stability of time series and determine whether it is a white noise sequence, it is necessary to carry out differential processing, and the initial time-series data are tested by ADF to detect whether the time series can meet the stability requirements. The trend diagram and the first-order difference diagram are shown in Figures 1 and 2.

It can be seen from the comparison between Figures 1 and 2 that the first-order difference has reached the stability requirements, but to ensure the accuracy of the data, it is necessary to perform the ADF test. If it cannot meet the stability requirements, further difference processing is carried out.

An augmented Dickey-Fuller test statistic [32] (ADF) is one of the important methods to detect whether the time series are stationary and white noise sequences. The test results are listed in Table 1.

It can be seen from Table 1 that the ADF test statistics for the time series are less than the detection values of the corresponding test levels of 1%, 5%, and 10%, and the probability  $P < 0.05$  meets the requirements of the test observations. Therefore, the time series is a stationary sequence and a nonwhite noise sequence.

*5.2. Parameter Selection.* After the first-order difference of the time series, when the trend of the time series disappears, the parameter  $d = 1$ , and when the first-order difference of the time series disappears seasonally, the parameter  $D = 1$ . Then, the ACF and the PACF are integrated to determine the values of the parameters  $p$  and  $q$ . The general range of parameters  $p$  and  $q$  is  $[0, 2]$ . The analysis of ACF and PACF is shown in Figures 3 and 4.

By analyzing the autocorrelation function diagram and partial autocorrelation function diagram, it can be concluded that there is no obvious tailing or truncation. Therefore, the time series are divided into seasonal trend, random fluctuation trend, and growth trend as shown in Figure 5.

It can be seen from Figure 5 that there are two peaks every 24 hours, so the time series is affected by periodicity and seasonality. Therefore, it can be determined that the period of the time series is  $S = 24$ , and there is no obvious law between the random fluctuation trend and the growth trend. After the known parameters  $d = 1$ ,  $D = 1$ ,  $S = 24$ , to optimize the combination of various parameters, the AIC minimum information criterion method [33] is used to select the optimal parameters. The smaller the AIC value, the better the model. The AIC calculation results of the time series are listed in Table 2.

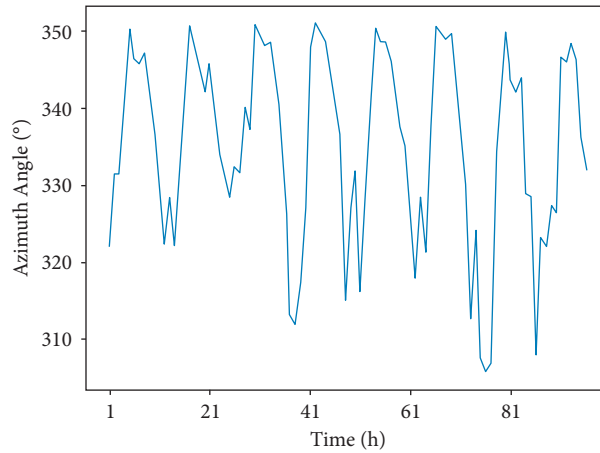


FIGURE 1: Azimuth trend diagram.

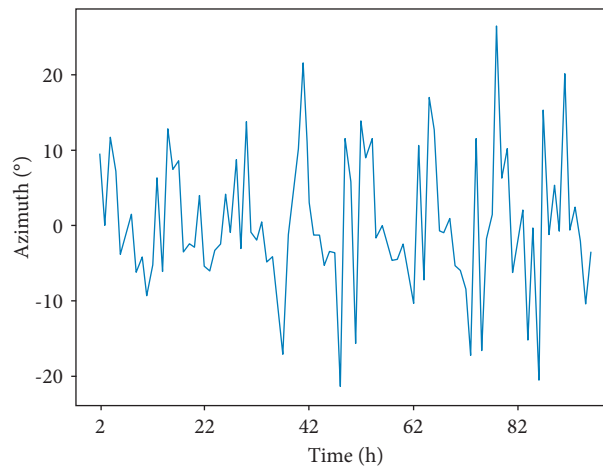


FIGURE 2: Azimuth angle first-order difference graph.

TABLE 1: ADF result of light buoy azimuth angle.

		t-statistic	Probability
Augmented Dickey–Fuller test statistic		-4.16208806	0.000763
Test critical values	1% level	-3.51273806	
	5% level	-2.89748987	
	10% level	-2.58594873	

Table 2 lists that the minimum AIC value is 320.38, so the parameters  $p=0$ ,  $q=1$ ,  $P=0$ , and  $Q=1$  can be determined, so the model combination should choose SARIMA  $(0,1,1) \times (0,1,1)_{24}$  as the best SARIMA multiplicative seasonal prediction model.

**5.3. Residual Acquisition.** According to the requirements of the Markov chain for time series data, the actual value data of 1–96 hours are taken as training data, and the multiplicative seasonal model is introduced to obtain the theoretical value data of 97–120 hours. Finally, the actual value is subtracted from the theoretical value to obtain the residual value. Some data are listed in Table 3.

**5.4. State Division.** State division is an important process in the Markov chain, which can divide each training data into different states, to make a theoretical basis for calculating the state probability. After the residual sequence is obtained, the mean value  $A$  and the mean square deviation  $M$  are calculated, and then the residual sequence is divided into four different states. The classification level is listed in Table 4.

**5.5. Probability Matrix and Markov Chain Test.** According to the state partition diagram, the frequency matrix  $(f_{ij})_{4 \times 4}$  is calculated and the one-step transition probability matrix  $P_1$  is obtained, and the results are as follows:

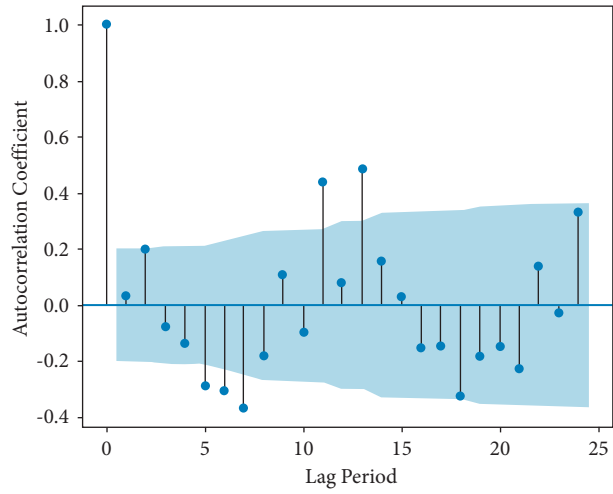


FIGURE 3: ACF analysis of light buoy azimuth.

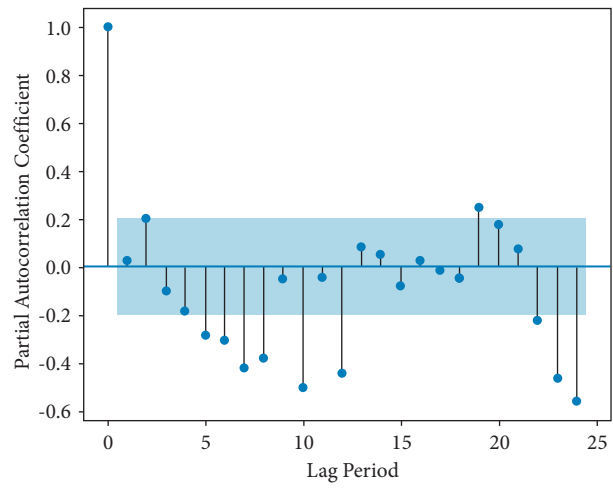


FIGURE 4: PACF analysis of light buoy azimuth.

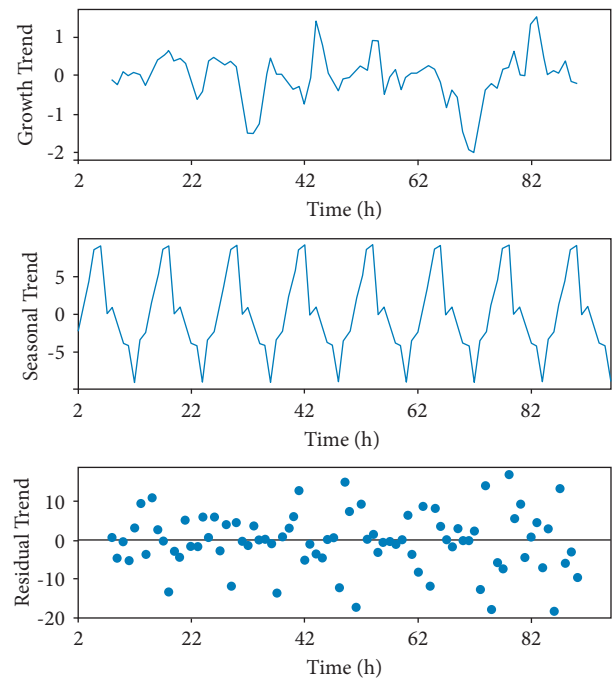


FIGURE 5: Analysis chart of azimuth angle trend of light buoy.

TABLE 2: AIC calculation results of light buoy azimuth angle.

Model parameter	AIC results
SARIMA (0,1,0) × (0,1,0,24)	AIC = 570.04
SARIMA (0,1,0) × (0,1,1,24)	AIC = 360.00
SARIMA (0,1,0) × (1,1,0,24)	AIC = 371.57
SARIMA (0,1,0) × (1,1,1,24)	AIC = 365.49
SARIMA (0,1,1) × (0,1,0,24)	AIC = 505.78
SARIMA (0,1,1) × (0,1,1,24)	AIC = 320.37
SARIMA (0,1,1) × (1,1,0,24)	AIC = 338.86
SARIMA (0,1,1) × (1,1,1,24)	AIC = 324.48
SARIMA (1,1,0) × (0,1,0,24)	AIC = 548.06
SARIMA (1,1,0) × (0,1,1,24)	AIC = 346.20
SARIMA (1,1,0) × (1,1,0,24)	AIC = 348.26
SARIMA (1,1,0) × (1,1,1,24)	AIC = 350.24
SARIMA (1,1,1) × (0,1,0,24)	AIC = 503.45
SARIMA (1,1,1) × (0,1,1,24)	AIC = 321.42
SARIMA (1,1,1) × (1,1,0,24)	AIC = 331.35
SARIMA (1,1,1) × (1,1,1,24)	AIC = 325.29

TABLE 3: Actual value, theoretical value, and residual value of azimuth angle.

Time	Actual values	Theoretical values	Residual values
2018/08/11 01:08	322.14239176	332.872897	-10.73050527
2018/08/11 02:08	331.44989702	330.7761572	0.67373980
2018/08/11 03:08	331.44989702	336.4254873	-4.97559026
2018/08/11 04:08	343.18230198	336.9629311	6.21937083
2018/08/11 05:08	350.26697980	339.4951977	10.77178205
2018/08/11 06:08	346.32674514	337.8502282	8.47651690
2018/08/11 07:08	345.65389600	333.6330618	12.02083420
2018/08/11 08:08	347.22796719	334.3270848	12.90088242
2018/08/11 09:08	340.87824255	325.2125488	15.66569373
2018/08/11 10:08	336.89065070	326.1651353	10.72551536
2018/08/11 11:08	327.45658909	323.685691	3.77089812
2018/08/11 12:08	322.14239176	318.6901163	3.45227541

$$(f_{ij})_{4 \times 4} = \begin{pmatrix} 5 & 6 & 3 & 0 \\ 5 & 11 & 12 & 0 \\ 2 & 10 & 19 & 8 \\ 1 & 1 & 6 & 6 \end{pmatrix}; P_1 = (P_{ij})_{4 \times 4} = \begin{pmatrix} 0.357142857 & 0.42857143 & 0.2142857 & 0 \\ 0.178571429 & 0.39285714 & 0.4285714 & 0 \\ 0.051282051 & 0.25641026 & 0.4871795 & 0.205128 \\ 0.071428571 & 0.07142857 & 0.4285714 & 0.428571 \end{pmatrix}. \quad (12)$$

After calculating the two-step to four-step transition probability matrix  $P_2$ ,  $P_3$ ,  $P_4$  and marginal probability  $P_{*1} = 0.136842105$ ,  $P_{*2} = 0.29473684$ ,  $P_{*3} = 0.4210526$ , and  $P_{*4} = 0.147368$ , the Markov chain test is done for the residual sequence.

The Markov chain test is an important prerequisite to test whether the trained time series conform to the one-dimensional Markov chain distribution. The statistical formula is shown in formula (9) and the results are listed in Table 5.

Given the explicitness level  $\alpha = 0.05$ , the statistics calculated according to Table 5,  $\chi^2 = 14.66091051$  and  $\chi_{\alpha}^2(7) = 14.067$ , from which it can be seen that  $\chi^2 > \chi_{\alpha}^2(7)$ , so the residual sequence satisfies the Markov chain test and the predicted value can be optimized by the Markov chain.

**5.6. State Prediction and Prediction Optimization.** According to the residual data of 96 hours from August 11 to August 14, 2018, and the corresponding state probability matrix, and combined with the weighted sum formula, the 12 hours on August 15, 2018, are predicted. The predicted results at 1:00 on August 15, 2018, are listed in Table 6.

It can be seen from Table 6 that  $\max(p_i) = 0.453017$ , corresponding to state  $i = 3$ , and the predicted value is optimized according to the different intervals of the predicted state, the number of states divided, and the corresponding interval value; the optimization formula is shown in equation (10).

According to the optimization formula, the optimization results at 1 o'clock on August 15, 2018, were 320.6522 m, and the data in the next 11 hours can be predicted using the same

TABLE 4: State partition diagram.

State $i$	Fundamentum divisionis	State interregional
1	[A-2M, A-M)	[-20.30674, -10.3859)
2	[A-M, A)	[-10.38587, -0.46499)
3	[A, A + M)	[-0.464995, 9.455879)
4	[A + M, A+2M)	[9.455879, 19.37675)

TABLE 5: Markov chain test diagram.

Statistic state	$\chi_1^2$	$\chi_2^2$	$\chi_3^2$	$\chi_4^2$	Total
1	0.959308117	0.374374521	0.675447603	0	2.009130241
2	0.266160936	0.287363144	0.017699577	0	0.571223658
3	0.981486931	0.139304172	0.145874771	0.330699458	1.597365331
4	0.650129795	1.417384948	0.017699577	1.067521702	3.152736022
Total	2.85708578	2.218426785	0.856721528	1.398221159	14.66091051

TABLE 6: The azimuth state probability chart at 1:00 on August 15, 2018.

Initialization time	Initial state	Lag order	Weight	$E_1$	$E_2$	$E_3$	$E_4$
2018/08/14 12:08	3	1	0.477522	0.051282	0.256410	0.487179	0.205128
2018/08/14 11:08	2	2	0.328658	0.156129	0.341064	0.415826	0.087945
2018/08/14 10:08	3	3	0.185709	0.119840	0.274692	0.432095	0.171957
2018/08/14 09:08	3	4	0.008109	0.126199	0.282190	0.427689	0.162349
$(p_i)$				0.099080	0.287836	0.453017	0.160108

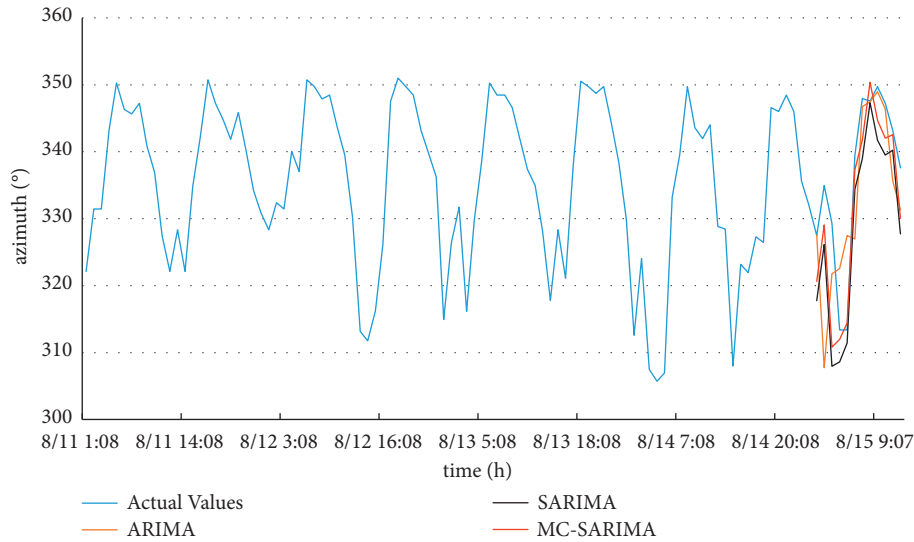


FIGURE 6: Comparison chart of actual value and optimization value of light buoy azimuth.

method and compared with the actual value, ARIMA prediction value, and SARIMA prediction value. The comparison figure is shown in Figure 6, and the specific data are listed in Table 7.

The prediction error is calculated based on the mean absolute error (MAE), sum of squares due to error (SSE), R-square ( $R^2$ ), mean absolute percentage error (MAPE), root mean square error (RMSE), and relative root mean square error (RRMSE) [34, 35], and the formula is shown in equations (11)–(16) and the comparison of prediction error is shown in Figure 7.

$$MAE = \frac{1}{n} \sum_{i=1}^n |x - x^*|, \tag{13}$$

$$SSE = \sum_{i=1}^n (x - x^*)^2, \tag{14}$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (x - x^*)^2}{\sum_{i=1}^n (x - \bar{x})^2}, \tag{15}$$

TABLE 7: Comparison diagram of actual value and optimized value.

Time	Actual values	ARIMA	SARIMA	MC-SARIMA
2018/08/15 01:08	327.456589	327.541026	317.748908	320.652286
2018/08/15 02:08	334.960322	307.729732	326.130433	329.059489
2018/08/15 03:08	329.335133	321.759505	307.967178	310.805353
2018/08/15 04:08	313.385821	322.552179	308.591898	311.972324
2018/08/15 05:08	313.385821	327.474060	311.432564	314.414731
2018/08/15 06:08	339.432520	326.956267	334.384477	337.296351
2018/08/15 07:08	347.957028	346.751462	338.976334	341.76127
2018/08/15 08:08	347.562032	347.476712	347.319574	350.401482
2018/08/15 09:08	349.721704	348.987890	341.703677	344.711572
2018/08/15 10:08	347.227967	346.402587	339.505950	342.047134
2018/08/15 11:08	343.182116	335.642142	340.225185	342.55644
2018/08/15 12:08	337.598977	331.335220	327.737422	330.062343

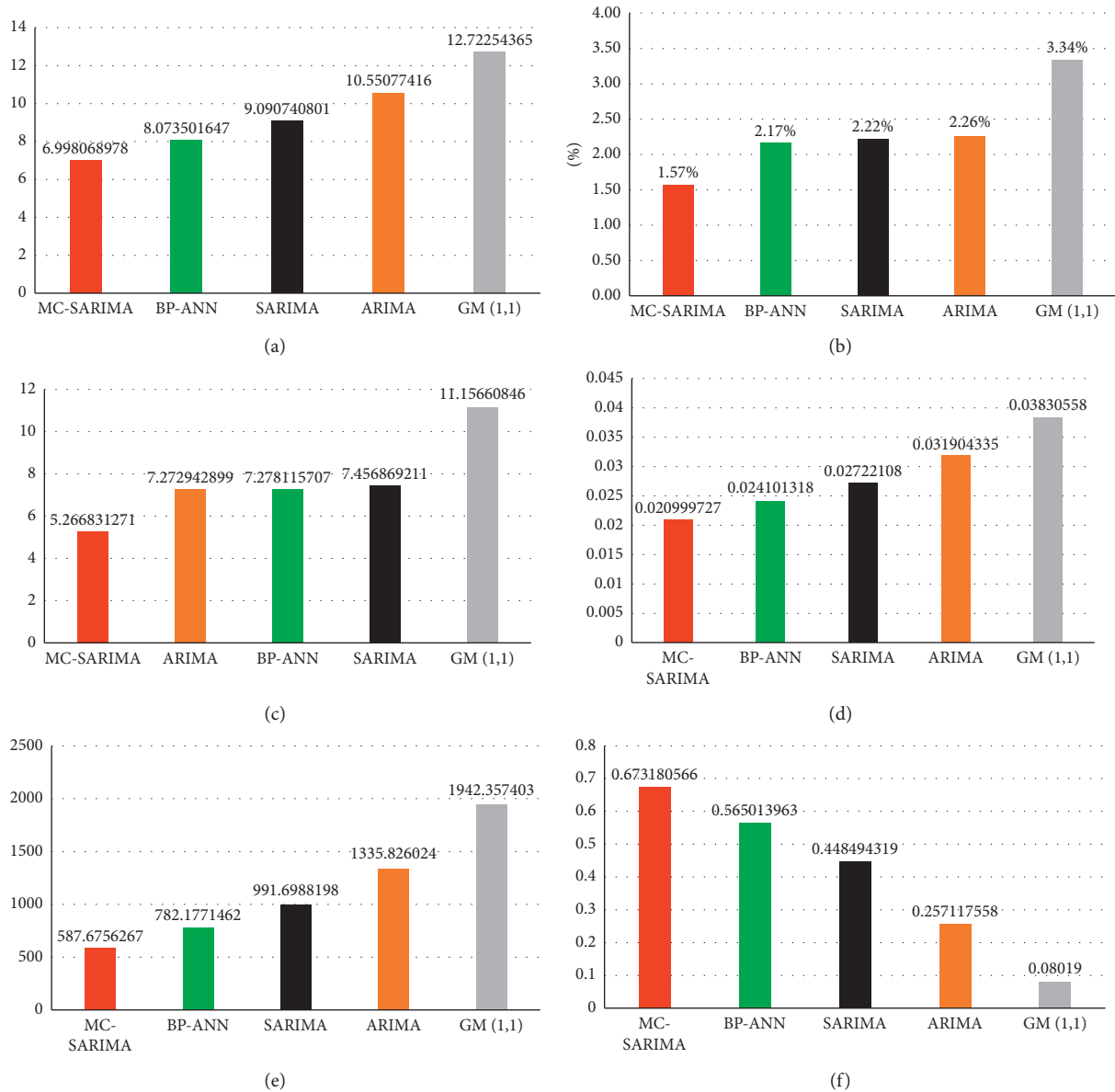


FIGURE 7: Error comparison.

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{x - x^*}{x} \right| \times 100\%, \quad (16)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (x - x^*)^2}{n}}, \quad (17)$$

$$\text{RRMSE} = \sqrt{\sum_{i=1}^n \left( \frac{x - x^*}{x} \right)^2 / n}, \quad (18)$$

where  $x^*$  is the prediction value,  $x$  is the actual value, and  $\bar{x}$  is the average of actual values.

Figure 7 shows that compared to the traditional multiplicative seasonal model and other prediction models, the prediction MAE of the MC-SARIMA model is decreased by 2.19003794 and the MAPE is decreased by 0.66%, the RMSE is decreased by 2.092671823, the RRMSE is decreased by 0.006221352, the SSE is decreased by 404.0231931, and the  $R^2$  is increased by 0.224686247.

It shows that the multiplicative seasonal model optimized by the Markov chain can predict the azimuth data of the light buoy more effectively.

## 6. Conclusion

In this study, a single multiplicative seasonal model is established and the time series is predicted based on the periodic characteristics of the offset azimuth of the light buoy. The azimuth prediction MAE is 7.456869211, MAPE is 2.22%, RMSE is 9.090740801, RRMSE is 0.02722108, SSE is 991.6988198, and  $R^2$  is 0.448494319, respectively. However, to further improve the prediction accuracy, the Markov chain is used to further optimize the multiplicative seasonal model. The azimuth prediction MAE of the light buoy azimuth is 5.266831271, MAPE is 1.57%, RMSE is 6.998068978, RRMSE is 0.020999727, SSE is 587.6756267, and  $R^2$  is 0.673180566, respectively. The MC-SARIMA model, ARIMA model, SARIMA model, BP-ANN, and GM (1, 1) are compared, and the results are obtained. The results show that compared with the single multiplicative seasonal model and other prediction models, the prediction results of the multiplicative seasonal model optimized by the Markov chain are more able to meet the needs of the navigator, and the prediction error is lower, which provides a new idea for the offset warning of the light buoy. However, the model does not fully take into account the special circumstances under the influence of nonseasonal cycles and will be followed up to study the possible changes in the azimuth of the buoy in special cases such as typhoons and ship collisions and make predictions according to their existing conditions.

## Data Availability

The light buoy position data used to support the findings of this study were supplied by Jinxing Shao under license and so cannot be made freely available. Requests for access to

these data should be made to Jinxing Shao; affiliation: Aids to Navigation Department, Xiamen; e-mail: 13606931987@139.com.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Natural Science Foundation of Fujian Province (Grant no.2020J01658, 2019J01325), Open Project Fund of National Local Joint Engineering Research Center for Ship Assisted Navigation Technology (Grant no. HHXY2020002), Doctoral Start-up Fund of Jimei University (Grant no. ZQ2019012).

## References

- [1] A. Blazquez-Garcia, A. Conde, and A. Milo, "Short-term office building elevator energy consumption forecast using SARIMA," *Journal of Building Performance Simulation*, vol. 13, no. 01, pp. 69–78, 2020.
- [2] R. Vahid, B. Saied, and H. Aliakbar, "Predicting cutaneous leishmaniasis using SARIMA and Markov switching models in Isfahan, Iran: a time-series study," *Asian Pacific Journal of Tropical Medicine*, vol. 14, no. 02, pp. 83–93, 2021.
- [3] R. Peirano, W. Kristjanpoller, and M. C. Minutolo, "Forecasting inflation in Latin American countries using a SARIMA-LSTM combination," *Soft Computing*, vol. 25, no. 16, Article ID 10851, 2021.
- [4] M. Farsi, D. Hosahalli, and B. R. Manjunatha, "Parallel genetic algorithms for optimizing the SARIMA model for better forecasting of the NCDC weather data," *Alexandria Engineering Journal*, vol. 60, no. 01, pp. 1299–1316, 2020.
- [5] M. K. Sharma, M. Omer, and S. Kiani, "Time series analysis on precipitation with missing data using stochastic SARIMA," *Mausam*, vol. 71, no. 04, pp. 617–624, 2020.
- [6] M. J. Qian and Y. Z. E. R. N. Li, "Method on monthly railway passenger traffic forecast based on seasonal decomposition and SARIMA-GARCH model," *Journal of the China Railway Society*, vol. 42, no. 06, pp. 25–34, 2020.
- [7] L. Parviz, "Comparative evaluation of hybrid SARIMA and machine learning techniques based on time varying and decomposition of precipitation time series," *Journal of Agricultural Science and Technology A*, vol. 22, no. 02, pp. 563–578, 2020.
- [8] S. S. Daisy, A. K. M. S. Islam, and A. S. Akanda, "Developing a forecasting model for cholera incidence in Dhaka megacity through time series climate data," *Journal of Water and Health*, vol. 18, no. 02, pp. 207–223, 2020.
- [9] H. Liu, C. X. Li, and Y. Q. Shao, "Forecast of the trend in incidence of acute hemorrhagic conjunctivitis in China from 2011–2019 using the seasonal autoregressive integrated moving average (SARIMA) and exponential smoothing (ETS) models," *Journal of Infection and Public Health*, vol. 13, no. 02, pp. 287–294, 2020.
- [10] T. Wang and Y. C. Zhou, "Forecasting the Yellow River runoff based on functional data analysis methods," *Environmental and Ecological Statistics*, vol. 28, no. 01, pp. 1–20, 2020.
- [11] O. Berke, L. Trotz-Williams, and S. De Montigny, "Good times bad times: automated forecasting of seasonal cryptosporidiosis in Ontario using machine learning," *Canada*



- Communicable Disease Report*, vol. 46, no. 06, pp. 192–197, 2020.
- [12] W. W. Yu and J. M. Xing, “Sports event model evaluation and prediction method using principal component analysis,” *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 9351522, 10 pages, 2022.
- [13] Z. Tian, “Short-term wind speed prediction based on LMD and improved FA optimized combined kernel function LSSVM,” *Engineering Applications of Artificial Intelligence*, vol. 91, Article ID 103573, 2020.
- [14] J. X. Li, A. J. Zhou, and Y. F. Liao, “Forest ecological diversity change prediction discrete dynamic model,” *Discrete Dynamics in Nature and Society*, vol. 2022, Article ID 4869363, 11 pages, 2022.
- [15] L. Liu, W. Z. Chen, and Y. B. Li, “Comprehensive analysis of pyroptosis-related long noncoding RNA immune infiltration and prediction of prognosis in patients with colon cancer,” *Journal of Oncology*, vol. 2022, 2022.
- [16] Z. D. Tian, S. J. Li, and Y. H. Wang, “A prediction approach using ensemble empirical mode decomposition-permutation entropy and regularized extreme learning machine for short-term wind speed[J],” *Wind Energy*, vol. 23, no. 02, pp. 177–206, 2020.
- [17] Z. Zeng, H. Yan, and A. M. N. Fu, “Time-series prediction based on pattern classification[J],” *Artificial Intelligence in Engineering*, vol. 15, no. 01, pp. 61–69, 2001.
- [18] K. X. Yuan, J. Liu, and S. C. Yang, “Time series forecasting based on kernel mapping and high-order fuzzy cognitive maps,” *Knowledge-Based Systems*, vol. 206, 2020.
- [19] R. Sebastian and N. Thomas, “A fractal interpolation approach to improve neural network predictions for difficult time series data,” *Expert Systems with Applications*, vol. 169, 2021.
- [20] M. T. Her, C. C. Chung, and C. T. Lin, “Ship price predictions of panamax second-hand bulk carriers using grey models,” *Journal Of Marine Science and Technology-Taiwan*, vol. 27, no. 03, pp. 229–234, 2019.
- [21] Y. Mei, J. Q. Yang, and Y. Lu, “BP-ANN model coupled with particle swarm optimization for the efficient prediction of 2-chlorophenol removal in an electro-oxidation system,” *International Journal of Environmental Research and Public Health*, vol. 16, no. 14, 2019.
- [22] L. Duan, T. Hu, and X. Y. Zhu, “SpatioTemporal prediction of suspect location by SpatioTemporal semantics,” *Geomatics and Information Science of Wuhan University*, vol. 44, no. 5, pp. 765–770, 2019.
- [23] X. D. Li and J. G. Ying, “Markov chain approximations for one dimensional diffusion[J],” *Osaka Journal of Mathematics*, vol. 58, no. 03, pp. 551–561, 2021.
- [24] Y. C. Lai and D. A. Dzombak, “Use of integrated global climate model simulations and statistical time series forecasting to project regional temperature and precipitation[J],” *Journal of Applied Meteorology and Climatology*, vol. 60, no. 05, pp. 695–710, 2021.
- [25] J. Huang, C. W. Chu, and Y. C. Tsai, “Container throughput forecasting for international ports in Taiwan,” *Journal of Marine Science and Technology-Taiwan*, vol. 28, no. 05, pp. 456–469, 2020.
- [26] P. P. Zhong, W. G. Yang, and Z. Y. Shi, “Some limit theorems of delayed averages for countable nonhomogeneous Markov chains,” *Probability in the Engineering and Informational Sciences*, vol. 35, no. 02, pp. 297–315, 2021.
- [27] X. R. Jiang, *Time Series Analysis and prediction*, pp. 42–43, Science Press, Beijing, China, 2020.
- [28] M. J. Song and L. H. Yong, “Forecasting the east sea rim container volume by SARIMA time series model,” *Korea Trade Review*, vol. 45, no. 05, pp. 75–89, 2020.
- [29] Z. G. Wu, S. L. Dong, and P. Shi, “Reliable filter design of takagi-sugeno fuzzy switched systems with imprecise modes,” *IEEE Transactions on Cybernetics*, vol. 50, no. 05, pp. 1941–1951, 2020.
- [30] Z. D. Tian, “Preliminary research of chaotic characteristics and prediction of short-term wind speed time series,” *International Journal of Bifurcation and Chaos*, vol. 30, no. 12, Article ID 2050176, 2020.
- [31] V. Rahmanian, S. Bokaie, and A. Haghdoost, “Predicting cutaneous leishmaniasis using SARIMA and Markov switching models in Isfahan, Iran: a time-series study,” *Asian Pacific Journal of Tropical Medicine*, vol. 14, no. 02, pp. 83–93, 2021.
- [32] L. X. Yang and Q. N. Li, “Spurious rejections of Dickey-Fuller tests under permanent breaks,” *Systems Engineering-Theory & Practice*, vol. 40, no. 06, pp. 1361–1370, 2020.
- [33] A. Choiruddin, J. F. Coeurjolly, and R. Waagepetersen, “Information criteria for inhomogeneous spatial point processes,” *Australian & New Zealand Journal of Statistics*, vol. 63, no. 01, pp. 119–143, 2021.
- [34] Z. D. Tian, “Approach for short-term traffic flow prediction based on empirical mode decomposition and combination model fusion,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 09, pp. 5566–5576, 2021.
- [35] Z. Tian and H. Chen, “A novel decomposition-ensemble prediction model for ultra-short-term wind speed,” *Energy Conversion and Management*, vol. 248, Article ID 114775, 2021.

## Research Article

# Chimp Optimization Algorithm to Optimize a Convolutional Neural Network for Recognizing Persian/Arabic Handwritten Words

Sara Khosravi  and Abdoloh Chalechale 

*Department of Computer Engineering and Information Technology, Razi University, Kermanshah, Iran*

Correspondence should be addressed to Abdoloh Chalechale; [chalechale@razi.ac.ir](mailto:chalechale@razi.ac.ir)

Received 31 January 2022; Revised 14 March 2022; Accepted 18 March 2022; Published 12 April 2022

Academic Editor: Araz Darba

Copyright © 2022 Sara Khosravi and Abdoloh Chalechale. 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.

Handwritten character recognition is an attractive subject in computer vision. In recent years, numerous researchers have implemented techniques to recognize handwritten characters using optical character recognition (OCR) approaches for many languages. One of the most common methods to improve the OCR accuracy is based on convolutional neural networks (CNNs). A CNN model contains several kernels accompanying with pooling layers and nonlinear functions. This model overcomes the problem of adjusting the value of weights and interconnections of the neural network (NN) for creating an appropriate pipeline to process the spatial and temporal information. However, the training process of a CNN is a challenging issue. Various optimization strategies have been recently utilized for optimizing CNN's biases and weights such as firefly algorithm (FA) and ant colony optimization (ACO) algorithms. In this study, we apply a well-known nature-inspired technique called chimp optimization algorithm (ChOA) to train a classical CNN structure LeNet-5 for Persian/Arabic handwritten recognition. The proposed method is tested on two known and publicly available handwritten word datasets. To deeply investigate and evaluate the approach, the results are compared with three optimization methods including ACO, FA, and particle swarm optimization (PSO). Outcomes indicated that the proposed ChOA technique considerably improves the performance of the original LeNet model and also shows a better performance than the others.

## 1. Introduction

Image processing and computer vision fields have been popular for solving numerous issues such as object recognition, face recognition, place recognition, emotion recognition, and so on. One of the widespread applications of image processing in the real world is optical character recognition (OCR) [1]. OCR is a technology developed with the purpose of detecting and recognizing text characters in any image containing printed or handwritten text. This technology has a wide range of applications, most notably in automatic identification of scanned documents, automatic reading of license plates, and automatic recognition of traffic signs [2, 3]. OCR can be considered as a compound problem due to the variety of styles, complex rules, fonts, orientation, and languages in which text can be written. Therefore, approaches from diverse strategies of computer

science (i.e., natural language processing, pattern recognition, and image processing) can be utilized to overcome diverse challenges. Persian and Arabic scripts are specific sets of characters employed by the Middle East countries and these characters are often found in the books, wall carvings, or street signboards. However, people these days face the problem that not everyone is able to read images of Persian and Arabic scripts and there is a need for a system to convert these images into texts that can be translated by machine. Nowadays, many techniques have been developed to create more efficient and robust systems in recognizing characters by employing OCR [4].

Deep learning (DL) is recently gaining attention, and it has been considered as the base pipeline for numerous artificial intelligence (AI) applications including emergency vehicle recognition [5], biometric authentication based on finger knuckle and fingernail [6], synthetic aperture radar

(SAR) image processing [7], the diagnosis of tumor location in breast cancer [8], speech recognition [9, 10], liver or lung tumor segmentation [11–13], and various applications in computer networks such as network traffic classification [14] and network intrusion detection system [15]. In the 1980s, researchers developed the first DL-based architecture built for computer vision applications on the foundations of artificial neural networks (ANNs) [16]. The ANNs with additional hidden layers have a large capacity for procedure of the feature extraction. However, an ANN often encounters gradient propagation and converges to the local optimum, when its structures become complex and deep. The core or fundamental benefit of DL is feature extraction from raw data and learning these features automatically. These extracted features are obtained by forming the composition of lower level features to generate the higher levels of the hierarchy [17].

DL is able to handle more complex challenges principally fast and well, due to employing more complex frameworks, which permit massive parallelization. These complex frameworks can be utilized to reduce the error of regression techniques and also enhance the classification efficiency. A DL pipeline entails several dissimilar layers (e.g., convolution layers (Conv layers), pooling layers, activation functions, fully connected (FC) layers, memory cells, gates, and so on), depending on the utilized model (i.e., deep belief network, recursive neural networks, recurrent neural networks, convolutional neural networks, or unsupervised pretrained networks). Among the several pipelines of DL, the focus of this study is on CNN.

Although a CNN framework is able to learn numerous complex patterns through several biases and weights inside the Conv layers, this framework encounters many challenges in the training step to produce the optimal outcomes. These biases and weights achieve their finest possible values through a learning step with a lot of input samples. In fact, in all CNN models, the number of samples in the training phase plays a crucial role to obtain the best possible outcome. To reach to this end, several optimization strategies have been implemented to manipulate the value of biases and weights. The most famous techniques are adaptive gradient strategies. Essentially, these techniques amend the learning rate (LR) by a backpropagation approach. These techniques reduce the LR if the gradient of a parameter is small or vice versa. The most preferred approach among adaptive gradient approaches is the stochastic gradient descent (SGD) [18]. However, due to tuning LR manually in the SGD, they demonstrated to have an inappropriate performance especially when the network is large like a CNN model. This significantly increases the training time of the models for large-scale input. To address this challenge and increase the efficiency of adaptive gradient approaches, new variants of adaptive gradient techniques were offered. These algorithms include YOGI [19], which improves the process of varying the LR in gaining better convergence, or Nostalgic Adam [20], which places bigger weights on the past gradient compared to the recent gradient. However, they have not achieved attractiveness in the field of image processing using CNNs, which leads to the use of metaheuristic algorithms as alternatives.

Recently, many metaheuristic schemes have been utilized to resolve complicated problems such as detection problems [21] and scheduling [22]. One popular way to classify the metaheuristics is based on the inspiration of human-based approaches, physics-based approaches, swarm intelligence approaches, and evolutionary approaches [23, 24]. Evolutionary techniques simulate patterns in natural evolution and apply operators motivated by biology manners like mutation and crossover. The genetic algorithm (GA) can be considered as a conventional evolutionary method, which is motivated by Darwinian evolutionary ideas. Conventional approaches of this group comprise evolution strategy [25], differential evolution [26], and evolutionary programming [27].

Swarm intelligence techniques which simulate the manners of animals in hunting or movement groups are another group of metaheuristics [28]. The distribution of organism details of all animals through the optimization procedure is the main characteristic of this group. Conventional algorithms of this group comprise dolphin echolocation [29], salp swarm approach [30], and krill herd approach [31].

Physics-based approaches originated from physical laws in real life. These strategies describe the communication of exploration solutions based on adjusting instructions rooted in physical approaches. The most popular employed strategies in these groups are charged system search [32], multiverse optimizer [33], and gravitational search algorithm [34]. Human-based approaches are considered the last optimizer strategy, which are motivated by human behavior and human cooperation in communities. One of the most utilized techniques which are motivated by the human sociopolitical growth practice is the imperialist competitive technique [35, 36].

Metaheuristic algorithms are cherished for their simplicity, gradient independence, and ability to evade local optima. These properties make them a superior choice in the training of ANNs with a great number of parameters [37]. Zhang et al. [38] utilized a whale optimization algorithm (WOA) to diminish the error rate during learning process during the pretraining process of the CNN for classification of skin cancer images. This optimization approach employs half value precision as the cost function. The outcomes of their analysis indicated that the accuracy of the approach is better compared to other recently published approaches.

In this paper, from the numerous kinds of the metaheuristic techniques, the chimpanzee optimization algorithm (ChOA) is employed to obtain weights and biases for improving the performance of the LeNet-5. This optimizer belongs to the swarm-based algorithms of metaheuristic algorithms inspired by the sexual motivation (SM) and individuals diversity (ID) of chimps in their group hunting. Also, to evaluate our results, some other well-known optimizers, such as firefly algorithm (FA), ant colony optimization (ACO), and particle swarm optimization (PSO), are used for comparison.

Although many metaheuristic optimization algorithms (MOAs) have been employed to improve the CNN performance, our contribution can be considered in four categories:

- (1) Based on no free lunch theorem (NFL) [39, 40] saying that “there is no metaheuristic algorithm that can solve all optimization problems well, so there is always a need to develop new algorithms,” this is the first time that the chimp optimization algorithm (ChOA) is used to improve the CNN performance by adjusting the learnable parameters.
- (2) We introduce an approach to reducing classification errors of the CNN based on local and global search capabilities of gradient descent (GD) and ChOA algorithms.
- (3) We improve the performance of GD algorithm by finding the optimal initial weights using the ChOA algorithm.
- (4) The CNN network improved by GD and ChOA is used to recognize handwritten words both in Persian and Arabic languages.

The rest of this paper is organized as follows. Section 2 presents a literature review of convolutional neural networks. Section 3 explains the chimp optimization algorithm. In Section 4, the proposed method is comprehensively explained. Section 5 represents and discusses the experimental results, and finally Section 6 gives the conclusion.

## 2. Convolutional Neural Network

Convolutional neural networks (CNNs) are considered the most common type of ANNs. Theoretically, as each unique neuron in the multilayer perceptron (MLP) maps the weighted inputs to the output using an activation function, CNNs resemble an MLP. They are specifically proposed to identify multidimensional data with a high degree of invariance to distortion and shift scaling [41–43].

The convolution layers (Conv layers) generate some new images from the input image called feature maps. These Conv layers operate in a quite different way in comparison to the other neural network layers. These layers use convolution kernels to convert images and do not use connection weights and a weighted sum. Convolution product recognizes visual patterns within the input image [38, 44]. All neurons in the feature maps are connected to the neurons of the previous layer within a region with the same dimension of the corresponding kernel. This region can be considered as the receptive field of the neuron. Weight sharing mechanism is also another beneficial property, which uses the identical convolutional kernel in the whole image. This weight sharing mechanism, by permitting the equivalent feature to be searched in the whole image, decreases the number of parameters to be trained [44, 45]. The output of convolution layers can be calculated as

$$I_i^l = \left( b_i^l + \sum_j w_{ij}^l \otimes I_j^{l-1} \right), \quad (1)$$

where  $I_i^l$  is the output,  $\otimes$  indicates the convolution operator,  $w_{ij}$  demonstrates convolution kernels, and  $b_i$  is the bias value [46].

The pooling layers reduce the dimension of the input data. This reducing strategy is implemented by combining neighboring pixels of a definite image region into a value. The well-known pooling operations are L2, average (Mean-Pooling), and maximum pooling (Max-Pooling). While L2 pooling computes the L2 norm of the inputs, Max-Pooling passes the maximum value throughout, and Mean-Pooling takes the average of the input values [47, 48].

FC layer is a layer of an ANN in which all neurons connect to all neurons of the following layer. For each connection to an FC, a linear grouping of the outputs of the prior layer with an added bias is considered the input to a node.

In this paper, to recognize correct words, we use the LeNet-5 model [49, 50] with eight layers. The initial Conv layer (Conv1) uses six different filters with the dimension of  $5 \times 5$  to extract key information from the input data (the input grayscale image size is  $32 \times 32$ ) for creating each feature map. Every neuron of Conv1 accepts achieved inputs from a  $5 \times 5$  receptive field at the former input (layer). Due to the weight sharing technique, the same bias and weights are used for all units in these feature maps for producing a linear position invariant kernel in all areas of the input image. During the training process, the sharing weights of the Conv layer are adjusted. Conv1 layer uses 6 convolutional  $5 \times 5$  filters with diverse biases including 122304 connections, 4704 number of neurons, and 156 trainable parameters [51, 52].

The succeeding layer is a downsampling layer (Pooling2) with a  $2 \times 2$  kernel for each feature map and six feature maps. Actually, after applying the downsampling strategy in the receptive field using the mean value, the outcome is multiplied (the value of the output pixel). Then, this outcome is added by two trainable coefficients which are different in different feature maps but are similar for the output pixels of a feature map. Pooling2 layer has 5880 connections and 12 trainable parameters.

The next convolutional layer is Conv3 which has a  $5 \times 5$  kernel for each feature map and 16 feature maps. The succeeding layer (Pooling4) is a downsampling layer with a  $2 \times 2$  filter for each feature map and 16 feature maps. This downsampling layer has 2000 connections and 32 trainable parameters. The next convolutional layer is Conv5 which has a  $5 \times 5$  kernel for each feature map, 120 feature maps, and 48120 connections and trainable parameters. The next layer is a fully connected layer (FC6) which accepts 84 neurons. Each CNN layer has a number of trainable parameters, connections, neurons, and feature maps, which are demonstrated in Table 1. The overall pipeline of LeNet-5 is demonstrated in Figure 1.

## 3. Chimp Optimization Algorithm

In order to design a metaheuristic algorithm, two criteria need to be considered: exploitation (intensification) and exploration (diversification). The solution accuracy and convergence speed of the strategy principally can be determined by the balance level among these two criteria. Exploitation can be described as discovering nearby

TABLE 1: Details of LeNet-5.

Layers	Dimension	Number of feature maps	Number of parameters	Number of connections
Input [51]	$32 \times 32$	...	...	...
Conv1	$28 \times 28$	6	156	122304
Pooling2	$14 \times 14$	6	12	5880
Conv3	$10 \times 10$	16	1516	151600
Pooling4	$5 \times 5$	16	32	2000
Conv5	$1 \times 1$	120	...	48120
FC6	$1 \times 1$	84	10164	...

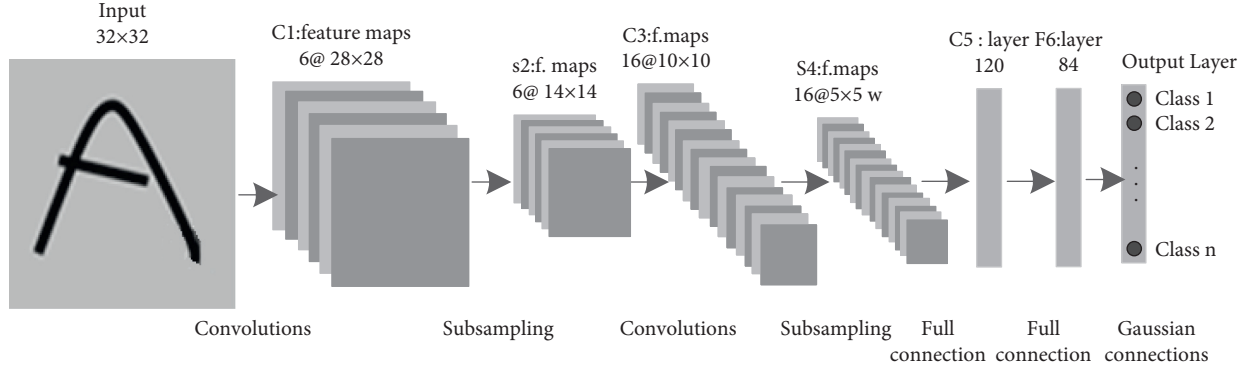


FIGURE 1: Design of LeNet-5 pipeline [51].

promising solutions to develop their quality locally. On the contrary, exploration can be described as exploring the search space globally. This ability is related to the avoidance of local optimum and resolving local optima entrapment. At the first step, an efficient optimizer approach needs to explore the search space to discover different solutions. After an adequate transition, the technique often employs local details for generating some improved solutions, which are commonly in the areas close to the existing solutions [53, 54].

There are some dissimilarities between the social behavior of chimps and any other flocking manners including SM and ID. Nutritional benefits of group hunting or SM implies that chimps' hunting can be influenced by the probable social advantages of achieving food (meat). Obtaining food offers a chance for trading it in return for social favors such as grooming and sex. This motivation in the ultimate phase leads chimps to forget their duties in the hunting practice. So, they attempt to gain food chaotically. This unconditional behavior in the ultimate phase leads to convergence rate progress and exploitation step [53, 54].

ID demonstrates that each individual does not have quite similar intelligence and ability compare to other group members, but they all accomplish their responsibilities in their best ways. The ability of each individual can be beneficial in a step of the hunting procedure. Hence, each part of the group based on his special ability accepts some responsibilities for a part of hunt.

In their colony, we can observe four kinds of chimps including attackers, chasers, barriers, and drivers. The responsibility of drivers is following the preys without trying to catch up with them. Barriers are responsible for placing themselves in trees to create a dam across the movement of

the prey. Chasers have responsibility to catch up the preys by moving rapidly after them. Lastly, attackers bring preys down into the lower canopy or back towards the chasers by anticipating the breakout route of preys to inflict them. This strategy is demonstrated in Figure 2.

Accordingly, attackers require much more cognitive skills to anticipating the next actions of the prey. So, this significant attacking role correlates positively with the physical ability, smartness, and age. Besides, all group members can keep their same responsibility during the entire procedure or alter duties during the same hunt [55, 57]. As mentioned before, the preys can be hunted during the exploitation and exploration steps. To this end, we formulate chasing and driving the preys as follows:

$$d = |c \cdot x_{\text{prey}} - m \cdot x_{\text{chimp}}|, \quad (2)$$

$$x_{\text{chimp}}(t+1) = x_{\text{prey}}(t) - a \cdot d,$$

where  $X_{\text{chimp}}$  indicates the position vector of a chimp,  $X_{\text{prey}}$  demonstrates the vector of prey position  $c$ ,  $m$  and  $a$  are the coefficient vectors, and  $t$  is the number of current iterations.  $a$ ,  $m$ , and  $c$  vectors are calculated by

$$a = 2 \cdot f \cdot r_1 - f,$$

$$c = 2 \cdot r_2, \quad (3)$$

$$m = \text{Chaotic\_value},$$

where  $r_1$  and  $r_2$  indicate the random vectors in the range of  $[0, 1]$  and  $f$  implies reduced nonlinearity in both exploration and exploitation steps from 2.5 to 0 through the iteration process. Lastly,  $m$  is a chaotic vector computed based on many chaotic maps to indicate the consequence of the SM in

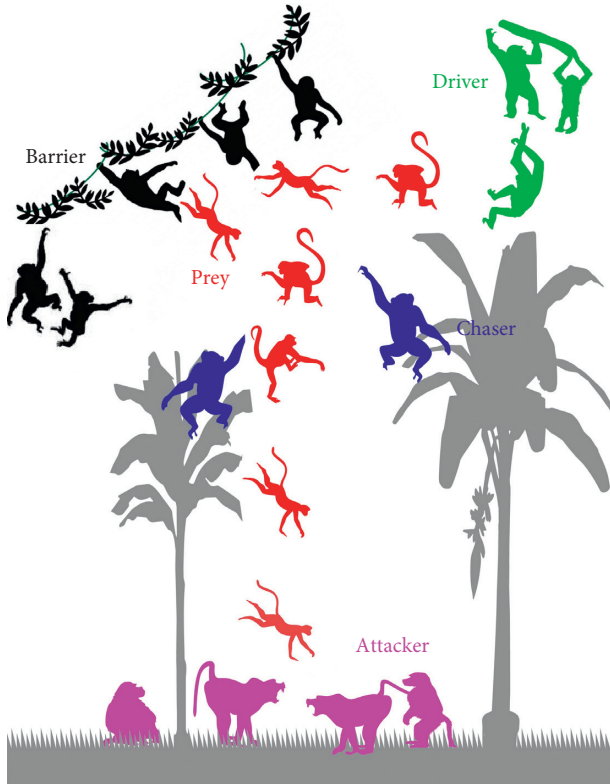


FIGURE 2: Demonstration of the hunting strategy [57].

the hunting procedure [57]. Six chaotic maps have been used in [57] as shown in Table 2 and Figure 3.

The Gauss/mouse map has been used in this study, since it has the most effect on the global minima findings and convergence speed.

There are two strategies for modeling the attacking behavior including blocking and encircling a prey. The hunting procedure can be typically done by attacker chimps. The chimps by driving, blocking, and chasing are able to discover the prey’s location and then encircle it. It is assumed that the first chaser, barrier, driver, and attacker are better notified about the position of a potential prey, to mathematically mimic the behavior of the chimps. Also, the first chaser, barrier, driver, and attacker are considered the best available solutions. Therefore, the positions of other chimps are updated based on the best chimps’ positions (best solutions yet obtained). This process is demonstrated in Figure 4.

In ChOA, when the attacker attacks the prey, the aforementioned operators are used to update the location of all chimps (i.e., driver, chaser, barrier, and the attacker) based on their current locations. However, the ChOA strategy may trap in minimum local, so other operators need to prevent this problem. As the exploration procedure between the chimps can be implemented by considering the position of driver, chaser, barrier, and attacker chimps, all members diverge for seeking preys and converge to attack preys. This divergence manner means that the vector  $a$  with some random values smaller than  $-1$  or higher than  $1$  can be employed, so that all variables (members) are forced to diverge and be away from preys. This

divergence strategy permits the ChOA to search globally and indicates the exploration procedure.

For mathematical modeling of the attacking practice, the value of  $f$  should be reduced. It is worth mentioning that the value of  $a$  is also reduced since it is dependent to  $f$ . In other words,  $a$  is a random value in  $[-2f, 2f]$ , and as the value of  $f$  is reducing from  $2.5$  to  $0$  in the period of iterations, the value of  $a$  is also reduced. In particular, when the value of  $a$  is in the range  $[-1, 1]$ , the next position of a chimp can be anywhere between its current position and the prey position. Figure 5 shows the inequality that forces the chimp to attack prey.

This divergence manner means that the vector  $a$  with some random values smaller than  $-1$  or higher than  $1$  can be employed, so that all variables (members) are forced to diverge and be away from preys. This divergence strategy permits the ChOA to search globally and indicates the exploration procedure.

To conclude, the procedure of searching preys in ChOA is initiated by creating some candidate solutions (a stochastic population of chimps). Next, all members of group are randomly separated into four self-regulating sets entitled driver, chaser, barrier, and attacker. Every group member updates its  $f$  coefficients utilizing the group technique. Throughout the repetition epoch, driver, chaser barrier, and attacker chimps approximate the potential prey positions. For every group member, solutions update their distance from the prey. Adjusting of  $a$  and  $c$  vectors is to accelerate convergence while also avoiding getting into local optima.

#### 4. Design of the Proposed Technique

As mentioned before, in this study, we propose to train a CNN model to recognize the Arabic and Persian words. This CNN model is optimized using a metaheuristic approach to obtain the best possible biases and weights for increasing the model efficiency.

In order to train CNNs, the final aim and the main problem need to be formulated mathematically in an appropriate way for any optimization approach. The significant variables of a CNN pipeline that lead to define the accuracy of the model are biases and weights. As biases and weights are the trainable variables, by altering these values of neurons in all convolutional layers, the output outcomes of the pipeline can have different results. Consequently, by directing the procedure of employing new biases and weights using an optimizer strategy, we can obtain higher accuracy. As the procedure of the ChOA accepts all input variables in the format of a vector, the variables of the CNN indicated for this approach can be given by

$$\vec{V} = \{ \vec{W}, \vec{\theta} \} = \{ \theta_1, \theta_2, \dots, \theta_h, W_{1,1}, W_{1,2}, \dots, W_{n,n} \}, \quad (4)$$

where  $\theta_j$  implies the bias (threshold) of the  $j$ th hidden node,  $n$  is the dimension of the input sample, and  $W_{ij}$  indicates the connection weight between  $i$ th and  $j$ th layers.

TABLE 2: Chaotic maps.

No	Name	Chaotic map	Range
1	Quadratic	$x_{i+1} = x_i^2 - c, c = 1$	(0, 1)
2	Gauss/mouse	$x_{i+1} = \begin{cases} 1x_i = 0 \\ 1/\text{mod}(x_i, 1) \text{ otherwise} \end{cases}$	(0, 1)
3	Logistic	$x_{i+1} = \alpha x_i (1 - x_i), \alpha = 4$	(0, 1)
4	Singer	$x_{i+1} = \mu (7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.302875x_i^4), \mu = 1.07$	(0, 1)
5	Bernoulli	$x_{i+1} = 2x_i \pmod{1}$	(0, 1)
6	Tent	$x_{i+1} = \begin{cases} x_i/0.7x_i < 0.7 \\ 10/3(1 - x_i)0.7 \leq x_i \end{cases}$	(0, 1)

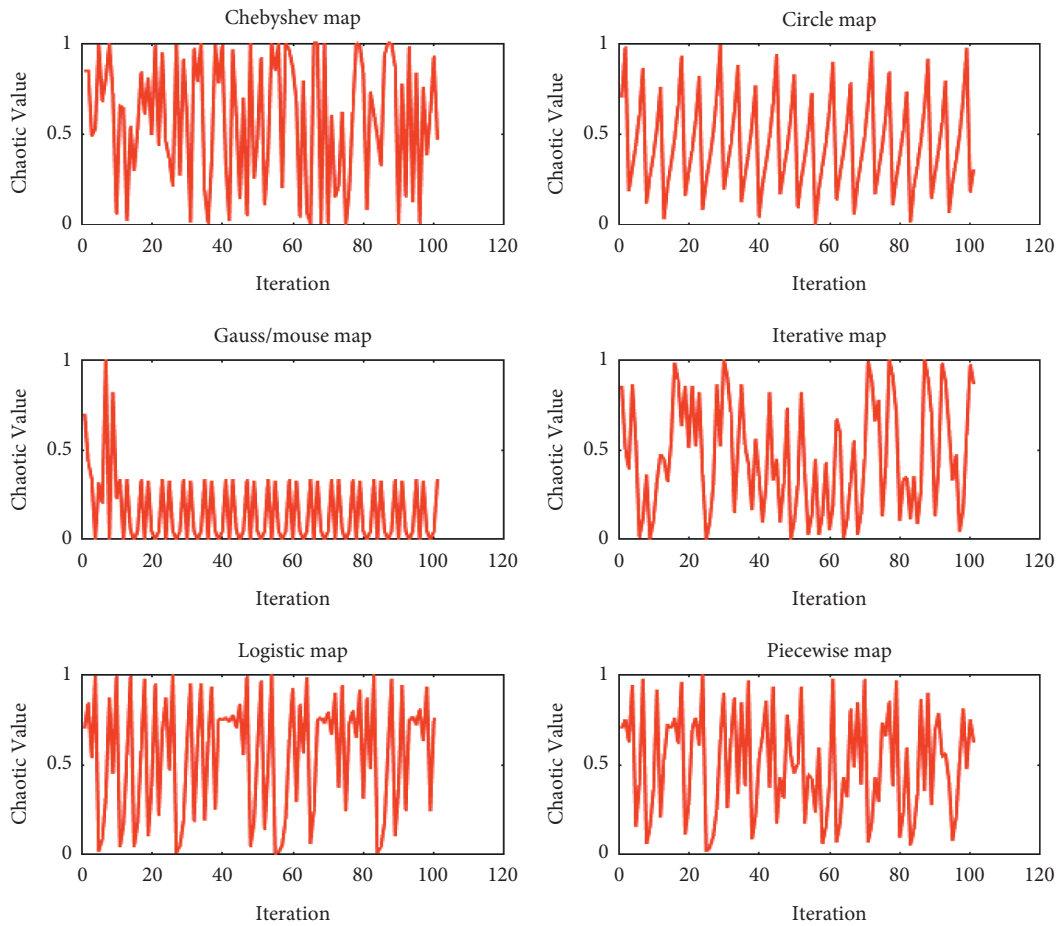


FIGURE 3: The chaotic map diagrams.

In the next step, we need to define the objective function for ChOA strategy. To implement effective CNNs, these networks need to be adjusted to the whole set of training data. To this end, the performance of a CNN can be evaluated using the average of mean squared error (MSE) with respect to the training data. By applying a set of training data (2D images or patches) to the CNN model (LeNet-5 in this study), we can measure the difference between the desirable values and the achieved result values through the following equation:

$$\text{MSE} = \sum_{i=2}^m (o_i^k - d_i^k)^2, \quad (5)$$

in which  $m$  indicates the output values, the ideal output of the  $i$ th input unit when the  $k$ th training data are employed is considered as  $d_i^k$ , and the actual output of the  $i$ th input unit when the  $k$ th training samples applied to the input is considered as  $o_i^k$ .

As at each iteration, the biases and weights accept the best possible values, the probability of an enhanced model

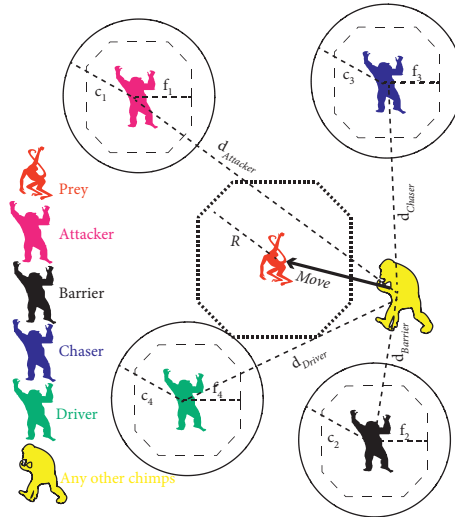


FIGURE 4: The procedure of position update [57].

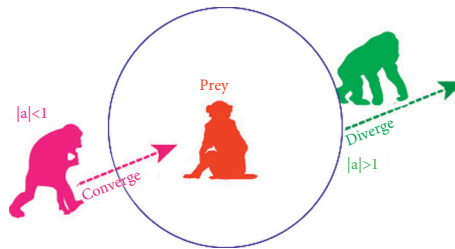


FIGURE 5: Position updating mechanism of chimps and effects of  $|a|$  on it [57].

increases progressively. However, due to stochastic nature of the ChOA strategy, there is no guarantee that the best CNN model is achieved. So, by employing adequate number of iterations, the ChOA strategy is able to reach a solution that works more efficient than a random preliminary solution.

Figure 6 shows the network training steps, using the proposed algorithm. The learnable parameters include all parameters both in convolution and in fully connected layers. The performance of the classical gradient descent (GD) algorithms strongly depends on the weight initialization while the random initialization can be trapped into a local optimum.

In this study, we use the general search property of the ChOA algorithm to find optimal weight initialization in order to achieve a more appropriate response. In the ChOA, the vector of the initial parameters is generated by the number of chimps. Here, in each iteration, the best answer is utilized in the CNN training phase. Thus, the probability of reaching the general optimization increases.

### 5. Experimental Results and Discussion

Standardized databases play a key role in all papers as they permit comparison of methods and repeatability. The suggested approach has been validated on two known and public datasets: ENIT/IFN (a dataset of Arabic handwritten

words) and Iranshahr (a dataset of Persian handwritten words) (see Figure 7).

The ENIT/IFN dataset contains 32,492 word-images of Tunisian village and town names and includes five subsets, namely, a, b, c, d, and e. In order to write the vocabulary, more than 1000 writers were employed and this vocabulary entails 946 unique village and city names [58, 59]. The known Iranshahr dataset includes nearly 17,000 images of handwritten names of 503 cities of Iran [60–62].

In the training, validation, and testing phases for both datasets, all images in the datasets were divided randomly into three groups: 70% of images for training, 15% of images for validation, and the remaining 15% for testing.

The parameters used in the ChOA algorithm and the other algorithms are shown in Table 3. It is noteworthy that these parameters were selected in an experimental process in an initial study. The results showed that these values are more suitable for the algorithms studied and compared in this research.

A Hewlett-Packard (HP) computer was utilized to evaluate our approach with the 64 bit operating system and Windows 10 Home, installed memory (RAM) of 8.00 GB, and Intel (R) Core (TM) i7-6500U CPU. For data processing, the MATLAB and Statistics Toolbox Release 2021a were employed.

To obtain the outcomes, two datasets were trained and tested 40 epochs by utilizing each strategy. The number of



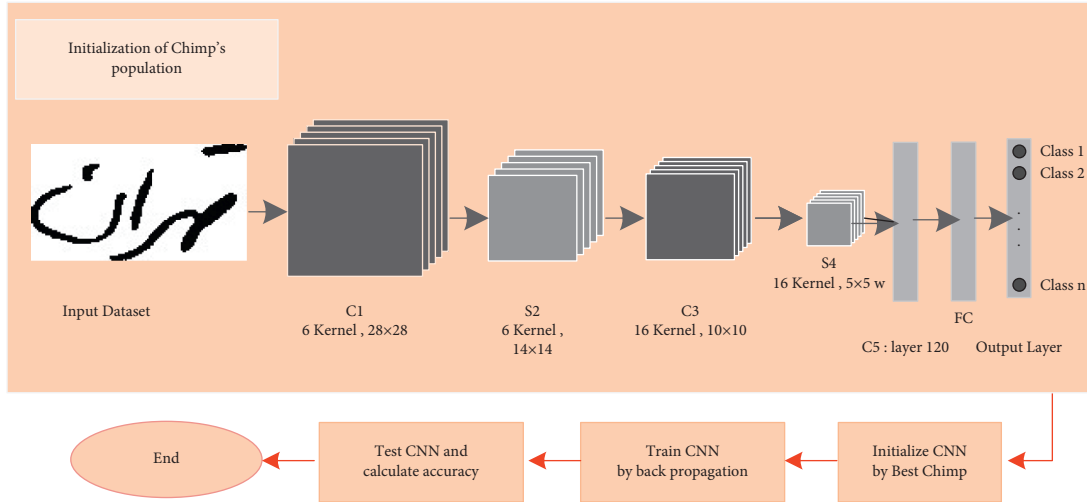


FIGURE 6: Handwritten recognition using CNN and ChOA.

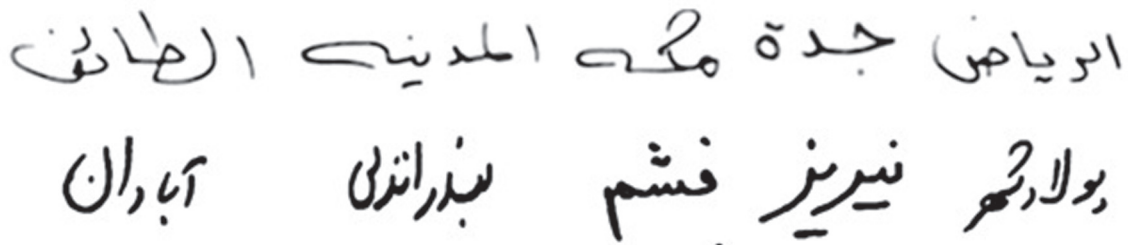


FIGURE 7: Some samples from the Arabic (first row) and the Persian (second row) datasets.

TABLE 3: The initial parameters of different techniques.

Algorithm	Number of populations	Epochs	Other parameters
FA	nAnts:10	40	$\alpha = 0.2, \beta = 2, Y = 1$
PSO	nParticles:10	40	$C_1 = 2, C_2 = 2, \omega = 0.7$
ACO	nAnts:10	40	$q = 0.5, \zeta = 1$
ChOA	nChimps:10	40	$f, c, d$ : according to Section 3 and $m = \text{Gauss/mouse}$

iterations in each period varies depending on the size of the images and the number of batches. Figure 8 shows that the final losses of all methods are similar, and Figure 9 depicts the first fifty iterations. As clearly illustrated in Figure 8, among these strategies, the ChOA technique converges more rapidly in the initial iterations. So, it can be the most efficient optimizer and can achieve higher accuracy.

Table 4 shows the CNN classifier performance after being trained with the GD and ChOA, separately, and also, with the GD and ChOA altogether. It can be seen that weight initialization using ChOA for the CNN classifier and training the classifier using the GD results in better performance. Furthermore, if the CNN classifier uses random initial weights and is trained with GD, the performance will

be better than the case where the CNN classifier only uses ChOA for initialization without training.

For each result in Tables 5 and 6, the best values of accuracy are highlighted in bold. According to Tables 5 and 6, we observe that optimized LeNet-5 using the ChOA method achieves the highest accuracy for both datasets. Moreover, the NN-based methods in [61, 64] gained the worst results among all methods for Persian and Arabic datasets, respectively. Also, for Arabic dataset, there is a minimum difference between the obtained accuracy by employing SIFT [65] and NN-based method [64]. For Persian dataset, there is a minimum difference between the achieved accuracy by utilizing the NN-based method [60] and LeNet-5.

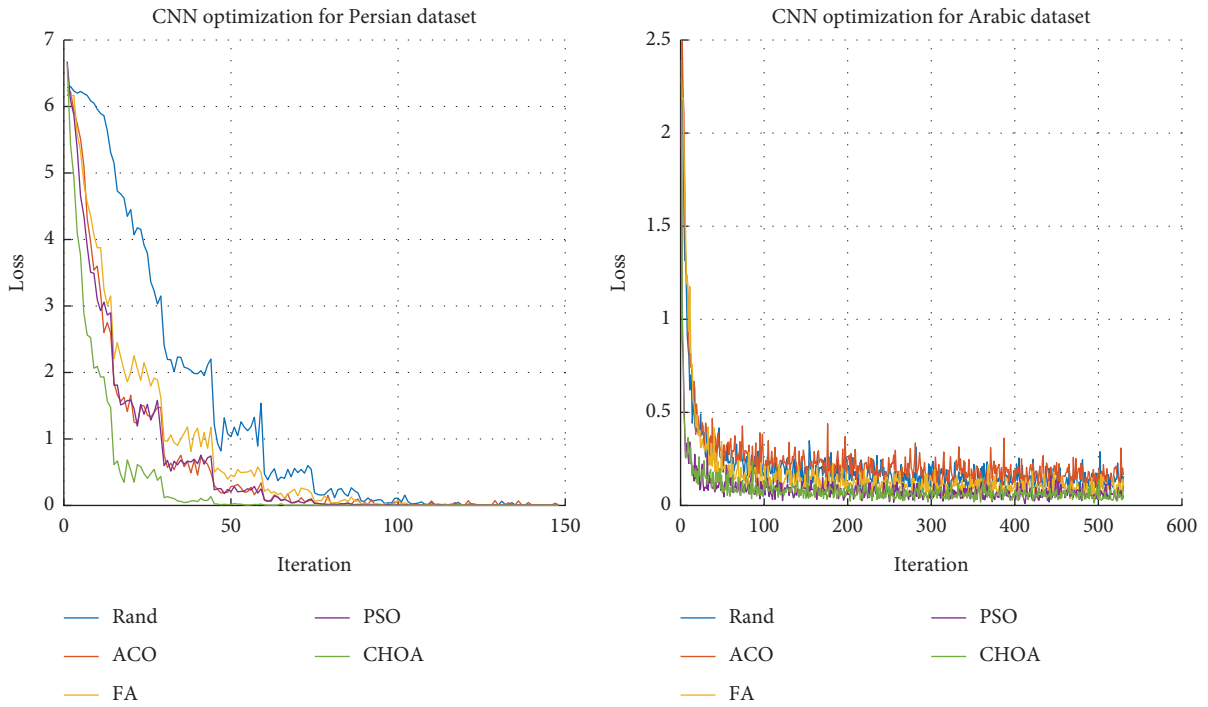


FIGURE 8: Error of classification employing two datasets and four optimization techniques for LeNet-5 for all iterations.

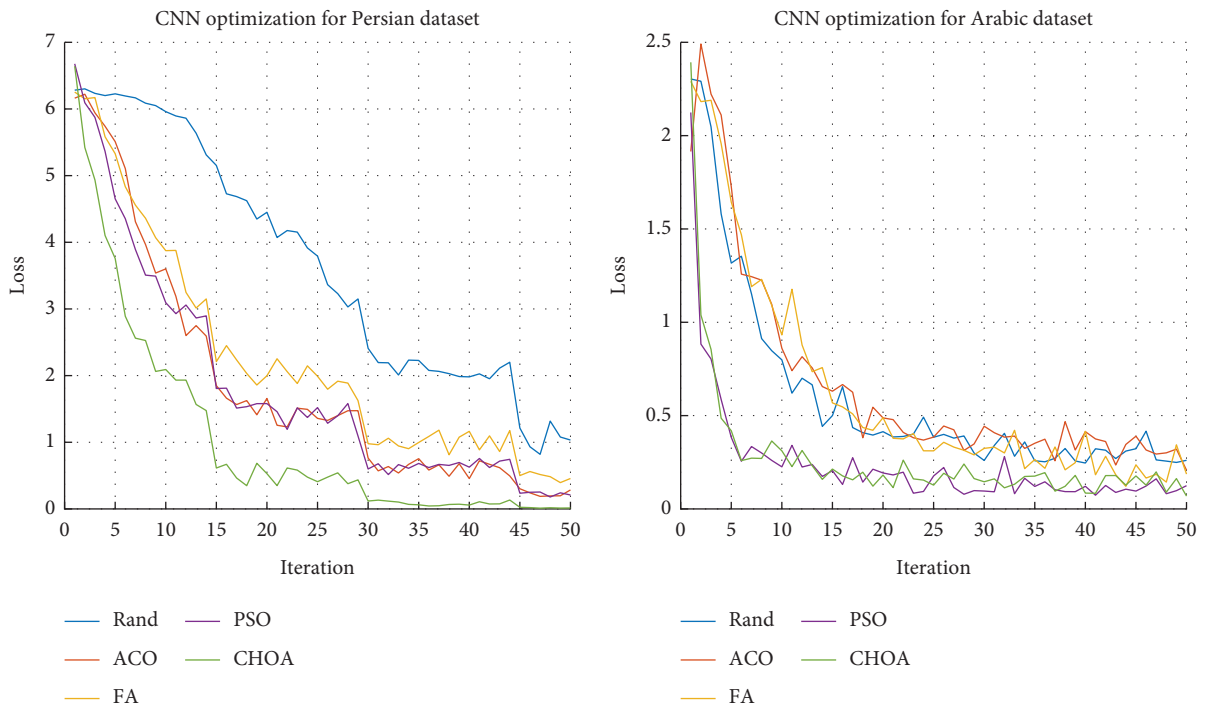


FIGURE 9: Error of classification employing two datasets and four optimization techniques for LeNet-5 for the first fifty iterations.

TABLE 4: Performance measure of CNN trained with the ChOA and GD algorithms.

Datasets	Initialization technique	Training technique	Accuracy (%)
Iranshahr	ChOA	—	72.01
	Random	GD	89.32
	<b>ChOA</b>	<b>GD</b>	<b>99.17</b>
ENIT/IFN	ChOA	—	69.84
	Random	GD	91.06
	<b>ChOA</b>	<b>GD</b>	<b>97.36</b>

TABLE 5: Experimental outcomes for the Persian dataset.

Technique	Recognition rate (%)
Recurrent neural network [63]	90.3
Deep learning [1]	98.99
NN-based method [61]	89.06
LeNet-5	89.32
FA-LeNet-5	93.46
ACO-LeNet-5	95.74
PCO-LeNet-5	96.34
ChOA-LeNet-5	<b>99.17</b>

TABLE 6: Experimental outcomes for the Arabic dataset.

Technique	Recognition rate (%)
External region features [59]	92.64
NN-based method [64]	90.10
SIFT [65]	90.61
LeNet-5	91.06
FA-LeNet-5	93.11
ACO-LeNet-5	94.07
PSO-LeNet-5	94.55
ChOA-LeNet-5	<b>97.36</b>

## 6. Conclusion

This study demonstrated the effect of applying the ChOA as one of metaheuristic strategies to optimize the LeNet-5 model. At first, the training issue of the CNN was formulated for the ChOA approach. Then, the ChOA technique was employed for defining the best values for weights and biases to increase the performance of the CNN model. The suggested ChOA approach was utilized for training two public datasets (handwritten Arabic, ENIT/IFN, and handwritten Persian, Iranshahr). In order to prove the accuracy of the ChOA technique, the outcomes were compared to three recently published methods and three other stochastic optimization approaches (ACO, PSO, and FA). The outcomes represented that the ChOA strategy can effectively train the CNN. This approach develops the probability of finding optimal values for weights and biases in a CNN model.

## Data Availability

The datasets used in this study are public datasets and their download links are <http://www.ifnenit.com/download.htm> and <https://drive.google.com/u/0/uc?id=1oF5S-BWsh5MU2SZOmdxmKI-pdSSwcpek&export=download>.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] M. Bonyani, S. Jahangard, and M. Daneshmand, "Persian handwritten digit, character and word recognition using deep learning," *International Journal on Document Analysis and Recognition*, vol. 24, pp. 1–11, 2021.
- [2] A. Chaudhuri, K. Mandaviya, P. Badelia, and S. K. Ghosh, "Optical character recognition systems," in *Optical Character Recognition Systems for Different Languages with Soft Computing*, Springer, Berlin, Germany, 2017.
- [3] S. Long, X. He, and C. Yao, "Scene text detection and recognition: the deep learning era," *International Journal of Computer Vision*, vol. 129, no. 1, pp. 161–184, 2021.
- [4] Z. Akhtar, J. W. Lee, M. A. Khan, M. Sharif, S. A. Khan, and N. Riaz, "Optical character recognition (OCR) using partial least square (PLS) based feature reduction: an application to artificial intelligence for biometric identification," *Journal of Enterprise Information Management*, vol. 33, 2020.
- [5] M. Asadi and A. Chalechale, "Emergency vehicles recognition based on deep learning for driver-less cars," *Journal of Machine Vision and Image Processing*, vol. 9, no. 2, pp. 19–32, 2022.
- [6] H. Heidari and A. Chalechale, "Biometric authentication using a deep learning approach based on different level fusion of finger knuckle print and fingernail," *Expert Systems with Applications*, vol. 191, Article ID 116278, 2022.
- [7] A. G. Mullissa, D. Marcos, D. Tuia, M. Herold, and J. Reiche, "Despecknet: generalizing deep learning-based SAR image despeckling," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, 2020.
- [8] S. Jafarzadeh Ghouschi, R. Ramin, A. N. Saeed, O. Elnaz, and B. T. Erfan, "An extended approach to the diagnosis of tumour location in breast cancer using deep learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1–11, 2021.
- [9] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, IEEE, Olomouc, Czech Republic, December 2013.
- [10] H. M. Fayek, M. Lech, and L. Cavedon, "Evaluating deep learning architectures for speech emotion recognition," *Neural Networks*, vol. 92, pp. 60–68, 2017.
- [11] T. Zhou, H. Lu, Z. Yang, S. Qiu, B. Huo, and Y. Dong, "The ensemble deep learning model for novel COVID-19 on CT images," *Applied Soft Computing*, vol. 98, Article ID 106885, 2021.
- [12] N. Coudray, P. S. Ocampo, T. Sakellaropoulos et al., "Classification and mutation prediction from non-small cell lung cancer histopathology images using deep learning," *Nature Medicine*, vol. 24, no. 10, pp. 1559–1567, 2018.
- [13] R. Ranjbarzadeh, D. Shadi, J. G. Saeid et al., "Nerve optic segmentation in CT images using a deep learning model and a

- texture descriptor,” *Complex & Intelligent Systems*, vol. 8, pp. 1–15, 2022.
- [14] S. Izadi, M. Ahmadi, and A. Rajabzadeh, “Network traffic classification using deep learning networks and Bayesian data fusion,” *Journal of Network and Systems Management*, vol. 30, no. 2, 2022.
- [15] R. Atefinia and M. Ahmadi, “Network intrusion detection using multi-architectural modular deep neural network,” *The Journal of Supercomputing*, vol. 77, no. 4, pp. 3571–3593, 2021.
- [16] K. Fukushima, “Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.
- [17] S. Di Cataldo and E. Ficarra, “Mining textural knowledge in biological images: applications, methods and trends,” *Computational and Structural Biotechnology Journal*, vol. 15, pp. 56–67, 2017.
- [18] D. Soydaner, “A comparison of optimization algorithms for deep learning,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 13, Article ID 2052013, 2020.
- [19] S. Reddi, M. Zaheer, D. Sachan, S. Kale, and S. Kumar, “Adaptive methods for nonconvex optimization,” in *Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS 2018)*, 2018.
- [20] H. Huang, C. Wang, and B. Dong, “Nostalgic Adam: weighting more of the past gradients when designing the adaptive learning rate,” 2018, <https://arxiv.org/abs/1805.07557>.
- [21] F. Altunbey Ozbay and B. Alatas, “A novel approach for detection of fake news on social media using metaheuristic optimization algorithms,” *Elektronika ir Elektrotechnika*, vol. 25, no. 4, pp. 62–67, 2019.
- [22] K. N. Reid, J. Li, A. Brownlee et al., “A hybrid metaheuristic approach to a real world employee scheduling problem,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2019.
- [23] N. Razmjooy, M. Ashourian, and Z. Foroozandeh, *Metaheuristics and Optimization in Computer and Electrical Engineering*, Springer, Berlin, Germany, 2020.
- [24] N. A. Kallioras, N. D. Lagaros, and D. N. Avtzis, “Pity beetle algorithm - a new metaheuristic inspired by the behavior of bark beetles,” *Advances in Engineering Software*, vol. 121, pp. 147–166, 2018.
- [25] N. Hansen, S. D. Müller, and P. Koumoutsakos, “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES),” *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.
- [26] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [27] X. Yao, Y. Liu, and G. Lin, “Evolutionary programming made faster,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [28] L. Abualigah, M. Shehab, M. Alshinwan, S. Mirjalili, and M. A. Elaziz, “Ant lion optimizer: a comprehensive survey of its variants and applications,” *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1397–1416, 2021.
- [29] A. Kaveh and N. Farhoudi, “A new optimization method: Dolphin echolocation,” *Advances in Engineering Software*, vol. 59, pp. 53–70, 2013.
- [30] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, “Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems,” *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.
- [31] A. L. A. Bolaji, M. A. Al-Betar, M. A. Awadallah, A. T. Khader, and L. M. Abualigah, “A comprehensive review: Krill Herd algorithm (KH) and its applications,” *Applied Soft Computing*, vol. 49, pp. 437–446, 2016.
- [32] A. Kaveh and S. Talatahari, “A novel heuristic optimization method: charged system search,” *Acta Mechanica*, vol. 213, no. 3, pp. 267–289, 2010.
- [33] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, “Multi-verse optimizer: a nature-inspired algorithm for global optimization,” *Neural Computing & Applications*, vol. 27, no. 2, pp. 495–513, 2016.
- [34] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, “GSA: a gravitational search algorithm,” *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
- [35] S. Mahdinia, M. Rezaie, M. Elveny, N. Ghadimi, and N. Razmjooy, “Optimization of PEMFC model parameters using meta-heuristics,” *Sustainability*, vol. 13, no. 22, Article ID 12771, 2021.
- [36] E. Atashpaz-Gargari and C. Lucas, “Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, Singapore, September 2007.
- [37] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, “Metaheuristic research: a comprehensive survey,” *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2191–2233, 2019.
- [38] L. Zhang, H. J. Gao, J. Zhang, and B. Badami, “Optimization of the convolutional neural networks for automatic detection of skin cancer,” *Open Medicine*, vol. 15, no. 1, pp. 27–37, 2019.
- [39] Y.-C. Ho and D. L. Pepyne, “Simple explanation of the no free lunch theorem of optimization,” in *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*, IEEE, Orlando, FL, USA, December 2001.
- [40] S. P. Adam, S.-A. N. Alexandropoulos, P. M. Pardalos, and M. N. Vrahatis, *No Free Lunch Theorem: A Review*, Springer, Berlin, Germany, 2019.
- [41] M. Taghizadeh and A. Chalechale, “A comprehensive and systematic review on classical and deep learning based region proposal algorithms,” *Expert Systems with Applications*, vol. 189, Article ID 116105, 2022.
- [42] A. Mahmood, M. Bennamoun, S. An et al., “Deep learning for coral classification,” in *Handbook of Neural Computation* Elsevier, Amsterdam, Netherlands, 2017.
- [43] J. Wang, L. Yu, K. Lai, and X. Zhang, “Dimensional sentiment analysis using a regional CNN-LSTM model,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, August 2016.
- [44] Y. Liu, L. Jin, S. Zhang, C. Luo, and S. Zhang, “Curved scene text detection via transverse and longitudinal sequence connection,” *Pattern Recognition*, vol. 90, pp. 337–345, 2019.
- [45] M. Sarigul, B. M. Ozyildirim, and M. Avci, “Differential convolutional neural network,” *Neural Networks*, vol. 116, pp. 279–287, 2019.
- [46] H. Sindi, M. Nour, M. Rawa, Ş. Öztürk, and K. Polat, “Random fully connected layered 1D CNN for solving the Z-bus loss allocation problem,” *Measurement*, vol. 171, Article ID 108794, 2021.
- [47] W. Shi, Y. Gong, and J. Wang, “Improving CNN performance with min-max objective,” in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, New York, NY, USA, July 2016.

- [48] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: a survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.
- [49] M. R. Islam and A. Matin, "Detection of COVID 19 from CT image by the novel LeNet-5 CNN architecture," in *Proceedings of the 23rd International Conference on Computer and Information Technology (ICCIT)*, IEEE, Dhaka, Bangladesh, December 2020.
- [50] Y. Zhu, G. Li, R. Wang, S. Tang, H. Su, and K. Cao, "Intelligent fault diagnosis of hydraulic piston pump combining improved LeNet-5 and PSO hyperparameter optimization," *Applied Acoustics*, vol. 183, Article ID 108336, 2021.
- [51] 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.
- [52] M. Kayed, A. Anter, and H. Mohamed, "Classification of garments from fashion MNIST dataset using CNN LeNet-5 architecture," in *Proceedings of the International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)*, Aswan, Egypt, February 2020.
- [53] R. P. França, C. B. M. Ana, V. E. Vania, and R. Navid, "Using metaheuristics in discrete-event simulation," in *Metaheuristics and Optimization in Computer and Electrical Engineering*, Springer, Berlin, Germany, 2021.
- [54] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, Article ID 113609, 2021.
- [55] E. H. Houssein, M. M. Emam, and A. A. Ali, "An efficient multilevel thresholding segmentation method for thermography breast cancer imaging based on improved chimp optimization algorithm," *Expert Systems with Applications*, vol. 185, Article ID 115651, 2021.
- [56] M. Kaur, R. Kaur, N. Singh, and G. Dhiman, "Schoa: a newly fusion of sine and cosine with chimp optimization algorithm for hls of datapaths in digital filters and engineering applications," *Engineering with Computers*, vol. 37, pp. 1–29, 2021.
- [57] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm," *Expert Systems with Applications*, vol. 149, Article ID 113338, 2020.
- [58] J. Al Abodi and X. Li, "An effective approach to offline Arabic handwriting recognition," *Computers & Electrical Engineering*, vol. 40, no. 6, pp. 1883–1901, 2014.
- [59] U. Saeed, T. Muhammad, A. S. Alghamdi, and M. S. Alkathiri, "Automatic recognition of handwritten Arabic using maximally stable extremal region features," *Optical Engineering*, vol. 59, no. 5, Article ID 051405, 2020.
- [60] M. Akhlaghi and V. Ghods, "Farsi handwritten phone number recognition using deep learning," *SN Applied Sciences*, vol. 2, no. 3, pp. 1–10, 2020.
- [61] S. A. A. Arani, E. Kabir, and R. Ebrahimpour, "Handwritten Farsi word recognition using NN-based fusion of HMM classifiers with different types of features," *International Journal of Image and Graphics*, vol. 19, no. 1, Article ID 1950001, 2019.
- [62] E. Bayesteh, A. Ahmadifard, and H. Khosravi, "A lexicon reduction method based on clustering word images in offline Farsi handwritten word recognition systems," in *Proceedings of the 7th Iranian Conference on Machine Vision and Image Processing*, IEEE, Tehran, Iran, November 2011.
- [63] M. F. Y. Ghadikolaie, E. Kabir, and F. Razzazi, "Sub-word based offline handwritten farsi word recognition using recurrent neural network," *ETRI Journal*, vol. 38, no. 4, pp. 703–713, 2016.
- [64] C. Leila, K. Maamar, and C. Salim, "Combining neural networks for Arabic handwriting recognition," in *Proceedings of the 10th International Symposium on Programming and Systems*, IEEE, Algiers, Algeria, April 2011.
- [65] L. Chergui and M. Kef, "SIFT descriptors for Arabic handwriting recognition," *International Journal of Computational Vision and Robotics*, vol. 5, no. 4, pp. 441–461, 2015.

## Research Article

# Federated Learning Algorithms to Optimize the Client and Cost Selections

Ali Alferaidi,<sup>1</sup> Kusum Yadav,<sup>1</sup> Yasser Alharbi ,<sup>1</sup> Wattana Viriyasitavat,<sup>2</sup> Sandeep Kautish ,<sup>3</sup> and Gaurav Dhiman <sup>4</sup>

<sup>1</sup>College of Computer Science and Engineering, University of Ha'il, Ha'il, Saudi Arabia

<sup>2</sup>Department of Statistics, Chulalongkorn Business School, Faculty of Commerce and Accountancy, Thailand

<sup>3</sup>LBEF Campus, Kathmandu, Nepal

<sup>4</sup>Department of Computer Science and Engineering, Graphic Era Deemed to Be University, Dehradun, India

Correspondence should be addressed to Sandeep Kautish; [dr.skautish@gmail.com](mailto:dr.skautish@gmail.com)

Received 25 October 2021; Revised 4 December 2021; Accepted 18 March 2022; Published 1 April 2022

Academic Editor: Araz Darba

Copyright © 2022 Ali Alferaidi 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 recent years, federated learning has received widespread attention as a technology to solve the problem of data islands, and it has begun to be applied in fields such as finance, healthcare, and smart cities. The federated learning algorithm is systematically explained from three levels. First, federated learning is defined through the definition, architecture, classification of federated learning, and comparison with traditional distributed knowledge. Then, based on machine learning and deep learning, the current types of federated learning algorithms are classified, compared, and analyzed in-depth. Finally, the communication from the perspectives of cost, client selection, and aggregation method optimization, the federated learning optimization algorithms are classified. Finally, the current research status of federated learning is summarized. Finally, the three major problems and solutions of communication, system heterogeneity, and data heterogeneity faced by federated learning are proposed and expectations for the future.

## 1. Introduction

As digital technology enters a period of rapid development, technologies such as big data and artificial intelligence are ushering in explosive development [1, 2]. However, on the one hand, this has brought new opportunities for upgrading and reformation of traditional businesses [3–5]. On the other hand, it inevitably brings new challenges to data and network security, and the problem of data islands [6, 7] is one of the critical challenges. Vertically, the top giant companies in the industry monopolize a large amount of data and information, and it is often difficult for small companies to obtain this data, which leads to the continuous widening of the level and gap between enterprises. The occlusion and barrier of business make it difficult to realize the communication and integration of data and information, and joint modeling needs to overcome many obstacles.

Federated Learning provides answers in response to the pain mentioned above currently facing the artificial

intelligence industry. Federated learning is a concept first proposed by Google Research in 2016 [8–10]. This technology can complete joint modeling without data sharing. Specifically, the own data of each data owner (individual/enterprise/organization) will not leave the local area through the parameter exchange method under the encryption mechanism in the federal system [11].

That is, without violating data privacy laws and regulations jointly establish a global sharing model, and the built model only serves local targets in their respective regions [12]. Although there are some similarities between federated learning [13–15] and distributed machine learning [16], federated learning has its characteristics in terms of application fields, system design, and optimization algorithms. When the amount of data is enormous, and the required computing resources are high, distributed machine learning (such as parameter servers) has obvious advantages. It stores independently identically distributed (IID) data or model parameters in each distributed node. On the Internet, the central server mobilizes data and

computing resources to train the model jointly [17]. Due to the differences in clients' geographical and temporal distribution, federated learning often has to deal with nonindependent and identically distributed (non-IID) data. This paper combines the status quo of federated learning, stratifies the federated learning system, and sorts out the relevant results of federated learning [18].

To integrate data from multiple sources, the currently common practice is to use data preprocessing ETL (extract-transform-load) tools to move data from different sources to a relational database and deploy tasks with a tremendous amount of calculation to multiple machines improve computing efficiency and reduce task energy consumption.

## 2. Federated Learning

In 2016, Google Research Institute proposed a new artificial intelligence solution for federated learning when optimizing the keyboard input method for individual users. The scenario of federation learning is distributed multi-user  $(f_1, f_2, \dots, f_n)$ . Each user client has the current user data set  $(d_1, d_2, \dots, d_n)$ . Traditional deep learning collects these data together to get the summary data set  $d = u_1 \cup u_2 \cup \dots \cup u_n$ , and the training model  $M_{SUM}$  is obtained. Participating users train a model  $M_{FED}$  together in the federated learning method, while the user data  $d_i$  is kept locally and not externally transmitted. If there is a non-negative real number  $\delta$ , the model accuracy  $V_{FED}$  of  $M_{FED}$  and the model accuracy  $V_{SUM}$  of  $M_{SUM}$  satisfy the following inequality:

$$|V_{FED} - V_{SUM}| < \delta. \quad (1)$$

It is said that the federated learning algorithm achieves  $\delta$ -accuracy loss [4]. Federated learning allows the training model to have a certain degree of performance deviation but provides all participants data security and privacy protection. There are two commonly used frameworks for federated learning, one is the client-server architecture [8], and the other is the peer-to-peer network architecture [2]. In the client-server architecture, the training method of federated learning allows each data holder to train the model locally according to their conditions and rules, then aggregate the desensitization parameters to the central server for calculation, and then send back each data. The holder updates its local model until the global model is stable. When performing federated learning training in peer-to-peer network architecture, participants can communicate directly without using a third party, further improving security. Still, more computing operations are required for encryption and decryption [1–4]. Current research is more based on the framework of third-party servers. Therefore, this paper focuses on the federated learning process of the client-server architecture as shown in Figure 1.

## 3. Federated Learning Process of Client-Server Architecture

The federated learning system is often built of data holders and central servers on a physical level. The quantity of local data or the number of features available to one data holder

may not be sufficient to allow effective model training, necessitating the assistance of other data holder. The Federal Learning Center server functions similarly to a distributed machine learning server in that it gathers the gradients from each data bearer. It returns a new gradient after server-side aggregating procedures [5]. To ensure data privacy, the data holder's training on local data is limited to local data in a cooperative modeling process of federated learning. Following desensitization, the gradient created by iterations is utilized as interactive data rather than local data and transferred to a third party. The trustworthy server then waits for the aggregated parameters to be returned before updating the model [8]. The client-server architecture's federated learning mechanism is shown in Figure 2.

*Step 1. System initialization:* first, the central server sends the modeling task, seeking to participate in the client. Next, the client data holder proposes a joint modeling vision based on its own needs. After reaching an agreement with other cooperative data holders, the joint modeling vision is established, and each data holder enters the collaborative modeling process. Finally, the central server releases the initial parameters to the data holders.

*Step 2. Partial calculation:* after the joint modeling task is started and the system parameters are initialized, the data holders will be required to perform local calculations locally based on their data. After the analysis is completed, the gradients obtained from the local calculations are desensitized and uploaded in the global model.

*Step 3. Center aggregation:* after receiving the calculation results from multiple data holders, the central server aggregates these calculated values. In the aggregation process, it is necessary to consider various issues such as efficiency, security, and privacy at the same time. For example, sometimes, due to the system's heterogeneity, the central server may not wait for all data holders to upload but select a suitable subset of data holders as the collection target. To aggregate parameters safely, use certain encryption techniques to encrypt the parameters.

*Step 4. Model update:* the central server updates the global model once based on the aggregation result and returns the updated model to the data holders participating in the modeling. The data holder updates the local model and starts the following partial calculation, and at the same time evaluates the performance of the updated model. When the performance is good enough, the training is terminated, and the joint modeling ends. The established global model will be kept on the central server for subsequent prediction or classification work.

The above process is a typical federated learning process based on client-server architecture. However, not every federal learning task must strictly follow this process. Sometimes the process may be changed for different

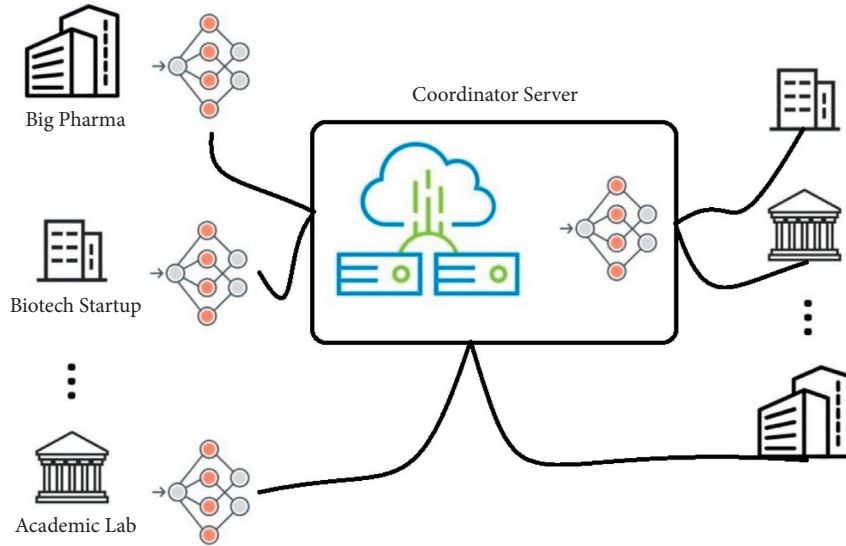


FIGURE 1: Structure of federated learning.

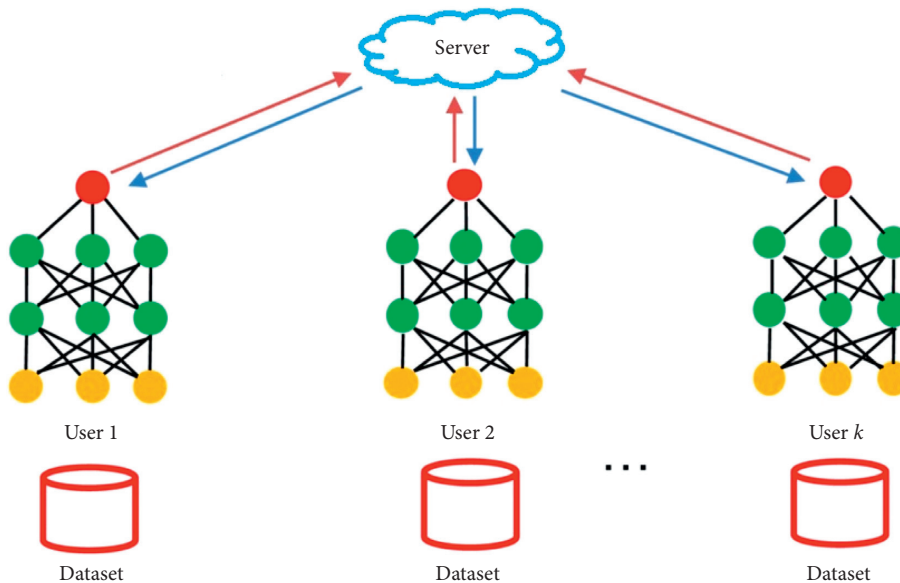


FIGURE 2: Federated learning process of client-server architecture [17].

scenarios. For example, the communication frequency is appropriately reduced to ensure learning efficiency [6, 7], or in after aggregation, a logical judgment is added to judge the quality of the received local calculation results [2–5] to improve the robustness of the federated learning system [19].

#### 4. Federated Vs Traditional Distributed Learning

Both federated learning and distributed machine learning based on the client-server architecture [8, 9, 20] are used to process distributed data. Still, they are different from distributed machine learning in terms of application fields, data attributes, and system components.

**4.1. Application Areas.** Large amounts of data or larger models often have higher requirements for computing resources. A single computing node can no longer meet the demand [10, 12]. Distributed machine learning distributes training data or model parameters to various computing or storage nodes and uses a central server to schedule nodes to accelerate model training. When the data has privacy-sensitive attributes, the central scheduling of distributed machine learning will bring a significant risk of privacy leakage to user data [21]. Federated learning permanently stores data locally, guarantee data privacy to the utmost extent compared to how information needs to be uploaded to the server.

**4.2. Data Attributes.** The primary purpose of machine learning is to find the probability distribution of data, which



is relatively easy when the data set satisfies independent and identical distribution. However, the data processed by distributed machine learning and classic machine learning are often disconnected and identically distributed, while federated learning is different. Due to the differences in the client's geographical location and time distribution, the raw data of the federated learning system is often non-independent and identically distributed. At the same time, horizontal federated learning and vertical federated learning are also classified according to different attributes of client data. Therefore, the clients' data characteristics and classification labels are quite different, and alignment work is required during training.

**4.3. System Composition.** In terms of physical composition, the federated learning system is similar to the distributed system, and both consist of a central server and multiple distributed nodes. In a distributed system, data calculation and model update is uniformly scheduled by the central server, and the data delay between nodes and the central server is relatively small. The model training time is mainly determined by the calculation time. In the federal system, each participant has equal status and can independently decide whether to participate in model training. And since distributed nodes are mostly clients with significant differences in computing capabilities, different network environments, and uncontrollable states, in system design, it is necessary to consider many factors such as data transmission delay, data nonindependent and identical distribution, and privacy and security. This requires the system to make adaptive changes to the federated learning algorithm [12–15]. Federated aggregation is an optimization algorithm that is different from distributed machine learning in a federated learning system. It provides new ideas for solving data nonindependent and identical distribution and reducing data heterogeneity. At the same time, because federated learning has excellent privacy protection capabilities, attention must be paid to applying encryption algorithms in all aspects of the system. The transmission of encrypted data, the calculation of the objective function loss, the analysis of the gradient and the information of the parameters of the transfer model all put forward new requirements for the traditional algorithms.

## 5. Federated Learning Classification

The island data of federated learning has different distribution characteristics. For each participant, the data they own can be represented by a matrix. Each row of the matrix represents each user or an independent research object, and each column represents a characteristic of the user or research object. At the same time, each row of data will have a label. People hope to learn a model to predict his label  $Y$  through his feature  $X$  for each user.

In reality, different participants may be different companies or institutions. Moreover, people don't want their data to be known by others. Still, people hope to jointly train a more robust model to predict the label  $Y$ . According to the

data characteristics of federated learning (that is, the degree of data overlap between different participants), federated learning can be divided into horizontal federated learning [13], vertical federated learning [14], and transfer federated knowledge [15].

When the user overlap of the two participants is minimal, but the user characteristics of the two data sets overlap more, the federated learning in this scenario is called horizontal federated learning. For example, the branches of a banking system in Shenzhen and Shanghai are the participants. The businesses on both sides are similar, and the characteristics of the collected user data are relatively identical. However, most of the users in the two branches are residents, and the user overlap is relatively small. Therefore, when a federated model is needed to classify users, it belongs to horizontal federated learning.

On the other hand, when there is a lot of overlap between the users of the two participants. But the overlap of the user characteristics of the two data sets is relatively tiny. Federated learning in this scenario is called vertical federated learning. For example, two institutions in the same area, one institution has user's consumption records; the other has user's bank records. The two institutions have many overlapping users, but the recorded data characteristics are different. The two institutions want to aggregate through different encryption characteristics of users to train a more powerful federated learning model jointly. This type of machine learning model belongs to longitudinal federated learning.

When the user overlap of the two participants is minimal, the user feature overlap of the two data sets is also relatively small, and some data still have missing labels. The federated learning in this scenario is called transfer federated learning. For example, two institutions in different regions have user consumption records in the area, and the other has bank records in the area. The two institutions have different users, and the data characteristics are also different. In this case, joint the trained machine learning model is transfer federated learning.

Most of the current researches are based on horizontal federated learning and vertical federated learning. However, there is not much research in the field of transfer federated learning for the time being. Therefore, this article will focus on the algorithm types of horizontal federated learning and vertical federated learning. In horizontal federated learning, there are many overlapping dimensions of data features. The alignment is performed according to the overlapping dimensions, and the parts of the participant data with the same features but not the same users are taken out for joint training; vertical federated learning users have many overlapping dimensions, matching according to user IDs, and removing The part of the participant data that has the same user but not the same characteristics is jointly trained.

## 6. Features of Federated Learning Algorithm

Based on the above introduction to federated learning, the characteristics of the following federated learning algorithms are summarized.

**6.1. Support Nonindependent and Identically Distributed Data.** This is an essential feature of federated learning algorithms. The federated learning algorithm must perform well in nonindependent and identically distributed data. In the actual use of federated learning, the data quality and distribution of the data holder is uncontrollable. The data holder's data cannot be required to meet independent and identical distribution [16], so the federated learning algorithm needs to support nonindependent and identically distributed data.

**6.2. Communication Efficiency.** The federated learning algorithm needs to consider the data holder's system heterogeneity, improve communication efficiency, and reduce communication loss without losing accuracy or loss.

**6.3. Fast Convergence.** In the process of joint modeling, it is first necessary to ensure model convergence, and at the same time, it is essential to increase the convergence speed.

**6.4. Security and Privacy.** Data privacy security is an essential feature of federated learning, so security and privacy are requirements for federated gradient updates. Security and privacy can be carried out in the aggregation process through encryption and other methods and can also be reflected in stand-alone optimization.

**6.5. Support for Complex Users.** Complex users refer to many users themselves and the imbalance or deviation of user data. This is possible in the practical application of federated learning, and the federated optimization algorithm needs to have a good compatibility effect for this situation.

## 7. Optimized Classification Method of Federated Learning Algorithm

Compared with distributed learning, federated learning has some unique attributes, as follows:

- (a) The communication of federated learning is relatively slow and unstable.
- (b) Participants in federated learning have heterogeneous devices, and different devices have different computing capabilities.
- (c) Federated learning pays more attention to privacy and security. At present, most studies assume that the participants and the server are trustworthy. However, in real life, it may be untrustworthy.

In implementing federated learning, it is necessary to consider how to optimize the federated learning algorithm to solve the existing practical problems. This article will introduce algorithms for optimizing federated learning from communication cost, client selection, and asynchronous aggregation. Before submitting the optimization algorithm, first, present the most traditional federated learning algorithm as FedAvg algorithm.

The FedAvg algorithm [8] is currently the most commonly used federated learning optimization algorithm. Unlike the conventional optimization algorithm [7], its essential idea is to use the local stochastic gradient descent method to optimize the local model on the data holder and perform aggregation operations on the central server-side [9]. The goal function is defined as follows:

$$f(\omega^*) = \min \left\{ \frac{1}{m} \sum_{n=1}^m E[f(\omega: x, x\epsilon n)] \right\}. \quad (2)$$

Among them,  $M$  represents the number of data holders participating in the joint modeling;  $\omega$  represents the current parameters of the model and represents the mean square error function. The FedAvg algorithm is a relatively basic federated optimization algorithm. Its deployment is relatively simple, and its application field is vast [6].

## 8. Cost Optimized FDA

Machine learning algorithms, incredibly complex deep learning algorithms, need to train many parameters during the training process. For example, CNN may need to train millions of parameters, and each update process needs to update millions of parameters; secondly, network communication. The status may also lead to remarkably high Communication costs, such as unstable network conditions and inconsistent speeds during parameter upload and download, leading to high model training costs for the entire algorithm. Therefore, it is necessary to consider optimizing the federated learning algorithm from the perspective of communication cost according to these characteristics [8]. The communication cost can be reduced from the following views.

**8.1. Client Computing Cost.** In the federated learning system, sometimes terminal nodes will only participate in federated learning training when Wi-Fi is available, or sometimes the network is in poor condition. In these cases, more calculations can be performed locally, thereby reducing the number of communications. Many algorithms optimize communication costs from this perspective. For example, optimizing the FedAvg algorithm [2], increasing the number of calculations of the local update parameters of each client in each iteration, and it was carried out with the Fed-SGD algorithm in which each round of server parameter update only requires one local update of the client. In contrast, the experiment shows through the MINSTCNN model test that the algorithm can significantly reduce the communication cost when the data is IID. However, when the information is non-IID, the algorithm can only slightly reduce the communication cost. Fed-Prox algorithm [9] more obvious optimization effect when the data is non-IID because the data and computing power of the joint training terminal participants are not balanced, so each During a parameter update, if the number of operations that different participants have to participate in is uniform, the computing resources of the client will not be fully utilized. To avoid this situation and optimize communication efficiency, the Fed-

Prox algorithm can dynamically update the number of local calculations required by different clients in each round, making the algorithm more suitable for nonindependent and codistributed joint modeling scenarios. Federated optimization algorithm [10] and learn under the framework of vertical federated learning. FedAvg algorithm [12] and proved that the heterogeneity of the data would cause the convergence rate of federated learning to decrease.

**8.2. Model Compression.** Some optimization algorithms aim to reduce the number of parameters in each round of communication. For example, model compression techniques (such as quantization and subsampling) reduce the total number of parameters to be transmitted for each parameter update. Structured model [15] update round of parameter communication, the size of the model update parameters passed by the participants to the server reduces transmission. The structured update refers to uploading the model by defining the matrix structure of the uploaded model parameters in advance. The contour update means that the parameters of each update need to be compressed and encoded by the participants. The model is finally verified by the CIFAR-10 image algorithm. Experiments show that participation the more parties, the better the compression effect. Optimization of model parameter transfers from the server to the participants and reduced the data from the server to the client through loss compression and federated dropout. However, the number of parameters that need to be passed at the end and the cost of reducing communication costs is to reduce the model's accuracy to a certain extent.

When implementing federated learning, communication is a bottleneck. Reducing communication costs is a vital optimization link. Some optimizations are at the expense of increasing the local calculations of the participants, and some optimizations are to reduce the accuracy of the entire model. In the actual optimization process, you can decide which way to reduce the communication cost according to the actual situation and demand.

## 9. Client Optimized FDA

The client devices of federated learning are heterogeneous, and the resources of different clients are limited. Usually, the client randomly chooses to participate in the model training process of federated learning. Therefore, in the federated learning and training process, some algorithms will consider the angle selected from the client for optimization.

Different clients have different network speeds, computing power, etc., and the data distribution owned by each client is also unbalanced. If all clients are allowed to participate in the training process of federated learning, there will be participants with backward iterations. If some clients do not respond for a long time, the whole system may not complete the joint training. Therefore, it is necessary to consider how to choose the clients to participate in the training. The FedAvg algorithm randomly selects the clients participating in the training. However, when the network structure is complex, and the data is not independent and

identically distributed, the FedAvg algorithm model does not necessarily perform well. The following two references introduce some optimization schemes.

Fed-CS algorithm [19] is a greedy algorithm protocol mechanism to select the client with the highest model iteration efficiency for aggregation update in each update of joint training. Thus, optimize the convergence efficiency of the entire federated learning algorithm. Experiments show that the Fed-CS algorithm can achieve higher accuracy. Still, the disadvantage is that it can perform well only when the model is relatively basic, such as a virtual dynamic neural network. For the case where the network structure or the number of parameters is more complex, Fed-CS will reduce the efficiency of selecting the optimal aggregation client, increasing the number of communications and decreasing time efficiency.

Hybrid-FL protocol [20] algorithm can process client data whose data set is non-IID and solve the problem of poor performance based on non-IID data on the FedAvg algorithm. The Hybrid-FL protocol enables the server to select some clients through the steps of the resource request to establish an approximately independent and identically distributed data set locally for training and iteration of federated learning. Furthermore, they show through experiments that Hybrid-FL has better accuracy performance for non-IID data type federated learning classification algorithms.

## 10. Result Evaluation

A classification model's performance may be quantified in terms of True Positive and False Positive rates. The TP rate is the difference between the patterns recognized by a system and the total patterns detected by system. A simple mathematical representation is presented as follows.

$$\text{TPR} = \frac{t_p}{t_p + f_n}. \quad (3)$$

Similarly the True negative rate is shown in the following formula:

$$\text{TNR} = \frac{t_n}{t_p + t_n}. \quad (4)$$

False Positive rates occur when any system incorrectly classifies a normal outcome. The FP rate may be calculated in this experiment by dividing the total number of self-antigens by the number of false positives generated by the system.

$$\text{FPR} = \frac{f_p}{f_p + t_n}. \quad (5)$$

Similarly false negative rate and accuracy calculated by

$$\text{FNR} = \frac{f_n}{t_p + f_n}, \quad (6)$$

$$\text{accuracy} = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}.$$

Table 1 compares the simulation results; it compares the categorization accuracy rate for the old approach with the

TABLE 1: Comparison of the simulation result of federated learning algorithm.

Methods	TPR	TNR	FPR	FNR	Accuracy
Traditional algorithm	64.52	54.23	42.12	52.69	75.24
Fed-CS algorithm	74.25	64.21	40.28	45.24	80.45
Hybrid-FL	79.45	70.52	30.51	32.46	86.47
Fed-average algorithm	84.21	74.25	23.45	25.64	90.45

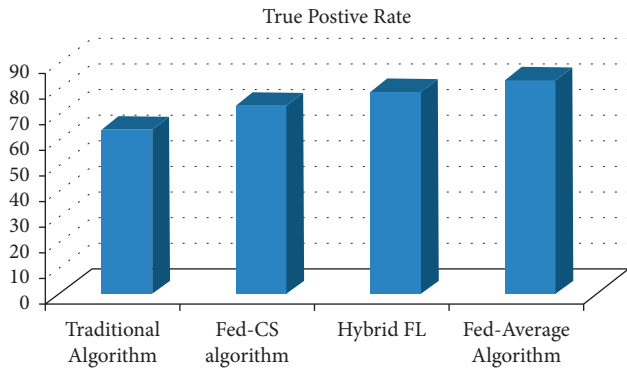


FIGURE 3: True positive rate.

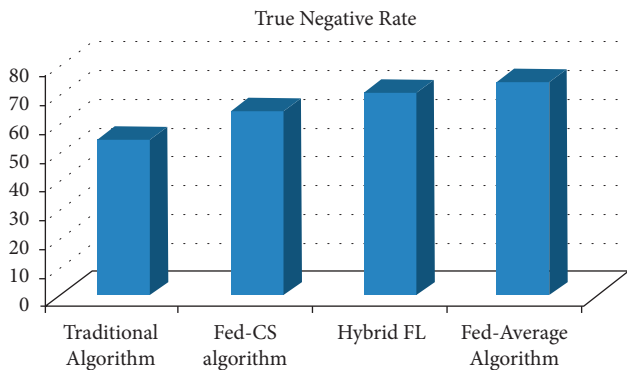


FIGURE 4: True negative rate.

FedAvg algorithm. The generating function, also known as the activation threshold value, was adjusted to the greatest accuracy rate attainable with our system utilising the FedAvg algorithm approach during simulation.

It compares the real positive rate for classification using the old technique to that utilising the FedAvg algorithm, as shown in Figure 3. In simulation, the generating function, also known as the activation threshold value, was set to the greatest TPR rate feasible with our approach.

As shown in Figure 4, it compares the genuine negative rate for categorization using the old technique to the FedAvg algorithm. In simulation, the generating function, also known as the activated threshold value, was set to the greatest TNR rate attainable with our technique.

It compares the false positive rate for classification using the old technique to the rate for classification using the FedAvg algorithm, as shown in Figure 5. In simulation, the generating function, also known as the activation threshold value, was set to the algorithm’s minimum FPR rate, which is only attainable using the FedAvg algorithm approach.

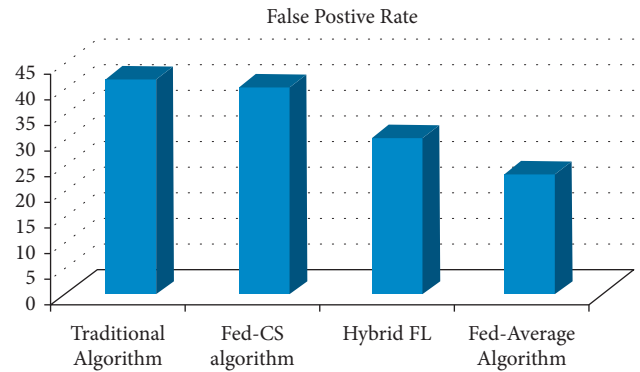


FIGURE 5: False positive rate.

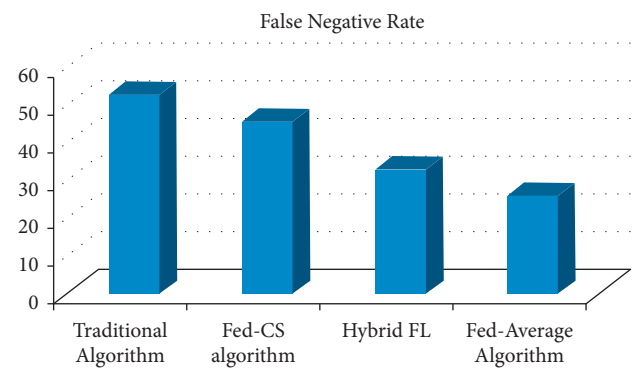


FIGURE 6: False negative rate.

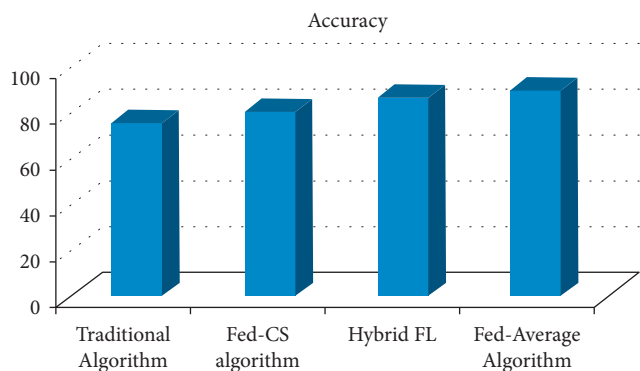


FIGURE 7: Accuracy.

It compares the false negative rate for classification using the old technique to the rate for classification using the FedAvg algorithm, as shown in Figure 6. In simulation, the generating function, also known as the activation threshold value, was set to the algorithm’s minimum FNR rate, which is only attainable using the FedAvg algorithm approach.

It compares the classification accuracy of the old technique to the FedAvg algorithm, as shown in Figure 7. The generating function, also known as the activation threshold value, was adjusted to the maximum accuracy feasible for our approach utilising the FedAvg algorithm technique during simulation.

## 11. Conclusion

This paper discusses the current development status of federated learning. From the perspective of federated learning algorithms, the federated learning-related algorithms are divided into federated optimization algorithms and federated machine learning algorithms. Related algorithms suitable for both central and decentralized federated learning structures are carried out. At the same time, the machine learning algorithm and the federated deep learning model under the federated learning framework will be summarized and discussed separately. In the process of federated algorithm optimization, the differences, advantages and disadvantages of existing federated optimization algorithms are discussed from the perspectives of reducing communication costs, selecting the best client, and optimizing model aggregation. Federated learning is still in a stage of rapid development. There has been a lot of research and discussion on the application of federated knowledge in practice, but there are still many problems and challenges in implementing federated learning. This work has been further extended to solve the real-life applications by using the recently developed optimization algorithms such as monarch butterfly optimization (MBO) [21], earthworm optimization algorithm (EWA) [22], elephant herding optimization (EHO) [23], moth search (MS) algorithm [24], Slime mould algorithm (SMA) [25], and Harris Hawks optimization (HHO) [26].

## Data Availability

The data used to support the findings of this study are available from the author upon request (gdhiman0001@hotmail.com).

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] M. Gharibi and P. Rao, "RefinedFed: a refining algorithm for federated learning," in *Proceedings of the 2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pp. 1–5, Washington DC, DC, USA, October 2020.
- [2] B. Gu, A. Xu, Z. Huo, C. Deng, and H. Huang, "Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning," *IEEE Transactions on Neural Networks and Learning Systems* pp. 1–13, 2021, In press.
- [3] A. R. Short, H. C. Leligou, and E. Theocharis, "Execution of a Federated Learning process within a smart contract," in *Proceedings of the 2021 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–4, Las Vegas, NV, USA, January 2021.
- [4] D. Jiang, C. Shan, and Z. Zhang, "Federated learning algorithm based on knowledge distillation," in *Proceedings of the 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, pp. 163–167, Beijing, China, October 2020.
- [5] X. Qu, J. Wang, and J. Xiao, "Quantization and knowledge distillation for efficient federated learning on edge devices," in *Proceedings of the 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 967–972, Yanuca Island, Cuvu, Fiji, December 2020.
- [6] S. Ek, F. Portet, P. Lalanda, and G. Vega, "A federated learning aggregation algorithm for pervasive computing: evaluation and comparison," in *Proceedings of the 2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–10, Kassel, Germany, March 2021.
- [7] A. Imteaj and M. Hadi Amini, "FedAR: activity and resource-aware federated learning model for distributed mobile robots," in *Proceedings of the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1153–1160, Miami, FL, USA, December 2020.
- [8] H. Kim, Y. Kim, and H. Park, "Reducing model cost based on the weights of each layer for federated learning clustering," in *Proceedings of the 2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 405–408, Jeju Island, Korea, August 2021.
- [9] S. K. Das and S. Beborrtta, "Heralding the future of federated learning framework: architecture, tools and future directions," in *Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 698–703, Noida, India, January 2021.
- [10] Y. Li, T.-H. Chang, and C.-Y. Chi, "Secure federated averaging algorithm with differential privacy," in *Proceedings of the 2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, Espoo, Finland, September 2020.
- [11] P. K. Vaishnav, S. Sharma, and P. Sharma, "Analytical review analysis for screening COVID-19 disease," *International Journal of Modern Research*, vol. 1, no. 1, pp. 22–29, 2021.
- [12] C. Chen, Z. Chen, Y. Zhou, and B. Kailkhura, "FedCluster: boosting the convergence of federated learning via cluster-cycling," in *Proceedings of the 2020 IEEE International Conference on Big Data (Big Data)*, pp. 5017–5026, Atlanta, GA, USA, December 2020.
- [13] D. Geng, H. He, X. Lan, and C. Liu, "An adaptive accuracy threshold aggregation strategy based on federated learning," in *Proceedings of the 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pp. 28–31, Nanchang, China, March 2021.
- [14] M. Servetnyk, C. C. Fung, and Z. Han, "Unsupervised federated learning for unbalanced data," in *Proceedings of the GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, Taipei, Taiwan, December 2020.
- [15] L. L. Pilla, "Optimal task assignment for heterogeneous federated learning devices," in *Proceedings of the 2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 661–670, Portland, OR, USA, May 2021.
- [16] X. Yao, T. Huang, C. Wu, R. Zhang, and L. Sun, "Towards faster and better federated learning: a feature fusion approach," in *Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP)*, pp. 175–179, Taipei, Taiwan, September 2019.
- [17] I. Chatterjee, "Patenting machine-learning: review and discussions," *International Journal of Modern Research*, vol. 1, no. 1, pp. 15–21, 2021.
- [18] V. K. Gupta, S. K. Shukla, and R. S. Rawat, "Crime tracking system and people's safety in India using machine learning approaches," *International Journal of Modern Research*, vol. 2, no. 1, pp. 1–7, 2022.

- [19] Y. Jin, L. Jiao, Z. Qian, S. Zhang, and S. Lu, "Learning for learning: predictive online control of federated learning with edge provisioning," in *Proceedings of the IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pp. 1–10, Vancouver, BC, Canada, May 2021.
- [20] S. S. Diwangkara and A. I. Kistijantoro, "Study of data imbalance and asynchronous aggregation algorithm on federated learning system," in *Proceedings of the 2020 International Conference on Information Technology Systems and Innovation (ICITSI)*, pp. 276–281, Bandung, Indonesia, October 2020.
- [21] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing and Applications*, vol. 31, no. 7, pp. 1995–2014, 2019.
- [22] G. G. Wang, S. Deb, and L. D. S. Coelho, "Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems," *International Journal of Bio-Inspired Computation*, vol. 12, no. 1, pp. 1–22, 2018.
- [23] G. G. Wang, S. Deb, and L. D. S. Coelho, "Elephant herding optimization," in *Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, pp. 1–5, IEEE, Bali, Indonesia, December 2015.
- [24] G.-G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2018.
- [25] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.
- [26] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

## Research Article

# Discrete Generalized Inverted Exponential Distribution: Case Study Color Image Segmentation

Mohamed Abd Elaziz <sup>1,2,3</sup> Nahla S. Abdelrahman <sup>2</sup> N. A. Hassan <sup>2</sup>  
and M. O. Mohamed <sup>2</sup>

<sup>1</sup>Faculty of Computer Science and Engineering, Galala University, Suez 435611, Egypt

<sup>2</sup>Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt

<sup>3</sup>Artificial Intelligence Research Center (AIRC), Ajman University, Ajman, P.O. Box 346, UAE

Correspondence should be addressed to Mohamed Abd Elaziz; [abd\\_el\\_aziz\\_m@yahoo.com](mailto:abd_el_aziz_m@yahoo.com)

Received 23 December 2021; Revised 26 January 2022; Accepted 2 February 2022; Published 23 March 2022

Academic Editor: Saeid Jafarzadeh Ghouschi

Copyright © 2022 Mohamed Abd Elaziz 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.

We present in this paper a discrete analogue of the continuous generalized inverted exponential distribution denoted by discrete generalized inverted exponential (DGIE) distribution. Since, it is cumbersome or difficult to measure a large number of observations in reality on a continuous scale in the area of reliability analysis. Yet, there are a number of discrete distributions in the literature; however, these distributions have certain difficulties in properly fitting a large amount of data in a variety of fields. The presented DGIE( $\beta, \theta$ ) has shown the efficiency in fitting data better than some existing distribution. In this study, some basic distributional properties, moments, probability function, reliability indices, characteristic function, and the order statistics of the new DGIE are discussed. Estimation of the parameters is illustrated using the moment's method as well as the maximum likelihood method. Simulations are used to show the performance of the estimated parameters. The model with two real data sets is also examined. In addition, the developed DGIE is applied as color image segmentation which aims to cluster the pixels into their groups. To evaluate the performance of DGIE, a set of six color images is used, as well as it is compared with other image segmentation methods including Gaussian mixture model, K-means, and Fuzzy subspace clustering. The DGIE provides higher performance than other competitive methods.

## 1. Introduction

In the field of reliability analysis, it is inconvenient or difficult to measure a lot of observations in nature on a continuous scale. For example, in many practical situations [1–9], reliability data are measured in terms of the number of cases, runs, or the number of days left for patients with the deadly disease since therapy. For more examples in reliability and lifetime applications, see Meeker and Escobar [10]. There are ways to build up a discrete distribution that has been recognized [11].

Indeed, this technique has been widely applied to generate new discrete distributions for example [12–18] and references cited therein. Abouammoh and Alshingiti [19] introduced a shape parameter to the inverted exponential distribution to get the generalized inverted exponential

(GIE) distribution. The GIE distribution is derived from the exponentiated Frechet distribution [20]. The hazard rate of the GIE distribution can be decreasing or increasing, based on its shape parameter. The GIE has effectiveness in modeling a lot of data and can be applied in several applications, such as horse racing, life testing, queues, and wind speeds [20]. Abouammoh and Alshingiti [19] show that the GIE distribution provides a better fit than Weibull, gamma, generalized exponential distribution, and gamma distribution. The GIE can be widely used in many fields, see for example, [21, 22]. However, there exist a number of discrete distributions in the literature; there are some limitations in these distributions in fitting a lot of data in many areas effectively, such as geometric, discrete Lindely, and discrete logistic distributions. There is still a need to develop new discretized distributions that are able to have

applications such as image segmentation. This motivated us to present a new distribution.

The presented distribution discrete generalized inverted exponential (DGIE) is constructed from a generalized inverted exponential distribution. Parameters are estimated using two methods, namely moments and maximum likelihood. The consistency of the estimated parameters is illustrated using simulation. Based on two data sets the proposed distribution is more convenient to analyze the given data more than competitive distributions. The proposed distribution is applied in color segmentation which helps in clustering the pixels into their groups. The DGIE provides higher performance than other competitive methods.

The main contribution of the current study can be summarized as follows:

- (1) Present a new distribution discrete generalized inverted exponential (DGIE) to avoid the limitations of other distributions
- (2) Compute the basic distributional properties, moments, probability function, reliability indices, characteristic function, and the order statistics of DGIE
- (3) Evaluate the applicability of DGIE by using it to improve the color segmentation

The paper is organized as follows: in Section 2, we introduce the DGIE( $\beta, \theta$ ) distribution and mention statistical properties, as failure function, survival function. In addition, we list some additional properties of the proposed

distribution such as moment generating function, moments, quantile, entropy, stress-strength, mean residual lifetime, and order statistics. We analyse the DGIE( $\beta, \theta$ ) using two real data set in Section 3. Finally, the conclusion is mentioned in Section 4.

## 2. Materials and Methods

### 2.1. Discrete Generalized Inverted Exponential Distribution

*Definition 1.* A random variable  $X$  is said to have a discrete generalized inverted exponential distribution with parameter  $\beta$  ( $\beta > 0$ ) and  $\theta = e^{-\lambda}$ ,  $0 < \theta < 1$ , if its probability mass function (PMF) has the form:

$$P(X = x) = (1 - \theta^{1/(x)})^\beta - (1 - \theta^{1/(x+1)})^\beta; x \in \mathbb{N}_0. \quad (1)$$

We denote this distribution as DGIE( $\beta, \theta$ ). Figure 1 illustrates several examples of the probability mass function of DGIE( $\beta, \theta$ ) distribution for various values of  $\beta$  and  $\theta$ .

*2.1.1. Cumulative Distribution Function.* The cumulative distribution function CDF of DGIE( $\beta, \theta$ ) is given by

$$F(x, \alpha, \theta) = 1 - S(x, \alpha, \theta) + P = 1 - (1 - \theta^{1/(x+1)})^\beta, \quad (2)$$

where  $\beta$  ( $\beta > 0$ ) and  $\theta = e^{-\lambda}$ ,  $0 < \theta < 1$ . Monotonic property simply, we find out

$$\frac{f_X(x+1; \beta, \theta)}{f_X(x; \beta, \theta)} = \frac{(1 - \theta^{1/(x+1)})^\beta - (1 - \theta^{1/(x+2)})^\beta}{(1 - \theta^{1/(x)})^\beta - (1 - \theta^{1/(x+2)})^\beta}, x \in \mathbb{N}_0, \beta > 0 \text{ and} \quad (3)$$

$$\theta = e^{-\lambda}, 0 < \theta < 1.$$

Is a decreasing function of which leads to  $\{f_X(x; \beta, \theta)\}^2 > f_X(x+1; \beta, \theta)f_X(x-1; \beta, \theta)$ ;  $x \in \mathbb{N}_0$ ,  $\beta > 0$  and  $\theta = e^{-\lambda}$ ,  $0 < \theta < 1$ .

The distribution is log-concave. Based on the log concavity (Mark, 1996), the proposed of DGIE( $\beta, \theta$ ) distribution is unimodal with increasing failure rate distribution, and it all its moments.

Furthermore, the quantile function of DGIE( $\beta, \theta$ ) distribution, say  $Q(p)$ , from  $F(x_p) = p$ , is given by

$$x_p = \frac{\ln \theta}{\ln(1 - (1 - p)^{1/\beta})} - 1, \quad (4)$$

where  $\beta$  ( $\beta > 0$ ),  $\theta = e^{-\lambda}$ ,  $0 < \theta < 1$  and  $0 < p < 1$ . Where  $n$  denotes the greatest integer function.

Hence the median can be obtained by putting  $p = 1/2$  in equation (4)

$$\text{Med}(X) = \frac{\ln \theta}{\ln(1 - (1/2)^{1/\beta})} - 1. \quad (5)$$

*Survival function:* The survival function of DGIE( $\beta, \theta$ ) distribution is defined as

$$S(x, \beta, \theta) = P(X \geq x) = (1 - \theta^{1/x})^\beta; x \in \mathbb{N}_0. \quad (6)$$

Hazard rate,  $r(x)$  is put as

$$h(x, \beta, \theta) = \frac{p(x)}{s(x)} = \frac{(1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta}{(1 - \theta^{1/x})^\beta}; x \in \mathbb{N}_0. \quad (7)$$

Figure 2 shows the HRF plots of DGIE( $\beta, \theta$ ) distribution for various values of  $\beta$  and  $\theta$ .

The reversed hazard rate function (RHRF) of the DGIE( $\beta, \theta$ ) distribution is put in the form



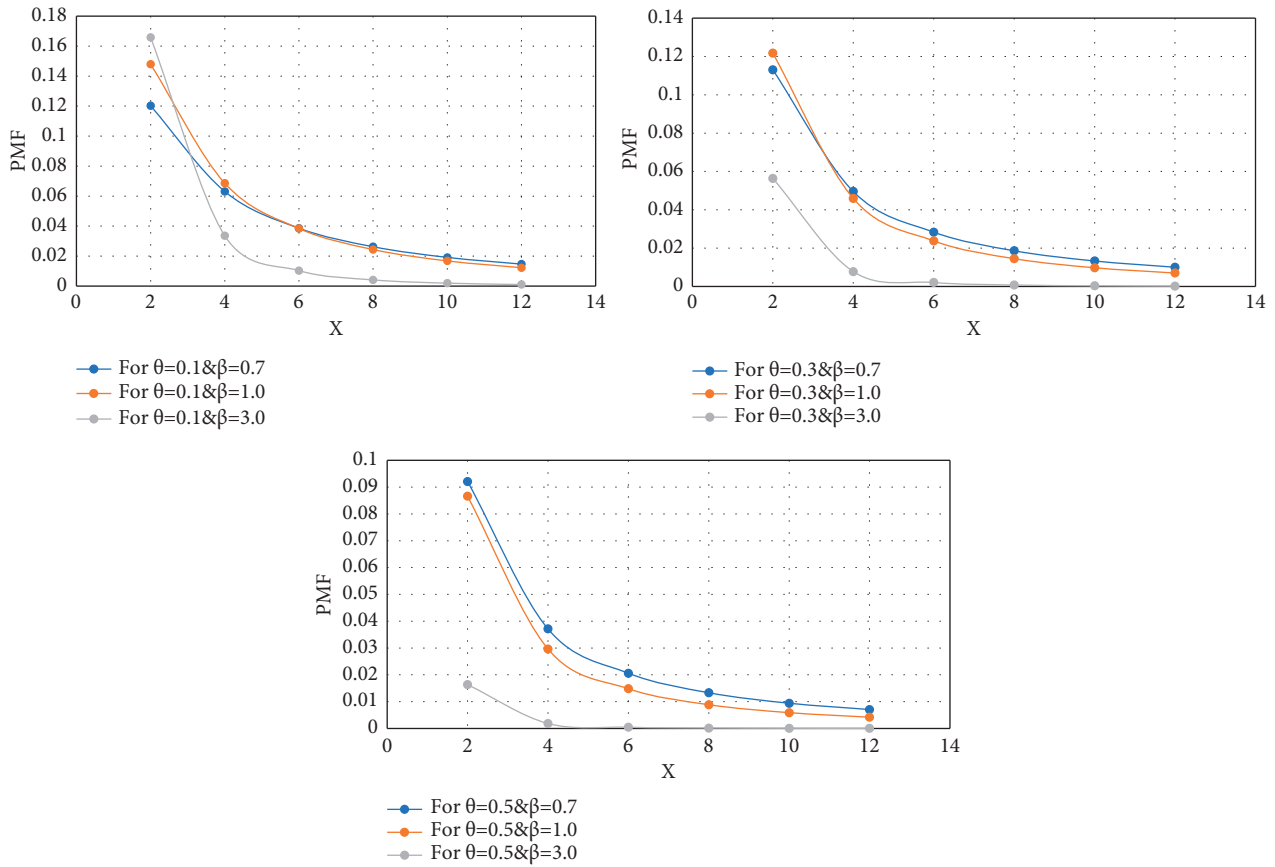


FIGURE 1: PMF of the DGIE( $\beta, \theta$ ).

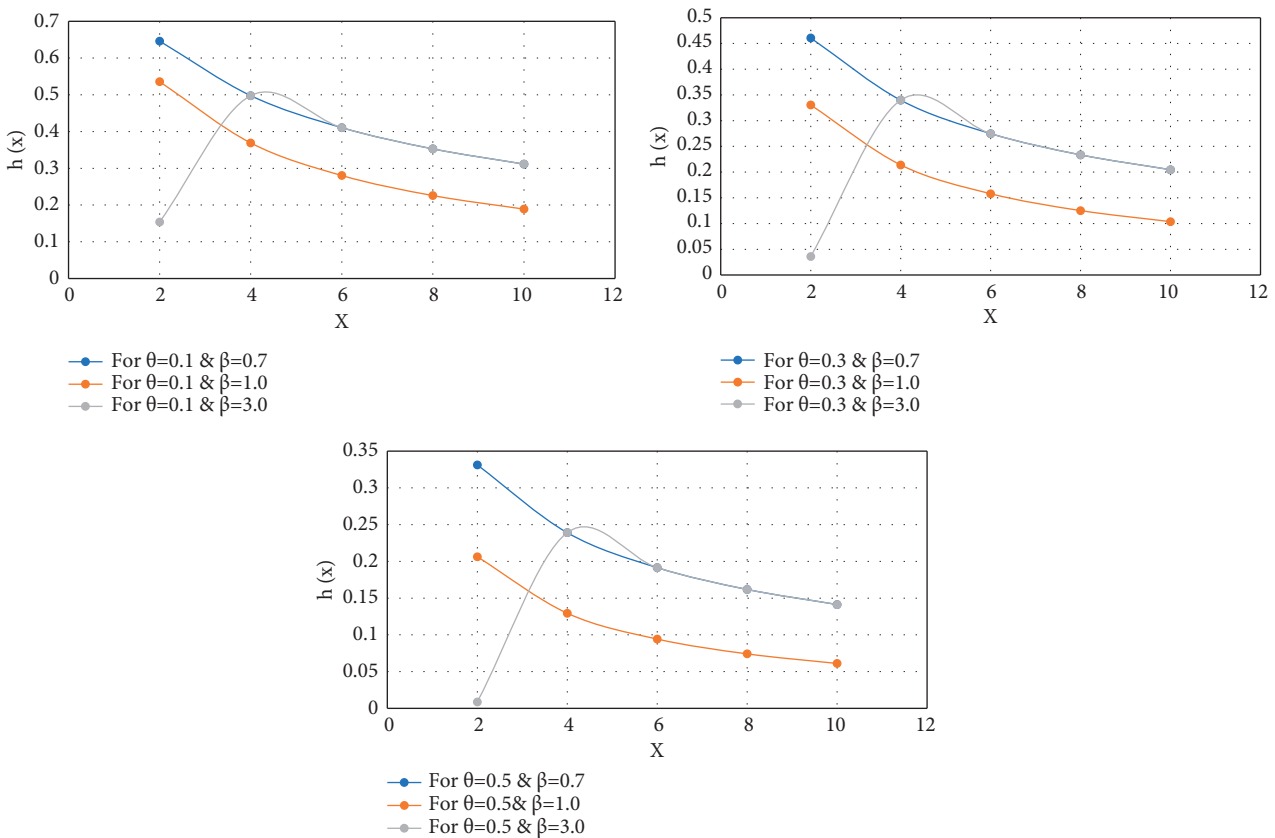


FIGURE 2: HRF of the DGIE( $\beta, \theta$ ).

$$r(x, \beta, \theta) = \frac{(1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta}{1 - (1 - \theta^{1/x})^\beta}; x \in \mathbb{N}_0. \quad (8)$$

Figure 3 indicate the RHRF plots of **DGIE**( $\beta, \theta$ ) distribution for various values of  $\beta$  and  $\theta$ .

**2.2. Statistical Properties.** The  $r^{\text{th}}$  moment  $\mu'_r$  of a discrete exponentiated exponential distribution **DGIE**( $\beta, \theta$ ) about the origin is obtained as follows:

$$\begin{aligned} \mu'_r &= \sum_{x=0}^{\infty} x^r \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right], \\ \mu'_r &= E[X^r] = \sum_{x=0}^{\infty} x^r P(X = x). \end{aligned} \quad (9)$$

The moment generating function (MGF)  $\mathbf{M}_X(\mathbf{t})$  of **DGIE**( $\beta, \theta$ ) distribution is computed as follows:

$$\begin{aligned} M_X(t) &= E[e^{tx}] = \sum_{x=0}^{\infty} e^{tx} P(X = x) \\ &= \sum_{x=0}^{\infty} e^{tx} P(X = x) \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right]. \end{aligned} \quad (10)$$

The mean ( $\mu$ ) of **DGIE**( $\beta, \theta$ ) distribution is derived as

$$\mu'_1 = \mu = E[X] = \sum_{x=0}^{\infty} x \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right]. \quad (11)$$

The second moment is obtained as

$$\mu'_2 = \mu = E[X^2] = \sum_{x=0}^{\infty} x^2 \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right]. \quad (12)$$

Hence, the variance ( $\sigma^2$ ) could be derived as

$$\text{var}(X) = \sum_{x=0}^{\infty} x^2 \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right] - \left( \sum_{x=0}^{\infty} x \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right] \right)^2. \quad (13)$$

The 3<sup>rd</sup> and 4<sup>th</sup> moments are, respectively are obtained as

$$\begin{aligned} \mu'_3 &= E[X^3] = \sum_{x=0}^{\infty} x^3 \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right], \\ \mu'_4 &= E[X^4] = \sum_{x=0}^{\infty} x^4 \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right]. \end{aligned} \quad (14)$$

The measure of skewness  $\alpha_3$  of **DGIE**( $\beta, \theta$ ) distribution is obtained as follows:

$$\begin{aligned} \alpha_3 &= \frac{\mu'_3 - 2\mu'_2\mu + \mu^3}{\sigma^3} \\ &= \frac{1}{\sigma^3} \left\{ \left[ \sum_{x=0}^{\infty} x^3 \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right] \right] - 2\mu \sum_{x=0}^{\infty} x^2 \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right] \right\} + \frac{\mu^3}{\sigma^3}. \end{aligned} \quad (15)$$

The measure of kurtosis  $\alpha_4$  of **DGIE**( $\beta, \theta$ ) distribution is obtained as follows:

$$\begin{aligned} \alpha_4 &= \frac{\mu'_4 - 4\mu'_3\mu + 6\mu'_2\mu^2 - 3\mu^4}{\sigma^4} \\ \alpha_4 &= \frac{1}{\sigma^4} \left\{ \sum_{x=0}^{\infty} x^4 \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right] - 4\mu \sum_{x=0}^{\infty} x^3 \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right] \right. \\ &\quad \left. + 6\mu^2 \sum_{x=0}^{\infty} x^2 \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right] - 3\mu^4 \right\}. \end{aligned} \quad (16)$$

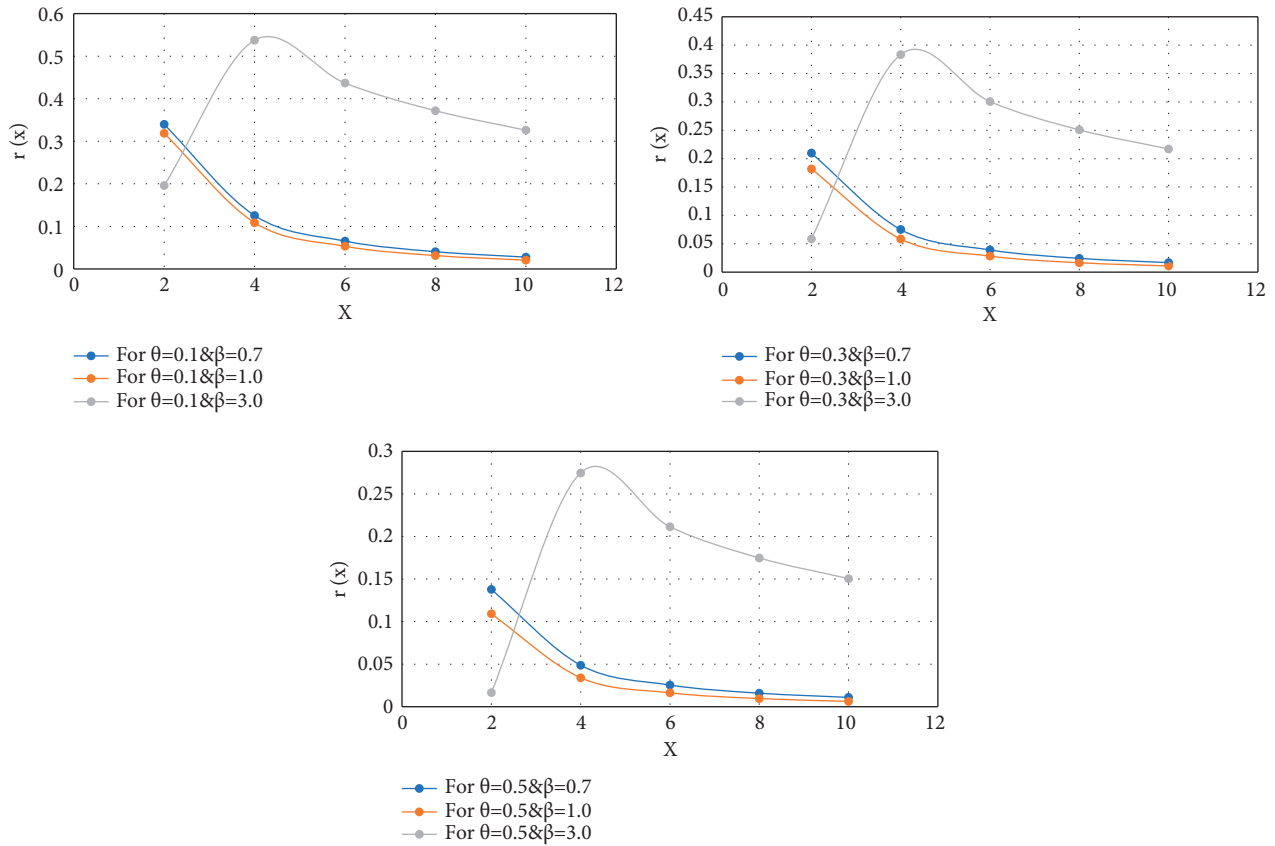


FIGURE 3: RHRF of the DGIE(β, θ).

The probability generating function (PGF),  $G(t)$ , of DGIE(β, θ) distribution is obtained as follows:

For simplicity, we compute the PGF numerically where the  $r^{\text{th}}$  the factorial moment is computed as

$$G(t) = E[t^X] = \sum_{x=0}^{\infty} t^x P(X = x) \tag{17}$$

$$= \sum_{x=0}^{\infty} t^x \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right].$$

$$\mu_{[r]} = G^{(r)}(1) = \frac{\alpha}{\alpha - 1} \sum_{x=0}^{\infty} x(x-1)\dots(x-r+1) \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right]. \tag{18}$$

The variance, the variance ( $\sigma^2$ ) of DGIE(β, θ) distribution is given by the following:

$$\text{var}(X) = \sigma^2 = G''(1) + G'(1) - \left( G'(1) \right)^2$$

$$= \frac{\alpha}{\alpha - 1} \sum_{x=0}^{\infty} x^2 \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right] - \left( \frac{\alpha}{\alpha - 1} \sum_{x=0}^{\infty} x \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right] \right)^2. \tag{19}$$

Characteristic function: the characteristic function (CF),  $\phi_X(w)$  of DGIE( $\beta, \theta$ ) distribution is of the form:

$$\phi_X(w) = E[e^{iwx}] = \sum_{x=0}^{\infty} e^{iwx} P(X = x) = \frac{\alpha}{\alpha - 1} \sum_{x=0}^{\infty} e^{iwx} \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right]. \tag{20}$$

Because moments do not have closed forms, the mean and variance can only be calculated numerically. We estimated mean and variance for various values of  $\beta$  and  $\theta$  in Tables 1 and 2, respectively.

2.3. *Order Statistics.* Order statistics has a deep reflection on theoretical and practical aspects of statistics. This

importance is shown in statistical inference and nonparametric statistics. Let  $X_1, X_2, \dots, X_n$  be a random sample from A DGIE( $\beta, \theta$ ) distribution, and let  $X_{1:n}, X_{2:n}, \dots, X_{n:n}$  be the order statistics. Then, the CDF of the  $i^{\text{th}}$  order statistics for  $x$  can be represented in the for

$$\begin{aligned} F_{i:n}(x, \alpha, \theta) &= \sum_{k=i}^n \binom{n}{k} [F_i(x, \beta, \theta)]^k [F_i(x, \beta, \theta)]^a \\ &= \sum_{k=i}^n \sum_{m=0}^{\beta k} (-1)^m \binom{n}{k} \binom{\beta k}{m} \theta^{m\beta/(x+1)} \left[ (1 - \theta^{1/(x+1)})^\beta \right]^{n+m-k}. \end{aligned} \tag{21}$$

Therefore, the PMF of the  $k^{\text{th}}$  Os has the form:

$$f_{k:n}(x, \alpha, \theta, \beta) = \sum_{m=0}^{k-1} \Theta_m^{(n, \beta(k-1))} \theta^{m/(x+1)} \left[ (1 - \theta^{(x+1)})^\beta \right]^{n+m-k} \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right], \tag{22}$$

where  $\Theta_m^{(n, k-1)} = (-1)^m \binom{\beta k}{m} n! / (k-1)! (n-k)!$ .

So, the  $q^{\text{th}}$  moments of  $X_{i:n}$  is written in the form:

$$E[X_{i:n}^q] = \sum_{x=0}^{\infty} \sum_{m=0}^{k-1} \Theta_m^{(n, \beta(k-1))} \theta^{m/(x+1)} x^q \left[ (1 - \theta^{(x+1)})^\beta \right]^{n+m-k} \left[ (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta \right], \tag{23}$$

where  $\Theta_m^{(n, k-1)} = (-1)^m \binom{\beta k}{m} n! / (k-1)! (n-k)!$ .

$$I_R(\gamma) = \frac{1}{1-\gamma} \log \sum_x (P(X = x))^\gamma, \tag{24}$$

2.3.1. *Renyi Entropy.* Renyi entropy plays a vital role in information theory. The Renyi entropy of a random variable  $X$  is defined as:

where  $\gamma > 0$  and  $\gamma \neq 1$  (Renyi, 1961). For the DGIE( $\beta, \theta$ ) distribution for  $\gamma$  is an integer number, we compute

$$\sum_{x=0}^{\infty} (P(X = x))^\gamma = \sum_{j=0}^{\gamma} \left[ (1 - \theta^{1/j})^\beta - (1 - \theta^{1/(j+1)})^\beta \right]^\gamma I_R(\gamma) = \frac{1}{1-\gamma} \log \sum_{j=0}^{\gamma} \left[ (1 - \theta^{1/j})^\beta - (1 - \theta^{1/(j+1)})^\beta \right]^\gamma. \tag{25}$$

2.4. *Unknown Parameters Estimation.* In this section, we used two methods to estimate DGIE( $\beta, \theta$ ) distribution unknown parameters using

2.4.1. *Maximum Likelihood Method.* Let  $X_1, X_2, \dots, X_n$  represents the lifetimes of  $n$  independent test units which

Following DGIE( $\beta, \theta$ ). So, its log-likelihood function is written as:  $p(x) = (1 - \theta^{1/x})^\beta - (1 - \theta^{1/(x+1)})^\beta$

TABLE 1: The mean of the DGIE( $\beta, \theta$ ) distribution.

$\beta/\theta$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
2	2.682	1.711	1.153	0.774	0.502	0.304	0.163	0.069
3	1.471	0.861	0.522	0.307	0.169	0.083	0.034	0.009
4	1.043	0.559	0.303	0.156	0.072	0.029	0.009	0.001
5	0.814	0.398	0.193	0.086	0.033	0.010	0.0025	0.0003

TABLE 2: The variance of the DGIE( $\beta, \theta$ ) distribution.

$\beta/\theta$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
3	6.317	2.822	1.449	0.7621	0.385	0.177	0.068	0.018
4	2.371	0.997	0.481	0.232	0.103	0.039	0.012	0.002
5	1.300	0.526	0.239	0.104	0.040	0.012	0.002	0.0003
6	0.845	0.334	0.141	0.054	0.017	0.004	0.001	0.001

$$L[P(X = x)] = \prod_{i=1}^n p(x_i) = \prod_{i=1}^n \left( (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right),$$

$$I(x, \alpha, \theta) = \sum_{i=1}^n \ln \left( (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right).$$

(26)

Likelihood equations are then obtained as follows:

$$\frac{\delta l}{\delta \theta} = \sum_{i=1}^n \frac{-1/x_i - 1\beta\theta^{1/x_i-1} (1 - \theta^{1/x_i})^\beta + \beta 1/(x_i + 1) (1 - \theta^{1/(x_i+1)})^{\beta-1} \theta^{1/(x_i+1)}}{(1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta} = 0,$$

$$\frac{\delta l}{\delta \beta} = \sum_{i=0}^n \frac{(1 - \theta^{1/x_i})^\beta \ln (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \ln (1 - \theta^{1/(x_i+1)})^\beta}{(1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta} = 0.$$

(27)

We can obtain the solution of these equations numerically then; we compute the Fisher's information matrix by finding the second partial derivatives

$$I_x(\beta, \theta) = \begin{bmatrix} -E \left[ \frac{\partial^2 l}{\partial \beta^2} \right] & -E \left[ \frac{\partial^2 l}{\partial \beta \partial \theta} \right] \\ -E \left[ \frac{\partial^2 l}{\partial \theta \partial \beta} \right] & -E \left[ \frac{\partial^2 l}{\partial \theta^2} \right] \end{bmatrix}. \tag{28}$$

One can infer that the DGIE( $\beta, \theta$ ) distribution satisfies the regularity conditions [23]. Then, the MLE vector  $(\hat{\beta}, \hat{\theta})^T$

is asymptotically normal and consistent. Fisher's information matrix can be approximated as

$$I_x(\alpha, \theta) = \begin{bmatrix} -\frac{\partial^2 l}{\partial \beta^2} \Big|_{(\hat{\beta}, \hat{\theta})} & -\frac{\partial^2 l}{\partial \beta \partial \theta} \Big|_{(\hat{\beta}, \hat{\theta})} \\ -\frac{\partial^2 l}{\partial \theta \partial \beta} \Big|_{(\hat{\beta}, \hat{\theta})} & -\frac{\partial^2 l}{\partial \theta^2} \Big|_{(\hat{\beta}, \hat{\theta})} \end{bmatrix}, \tag{29}$$

where  $\hat{\beta}$  and  $\hat{\theta}$  are the MLEs of  $\beta$  and  $\theta$  [24].

The element of the hessian matrix  $I_x(\alpha, \theta)$  are obtained from

$$\begin{aligned}
\frac{\partial^2 l}{\partial \beta^2} &= \sum_{i=0}^n \frac{\left[ (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right] \left[ (1 - \theta^{1/x_i})^\beta (\ln(1 - \theta^{1/x_i}))^2 - (1 - \theta^{1/(x_i+1)})^\beta (\ln(1 - \theta^{1/(x_i+1)}))^2 \right]}{\left( (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right)^2} a \\
&\quad - \frac{\left[ (1 - \theta^{1/x_i})^\beta \ln(1 - \theta^{1/x_i}) - (1 - \theta^{1/(x_i+1)})^\beta \ln(1 - \theta^{1/(x_i+1)}) \right] \left[ (1 - \theta^{1/x_i})^\beta \ln(1 - \theta^{1/x_i}) - (1 - \theta^{1/(x_i+1)})^\beta \ln(1 - \theta^{1/(x_i+1)}) \right]}{\left( (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right)^2} \\
\frac{\partial^2 l}{\partial \theta \partial \beta} &= \sum_{i=0}^n \frac{\left[ (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right] \left[ -1/x_i \theta^{1/x_i} (1 - \theta^{1/x_i})^\beta - 1/x_i \beta \theta^{1/x_i - 1} (1 - \theta^{1/x_i})^\beta \ln(1 - \theta^{1/x_i}) \right]}{\left( (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right)^2} \\
&\quad + \frac{1/(x_i + 1) \theta^{1/(x_i+1)} (1 - \theta^{1/(x_i+1)})^{\beta-1} + \beta 1/(x_i + 1) \theta^{1/(x_i+1)} (1 - \theta^{1/(x_i+1)})^{\beta-1} \ln(1 - \theta^{1/(x_i+1)})}{\left( (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right)^2} \\
&\quad - \frac{\left( -1/x_i \beta \theta^{1/x_i - 1} (1 - \theta^{1/x_i})^\beta + \beta 1/(x_i + 1) \theta^{1/(x_i+1)} (1 - \theta^{1/(x_i+1)})^{\beta-1} \right) \left( (1 - \theta^{1/x_i})^\beta \ln(1 - \theta^{1/x_i}) - (1 - \theta^{1/(x_i+1)})^\beta \ln(1 - \theta^{1/(x_i+1)}) \right)}{\left( (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right)^2} \\
\frac{\partial^2 l}{\partial \theta^2} &= \sum_{i=0}^n \frac{\left[ (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right] \left[ -\beta 1/x_i (1/x_i - 1) \theta^{1/x_i - 2} (1 - \theta^{1/x_i})^\beta + 1/x_i^2 \beta^2 \theta^{2(1/x_i - 1)} (1 - \theta^{1/x_i})^\beta \right]}{\left( (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right)^2} \\
&\quad + \frac{\beta 1/(x_i + 1)^2 \theta^{1/x_i - 1 - 1} (1 - \theta^{1/(x_i+1)})^{\beta-1} + \beta(\beta - 1) 1/(x_i + 1)^2 \theta^{1/x_i - 1 - 1} (1 - \theta^{1/(x_i+1)})^{\beta-2}}{\left( (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right)^2} \\
&\quad - \frac{\left( -1/x_i \beta \theta^{1/x_i - 1} (1 - \theta^{1/x_i})^\beta + \beta 1/(x_i + 1) \theta^{1/x_i - 1} (1 - \theta^{1/(x_i+1)})^{\beta-1} \right) - \left( \beta 1/x_i (1 - \theta^{1/x_i - 1})^{\beta-1} + \beta 1/(x_i + 1) (1 - \theta^{1/(x_i+1)})^{\beta-1} \right)}{\left( (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right)^2}.
\end{aligned} \tag{30}$$

2.4.2. *Method of Moments Estimation.* We can find moments' estimates (MM  $E_s$ ) of  $(\beta, \theta)$  by solving the equations

$$\begin{aligned}
\sum_{i=1}^{\infty} x_i \left[ (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right] &= \mu_1^{[1]}, \\
\sum_{i=1}^{\infty} x_i^2 \left[ (1 - \theta^{1/x_i})^\beta - (1 - \theta^{1/(x_i+1)})^\beta \right] &= \mu_2^{[2]},
\end{aligned} \tag{31}$$

where  $\mu_1^{[1]}$  and  $\mu_2^{[2]}$  represent the first and the second sample moments.

### 3. Results and Discussion

3.1. *A Simulation Study.* In this section, we assess the performance of the maximum-likelihood estimate with respect to sample size  $n$ . The assessment is based on a simulation study:

- (1) Generate 10000 samples of size  $n$  from equation (1). The inversion method is used to generate samples; that is, varieties of the discrete generalized inverted exponential distribution are generated using

$$X = \frac{\ln \theta}{\ln(1 - (1 - u)^{1/\beta})} - 1; 0 < u < 1, \quad (32)$$

where  $U \sim U(0, 1)$  is a uniform variable on the unit interval.

- (2) Compute the Maximum-likelihood Estimates for 10000 Samples, Say  $\hat{\theta}_i$  for

$$i = 1, 2, \dots, 10000. \quad (33)$$

- (3) Compute the biases and mean-squared errors given by

$$\begin{aligned} \text{bias}(n) &= \frac{1}{10000} \sum_{i=1}^{10000} (\hat{\theta}_i - \theta_i), \\ \text{MSE}(n) &= \frac{1}{10000} \sum_{i=1}^{10000} (\hat{\theta}_i - \theta_i)^2. \end{aligned} \quad (34)$$

The empirical results are given in Table 3. From Table 3, the following observations can be noted: the magnitude of the bias always decreases to zero as  $n \rightarrow \infty$ . The MSEs always decrease to zero as  $n \rightarrow \infty$ . This shows the consistency of the estimators.

3.1.1. *Data Application.* We pointed out here, the notability of a discrete generalized inverted exponential distribution on distributions: geometric distribution, discrete logistic distribution and discrete Lindley distribution. Two real data sets are applied. The first data are in Table 4 is for 30 failure times of the air conditioning system of an airplane. These data are taken from [25].

The MLE of  $(\beta, \theta)$  values in all these cases has been computed. The Kolmogorov–Smirnov (K–S) measure in each case and the associated  $P$  value are computed. The result is put in Table 5.

The Akaike information criterion (AIC), correct Akaike information criterion (CAIC) and Bayesian Akaike information criterion (BIC) values for the models have been computed. The result is reported in Table 6. The Akaike’s measures indicate that the GIED distribution fits the data better than some existing distributions for this data set.

The data set given in Table 7 consists of uncensored data from [23]. The data gives 100 observations on breaking stress

of carbon fibers (in Gba). The MLE of  $(\beta, \theta)$  values in all these cases have been computed. The Kolmogorov–Smirnov (K–S) measure in each case and the associated  $p$ -value are computed. The result is put in Table 8. A comparison between the observed and the fitted distributions are shown in Figures 4 and 5.

The Akaike’s measures indicate that the GIED distribution fits the data better than some existing distributions for this data set, as in Table 9. For the first, second data sets, the discrete generalized inverted exponential distribution shows the best convenient  $p$  values. The distribution plots propose that the discrete generalized inverted exponential distribution offers the best fit between the competitor distributions. On the basis of the tabulated results, we infer that the discrete generalized inverted exponential distribution provides the best fit compared to its submodels. Some summary statistics of data sets 1 and 2 are listed in Table 10.

3.2. *Image Segmentation.* In this section, we assess the ability of DGIED to improve the performance of segmentation the image. This can be performed by considering it as a clustering method.

3.2.1. *Clustering Problem Formulation for Image Segmentation.* In this part, we introduce the mathematical formulation of the automatic clustering-based image segmentation problem. In general, the main aim of AC is to split the given image  $\mathbf{I}$  into a set of  $K_{\max}$  groups. To perform this task, the between-cluster variation must be maximized at the same time with minizing within-cluster variation.

Therefore, the mathematical representation of AC can be given by dividing the image into  $K_{\max}$  cluster (i.e.,  $C_1, C_2, \dots, C_{K_{\max}}$ ) with satisfied the following criteria:

$$\begin{aligned} \bigcup_{l=1}^{K_{\max}} C_l &= \mathbf{I}, C_l \neq \phi, l = 1, \dots, K_{\max} \\ C_l \cap C_{ll} &= \phi, ll = 1, 2, \dots, K_{\max}, l \neq ll. \end{aligned} \quad (35)$$

Gaussian mixture models (GMM): it is one of the most popular clustering techniques, and it has been used as an image segmentation method in different applications, for example, image retrieval [26], chemical and physical properties of Italian wines, and the chemical [27, 28] and others [29].

The mathematical formulation of the Gaussian mixture model (GMM) can be represented by considering the given image  $\mathbf{I}$  consisting of a set of pixels  $\mathbf{X}$  that are represented as a random variable. So, the GMM can be defined as

$$f(x) = \sum_{i=1}^K w_i N(x|\mu_i, \sigma_i^2). \quad (36)$$

In equation (36),  $K$  represents the number of objects and  $w_i > 0$  refer to the weights where  $\sum_{i=1}^K w_i = 1$ . In addition, the  $N(x|\mu_i, \sigma_i^2)$  is defined as

$$N(x|\mu_i, \sigma_i^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}, \quad (37)$$

TABLE 3: The averages bias and averages MSE for simulated results of ML estimates.

Sample size	$\beta = 0.779$			$\theta = 1.671 * 10^{-5}$			$\theta = 2.823 * 10^{-6}$		
	Bias	MSE		Bias	MSE		Bias	MSE	
10	-4.064 * 10 <sup>-5</sup> (4.224 * 10 <sup>-5</sup> )	1.651 * 10 <sup>-9</sup> (1.785 * 10 <sup>-9</sup> )		-0.05 (1.178 * 10 <sup>-4</sup> )	2.486 * 10 <sup>-3</sup> (1.388 * 10 <sup>-8</sup> )		-3.541 * 10 <sup>-8</sup> (1.504 * 10 <sup>-5</sup> )	1.254 * 10 <sup>-15</sup> (2.263 * 10 <sup>-10</sup> )	
70	-6.59 * 10 <sup>-10</sup> (9.092 * 10 <sup>-7</sup> )	0 (8.266 * 10 <sup>-13</sup> )		-0.017 (5.958 * 10 <sup>-6</sup> )	2.933 * 10 <sup>-4</sup> (3.55 * 10 <sup>-11</sup> )		-3.788 * 10 <sup>-12</sup> (8.225 * 10 <sup>-8</sup> )	0 (6.765 * 10 <sup>-15</sup> )	
130	3.139 * 10 <sup>-10</sup> (-3.257 * 10 <sup>-7</sup> )	0 (1.061 * 10 <sup>-13</sup> )		0.041 (-5.469 * 10 <sup>-6</sup> )	1.656 * 10 <sup>-3</sup> (2.991 * 10 <sup>-11</sup> )		4.173 * 10 <sup>-11</sup> (-1.977 * 10 <sup>-7</sup> )	0 (3.91 * 10 <sup>-14</sup> )	
210	-9.771 * 10 <sup>-9</sup> (1.728 * 10 <sup>-5</sup> )	0 (2.987 * 10 <sup>-10</sup> )		0.039 (1.812 * 10 <sup>-5</sup> )	1.499 * 10 <sup>-3</sup> (3.285 * 10 <sup>-10</sup> )		2.595 * 10 <sup>-10</sup> (5.774 * 10 <sup>-6</sup> )	0 (3.334 * 10 <sup>-11</sup> )	
					$\theta = 8.763 * 10^{-6}$				
					$\beta = 0.779$				
Sample size					$\beta = 26.715$				
10	-2.274 * 10 <sup>-4</sup> (5.067 * 10 <sup>-5</sup> )	5.172 * 10 <sup>-8</sup> (2.568 * 10 <sup>-9</sup> )		-17.327 (1.392 * 10 <sup>-3</sup> )	3.232 * 10 <sup>-14</sup> (1.937 * 10 <sup>-6</sup> )		-4.064 * 10 <sup>-5</sup> (4.224 * 10 <sup>-5</sup> )	1.651 * 10 <sup>-9</sup> (1.785 * 10 <sup>-9</sup> )	
70	-5.551 * 10 <sup>-10</sup> (3.192 * 10 <sup>-6</sup> )	0 (1.019 * 10 <sup>-11</sup> )		-6.203 * 10 <sup>-7</sup> (3.765 * 10 <sup>-5</sup> )	3.847 * 10 <sup>-13</sup> (1.418 * 10 <sup>-9</sup> )		-5.793 * 10 <sup>-10</sup> (1.017 * 10 <sup>-6</sup> )	0 (1.034 * 10 <sup>-12</sup> )	
130	4.006 * 10 <sup>-10</sup> (1.893 * 10 <sup>-6</sup> )	0 (3.584 * 10 <sup>-12</sup> )		-5.778 * 10 <sup>-7</sup> (4.017 * 10 <sup>-5</sup> )	3.338 * 10 <sup>-13</sup> (1.613 * 10 <sup>-9</sup> )		3.436 * 10 <sup>-10</sup> (-2.201 * 10 <sup>-7</sup> )	0 (4.844 * 10 <sup>-14</sup> )	
210	-2.831 * 10 <sup>-8</sup> (1.876 * 10 <sup>-5</sup> )	0 (3.518 * 10 <sup>-10</sup> )		-6.466 * 10 <sup>-7</sup> (5.427 * 10 <sup>-5</sup> )	4.181 * 10 <sup>-13</sup> (2.945 * 10 <sup>-9</sup> )		-1.065 * 10 <sup>-8</sup> (1.735 * 10 <sup>-5</sup> )	0 (3.011 * 10 <sup>-10</sup> )	



TABLE 4: Data set 1.

23	62	42	3	16
261	47	20	14	90
87	225	5	71	1
7	71	12	11	16
120	246	120	14	52
14	21	11	11	95

TABLE 5: The results of data set 1.

Distribution	$p(x)$	Estimates of parameters	$p$ value	K-S statistics
Discrete generalized inverted exponential distribution	(2.2)	$\theta = 8.763 \times 10^{-6}$ $\beta = 0.779$	$0.237755 * 10^{-2}$	0.326379
Geometric	$p(1-p)^x$	$p = 0.017$	$0.5382 * 10^{-5}$	0.45039
Discrete logistic	$(1-p)p^{x-\mu}/(1+p^{x-\mu})(1+p^{x-\mu+1})$	$p = 0.083, \mu = -30.999$	$6.968 \times 10^{-28}$	1
Discrete lindley	$\lambda^x/1 - \log \lambda [\lambda \log \lambda + (1-\lambda)(1 - \log \lambda^{x+1})]$	$\lambda = 0.401$	$1.198124 \times 10^{-10}$	0.610163

TABLE 6: AIC, CAIC and BIC measures for data set 1.

	Distribution	AIC	CAIC	BIC
N = 30	DGIE	312.880	313.324	315.683
	Geometric	314.397	314.539	315.798
	Discrete lindely	478.532	478.673	479.9314
	Discrete logistic	13628.219	13628.659	13631.017

TABLE 7: Data set 2.

3.7	2.74	2.73	2.5	3.6
3.11	3.27	2.87	1.47	3.11
4.42	2.41	3.19	3.22	1.69
3.28	3.09	1.87	3.15	4.9
3.75	2.43	2.95	2.97	3.39
2.96	2.53	2.67	2.93	3.22
3.39	2.81	4.2	3.33	2.55
3.31	3.31	2.85	2.56	3.56
3.15	2.35	2.55	2.59	2.38
2.81	2.77	2.17	2.83	1.92
1.41	3.68	2.97	1.36	0.98
2.76	4.91	3.68	1.84	1.59
3.19	1.57	0.81	5.56	1.73
1.59	2	1.22	1.12	1.71
2.17	1.17	5.08	2.48	1.18
3.51	2.17	1.69	1.25	4.38
1.84	0.39	3.68	2.48	0.85
1.61	2.79	4.7	2.03	1.8
1.57	1.08	2.03	1.61	2.12
1.89	2.88	2.82	2.05	3.65

TABLE 8: The results of data set 2.

Distribution	$p(x)$	Estimates of parameters	$p$ value	K-S statistics
Discrete generalized inverted exponential distribution	(2.2)	$\theta = 1.671 \times 10^{-5} \beta = 26.715$	$2.19 * 10^{-7}$	0.279787
Geometric	$p(1-p)^x$	$p = 0.381$	$1.653 * 10^{-31}$	0.591263
Discrete logistic	$(1-p)p^{x-\mu}/(1+p^{x-\mu})(1+p^{x-\mu+1})$	$p = 0.132, \mu = -38.28$	$2.45 \times 10^{-89}$	1
Discrete lindley	$\lambda^x/1 - \log \lambda [\lambda \log \lambda + (1-\lambda)(1 - \log \lambda^{x+1})]$	$\lambda = 0.599$	$8.6704 \times 10^{-13}$	0.372892

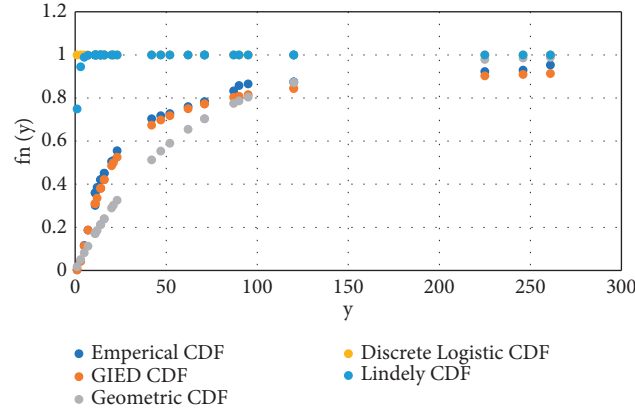


FIGURE 4: Distribution plots for data set 1.

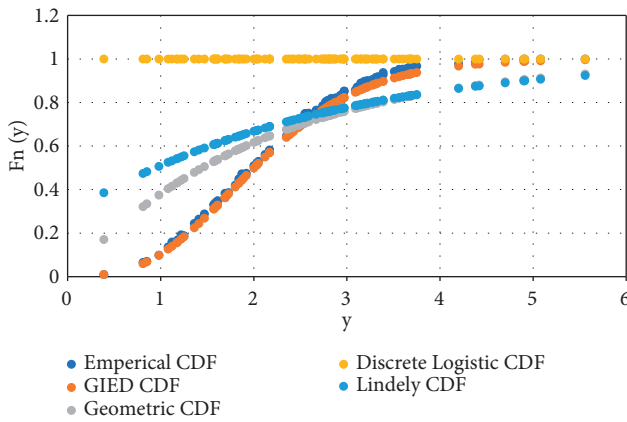


FIGURE 5: Distribution plots for data set 2.

TABLE 9: AIC, CAIC and BIC measures for data set 2.

	Distribution	AIC	CAIC	BIC
N = 100	DGIE	314.445	314.569	319.655
	Geometric	400.411	400.541	403.016
	Discrete lindely	422.027	422.067	424.632
	Discrete logistic	16832.972	16833.095	16838.182

where  $\mu_i$  and  $\sigma_i$  are mean and the standard deviation of class  $i$ . For image  $\mathbf{X}$ , the parameters are  $\theta = (\mathbf{w}_1, \dots, \mathbf{w}_k, \mu_1, \dots, \mu_k, \sigma_1^2, \dots, \sigma_k^2)$  are required to determine and to achieve this estimation, the Expectation-Maximization (EM) method is used. The steps of EM can be summarized as in Algorithm 1:

However, the traditional GMM has some limitations that influence its performance, such as inefficiency in

modeling all the data types, including discrete data in the application such as machine learning [30]. To avoid these limitations, we use the new distribution named Discrete Generalized Inverted Exponential Distribution. In general, DGIED has the ability to tackle the non-inaccuracy of using general distributions such as mixture Gaussian distribution.

**3.2.2. Dataset Description.** In this study, the performance of the developed clustering-based color image segmentation using DGIED mixture model (DGIEMM) is evaluated using a set of six color images (as in Figures 6(a)–6(f)) [31]. In addition, we compared the results of DGIEMM with GMM, K-means, and Fuzzy subspace clustering (FSC).

**3.2.3. Performance Measures.** To evaluate the efficacy of the developed image segmentation, a set of performance measures is used. For example, Accuracy, Adjust Rand Index, Hubert, and Normalized mutual information. The details of these measures are given as follows:

**Accuracy:** It is a measure used to assess the ability of the method to determine the optimal cluster for each pixel. It is formulated as

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \quad (38)$$

where TP, FP, TN, and FN are the True positive, False positive, True negative, and False negative.

**Adjust Rand Index:** It is a measure used to assess the similarity between two groups, and it is defined as:

$$\text{ARI} = \frac{\sum_{ij} \binom{\mathbf{n}_{ij}}{2} - \left[ \sum_i \binom{\mathbf{a}_i}{2} \sum_j \binom{\mathbf{b}_j}{2} \right] / \binom{\mathbf{n}}{2}}{0.5 \left[ \sum_i \binom{\mathbf{a}_i}{2} + \sum_j \binom{\mathbf{b}_j}{2} \right] - \left[ \sum_i \binom{\mathbf{a}_i}{2} \sum_j \binom{\mathbf{b}_j}{2} \right] / \binom{\mathbf{n}}{2}}, \quad (39)$$

TABLE 10: Some statistical measures for data sets.

Data set	Mean	Median	Std. deviation	Variance	Skewness	Kurtosis	Minimum	Maximum
I	28.73429	22	70.67654	5167.421	1.784077	2.568813	1	261
II	2.404559	2.7	1.008803	1.027964	0.373784	0.172868	0.39	5.56

(1) Input: image  $X_j$ ,  $j = 1, \dots, n$  and  $i \in \{1, 2, \dots, k\}$  are the label set  $Y = \{y_1, y_2, \dots, y_N\}$ ,  $y_n \in \{1, \dots, K\}$   
 (2) Initialize  $\theta^0 = (\mathbf{p}_1^0, \dots, \mathbf{p}_k^0, \mu_1^0, \dots, \mu_k^0, \sigma_1^{2(0)}, \dots, \sigma_k^{2(0)})$ :  
 (3) While (condition not met)  
 (4) E-Step:  

$$\mathbf{p}_{ij}^{r+1} = \mathbf{p}_{ij}^{r+1}(i|x_j) = \mathbf{w}_i^r \mathbf{N}(x_j|\mu_i^{(r)}, \sigma_i^{2(r)})/f(x_j)$$
  
 (5) M-Step  

$$\hat{\mathbf{w}}_{ij}^{r+1} = 1/n \sum_{j=1}^n \mathbf{w}_{ij}^r$$

$$\hat{\mu}_{ij}^{r+1} = \sum_{j=1}^n \mathbf{w}_{ij}^{r+1} x_j / n \hat{\mathbf{w}}_{ij}^{r+1}$$

$$\hat{\sigma}_i^{2(r+1)} = \sum_{j=1}^n \mathbf{w}_{ij}^{r+1} (x_j - \hat{\mu}_i^{r+1})^2 / n$$
  
 End While  
 (6) For each data vector  $x_n$ , set.  

$$y_n = \arg \max_i (\mathbf{w}_i \mathbf{N}(x_n|\mu_i, \sigma_i^2))$$

ALGORITHM 1: Steps of EM method.



I1  
(a)



I2  
(b)



I3  
(c)



I4  
(d)

FIGURE 6: Continued.



FIGURE 6: The original tested images used in this study. (a) I1, (b) I2, (c) I3, (d) I4, (e) I5, (f) I6.

TABLE 11: Comparison between developed method and other segmentation methods.

	DGIEMM	FSC	GMM	K-means		DGIEMM	FSC	GMM	K-means	
I1	Accuracy	0.9951	0.9494	0.6259	0.6259	I4	0.9886	0.9086	0.9013	0.9515
	AR	0.9803	0.8068	0.9572	0.9134		0.9548	0.6679	0.6441	0.7422
	Hubert	0.9805	0.8078	0.8634	0.8886		0.9548	0.6679	0.6442	0.7141
	NMI	0.9574	0.7474	0.8373	0.8726		0.9208	0.6419	0.5919	0.6986
I2	Accuracy	0.9995	0.9880	0.8956	0.8956	I5	0.9998	0.9964	0.9899	0.5614
	AR	0.9978	0.9510	0.6224	0.6224		0.9993	0.9858	0.9600	0.0785
	Hubert	0.9979	0.9526	0.6260	0.6260		0.9993	0.9858	0.9601	0.0543
	NMI	0.9931	0.9076	0.5820	0.5820		0.9978	0.9691	0.9276	0.1884
I3	Accuracy	0.9913	0.9867	0.9280	0.9274	I6	0.9982	0.9928	0.8972	0.9198
	AR	0.9647	0.9468	0.7227	0.7206		0.9915	0.9666	0.6047	0.6251
	Hubert	0.9653	0.9476	0.7326	0.7306		0.9928	0.9716	0.6311	0.7240
	NMI	0.9306	0.8926	0.6554	0.6537		0.9762	0.9160	0.5463	0.6547

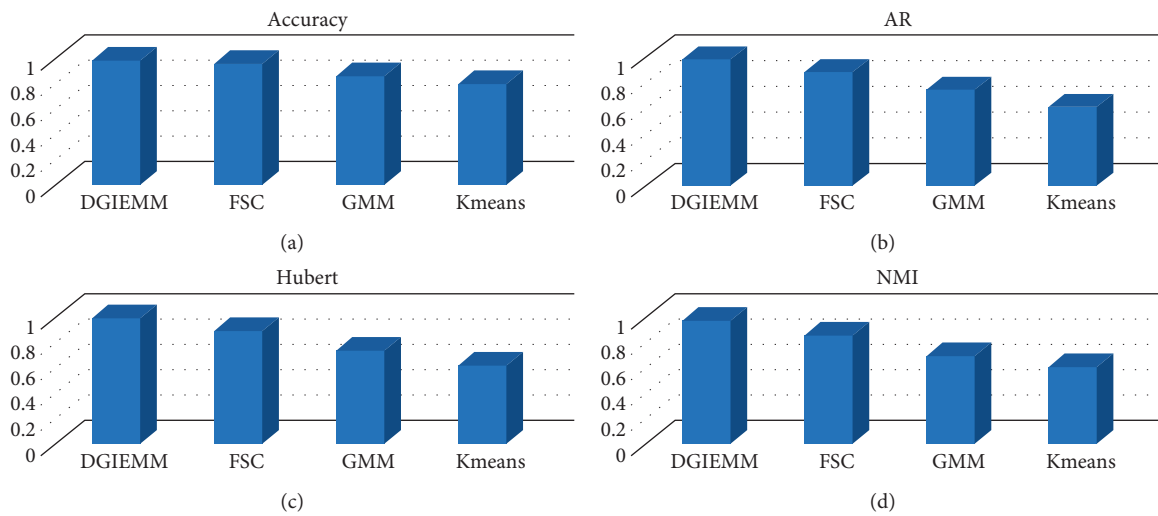


FIGURE 7: Average of each method in terms of (a) accuracy, (b) AR, (c) Hubert, and (d) NMI.

TABLE 12: Mean-rank obtained by each algorithm.

	DGIEMM	FSC	GMM	K-means
Accuracy	4	2.8333	1.5000	1.6667
AR	4	2.5000	1.7500	1.7500
Hubert	4	2.5000	1.5833	1.9167
NMI	4	2.5000	1.5833	1.9167

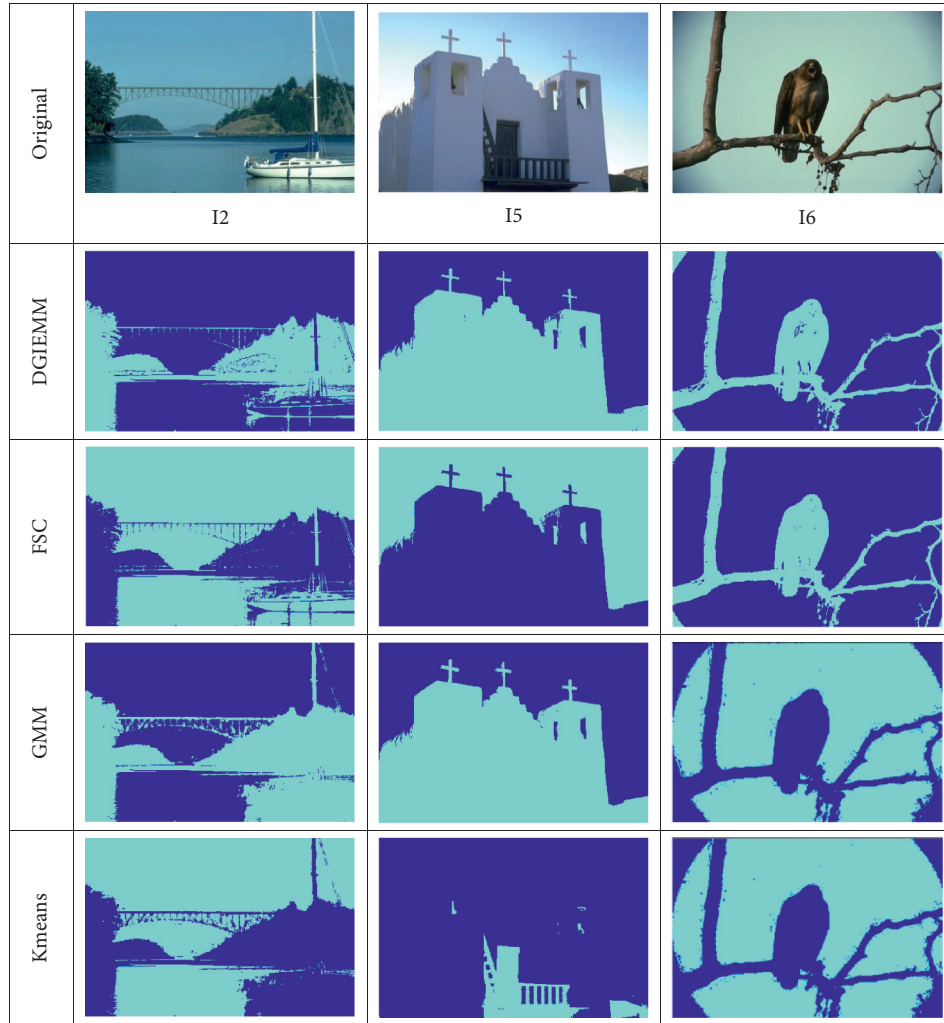


FIGURE 8: Segmented images (I1, I5, I6) using competitive algorithms.

where  $n_{ij}$  denotes the number of objects in common between classes.  $a_i$  and  $b_j$  are the sum of rows and columns of contingency table, respectively.

Hubert: it is a measure used to compute the correlation coefficient between classes and it is defined as:

$$\text{Hubert} = \frac{\hat{a}_{i < j} (X_{ij} - \mu_X)(Y_{ij} - \mu_Y)}{N \sigma_X \sigma_Y}, \quad (40)$$

where  $\sigma_X$  and  $\sigma_Y$  are the standard deviation of cluster  $X$  and cluster  $Y$ , respectively.

Normalized mutual information (NMI): is defined as a normalization of the Mutual Information that defined as:

$$\text{NMI} = \frac{2I(C_T; C)}{H(C_T) + H(C)}, \quad (41)$$

where  $C_T$  and  $C$  are the class label and its cluster label, respectively.  $H$  is the Entropy and Mutual Information between  $C_T$  and  $C$ , respectively.

3.2.4. Results and Discussion. The comparison between the developed color image segmentation method (i.e., DGIEMM) and the other methods is given in Table 11. It can be noticed from these results the high ability of the developed method to cluster the images into their objects overall the other methods. For example, according to the results in

terms of accuracy, it can be seen from these values that the DGIEMM has a high ability to assign each pixel into its true label (i.e., the object that contains it). The FSC and GMM provide results better than K-means, and this observation can be noticed from Figure 7(a) that shows the average overall of the tested six images.

In terms of AR, it can be seen that the DGIEMM still provides results better than other methods. The same observations are noticed in the other three measures (i.e., RI, NMI, and Hubert); also, Figures 7(b)–7(d) shows the superiority of DGIEMM.

To justify the superiority of DGIEMM, the nonparametric Friedman test is used. In general, this test is applied to make a decision about the difference between the DGIEMM and other methods is significant or not. There are two hypotheses; the first one is named null, and it is assumed that there is no difference between the tested methods. In contrast, the second hypothesis, called alternative, is considered there is a difference between the method. We accept the alternative hypothesis, when the obtained value is less than significant level 0.05.

Table 12 shows the mean-rank obtained using the Friedman test in terms of the performance measures (i.e., accuracy, AR, RI, Hubert, NMI). From these values, it can be seen that the developed color image segmentation method has the highest mean rank in terms of performance measures. In addition, FSC allocates the second mean rank, followed by K-means that provides results better than traditional GMM. Finally, Figure 8 shows an example of Segmented image using competitive algorithms.

#### 4. Concluding Remarks

In this study, a new two-parameters distribution for modeling a lot of observations in nature has been presented. It is constructed from continuous generalized inverted exponential distribution, so called discrete generalized inverted exponential distribution DGIE distribution. Some important probabilistic properties of this distribution have been studied. Using two methods namely the moment's method and the maximum likelihood technique, the parameters of the  $\text{DGIE}(\beta, \theta)$  distribution have been estimated. To evaluate the quality of  $\text{DGIE}(\beta, \theta)$ , a set of experimental series has been conducted using synthetic and real data. The results have been shown the efficiency of  $\text{DGIE}(\beta, \theta)$  in fitting data better than some existing distribution in case of synthetic data. In addition, the developed DGIE has been applied as image segmentation based on clustering technique, which aims to avoid the limitations of the traditional Gaussian mixture model (GMM). This is achieved by using DGIE instead of Gaussian distribution. The developed image segmentation has been established its performance using a set of color images which provides results better than GMM, K-means, and Fuzzy subspace clustering (FSC).

According to these properties and results, DGIE can be applied to a wide range of applications, including reliability, physics, and machine learning techniques.

#### Data Availability

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

#### Conflicts of Interest

The authors declare no conflict of interest.

#### References

- [1] M. A. Ilani and M. Khoshnevisan, "Powder mixed-electrical discharge machining (EDM) with the electrode is made by fused deposition modeling (FDM) at Ti-6Al-4V machining procedure," *Multiscale and Multidisciplinary Modeling, Experiments and Design*, vol. 3, no. 3, pp. 173–186, 2020.
- [2] N. H. Phan, V. N. Pi, N. Q. Tuan et al., "Tool wear rate analysis of uncoated and AlCrNi coated aluminum electrode in EDM for Ti-6Al-4 V titanium alloy," *Advances in Engineering Research and Application*, pp. 832–838, 2021.
- [3] "Material removal rate during b4c abrasives mixed electrical discharge machining of titanium alloy (ti-6al-4v)," *International Journal of Latest Trends in Engineering and Technology*, vol. 7, no. 3, 2016.
- [4] R. Kwitt, P. Meerwald, and A. Uhl, "Color-image watermarking using multivariate power-exponential distribution," in *Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 4245–4248, IEEE, Cairo, Egypt, 7 November 2009.
- [5] M. A. Ilani and M. Khoshnevisan, "Study of surfactant effects on intermolecular forces (IMF) in powder-mixed electrical discharge machining (EDM) of Ti-6Al-4V," *International Journal of Advanced Manufacturing Technology*, vol. 116, no. 5-6, pp. 1763–1782, 2021.
- [6] G. Paul, J. Cardinale, and I. F. Sbalzarini, "Coupling image restoration and segmentation: a generalized linear model/bregman perspective," *International Journal of Computer Vision*, vol. 104, no. 1, pp. 69–93, 2013.
- [7] B. Ghoshal, A. Tucker, B. Sanghera, and W. Lup Wong, "Estimating uncertainty in deep learning for reporting confidence to clinicians in medical image segmentation and diseases detection," *Computational Intelligence*, vol. 37, no. 2, pp. 701–734, May 2021.
- [8] M. A. Ilani and M. Khoshnevisan, "Mathematical and physical modeling of FE-SEM surface quality surrounded by the plasma channel within Al powder-mixed electrical discharge machining of Ti-6Al-4V," *International Journal of Advanced Manufacturing Technology*, vol. 112, no. 11-12, pp. 3263–3277, 2021.
- [9] A. Taherkhani, M. A. Ilani, F. Ebrahimi et al., "Investigation of surface quality in Cost of Goods Manufactured (COGM) method of  $\mu$ -Al<sub>2</sub>O<sub>3</sub> Powder-Mixed-EDM process on machining of Ti-6Al-4V," *International Journal of Advanced Manufacturing Technology*, vol. 116, no. 5-6, pp. 1783–1799, 2021.
- [10] W. Nelson, W. Q. Meeker, and L. A. Escobar, "Statistical methods for reliability data," *Technometrics*, vol. 40, no. 3, p. 255, 1998.
- [11] S. Kotz, C. B. Read, N. Balakrishnan, B. Vidakovic, and N. L. Johnson, *Encyclopedia of Statistical Sciences*, John Wiley & Sons, Hoboken, NJ, USA, 2004.

- [12] D. Roy and R. P. Gupta, "Classifications of discrete lives," *Microelectronics Reliability*, vol. 32, no. 10, pp. 1459–1473, Oct. 1992.
- [13] D. Roy, "Reliability measures in the discrete bivariate set-up and related characterization results for a bivariate geometric distribution," *Journal of Multivariate Analysis*, vol. 46, no. 2, pp. 362–373, Aug. 1993.
- [14] H. Krishna and P. Singh Pundir, "Discrete burr and discrete pareto distributions," *Statistical Methodology*, vol. 6, no. 2, pp. 177–188, 2009.
- [15] B. A. Para and T. R. Jan, "Discrete generalized burr-type XII distribution," *Journal of Modern Applied Statistical Methods*, vol. 13, no. 2, pp. 244–258, 2014.
- [16] B. A. Para and T. R. Jan, "Discrete inverse Weibull minimax distribution: properties and applications," *Journal of Statistics Applications & Probability*, vol. 6, no. 1, pp. 205–218, 2017.
- [17] E. Gómez-Déniz and E. Calderín-Ojeda, "The discrete Lindley distribution: properties and applications," *Journal of Statistical Computation and Simulation*, vol. 81, no. 11, pp. 1405–1416, 2011.
- [18] V. Nekoukhou and H. Bidram, "Exponential-discrete generalized exponential distribution: a new compound model," *Journal of Statistical Theory and Applications*, vol. 15, no. 2, p. 169, 2016.
- [19] A. M. Abouammoh and A. M. Alshingiti, "Reliability estimation of generalized inverted exponential distribution," *Journal of Statistical Computation and Simulation*, vol. 79, no. 11, pp. 1301–1315, 2009.
- [20] A. M. Abd-Elfattah, S. M. Assar, and H. I. Abd-Elghaffar, "Exponentiated generalized Frechet distribution," *Int. J. Math. Anal. Appl.* vol. 3, no. 5, pp. 39–48, 2016.
- [21] S. Dey and T. Dey, "On progressively censored generalized inverted exponential distribution," *Journal of Applied Statistics*, vol. 41, no. 12, pp. 2557–2576, Dec. 2014.
- [22] H. Krishna and K. Kumar, "Reliability estimation in generalized inverted exponential distribution with progressively type II censored sample," *Journal of Statistical Computation and Simulation*, vol. 83, no. 6, pp. 1007–1019, Jun. 2013.
- [23] M. D. Nichols and W. J. Padgett, "A bootstrap control chart for Weibull percentiles," *Quality and Reliability Engineering International*, vol. 22, no. 2, pp. 141–151, 2006.
- [24] E. Gómez-Déniz, "Another generalization of the geometric distribution," *Test*, vol. 19, no. 2, pp. 399–415, 2010.
- [25] G. Blom, H. Linhart, and W. Zucchini, "Model selection," *Biometrics*, vol. 45, no. 1, p. 340, 1989.
- [26] P. D. McNicholas and T. B. Murphy, "Parsimonious Gaussian mixture models," *Statistics and Computing*, vol. 18, no. 3, pp. 285–296, Sep. 2008.
- [27] J. Diaz-Rozo, C. Bielza, and P. Larranaga, "Clustering of data streams with dynamic Gaussian mixture models: an IoT application in industrial processes," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3533–3547, Oct. 2018.
- [28] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, Article ID 113609, 2021.
- [29] X.-M. Zhao, "Bayesian information criterion (BIC)," in *Encyclopedia of Systems Biology* p. 73, Springer New York, New York, NY, 2013.
- [30] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): a nature-inspired meta-heuristic optimizer," *Expert Systems with Applications*, vol. 191, Article ID 116158, 2022.
- [31] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," vol. 2, pp. 416–423, in *Proceedings of the Eighth IEEE International Conference on Computer Vision*, vol. 2, IEEE, Vancouver, BC, Canada, 7 July 2001.

## Research Article

# Distributed Deep CNN-LSTM Model for Intrusion Detection Method in IoT-Based Vehicles

Ali Alferaidi,<sup>1</sup> Kusum Yadav,<sup>1</sup> Yasser Alharbi ,<sup>1</sup> Navid Razmjooy ,<sup>2</sup>  
Wattana Viriyasitavat ,<sup>3</sup> Kamal Gulati ,<sup>4</sup> Sandeep Kautish ,<sup>5</sup> and Gaurav Dhiman ,<sup>6,7</sup>

<sup>1</sup>College of Computer Science and Engineering, University of Ha'il, Ha'il, Saudi Arabia

<sup>2</sup>Department of Electrical Engineering, Ardabil Branch, Islamic Azad University, Ardabil, Iran

<sup>3</sup>Department of Statistics, Chulalongkorn Business School, Faculty of Commerce and Accountancy, Bangkok, Thailand

<sup>4</sup>Amity School of Insurance, Banking and Actuarial Science, Amity University, Noida, India

<sup>5</sup>LBEF Campus, Kathmandu, Nepal

<sup>6</sup>Department of Computer Science, Government Bikram College of Commerce, Patiala 147001, Punjab, India

<sup>7</sup>University Centre for Research and Development, Department of Computer Science and Engineering, Chandigarh University, Gharuan, Mohali, India

Correspondence should be addressed to Sandeep Kautish; [dr.skautish@gmail.com](mailto:dr.skautish@gmail.com)

Received 25 November 2021; Revised 15 January 2022; Accepted 3 February 2022; Published 19 March 2022

Academic Editor: Ramin Ranjbarzadeh

Copyright © 2022 Ali Alferaidi 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.

As 5G and other technologies are widely used in the Internet of Vehicles, intrusion detection plays an increasingly important role as a vital detection tool for information security. However, due to the rapid changes in the structure of the Internet of Vehicles, the large data flow, and the complex and diverse forms of intrusion, traditional detection methods cannot ensure their accuracy and real-time requirements and cannot be directly applied to the Internet of Vehicles. A new AA distributed combined deep learning intrusion detection method for the Internet of Vehicles based on the Apache Spark framework is proposed in response to these problems. The cluster combines deep-learning convolutional neural network (CNN) and extended short-term memory (LSTM) network to extract features and data for detection of car network intrusion from large-scale car network data traffic and discovery of abnormal behavior. The experimental results show that compared with other existing models, the algorithm of this model can reach 20 in the fastest time, and the accuracy rate is up to 99.7%, with a good detection effect.

## 1. Introduction

With the practical application of emerging technologies in the field of the Internet of Vehicles, the development of the Internet of Vehicles has become more rapid. Due to its particularity, that is, the car itself does not consider network security enough, the capacity of the vehicle is limited, the application environment is complex, the number of distributed nodes and sensor networks are many, and the safety requirements are incredibly high. Therefore, the security issue of the Internet of Vehicles has increasingly become a stumbling block to its application. Ensuring the security of car G road G cloud communications in the car networking security system, identifying various malicious attacks, has

become the focus of close attention by industry insiders and information security experts. Intrusion detection is a network security technology used to detect intruders and aggression in any communication system through various identifications or detections.

Attack behavior, monitor and analyze network traffic, classify normal and abnormal behavior, and identify strange activities such as threats in the network are all roles played by the Internet of Vehicles. As an active defense technology, this technology has become one of the primary mechanisms to ensure the safety of the Internet of Vehicles. The application of machine learning algorithms in traditional Internet intrusion detection systems is the current mainstream research direction. Wisanwanichthan and Thammawichai [1]



apply the machine learning method to intrusion detection systems (IDS), and use SVM and Naive Bayes algorithms for normalization and feature reduction for analysis and comparison. However, the key disadvantage of the machine learning-based intrusion detection mechanism is that it requires a lot of training time to process many datasets of previous data streams in the network. In the network environment, deep learning technology has good self-learning functions, Lenovo storage functions, and high-speed optimization functions, which are very suitable for processing the current complex network traffic data, especially in the complex car networking environment.

At the moment, there is a great deal of research being conducted on intrusion detection using deep learning and distributed big data technologies. Chen et al. [2] developed a hybrid deep neural network (DNN) model for classifying and detecting unknown network threats. Chen et al. [2] think that deep learning has received a lot of attention recently, and they compared conventional techniques to new deep learning methods. Chen et al. [3] constructed an intelligent intrusion detection system using deep learning's intelligent capabilities. Vijayanand et al. [3] presented a technique for detecting anomalous intrusions using a hybrid MLP/CNN. Parimala and Kayalvizhi [4] developed a deep learning-based technique for detecting network intrusions. The KDD-CUP99 dataset was examined using the BP neural network to identify the kinds of invasions. Karatas et al. [5] developed an intrusion detection technique based on deep convolutional neural networks, which lowers the dimensionality of network data by converting it to pictures. The detection accuracy, false alarm rate, and detection rate are all enhanced via training and recognition. Shettar et al. [6] utilized Keras on top of TensorFlow to categorize various assaults using supervised deep learning and achieved the best accuracy using RNN deep learning technology. Zhang et al. [7] implemented random forests and SVMs using the Spark framework. Other machine learning methods were evaluated and compared to multilayer deep perceptions. We may conclude from studies that although deep learning algorithms are more accurate than conventional machine learning algorithms, they need more time to examine data. The static network in its traditional form intrusion detection is often classified as either host-based or network-based. The Internet of cars' intrusion detection is accomplished by filtering the data transferred between vehicles. Due to the fact that the Internet of Vehicles is also linked to the Internet or to a specialized network, traditional harmful attack techniques are also successful on the Internet. They are more damaging, which necessitates more stringent standards for intrusion detection protection. Combining the features of the Internet of Vehicles' massive traffic and multidimensional complexity, the application of deep neural network detection. Due to the benefits of distributed parallel computing and its rapid and influential features, this article proposes using a combined deep learning algorithm with the Spark framework [8, 9] for intrusion detection. By utilizing the Spark architecture, the traditional deep learning algorithm is improved. Combining CNN and LSTM, Dey [10] proposed the CNNGLSTM algorithm model, which was used to analyze the NSL-KDD dataset [11] and the UNSW-NB15 dataset [12–15] in order to minimize security attacks on

connected vehicles. Its primary objective is to decrease the time needed to identify assaults and increase the accuracy of classification jobs, which is more appropriate for the Internet of Vehicles' real environment. Each indication has been enhanced as a result of experimental research. Researchers are proposing various protocol schemes [16–20] to maintain the integrity, confidentiality, and security of the information shared among users and servers.

The rest of this paper is structured as follows: Section 2 describes the CNN-LSTM algorithm. The Spark framework and NSL-KDD dataset are mentioned in Sections 3 and 4, respectively. Result analysis is given in Section 5, followed by the conclusions in Section 6.

## 2. CNN-LSTM Algorithm

CNN is suitable for extracting data features; LSTM is suitable for processing time series, solving the dependency problem between time-series data, and improving recognition accuracy. This paper combines the advantages of the two algorithms and proposes the CNNGLSTM algorithm. Convolution neural network (CNN) [21] evolved from multilayer perception (MLP) [22]. Compared with traditional feature selection algorithms, this algorithm can learn features better. The more traffic data CNN can learn, the more useful features there are, the better the classification, which is suitable for large-scale network environments. As shown in Figure 1, its structure is divided into a convolution layer, a pooling layer, and a fully connected layer. The role of the convolution layer is to extract features, and the role of the pooling layer is to sample the features. Finally, the fully connected layer is responsible for connecting the extracted features and obtaining the classification results through the classifier.

The long-term memory network (LSTM) is an improved recurrent neural network (RNN) method, which aims to alleviate the explosion gradient problem. Compared with traditional RNN units, LSTM uses a set of gate functions to control feedback so that short-term errors will eventually be deleted while persistent features will be retained. The data processing flow is shown in Figure 2.

The LSTM is abstracted into four subnets (p-net, g-net, f-net, and q-net), a collection of gate controllers, and a link to the memory component. The figure's input and output are controlled by the vector's size,  $x(t)$ . The state  $s(t)$  contains information about the present learning.

The CNN-LSTM method is capable of expressing both temporal and spatial information. Due to the fact that an intrusion assault occurs in real time, the methods of attack are varied, as is the target or point of attack. To extract features, a CNN is utilized, and high-level features may be retrieved using the convolution kernel operation, which has been successfully used in image processing [23–25]. Additionally, LSTM utilizes gate functions to regulate the remembering and forgetting of previous data, making it ideal for processing long-term sequence data and increasing detection accuracy [26, 27]. As a result, the CNN-LSTM algorithm model is suitable for intrusion detection processing in this study. Figure 3 illustrates the CNN-LSTM algorithm paradigm, and the particular stages are as follows:

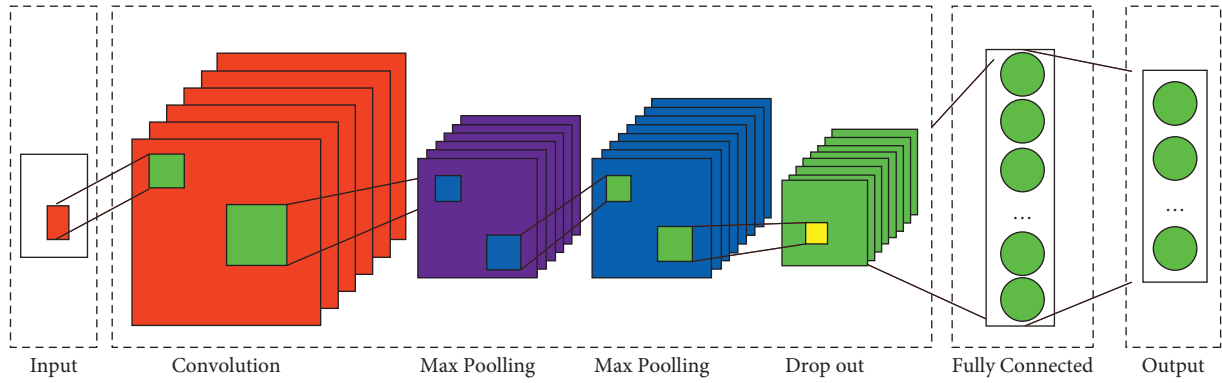


FIGURE 1: CNN architecture.

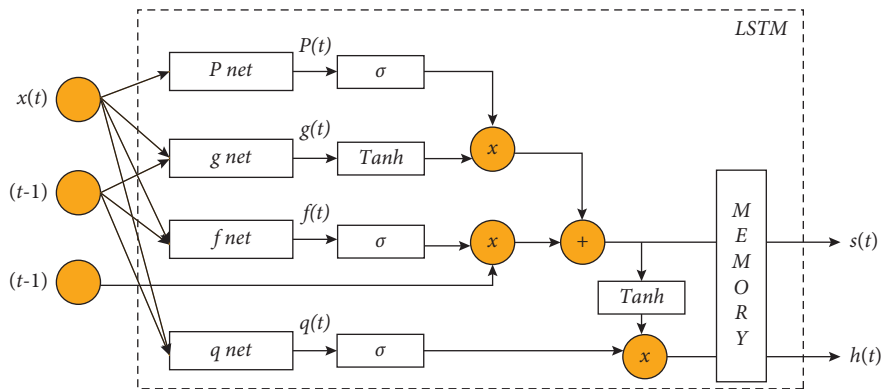


FIGURE 2: LSTM data processing diagram.

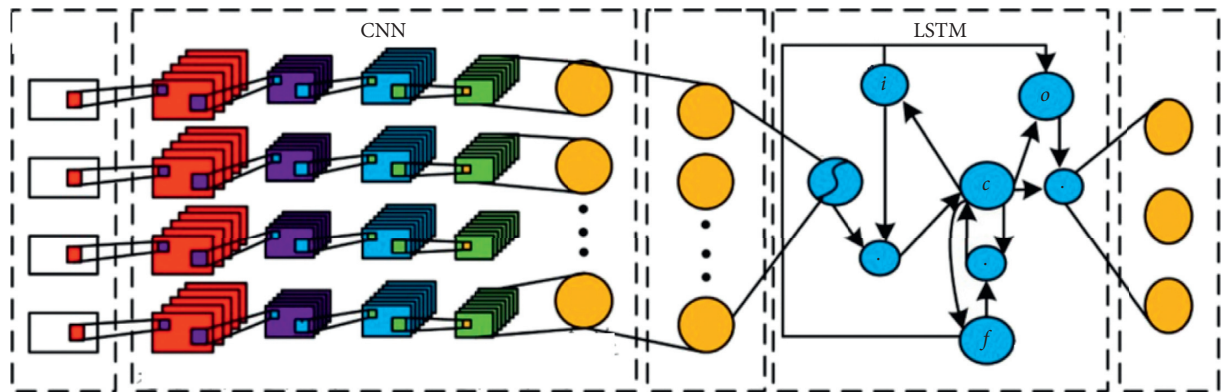


FIGURE 3: CNN-LSTM model architecture.

- (1) The input layer collects real-time Internet of Vehicles data through the flow data collection module. This article uses the dataset to analyze characteristics, including network protocol types, network service types, network connection status, and connection time [28, 29].
- (2) According to the data processing steps, the data are respectively preprocessed, digitized, and normalized. The specific operation steps will be described in detail later.
- (3) It sends the processed data to the convolution layer for feature extraction and outputs the features

through a one-dimensional convolution operation. Each convolution layer is accompanied by a pooling layer to reduce feature dimensions, accelerate convergence, and remove redundancy features to prevent network overfitting. Then all local features are integrated through the fully connected layer to form an overall feature. Finally, the leaky ReLU activation function in the fully connected layer is operated [30–32].

- (4) Input the features extracted by CNN into LSTM. After the SoftMax function, the classification result of network data is obtained [23, 33, 34].

### 3. Spark Framework

To enhance detection efficiency, this study makes use of the Apache Spark framework, a large data processing platform focused on speed, simplicity of use, and sophisticated analysis. It was created in 2009 [8–10] at the University of California, Berkeley, and became one of the Apache open-source projects. In comparison to other big data technologies such as Hadoop, Storm, and MapReduce, Spark offers the following advantages [35–37]:

- (a) Spark offers a consistent and comprehensive framework for handling diverse datasets and data sources (batch or real-time streaming data) with varying characteristics (text data, chart data, etc.) [38, 39].
- (b) Spark improves the performance of Hadoop cluster apps operating in memory by 100 times and the speed of Hadoop cluster applications running on the disc by ten times [14, 40].
- (c) When compared to MapReduce, Spark performs quicker data calculations and offers more robust functions [25, 41].

When the quantity of processed data surpasses the capacity of a single machine (for example, a computer with 4 GB of memory must process more than 100 GB of data), or when the amount of processed data is trivial, nonetheless, the calculation is difficult and time-consuming. As a result, the Spark cluster can use its massive computational capabilities to perform the analysis in an organized fashion. The architecture's schematic design is shown in Figure 4.

Using the Spark distributed open-source framework, the experimental PCs are connected to form a master-slave control structure. The master node performs task scheduling, distribution, and fault tolerance on the slave nodes, and the slave nodes realize parallel computing. This structure has been proven to be an owner of a distributed design with high reliability, high concurrency, and high-performance computing capabilities. The HDFS storage system of the node is then used to store the data, and the combined deep learning algorithm is used for intrusion detection.

### 4. NSL-KDD Dataset

In contrast to a conventional network, the heterogeneous communication network created by vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication are formed by the self-organization of vehicle nodes. Driving, fast channel fading, strong Doppler effect, and rapid network topology changes are all examples of rapid network topology changes. However, the attack techniques used against the Internet of Vehicles throughout the communication process are very similar to those used against conventional networks, including backdoor assaults and denial of service attacks. To evaluate the proposed Spark-based distributed combined deep learning intrusion detection method for the Internet of Vehicles, the proposed deep learning algorithm is applied to two intrusion detection benchmark datasets, namely, NSL-

KDD [11] and UNSW-NB15 [12], in order to develop an effective intrusion detection system for the Internet of Vehicles' external communication. There are a total of 21,473 pieces of training data and 51,025 pieces of test data in the experimental dataset.

The NSL-KDD dataset is a refinement of the KDD CUP 99 data collection [12]. It eliminates redundant records from the CUP 99 dataset and addresses the classifier's bias for repeating records. In comparison to the KDD 99 dataset, the usage of NSL classification of the KDD dataset will provide comparable or superior accuracy. As a result, it is widely regarded as one of the most effective datasets for intrusion detection studies. The dataset's assaults are classified into four groups.

- (1) Denial of service (DoS): the intruder will send many malicious requests to the server, causing the machine's memory and computational resources to become insufficiently full or busy to handle genuine traffic, thus denying regular users services.
- (2) User-to-root (U2R): this is a kind of attack in which the attacker tries to acquire administrator privileges through regular user access.
- (3) Remote-to-local attack (R2L): the attacker wishes to transmit data to a computer via a network in order to obtain access to the machine fraudulently.
- (4) Detection attack (Probe): the network is scanned to obtain detailed information about the user's device.

In addition, the dataset contains 49 features, which constitute the traffic that exists between the host and the network data packet and are used to distinguish normal or abnormal observation results. Compared with other datasets, it contains both real-scene data and synthetic data. Attack behavior and the complexity of UNSW-NB means dataset are valid and reliable.

### 5. Result Analysis

The CNN-LSTM algorithm and SVM, RNN, CNN, and LSTM algorithms are used to compare the accuracy rate (AC) and false alarm rate (FPR) of different attack types. The CNN-LSTM method has a high classification detection rate. Compared with other algorithms, it has a lower false alarm rate.

To verify the overall effectiveness and comparison of the experiment, this paper uses two datasets of NSL-KDD and UNSW GNB15 to compare the accuracy rate (AC) and false alarm rate (FPR) of the above five algorithms. The experimental results are shown in Figures 5 and 6.

It can be seen from Figures 7 and 8 that CNN-LSTM performs well in the NSL-KDD dataset and UNSWGNB15 dataset reaching 7% and 99%, respectively. The accuracy rate of 4% also has the lowest false alarm rate of two, respectively, 24% and 2.17%. Therefore, this algorithm has better performance characteristics among similar algorithms.

All the deep learning algorithms discussed in this article are implemented in a distributed manner under Apache Spark. The experimental results are shown in Figures 7 and 8.

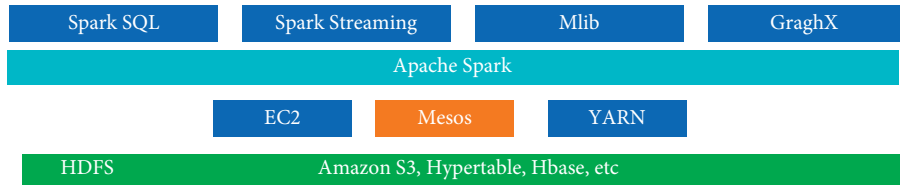


FIGURE 4: Apache spark architecture.

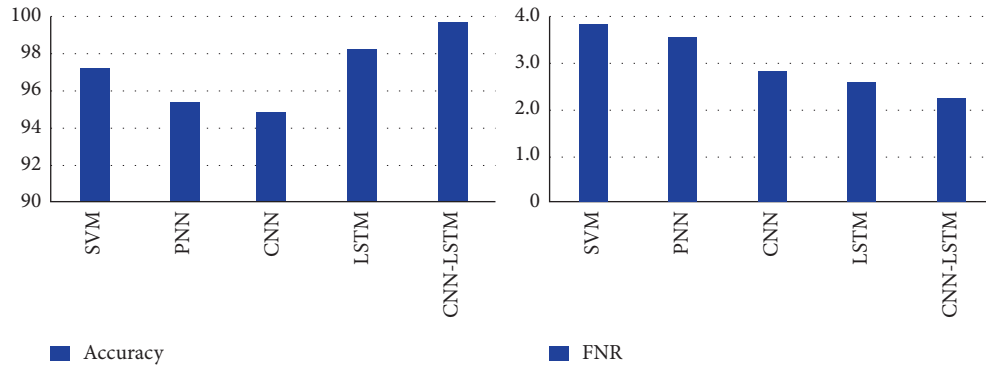


FIGURE 5: Performance comparison of various algorithms under NSL-KDD dataset.

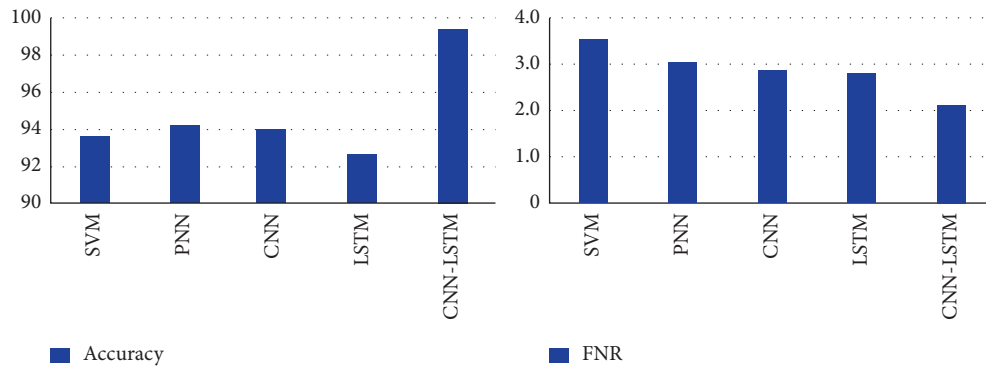


FIGURE 6: UNSW-NB15 dataset performance comparison of each algorithm.

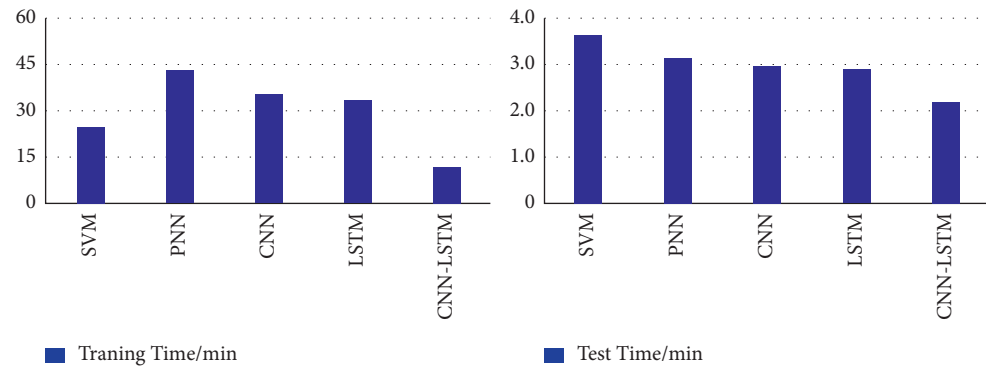


FIGURE 7: Detection time of each algorithm under the NSL GKDD dataset.

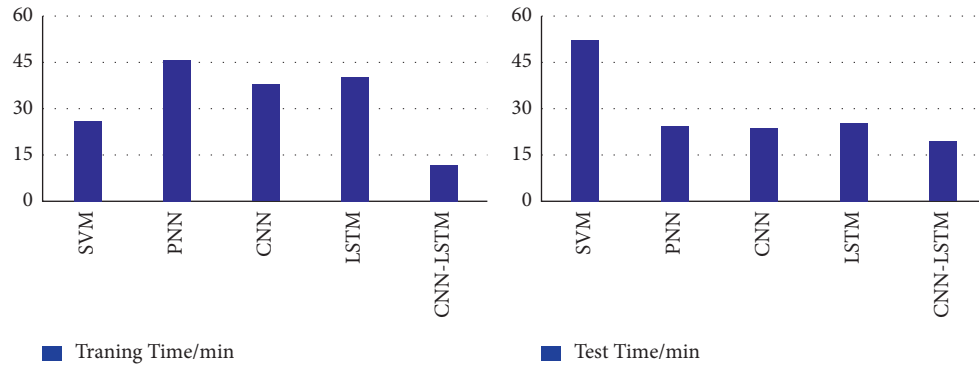


FIGURE 8: Detection time of each algorithm under UNSWGNB15 dataset.

It can be seen that compared with traditional nonparallel machines and deep learning algorithms, the training and testing time is significantly shortened. Furthermore, the experimental results show that the training time and test time used by the CNN-LSTM algorithm are the shortest.

## 6. Conclusion

Comparative experiments found that because of the slow detection speed and low detection efficiency of big data in intrusion detection systems, the advantages of distributed frameworks and deep learning algorithms are fully considered, and the distributed architecture is combined with the deep learning CNN-LSTM algorithm. Through data, the detection efficiency and detection time are improved after the data is standardized by preprocessing and other methods. Experimental verification on the NSL-KDD dataset and the UNSW-NB15 dataset shows that the deep learning algorithm of CNN-LSTM using the Spark framework is comparable to other deep learning algorithms. It reduces the training time and test time, improves the detection rate, can well meet the real-time requirements of intrusion detection, and more satisfies the actual needs of the Internet of Vehicles for intrusion detection. In the next step, this article will improve based on intrusion detection performance and reduction of detection time, and we will further focus on the detection capabilities of deep learning algorithms, conduct intrusion detection on distributed platforms, and explore suitable distributed deep learning algorithms to meet the needs of intrusion detection for car network information security. A more efficient algorithm handles the network data traffic of the Internet of Vehicles and enhances the adaptability of the algorithm [42].

## Data Availability

The data used to support the findings of this study are available from the author upon request (kusumasyadav0@gmail.com).

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] T. Wisanwanichthan and M. Thammawichai, "A double-layered hybrid approach for network intrusion detection system using combined naive Bayes and SVM," *IEEE Access*, vol. 9, pp. 138432–138450, 2021.
- [2] P. Chen, Y. Guo, J. Zhang, Y. Wang, and H. Hu, "A novel preprocessing methodology for DNN-based intrusion detection," in *Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pp. 2059–2064, Chengdu, China, December 2020.
- [3] L. Chen, X. Kuang, A. Xu, S. Suo, and Y. Yang, "A novel network intrusion detection system based on CNN," in *Proceedings of the 2020 Eighth International Conference on Advanced Cloud and Big Data (CBD)*, pp. 243–247, Taiyuan, China, December 2020.
- [4] R. Vijayanand, D. Devaraj, and B. Kannapiran, "A novel deep learning based intrusion detection system for smart meter communication network," in *Proceedings of the 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, pp. 1–3, Tamilnadu, India, April 2019.
- [5] G. Parimala and R. Kayalvizhi, "An effective intrusion detection system for securing IoT using feature selection and deep learning," in *Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–4, Coimbatore, India, January 2021.
- [6] G. Karatas, O. Demir, and O. Koray Sahingoz, "Deep learning in intrusion detection systems," in *Proceedings of the 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, pp. 113–116, Ankara, Turkey, December 2018.
- [7] P. Shettar, A. V. Kachavimath, M. M. Mulla, and D. G. Narayan, "Intrusion detection system using MLP and chaotic neural networks," in *Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–4, Coimbatore, India, January 2021.
- [8] H. Zhang, S. Dai, Y. Li, and W. Zhang, "Real-time distributed-random-forest-based network intrusion detection system using Apache Spark," in *Proceedings of the 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–7, Orlando, FL, USA, November 2018.
- [9] S. V. Siva reddy and S. Saravanan, "Performance evaluation of classification algorithms in the design of Apache Spark based intrusion detection system," in *Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 443–447, Coimbatore, India, June 2020.

- [10] A. Dey, "Deep IDS: a deep learning approach for Intrusion detection based on IDS 2018," in *Proceedings of the 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pp. 1–5, Dhaka, Bangladesh, December 2020.
- [11] P. S. Bhattacharjee, A. K. Md Fujail, and S. A. Begum, "A comparison of intrusion detection by K-means and fuzzy C-means clustering algorithm over the NSL-KDD dataset," in *Proceedings of the 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICIC)*, pp. 1–6, Coimbatore, India, December 2017.
- [12] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, Canberra, ACT, Australia, November 2015.
- [13] T. Alladi, V. Kohli, V. Chamola, F. R. Yu, and M. Guizani, "Artificial intelligence (AI)-empowered intrusion detection architecture for the Internet of Vehicles," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 144–149, 2021.
- [14] M. Mahdavisarif, S. Jamali, and R. Fotohi, "Big data-aware intrusion detection system in communication networks: a deep learning approach," *Journal of Grid Computing*, vol. 19, no. 4, pp. 1–28, 2021.
- [15] S. Jamali and R. Fotohi, "DAWA: defending against wormhole attack in MANETs by using fuzzy logic and artificial immune system," *The Journal of Supercomputing*, vol. 73, no. 12, pp. 5173–5196, 2017.
- [16] R. Nair, M. Soni, B. Bajpai, G. Dhiman, and K. M. Sagayam, "Predicting the death rate around the world due to COVID-19 using regression analysis," *International Journal of Swarm Intelligence Research*, vol. 13, no. 2, pp. 1–13, 2022.
- [17] P. K. Vaishnav, S. Sharma, and P. Sharma, "Analytical review analysis for screening COVID-19 disease," *International Journal of Modern Research*, vol. 1, no. 1, pp. 22–29, 2021.
- [18] I. Chatterjee, "Artificial intelligence and patentability: review and discussions," *International Journal of Modern Research*, vol. 1, no. 1, pp. 15–21, 2021.
- [19] R. Kumar and G. Dhiman, "A comparative study of fuzzy optimization through fuzzy number," *International Journal of Modern Research*, vol. 1, no. 1, pp. 1–14, 2021.
- [20] M. Soni, G. Dhiman, B. S. Rajput, P. Rajan, and T. Nitesh Kumar, "Energy-Effective and Secure Data Transfer Scheme for Mobile Nodes in Smart City Applications," *Wireless Pers Commun*, 2021.
- [21] W. Cao, "CNN-based intelligent safety surveillance in green IoT applications," *China Communications*, vol. 18, no. 1, pp. 108–119, 2021.
- [22] M. Ganesan and N. Sivakumar, "IoT based heart disease prediction and diagnosis model for healthcare using machine learning models," in *Proceedings of the 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, pp. 1–5, Pondicherry, India, March 2019.
- [23] R. Fotohi, S. Firoozi Bari, and M. Yusefi, "Securing wireless sensor networks against denial-of-sleep attacks using RSA cryptography algorithm and interlock protocol," *International Journal of Communication Systems*, vol. 33, no. 4, Article ID e4234, 2020.
- [24] A. Derhab, A. Aldweesh, A. Z. Emam, and F. A. Khan, "Intrusion detection system for Internet of Things based on temporal convolution neural network and efficient feature engineering," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 6689134, 16 pages, 2020.
- [25] R. Fotohi and S. F. Bari, "A novel countermeasure technique to protect WSN against denial-of-sleep attacks using firefly and Hopfield neural network (HNN) algorithms," *The Journal of Supercomputing*, vol. 76, pp. 1–27, 2020.
- [26] M. Zaminkar and R. Fotohi, "SoS-RPL: securing internet of things against sinkhole attack using RPL protocol-based node rating and ranking mechanism," *Wireless Personal Communications*, vol. 114, 2020.
- [27] G. D. L. T. Parra, P. Rad, K. K. R. Choo, and N. Beebe, "Detecting Internet of Things attacks using distributed deep learning," *Journal of Network and Computer Applications*, vol. 163, Article ID 102662, 2020.
- [28] R. Fotohi, E. Nazemi, and F. S. Aliee, "An agent-based self-protective method to secure communication between UAVs in unmanned aerial vehicle networks," *Vehicular Communications*, vol. 26, Article ID 100267, 2020.
- [29] M. Roopak, G. Y. Tian, and J. Chambers, "An intrusion detection system against ddos attacks in iot networks," in *Proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0562–0567, IEEE, Las Vegas, NV, USA, January 2020.
- [30] T. Alladi, A. Agrawal, B. Gera, V. Chamola, B. Sikdar, and M. Guizani, "Deep neural networks for securing IoT enabled vehicular ad-hoc networks," in *Proceedings of the ICC 2021-IEEE International Conference on Communications*, pp. 1–6, IEEE, Montreal, QC, Canada, June 2021.
- [31] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, "A hierarchical hybrid intrusion detection approach in iot scenarios," in *Proceedings of the GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–7, IEEE, Taipei, Taiwan, December 2020.
- [32] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," 2018, <https://arxiv.org/abs/1802.09089>.
- [33] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine," *Electronics*, vol. 9, no. 1, p. 173, 2020.
- [34] G. J. M. Ariyathilake, M. H. R. Sandeepanie, and P. L. Rupasinghe, "SQL injection detection and prevention solution for web applications," 2021, <http://ir.kdu.ac.lk/handle/345/5253>.
- [35] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Generation Computer Systems*, vol. 82, pp. 761–768, 2018.
- [36] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, "A big data-enabled hierarchical framework for traffic classification," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2608–2619, 2020.
- [37] G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé, "Know your big data trade-offs when classifying encrypted mobile traffic with deep learning," in *Proceedings of the 2019 Network traffic measurement and analysis conference (TMA)*, pp. 121–128, IEEE, Paris, France, June 2019.
- [38] Z. S. Alwan and M. F. Younis, "Detection and prevention of SQL injection attack: a survey," *International Journal of Computer Science and Mobile Computing*, vol. 6, no. 8, pp. 5–17, 2017.
- [39] M. Nasereddin, A. ALKhamaiseh, M. Qasaimeh, and R. Al-Qassas, "A systematic review of detection and prevention techniques of SQL injection attacks," *Information Security Journal: A Global Perspective*, pp. 1–14, 2021.

- [40] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Computer Networks*, vol. 136, pp. 37–50, 2018.
- [41] R. Fotohi and H. Pakdel, "A lightweight and scalable physical layer attack detection mechanism for the internet of things (IoT) using hybrid security schema," *Wireless Personal Communications*, vol. 119, pp. 1–18, 2021.
- [42] M. Lei, X. Li, B. Cai, Y. Li, L. Liu, and W. Kong, "P-DNN: an effective intrusion detection method based on pruning deep neural network," in *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, Glasgow, UK, July 2020.

## Research Article

# Identification of Chaos in Financial Time Series to Forecast Nonperforming Loan

Farid Ahmadi <sup>1</sup>, Mohammad Pourmahmood Aghababa,<sup>2</sup> and Hashem Kalbkhani<sup>2</sup>

<sup>1</sup>Information Technology Engineering Department, Urmia University of Technology, Urmia, Iran

<sup>2</sup>Faculty of Electrical Engineering, Urmia University of Technology, Urmia, Iran

Correspondence should be addressed to Farid Ahmadi; f.ahmadi@uut.ac.ir

Received 26 October 2021; Revised 24 December 2021; Accepted 19 January 2022; Published 4 February 2022

Academic Editor: Yong Aaron Tan

Copyright © 2022 Farid Ahmadi 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 discusses the importance of modeling financial time series as a chaotic dynamic rather than a stochastic system. The dynamical properties of a financial time series of an economic institution in Iran were analyzed to identify the potential occurrence of the low-dimensional deterministic chaos. This paper applies several classic nonlinear techniques for detecting the chaotic nature of the time series of loan payment portion and proposes a modified nonlinear predictor scheme for forecasting the future levels of the nonperforming loan. The auto mutual information was implemented to estimate the delay time dimension, and Cao's approach, along with correlation dimension methodology, quantified the embedding dimension of the time series. The results reveal a low embedding dimension implying the chaotic nature exists in the financial data. The maximum Lyapunov exponent measure is also adopted to investigate the divergence or convergence of the trajectories. Since positive Lyapunov exponents are revealed, the long-term unpredictability of the time series is proved. Lastly, a modified nonlinear local approximator is developed to forecast the short-term history of the time series. Numerical simulations are provided to illustrate the adopted nonlinear techniques. The results reported in this paper could have implications for commercial bank managers who could use the nonlinear models for early detection of the possible nonperforming loans before they become uncontrollable.

## 1. Introduction

Banks and financial institutions have played a significant role in balancing the economic life of the people in recent decades owing to the development of the countries and the development of new financial opportunities for the merchants. Lending is one of the primary and popular approaches in such financial organizations aiming to make a loan to somebody on the condition that the amount borrowed is to be returned, usually with an interest fee. However, in some cases, the nonperforming loan (NPL) problem occurs when the borrowed money is not returned in the scheduled period. High levels of NPL mean reducing the income of the banks, which in turn leads to severe economic losses. Therefore, governments have paid more attention to this issue in recent years. Accordingly, a weakening in bank loan services may cause a delay in economic growth and can be a good reason for the economic

crisis. It has been argued that NPLs may create economic stagnation and, therefore, can deter economic growth and weaken financial efficacy [1]. On the other hand, high levels of NPLs harm the growth rate of gross domestic product [2]. Given the importance of this issue, the governments and financial sectors should encourage researchers to conduct studies on this issue and banks to implement research results in practice.

As a result, NPL ratios should stand at low or manageable levels before the crisis. The NPL ratio prediction method is necessary to achieve this goal based on previous information. Various studies have attempted to study the relationship between various economic factors and the NPLs to forecast the NPLs. In [1], a heuristic hybrid classification method has been used to predict banks' nonperforming loans using some macroeconomic and bank-specific features. Tang et al. [3] have used a stepwise discrimination algorithm to find essential factors for building distance



discrimination and Bayesian discrimination models to determine whether an NPL has a zero or positive recovery rate. In [4], three two-phase mixture models of logistic regression and artificial neural networks have been developed to create an economic distress warning system appropriate for Taiwan's banking business. The application of a neural network predictor in forecasting loan recovery in the Nigerian financial institutions has been reported in [5]. In [6], the principal component analysis (PCA) technique has been adopted for feature selection for prediction models of Chinese bank loans. Then, the models have been assessed using the TOPSIS multicriteria decision-making approach. Saha et al. [7] have utilized machine learning strategies such as ontology, text, data mining, and multiagent approaches to develop a knowledge-determined automatic acquiescence auditing method for bank loans. Some studies have used other techniques, such as nonlinear regression models for loan prediction [8, 9].

Most of the studies mentioned above have tried to check whether a particular financial or banking attribute influences NPLs quotient or not. In most cases, just some measurable quantities which depend on the underlying and usually unknown dynamics of the NPL rate are available. Since many factors affect the ratio of NPLs, the inherent nature of NPLs becomes more complicated and nondeterministic. In such situations, linear estimators that minimize the variance fail to reach the best fitting or forecasting purposes [10]. Moreover, the designer should correctly set many model parameters in the previous nonlinear models (such as neural networks). Therefore, it is hard to predict and interpret the long-term future of NPLs with limited features. The complex behavior of the NPLs cannot be easily modeled using the common linear or nonlinear statistical approaches, such as autoregressive methods and neural networks. Thus, it would be better to use alternative nonlinear powerful techniques that utilize the inherent attributes of the NPLs' nature, such as chaos [11]. In the past few decades, the concept of nonlinear dynamical systems and chaos theory has changed the treated manner of financial systems. The complicated behavior of NPLs can be interpreted using a chaotic system (as well as chaos theory) with high sensitivity to the initial conditions.

Chaos is a particular version of nonlinear dynamics that possesses unique attributes such as incredible sensitivity to the initial states (a tiny change in the starting point will produce a significant diversity in the future), broad Fourier transform spectrum, irregular attractors (the states are locally unbounded but globally bounded), and fractality property. A chaotic phenomenon is neither stochastic nor random; it is a deterministic system in which some equations are available to determine its behavior. Moreover, although the chaotic time series are not periodic and random, they exhibit a sense of order and pattern. Such unique and complicated appearances can be detected using nonlinear techniques, including phase space reconstruction, false nearest neighbor (FNN) algorithm, correlation dimension method, and Lyapunov exponent. Consequently, many real-world applied systems that may seem to have random nature can possess a nonlinear deterministic and potentially chaotic

behavior. In this line, with the introduction of the ability to model and predict the future of chaotic time series through nonlinear deterministic system theory, many researchers have attempted to study and detect chaos in time series of multidisciplinary fields [12, 13]. Also, complex nonlinear dynamical modeling and analysis have recently become more important in advanced science approaches [14–16] and [17]. However, a chaotic time series analysis is still a good choice in many economic-related issues.

In [18], Novikov's theorem has been utilized to model the complicated dynamics of noisy credit risk contagion with time delay, and the Hopf bifurcation and chaotic behaviors are evaluated. In [19], some numerical approaches have been applied to discover Hopf bifurcation, inverse bifurcation, and chaos phenomena in the credit risk contagion dynamics. Lahmiri [20] has investigated the fractal inherence and chaotic behavior in returns and volatilities of family business companies of Morocco, using Hurst exponent and an autoregressive model. In [21], the phase synchronization method has been introduced for analyzing the chaotic behavior of stock price and index movements in crisis stages. For identifying the quality of similarity measure of financial time series, three techniques including information categorization approach, reconstructed phase space clustering strategy, and system methodology with squared Euclidean distances have been used in [22]. In addition to these works, some research in the literature aims to predict the chaotic financial time series. In [23], a self-organizing map neural network along a recommender system has been proposed to cluster and predict stock price time series. Yang and Lin [14] have applied empirical mode decomposition and phase space reconstruction methods combined with extreme learning machines for predicting financial exchange rates' time series forecasting. The artificial neural network method is the main and mostly applied technique in the literature for predicting the future of the financial chaotic time series [24–27]. To the best of our knowledge, no single review article detecting and predicting chaos in NPLs' time series has been published; therefore, it will be addressed in the present work.

This research aims to investigate the financial time series behavior of the primary interest-free institution of Iran named Omid Entrepreneurship Fund (OEF). The main purpose is to identify the potential occurrence of low-dimensional deterministic chaos in the financial time series of OEF and to propose an efficient forecasting technique to predict the future events of NPLs. The dataset is collected during the last five years of the OEF activity. We adopt the loan payment percent (LPP) as the time series to be analyzed. We try to predict the future of the LPP with a modified chaos-based nonlinear methodology. As a result, the predicted values can be utilized to detect and predict NPLs (an NPL can be determined by an LPP less than a specific value in a given time) before they become uncontrollable. The superposition of all exogenous and endogenous variables affects the loan payment percent. Hence, in this paper, the effects of exogenous and endogenous were considered indirectly. The delay time dimension is discovered using auto mutual information to investigate the chaos in the time

series. It is employed to rebuild the irregular attractors. The embedding dimension is calculated via Cao's technique and correlation dimension approach. The quantity of the sensitivity to the initial states is computed with the help of the largest Lyapunov exponent (LLE) principle. The given positive Lyapunov exponents assure the exponential deviation of the trajectories and, therefore, the unpredictability of the time series. Finally, a modified nonlinear predictor is realized to forecast the financial time series. All the mentioned approaches are illustrated using computer simulations.

The rest of this article is organized as follows. In Section 2, the nonlinear chaos-based techniques are presented. Section 3 deals with the description of a modified prediction scheme. In Section 4, some computer simulations are carried out. Finally, concluding remarks are provided in Section 5.

## 2. Nonlinear Techniques for Analyzing Chaotic Time Series

This section discusses the adopted nonlinear techniques for chaos detection in the financial time series.

**2.1. Phase Space Rebuilding.** Since the chaotic dynamics of an irregular time series are not precisely known, a phase space estimation of the state space of a time series can be effectively used to rebuild the equivalent irregular attractor. To this end, the time series is converted into the geometry of a single moving point along a chaotic trajectory, where each of its points corresponds to a state of the analogous chaotic system. As a result, the phase space rebuilding can be inferred as a multidimensional depiction of a single-dimensional nonlinear time series. A delay-based applied method for rebuilding the phase space has been introduced by Takens [28]. The main idea of the Takens method is that the matching trajectories of the systems with a set of naturally deterministic dynamics can reach toward the subset of the phase space (i.e., the attractor). So, that technique assures that an  $m$ -dimensional space can be created to embed the original behavior of a given time series  $X_i, i = 1, 2, \dots, N$ , in which each element of the state vector is obtained via the delay coordinates as below [28]:

$$Y_j = (X_j, X_{j+\tau}, X_{j+2\tau}, \dots, X_{j+(m-1)\tau}), \quad (1)$$

where  $j = 1, 2, \dots, N - (m-1)\tau/\Delta t$ ,  $m$  is named the embedding dimension ( $m \geq d$  where  $d$  shows the dimension of the attractor),  $\tau$  denotes the delay time, and  $\Delta t$  represents the sampling time. Computational errors caused by a finite precision arithmetic allow to consider one pseudotrajectory computed for a sufficiently large time interval [29–31].

**2.2. Time Delay Estimation.** Two main approaches can carry out a proper estimation of the time delay  $\tau$ . In the first strategy, one can calculate the autocorrelation function of the time series and choose the first zero-crossing time. In this strategy, the  $X_{i+\tau}$  sample can be fully decorrelated from the  $X_i$  sample, once the autocorrelation function reaches zero at

spots the past point [32, 33]. This technique reproduces only linear features of the dynamics and usually requires supplementary data. At the same time, the other methodology involves a nonlinear autocorrelation function named mutual information (MI) to compute the delay from the time series [34]. Shannon's information theory inspires the main motivation of this technique to produce the information accomplished from examinations of one random event on another using the MI criterion. The MI is a nonlinear equivalent to the correlation function, and both linear and nonlinear reliance among two time series can be measured by it. Once the MI is adopted for time-delayed translations of the identical sequence, it is named auto MI (AMI).

Usually, the MI recognizes the quantity of information a signal gives regarding the other signal. Thus, the AMI calculates the approximate degree of forecasting  $X_{i+\tau}$  from  $X_i$ . From an information-theoretic point of view, the AMI discovers how the measurements  $X_i$  are joined  $X_{i+\tau}$ . The following formula gives the AMI [34]:

$$I(\tau) = \sum_{X_i, X_{i+\tau}} P(X_i, X_{i+\tau}) \log_2 \left( \frac{P(X_i, X_{i+\tau})}{P(X_i)P(X_{i+\tau})} \right), \quad (2)$$

where  $i$  denotes total sample number,  $P(X_i)$  and  $P(X_{i+\tau})$  show the marginal probabilities for measurements  $X_i$  and  $X_{i+\tau}$ , respectively, and  $P(X_i, X_{i+\tau})$  gives their connection probability density for measurements  $X_i$  and  $X_{i+\tau}$ .

*Remark 1.* A delay time  $\tau$  that minimizes  $I(\tau)$  for  $t = \tau$  and  $X_{i+\tau}$  appends the highest information on  $X_i$  has the optimal value.

**2.3. Embedding Dimension.** The minimum number of the state variables needed to display the system behavior is called the embedding dimension  $m$ . The Grassberger–Procaccia (GP) [35], the singular value decomposition (SVD) [36], the FNN [37], and Cao's scheme [38] are the main approaches for obtaining the minimum embedding dimension from a scalar time series. The delay coordinates at a specified time delay  $\tau$  are used in the GP method to rebuild the dynamics of a scalar time series in an embedding space of dimension  $m$ . Although this procedure is data demanding and subjective and consumes more time in the simulation, it can determine the time series's chaotic and/or random nature. With the help of singular values of embedding, the quantity of variance of the trajectory's projection on the orthogonal directions in the embedding space is used in the SVD method. The minimum dimension is computed by the number of directions the rebuilt trajectory sees and determined by the large singular values. The main drawback of the SVD scheme is its subjectivity to the number of singular values.

The FNN algorithm is inspired by the fact that the orbits of a chaotic attractor cannot cross or go beyond each other. In contrast, when a lower dimension than the adequate characterizes an irregular attractor, a junction and/or an overlap occurs. In other words, the FNN algorithm relies on that if a too low embedding dimension is chosen, the points distant from each other in the original phase space will be

closed in the rebuilt phase space. Subjectivity and the need for supplementary data are weaknesses of this approach.

The drawbacks of the algorithms above are tackled using Cao's strategy. This scheme adopts a scalar time series  $X_1, X_2, \dots, X_N$  and the delay time vector (1) with the following norm [38]:

$$a(i, m) = \frac{\|y_i(m+1) - y_{n(i,m)}(m+1)\|}{\|y_i(m) - y_{n(i,m)}(m)\|}, \quad i = 1, 2, \dots, N - m\tau, \quad (3)$$

where  $y_i(m+1)$  denotes the  $i$ th reconstructed vector with embedding dimension  $m+1$ , i.e.,  $y_i(m+1) = (X_i, X_{i+\tau}, \dots, X_{i+m\tau})$ ,  $1 \leq n(i, m) \leq n - m\tau$  stands for an integer such that  $y_{n(i,m)}(m)$  is the nearest neighbor of  $y_i(m)$  in the  $m$ -dimensional rebuilt phase space, and  $\|y_k(n) - y_l(n)\| = \max_{0 \leq j \leq n-1} |x_{k+j\tau} - x_{l+j\tau}|$ .

*Remark 2.* If  $y_{n(i,m)}(m)$  is equal to  $y_i(m)$ , the second nearest neighbor is adopted instead of it.

In Cao's and FNN approaches, any two points represent true neighbors that their nearness in the  $m$ -dimensional rebuilt phase space guarantees that they are still near in the  $(m+1)$ -dimensional rebuilt phase space. If this condition is not met, the two points will be interpreted as false neighbors. In a perfect embedding, no false neighbors should be supposed. In [32], a false neighbor has been recognized by checking whether  $a(i, m)$  is larger than some threshold rate. Nonetheless, it is clear from (3) that this requires the derivative of the original signal. Hence, different threshold values are essential for different phase points  $i$  implying that it is difficult to gain an appropriate threshold value independent of the values of  $m$ ,  $X_i$ , and the time series.

To circumvent the issue above, Cao has applied the average value of all  $a(i, m)$   $s$  as another quantity as follows:

$$E(m) = \frac{1}{N - m\tau} \sum_{i=1}^{N-m\tau} a(i, m). \quad (4)$$

The variation of  $E(m)$  from  $m$  to  $m+1$  can be evaluated using the following formula [35]:

$$E_1(m) = \frac{E(m+1)}{E(m)}. \quad (5)$$

Once the time series appears from a chaotic attractor,  $E_1(m)$  ends variation when  $m$  is greater than a fixed value  $m_0$ . In this situation,  $m = m_0 + 1$  is taken as the minimum embedding dimension.

**2.4. Largest Lyapunov Exponent Measure.** It is well known that once a time series is susceptible to the initial states, it will be unpredictable, at least for the long term. The highly sensitive time series have divergent exponential trajectories varying with small fluctuations of the initial states. A mean norm of this divergence and the unpredictability of a chaotic time series are given by the Lyapunov exponent measure, in which it expresses the rate of division of infinitesimally close states. The LLE calculates the deviation of close trajectories

in the phase space. Therefore, a positive Lyapunov exponent implies the occurrence of chaos.

Suppose that  $s_{t_1}$  and  $s_{t_2}$  are two points in two trajectories in the state space with  $\|s_{t_1} - s_{t_2}\| = \partial_0 \ll 1$ . If  $\Delta t$  time steps are continued, one has  $\partial_{\Delta t} \cong \|s_{t_1+\Delta t} - s_{t_2+\Delta t}\|$ ,  $\partial_{\Delta t} \ll 1$  and  $\Delta t \gg 1$ , which means that the initial division  $\partial_0$  will diverge the trajectories with an exponential rate,  $\partial_{\Delta t} \cong e^{\lambda \Delta t} \partial_0$ , where  $\lambda$  is the Lyapunov exponent [39]. As a result, a positive  $\lambda$  assigns an exponential divergence of the closed trajectories, implying a chaotic behavior. On the other side, dissipative and nonconservative time series show negative Lyapunov exponents, and their trajectories will attain a stable equilibrium point or a periodic orbit. Also, the Lyapunov exponent of the conservative time series is equal to zero.

In work [40], a numerical algorithm has been given for determining the LLE of the scalar time series. The algorithm looks for every neighbor within a neighborhood of command trajectory and gets the average distance of neighbors and the command trajectory as a function of time. This algorithm computes a stretching factor  $S(\tau)$  whose slope is equal to the LLE. The methodology in [39] takes a  $t$  from the set  $T_\tau = \{m, m+1, \dots, T-\tau\}$  to find  $U_t = \varepsilon$ -neighborhood of an arbitrary point  $X_t$  in the time series. Afterward, for all  $i \in U_t$ , the distance of  $X_t$  and a neighbor of it is computed as  $|X_{t+\tau} - X_{i+\tau}|$ , and the logarithm of the mean of these distances is found. This process is repeated for all  $t \in T_\tau$ , and the stretching factor is achieved.

$$S(\tau) = \text{mean}_{t \in T_\tau} \left( \ln \left( \text{mean}_{i \in U_t} (|X_{t+\tau} - X_{i+\tau}|) \right) \right). \quad (6)$$

The other algorithm for estimating the LLE of a scalar time series has been introduced by Rosenstein et al. [41]. This algorithm utilizes all the information of the entire dataset rather than relying on one trajectory. Once the phase space is built by supposed  $\tau$  and  $m$ , a point  $X_{n_0}$  is taken, all neighbor points  $X_n$  closer than a distance  $r$  are recognized, and the average distance from that point between them is calculated. This process is reiterated for  $N$  points along the trajectory to find an average value of  $S$  as follows:

$$S = \frac{1}{N} \sum_{n_0=1}^N \ln \left( \frac{1}{|U_{X_{n_0}}|} \sum |X_{n_0} - X_n| \right), \quad (7)$$

where  $|U_{X_{n_0}}|$  denotes the number of neighbors recognized around the point  $X_{n_0}$ . The diagram of the stretching factor against time  $t = N\Delta t$  appears an arc with a linear increase at the start, continued by an approximately smooth area.

**2.5. Surrogate Data.** One of powerful tools for the nonlinearity examination is surrogate data test [12, 42], in which the null hypothesis indicates that the observed spatial series is produced by a Gaussian (linear) process with a possibly nonlinear static transform. This method considers the mean, the standard deviation, the cumulative distribution function, and the power spectrum of the original data. The surrogate approach produces substitute data with the identical probabilistic organization as the original data [12].

The iterative amplitude-adjusted Fourier transform (IAAFT) has been introduced in [42] to produce the surrogate datasets. In this approach, the created surrogates rely on the idea of generating constrained realizations, where the measurable features of the spatial series are taken into account rather than the basic model equations [12]. The null hypothesis of underlying Gaussian linear stochastic processes can also be formulated, denoting that randomized samples can be created by generating orders with the same linear attributes as the observed data but which are otherwise random [12]. In the surrogate data test, if the squared amplitudes of the discrete Fourier transform are adopted to denote the linear characteristics of the data, the surrogate spatial series will be produced by multiplying the Fourier transform of the data by random phases and subsequently transforming back to the time domain [12]. An appropriate subsequence of the data is chosen before creating the surrogates to avoid the periodicity artifact from leading to false nonlinearity test outcomes [12]. Moreover, windowing and zero-padding techniques can be applied to suppress the edge effects' problems. However, these methods degrade the invertibility of transform and, therefore, cannot be effectively applied for the phase randomization of the surrogates [12].

In the surrogate data test, the IAAFT is applied to preserve the probability density function and the correlation structure (and therefore the power spectrum) of the original data by iteratively minimizing the deviation. The main procedure of the adopted algorithm is given below [12, 43].

- (a) A sorted record of the original spatial series  $\{s_n\}$  and the squared amplitudes of its Fourier transform,  $S_k^2 = |\sum_{n=0}^{N-1} e^{j2\pi kn}|^2$ , are saved.
- (b) The data are randomly shuffled (without substitution)  $\{S_n^{(0)}\}$  to corrupt any nonlinear associations and correlations.
- (c) The Fourier transform of  $\{S_n^{(i)}\}$  is calculated, and its squared amplitude is substituted by  $S_k^2$ . The result is transformed back to the time domain.
- (d) The resulting series is ranked in order, and each value is replaced with the original series value with an equal rank. This work corrects the probability density function of the data and modifies the power spectrum once more.
- (e) Steps (c) and (d) are repeated to achieve a given accuracy.

### 3. Nonlinear Predictor

In nonlinear dynamics theory, although chaos is deterministic, it cannot be cast further than short intervals. The average forecast horizon of a chaotic time series can be reached by the LLE norm as follows [44]:

$$\Delta t_{\max} = \frac{1}{\lambda_{\max}}. \quad (8)$$

In recent years, some approximate linear and/or nonlinear algorithms have been presented in the literature to forecast the future of the chaotic time series. In [45], a simple

nonlinear technique has been provided, restated as follows. First, the delay time of the time series  $X_i$ ,  $i = 1, 2, \dots, N$ , is calculated to rebuild the phase space. Then, the irregular attractor is established with an embedding dimension  $m$ , and the following  $m$ -dimensional map  $f_T$  is applied to model the dynamics of the time series.

$$Y_{j+T} = f_T(Y_j), \quad (9)$$

where  $Y_j$  and  $Y_{j+T}$  are vectors of dimension  $m$  in which they denote the state at the current time  $j$  and the state at the future time  $j + T$ , respectively.

The next step is to get the help of the observed time series to discover a proper estimation of  $f_T$ . In the local approximation procedure [44], a locally piecewise structure is adopted to build the dynamics in the embedding space. Moreover, the domain is divided into some local neighborhoods, and the model is accomplished for each neighborhood separately, resulting in a different  $f_T$  for each subset. Hence, the dynamics of the system are governed part by part, and the complexity of  $f_T$  is considerably reduced without affecting the accuracy of the forecast.

The deviation of the trajectory concerning the time should be estimated to forecast in the  $m$ -dimensional space. Regarding the relation of the  $X_t$  and  $X_{t+p}$  points, the future of the system at the time  $p$  on the irregular attractor can be estimated via a nonlinear function  $F$  as follows:

$$X_{t+p} \cong F(X_t). \quad (10)$$

The primary assumption of this forecast algorithm is that the variation of  $X_t$  with time on the irregular attractor is identical to those of close points  $(X_{T_h}, h = 1, 2, \dots, n)$ . Afterward,  $X_{t+p}$  is built by the order  $d$  polynomial  $F(X_t)$  as follows.

$$\begin{aligned} X_{t+p} \cong & f_0 + \sum_{k_1=0}^{m-1} f_{1k_1} X_{t-k_1\tau} + \sum_{\substack{k_2=k_1 \\ k_1=0}}^{m-1} f_{2k_1k_2} X_{t-k_1\tau} X_{t-k_2\tau} \\ & + \dots + \sum_{\substack{k_d=k_{d-1} \\ k_1=0}}^{m-1} f_{dk_1k_2\dots k_d} X_{t-k_1\tau} X_{t-k_2\tau} \dots X_{t-k_d\tau}. \end{aligned} \quad (11)$$

In [44], it has been suggested that using  $n$  of  $X_{T_h}$  and  $X_{T_h+p}$  with known values, the coefficients  $f$  can be reached by the following equation:

$$X \cong Af, \quad (12)$$

where  $X = (X_{T_1+p}, X_{T_2+p}, \dots, X_{T_n+p})$ ,  $f = (f_0, f_{10}, f_{11}, \dots, f_{1(m-1)}, \dots, f_{d(m-1)(m-1)\dots(m-1)})$  and  $A$  is a  $n(m+d)!/m!d!$  Jacobian matrix described below.

$$A = \begin{bmatrix} X_{T_1} & X_{T_1-\tau} & \dots & X_{T_1-(m-1)\tau} & X_{T_1}^2 & \dots & X_{T_1-(m-1)\tau}^2 \\ X_{T_2} & X_{T_2-\tau} & \dots & X_{T_2-(m-1)\tau} & X_{T_2}^2 & \dots & X_{T_2-(m-1)\tau}^2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ X_{T_n} & X_{T_n-\tau} & \dots & X_{T_n-(m-1)\tau} & X_{T_n}^2 & \dots & X_{T_n-(m-1)\tau}^2 \end{bmatrix}. \quad (13)$$

However, in this paper, the idea of memory usage is proposed. In our approach, the Jacobian matrix  $A$  not only is constructed by  $m$  points of the current attractor, but also benefits the other  $m$  points of the past attractors to increase the accuracy of the approximate model. Thus, the size of the Jacobian matrix  $A$  is modified as  $n(K(m+d))!/(Km)!(Kd)!$ , where  $1 \leq K \leq m$  is selected as an integer. The rows matrix  $A$  must approve the following inequality to attain a sure solution:

$$n \geq \frac{(K(m+d))!}{(Km)!(Kd)!} \quad (14)$$

## 4. Data and Numerical Results

In this section, a brief description of the adopted data is given. Then, comprehensive numerical simulation and analysis are provided to verify the data's complex behavior and forecast the short-term future of the chaotic financial time series.

**4.1. Financial Data.** Recently, some microfinance institutes have the mission of providing limited loans with no (or at least a minimum) interest. The target population of such interest-free organizations is low-income people who lack access to the financial services of other banks or traditional financial institutions. The essential condition for requesting an interest-free loan is that the recipient must prove setting up a small-scale enterprise. The essential goal of such microfinance institutions is to help low-income people get better access to financial services and finance small or medium projects. Omid Entrepreneurship Fund (OEF) is the biggest and the most important interest-free institution in Iran. OEF has at least one branch in each state of Iran. We have taken the five-year loan payment percent of two branches of OEF that have the highest activity in the considered time horizon and call them  $B_1$  and  $B_2$  branches. For the considered loans, the due of the first installment is in the range of years 2011–2017. The considered period does not contain any crisis years and other shocks. Since an NPL is defined as a loan with no payment for at least 18 months for the OEF policies, the two last years' data (which do not contain an NPL) are removed. The time series of the two time series are depicted in Figure 1. Furthermore, the significant statistical attributes of these time series are given in Table 1.

**4.2. Phase Space.** We choose  $m = 2$  and  $\tau = 1$  to rebuild the phase space and make an illustrative attractor. In this case, the irregular attractor will be projected to the plane  $\{X_i, X_{i+1}\}$ . Figure 2 depicts the phase space reconstruction of the  $B_1$  and  $B_2$  time series. One sees that an irregular attractor occurs in both the phase space plots. Moreover, these plots deny the requirement for a stochastic modeling approach for the time series because the attractors are in well-defined regions, implying that the deterministic chaos can effectively elucidate the system dynamics.

**4.3. Time Delay Assignment.** The time delay is estimated using the AMI algorithm with a delay time belonging to  $[1, 30]$ . The results for the two datasets are illustrated in Figure 3. Finding the first local minima on the diagrams is needed to estimate an appropriate time delay. This happens in the delay time of 4 and 3 for  $B_1$  and  $B_2$ , respectively.

**4.4. Embedding Dimension Estimation.** Cao's algorithm is implemented to find the minimum value of the embedding dimension to determine the sufficient embedding dimension for the phase space reconstruction. The maximum value of the embedding dimension is assigned equal to 50 to achieve a good result. Then, the criterion  $E_1(m)$  is depicted versus the variation of  $m$ . The simulation results are plotted in Figure 4. Figure 4(a), which stands for  $B_1$  time series, shows that  $E_1(m)$  end varies  $m_0 = 16$ . Thus, based on Cao's technique, the minimum embedding dimension for  $B_1$  is  $m = m_0 + 1 = 17$ .

On the other hand, as Figure 4(b) shows, the curve  $E_1(m)$  ends vary with  $m_0 = 15$  in the case of  $B_2$ . So, the minimum embedding dimension of these data equals to  $m = m_0 + 1 = 16$ . The most important issue is that since  $E_1(m)$  does not vary after some finite value of  $m_0$ , both the time series  $B_1$  and  $B_2$  are originated from an irregular attractor, implying the existence of chaos.

**4.5. Largest Lyapunov Exponent.** The computation of the LLE is carried out using the algorithm proposed in [41]. The time delay and minimum embedding dimension parameters computed in the previous subsections are used in the numerical simulation. The curves for the stretching factor  $S$  versus the number of points  $N$  are displayed in Figure 5. One can see that linear increased regions with some fluctuations placed over the linear parts occur. It is noted that owing to the stretching factor as an average value of the local stretching or shrinking rates in the irregular attractor, some fluctuations typically emerge, and the different rates cannot be continually flattened by the averaging process of the algorithm [46].

Accordingly, the slopes of the curves about the linear regions are determined using a least-squares line fitting approach aiming to achieve the LLEs of the time series. The LLEs of the  $B_1$  and  $B_2$  time series are achieved as 0.0205 and 0.0178, respectively. These positive values of the LLEs confirm the exponential divergence rate of the trajectories and, then, the chaotic behavior of the financial time series. Besides, the inverse of the LLEs is computed to get the forecast horizon. In this situation, it is revealed that the forecast horizons are 48 and 56 samples for  $B_1$  and  $B_2$ , respectively. The predictions over these values are subject to extreme uncertainties.

**4.6. Surrogate Data.** The potential nonlinearity of the time series can be tested for the surrogate data procedure. At  $\alpha = 1\%$  the significance level, to gather  $1/\alpha - 1 = 99$  surrogate time series, the iterative algorithm given in the previous subsections is adopted to verify that if the original data come from a linear

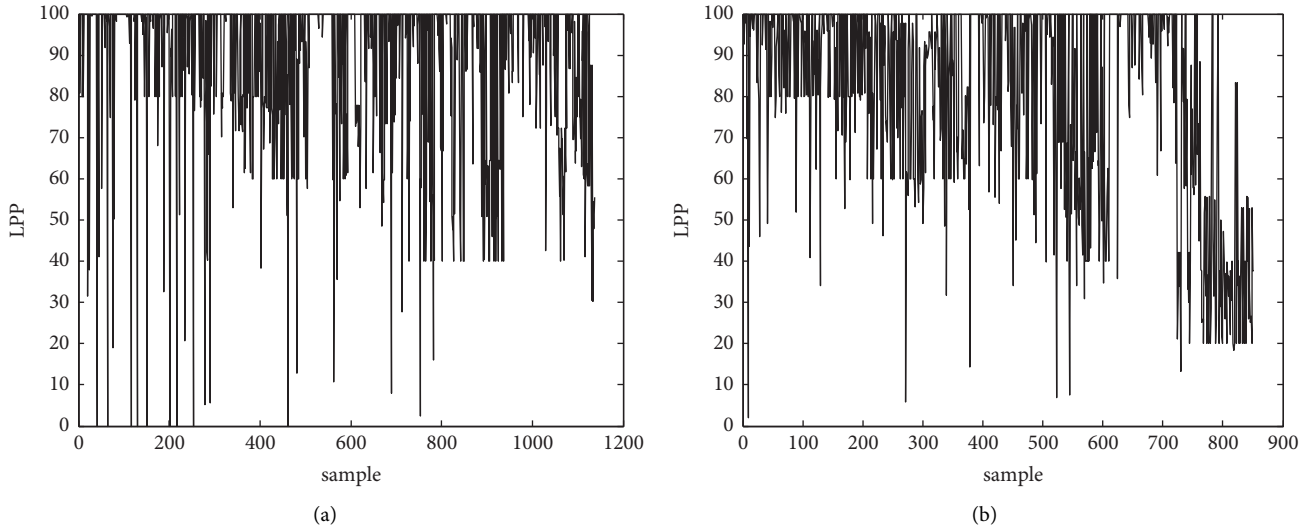


FIGURE 1: Time series of (a)  $B_1$  and (b)  $B_2$ .

TABLE 1: Statistical attributes of the time series.

Statistical attribute	$B_1$	$B_2$
Number of data	1138	851
Mean	87.9281	80.0138
Median	100	92.3100
Min	0	1.9300
Max	100	100
Standard deviation	19.8204	24.4088
Variance	392.8466	595.7916
Skewness	-1.9952	-1.0641
Kurtosis	7.2104	3.0071

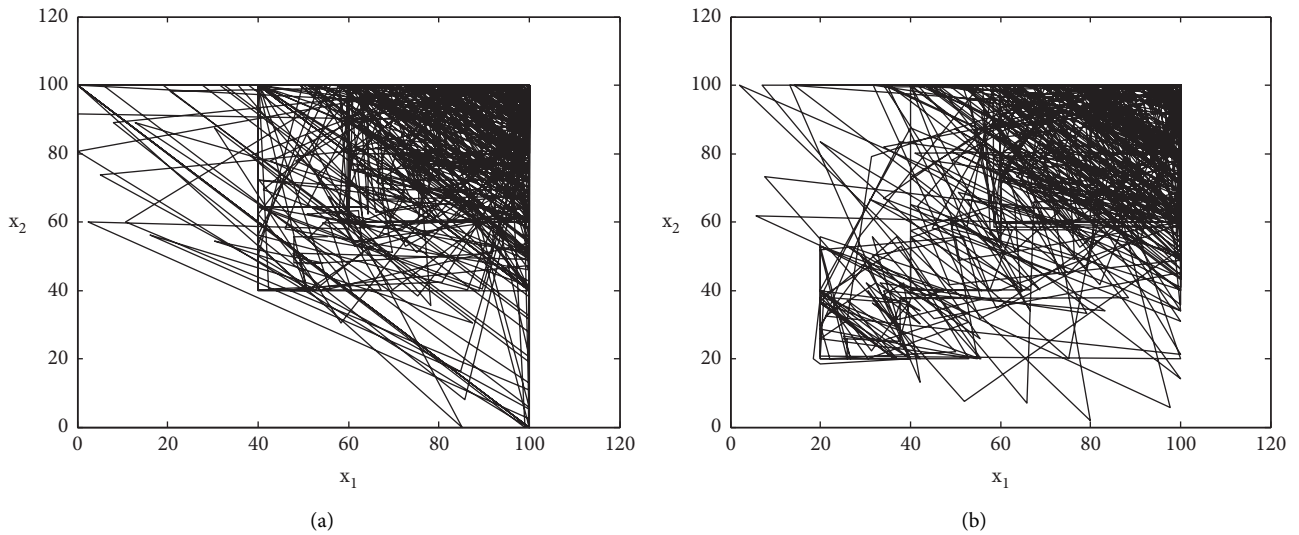


FIGURE 2: Phase spaces of the time series (a)  $B_1$  and (b)  $B_2$ .

stochastic process or not. In this regard, the nonlinear prediction error examination is applied for the test statistic. The diagram of the nonlinear prediction errors of 99 surrogates,

and the original data are plotted in Figure 6. The lower thick long line stands for the prediction error, and the other thin lines show the other 99 surrogates, indicating that the null

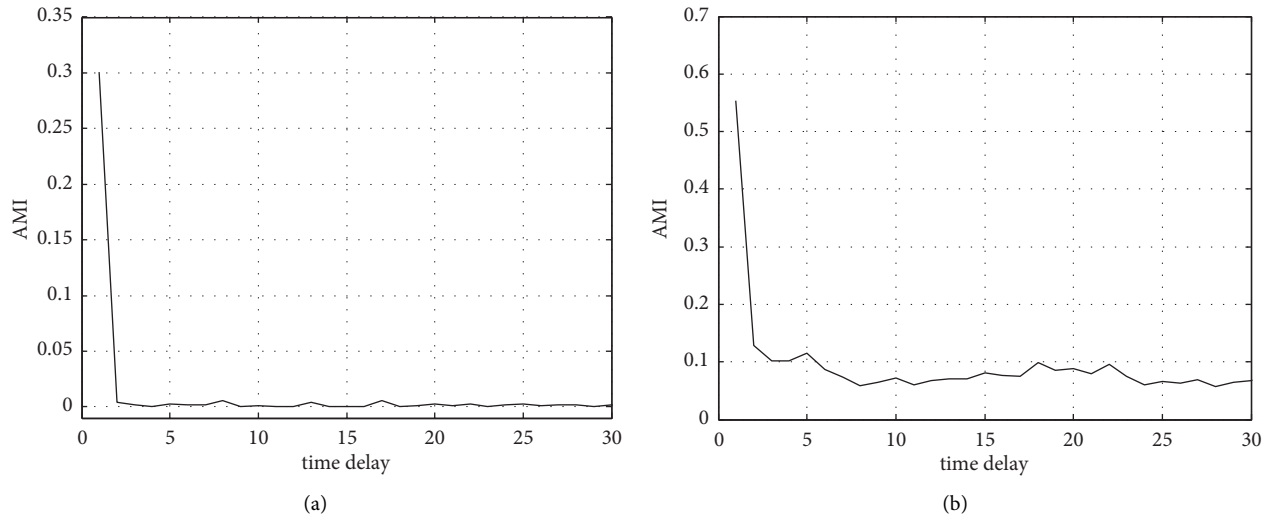


FIGURE 3: AMI for the time series (a) B<sub>1</sub> and (b) B<sub>2</sub>.

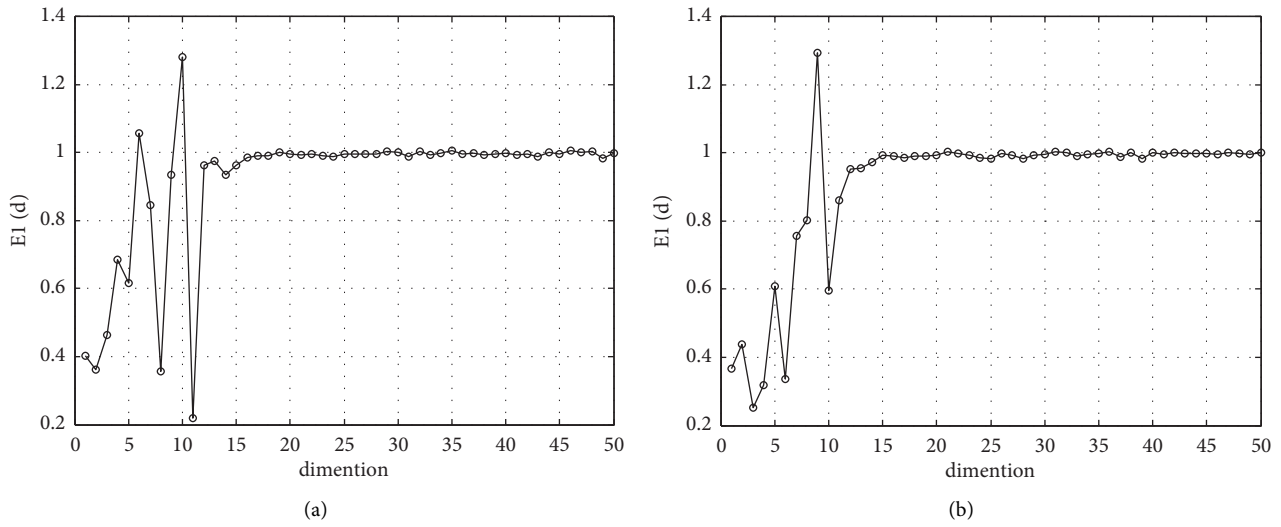


FIGURE 4: Cao's results for the time series (a) B<sub>1</sub> and (b) B<sub>2</sub>.

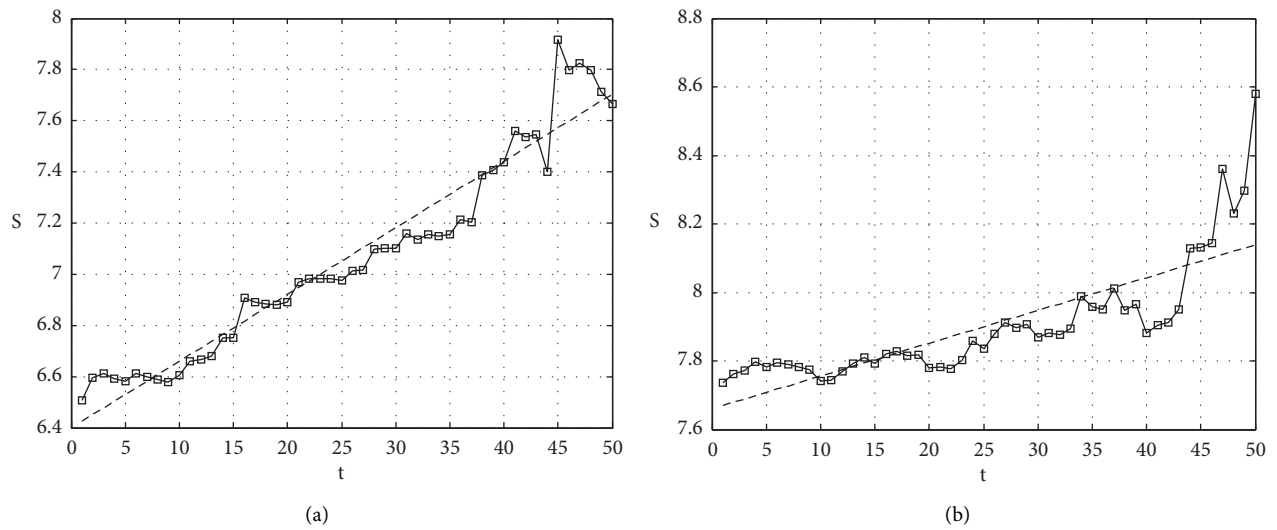


FIGURE 5: LLEs for the time series (a) B<sub>1</sub> and (b) B<sub>2</sub>.

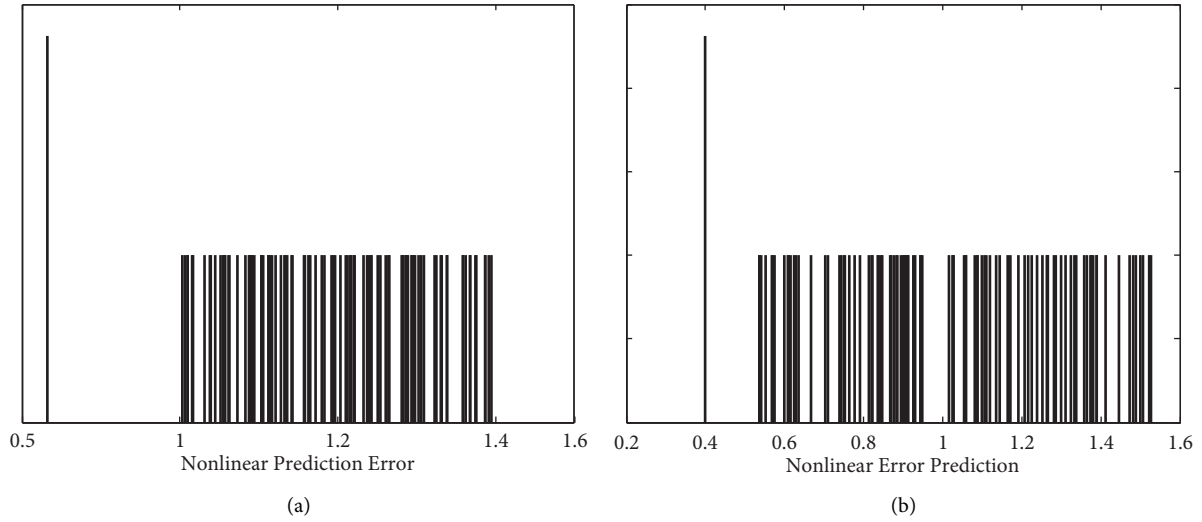


FIGURE 6: Surrogate data test for the time series (a)  $B_1$  and (b)  $B_2$ .

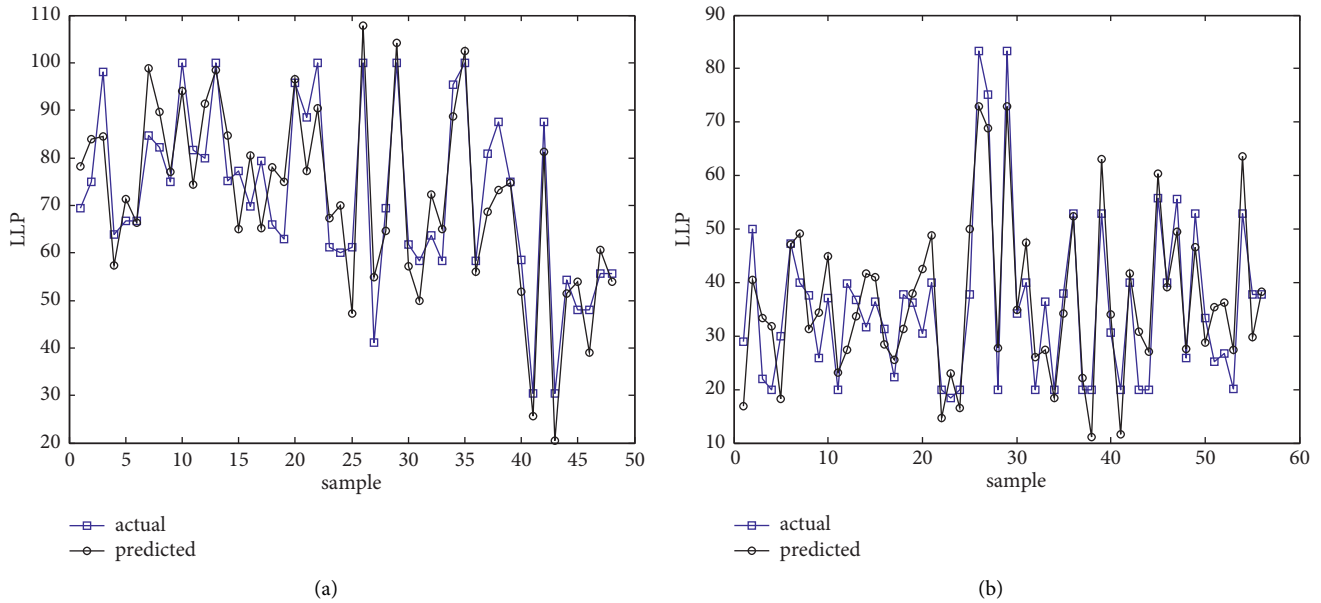


FIGURE 7: Forecast results for the time series (a)  $B_1$  and (b)  $B_2$ .

hypothesis is declined at the 99% significance level. This means that the hypothesis that the data originality is discarded from a linear stochastic process. Therefore, the surrogate data test further validates that the correlation dimension's convergence occurs due to the low-dimensional dynamics that are dominant in the chaotic systems; i.e., it is not due to the linear stochastic nature of the data.

**4.7. Prediction.** We attempted to forecast the future of the LPP with the modified chaos-based nonlinear prediction method. The performed analysis in the prior sections has proved that the financial time series exhibit a chaotic behavior with a long-term unpredictability feature. On the other hand, the modified local forecasting algorithm can

model their future values quickly. The inverse of LLEs suggests that the maximum window for accurate forecasting is 48 and 56 samples for  $B_1$  and  $B_2$ , respectively. The motivation for such prediction is clear: the developments of many financial processes have become ever more demanding for the exact short-term prediction. It is noticed that the partial predictability of a chaotic time series owes to its high sensitivity to initial states. This fact indicates that the information is built in the whole time series and that short-time predictions are reasonable. The forecasting results of the time series (with  $K = 2$ ) and the corresponding actual values are depicted in Figure 7. It is viewed that the forecasting results are satisfactory, and they follow the actual data reasonably accurately, except for extreme value. As a result, the predicted values can be utilized to identify and



predict NPLs (an NPL is an LPP whose value is less than a particular fixed constant in a specified time) before they become uncontrollable. Therefore, these predictions can help the managers of economic institutions to develop the necessary policies regarding the budget and the remainder of the loans.

## 5. Concluding Remarks

This paper has introduced the concept of chaotic predictions over short periods for the loan time series. The main idea is that if one can forecast the future behavior of the loan payment percent, then the nonperforming loans will be identified. So, the policies can be modified according to the current state of the budget. Various nonlinear dynamic methods have been realized to identify the existence of low-dimensional chaos in the data to implement this idea. First, the well-known AMI algorithm has discovered the delay time to rebuild the possible irregular attractors. Afterward, the dimensionality of the trajectories was detected using Cao's technique. A low embedding dimension has validated the low-dimensional chaos in the financial data. Accordingly, based on the largest Lyapunov exponent norm, it has been revealed that the time series is susceptible to tiny fluctuations of the initial states. This proves the exponential divergence of the trajectories and the unpredictability of the time series. Subsequently, the surrogate data examination has been adopted to verify that the financial time series do not originate from a stochastic process. Lastly, a modified local nonlinear approximator has been presented to forecast the short-term behavior of the time series. The numerical simulations on the data collected from an interest-free economic institution in Iran have confirmed the complex nonlinear structure of the LPPs. The findings of this article may help the managers of the banks and economic organizations forecast the short-term horizon of the NPLs and, therefore, balance the budget and loans.

## Data Availability

Access to data is restricted due to Omid Entrepreneurship Fund (OEF) policy. To access data, a request should be sent to OEF.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The Omid Entrepreneurship Fund of Iran financially supported this research.

## References

- [1] Y. Zhang, G. Yu, and D. Yang, "Predicting non-performing loan of business bank by multiple classifier fusion algorithms," *Journal of Interdisciplinary Mathematics*, vol. 19, no. 4, pp. 657–667, 2016.
- [2] A. Belgrave, G. Kester, and M. Jackman, "Industry specific shocks and non-performing loans in Barbados," *the review of finance and banking*, vol. 4, no. 2, 2012.
- [3] Y. Tang, H. Chen, B. Wang, M. Chen, M. Chen, and X. Yang, "Discriminant analysis of zero recovery for China's NPL," *Advances in Decision Sciences*, vol. 2009, Article ID 594793, 16 pages, 2009.
- [4] S. L. Lin, "A new two-stage hybrid approach of credit risk in banking industry," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8333–8341, 2009.
- [5] A. O. Adewusi, T. B. Oyedokun, and M. O. Bello, "Application of artificial neural network to loan recovery prediction," *International Journal of Housing Markets and Analysis*, vol. 9, no. 2, pp. 222–238, 2016.
- [6] G. Kou, Y. Peng, and C. Lu, "MCDM approach to evaluating bank loan default models," *Technological and Economic Development of Economy*, vol. 20, no. 2, pp. 292–311, 2014.
- [7] P. Saha, I. Bose, and A. Mahanti, "A knowledge based scheme for risk assessment in loan processing by banks," *Decision Support Systems*, vol. 84, pp. 78–88, 2016.
- [8] Z. Bitvai and T. Cohn, "Predicting peer-to-peer loan rates using bayesian non-linear regression," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2203–2209, Austin, TX, USA, January 2015.
- [9] R. Calabrese, "Predicting bank loan recovery rates with a mixed continuous-discrete model," *Applied Stochastic Models in Business and Industry*, vol. 30, no. 2, pp. 99–114, 2014.
- [10] P. Tahmasebi and A. Hezarkhani, "A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation," *Computers & Geosciences*, vol. 42, pp. 18–27, 2012.
- [11] B. LeBaron, "Chaos and nonlinear forecastability in economics and finance," *Philosophical Transactions of the Royal Society of London, Series A: Physical and Engineering Sciences*, vol. 348, no. 1688, pp. 397–404, 1994.
- [12] M. Pourmahmood Aghababa and J. Abdollahi Sharif, "Chaos and complexity in mine grade distribution series detected by nonlinear approaches," *Complexity*, vol. 21, no. S2, pp. 355–369, 2016.
- [13] S. Ramdani, F. Bouchara, and O. Caron, "Detecting high-dimensional determinism in time series with application to human movement data," *Nonlinear Analysis: Real World Applications*, vol. 13, no. 4, pp. 1891–1903, 2012.
- [14] H.-L. Yang and H.-C. Lin, "Applying the hybrid model of EMD, PSR, and ELM to exchange rates forecasting," *Computational Economics*, vol. 49, no. 1, pp. 99–116, 2017.
- [15] A. Papan, C. Kyrtsov, D. Kugiumtzis, and C. Diks, "Detecting causality in non-stationary time series using partial symbolic transfer entropy: evidence in financial data," *Computational Economics*, vol. 47, no. 3, pp. 341–365, 2016.
- [16] V. K. Dabhi and S. Chaudhary, "Financial time series modeling and prediction using postfix-GP," *Computational Economics*, vol. 47, no. 2, pp. 219–253, 2016.
- [17] P. Flaschel, F. Hartmann, C. Malikane, and C. R. Proaño, "A behavioral macroeconomic model of exchange rate fluctuations with complex market expectations formation," *Computational Economics*, vol. 45, no. 4, pp. 669–691, 2015.
- [18] T. Chen, X. Li, and J. He, "Complex dynamics of credit risk contagion with time-delay and correlated noises," *Abstract and Applied Analysis*, vol. 2014, Article ID 456764, 10 pages, 2014.
- [19] T. Chen, J. He, and J. Wang, "Bifurcation and chaotic behavior of credit risk contagion based on fitzhugh-nagumo system," *International Journal of Bifurcation and Chaos*, vol. 23, no. 07, Article ID 1350117, 2013.

- [20] S. Lahmiri, "On fractality and chaos in Moroccan family business stock returns and volatility," *Physica A: Statistical Mechanics and Its Applications*, vol. 473, pp. 29–39, 2017.
- [21] S. Radhakrishnan, A. Duvvuru, S. Sultornsane, and S. Kamarthi, "Phase synchronization based minimum spanning trees for analysis of financial time series with nonlinear correlations," *Physica A: Statistical Mechanics and Its Applications*, vol. 444, pp. 259–270, 2016.
- [22] Q. Tian, P. Shang, and G. Feng, "The similarity analysis of financial stocks based on information clustering," *Nonlinear Dynamics*, vol. 85, no. 4, pp. 2635–2652, 2016.
- [23] B. B. Nair, P. K. S. Kumar, N. R. Sakthivel, and U. Vipin, "Clustering stock price time series data to generate stock trading recommendations: an empirical study," *Expert Systems with Applications*, vol. 70, pp. 20–36, 2017.
- [24] A. Parida, R. Bisoi, P. Dash, and S. Mishra, "Financial time series prediction using a hybrid functional link fuzzy neural network trained by adaptive unscented kalman filter," in *Proceedings of the 2015 IEEE Power, Communication and Information Technology Conference (PCITC)*, pp. 568–575, Bhubaneswar, India, 15 October 2015.
- [25] R. Singh and S. Srivastava, "Stock prediction using deep learning," *Multimedia Tools and Applications*, vol. 76, no. 18, pp. 18569–18584, 2017.
- [26] T. Zhou, S. Gao, J. Wang, C. Chu, Y. Todo, and Z. Tang, "Financial time series prediction using a dendritic neuron model," *Knowledge-Based Systems*, vol. 105, pp. 214–224, 2016.
- [27] R. Chandra and S. Chand, "Evaluation of co-evolutionary neural network architectures for time series prediction with mobile application in finance," *Applied Soft Computing*, vol. 49, pp. 462–473, 2016.
- [28] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence*, pp. 366–381, Springer, Warwick, England, 1981.
- [29] T. A. Alexeeva, N. V. Kuznetsov, and T. N. Mokaev, "Study of irregular dynamics in an economic model: attractor localization and Lyapunov exponents," *Chaos, Solitons & Fractals*, vol. 152, Article ID 111365, 2021.
- [30] N. V. Kuznetsov, T. N. Mokaev, O. A. Kuznetsova, and E. V. Kudryashova, "The Lorenz system: hidden boundary of practical stability and the Lyapunov dimension," *Nonlinear Dynamics*, vol. 102, no. 2, pp. 713–732, 2020.
- [31] N. V. Kuznetsov, G. A. Leonov, T. N. Mokaev, A. Prasad, and M. D. Shrimali, "Finite-time Lyapunov dimension and hidden attractor of the Rabinovich system," *Nonlinear Dynamics*, vol. 92, no. 2, pp. 267–285, 2018.
- [32] J. Holzfuss and G. Mayer-Kress, "An approach to error-estimation in the application of dimension algorithms," in *Dimensions and Entropies in Chaotic Systems*, pp. 114–122, Springer, New York, NY, USA, 1986.
- [33] W. Liebert and H. Schuster, "Proper choice of the time delay for the analysis of chaotic time series," *Physics Letters A*, vol. 142, no. 2-3, pp. 107–111, 1989.
- [34] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical Review A*, vol. 33, no. 2, pp. 1134–1140, 1986.
- [35] K. P. Harikrishnan, R. Misra, G. Ambika, and A. K. Kembhavi, "A non-subjective approach to the GP algorithm for analysing noisy time series," *Physica D: Nonlinear Phenomena*, vol. 215, no. 2, pp. 137–145, 2006.
- [36] D. S. Broomhead and G. P. King, "Extracting qualitative dynamics from experimental data," *Physica D: Nonlinear Phenomena*, vol. 20, no. 2-3, pp. 217–236, 1986.
- [37] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical Review A*, vol. 45, no. 6, pp. 3403–3411, 1992.
- [38] L. Cao, "Practical method for determining the minimum embedding dimension of a scalar time series," *Physica D: Nonlinear Phenomena*, vol. 110, no. 1-2, pp. 43–50, 1997.
- [39] S. Mehdizadeh, "A robust method to estimate the largest Lyapunov exponent of noisy signals: a revision to the Rosenstein's algorithm," *Journal of Biomechanics*, vol. 85, pp. 84–91, 2019.
- [40] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," *Physica D: Nonlinear Phenomena*, vol. 16, no. 3, pp. 285–317, 1985.
- [41] M. T. Rosenstein, J. J. Collins, and C. J. De Luca, "A practical method for calculating largest Lyapunov exponents from small data sets," *Physica D: Nonlinear Phenomena*, vol. 65, no. 1-2, pp. 117–134, 1993.
- [42] T. Schreiber and A. Schmitz, "Improved surrogate data for nonlinearity tests," *Physical Review Letters*, vol. 77, no. 4, pp. 635–638, 1996.
- [43] C. T. Dhanya and D. Nagesh Kumar, "Nonlinear ensemble prediction of chaotic daily rainfall," *Advances in Water Resources*, vol. 33, no. 3, pp. 327–347, 2010.
- [44] H. Abarbanel, *Analysis of Observed Chaotic Data*, Springer Science & Business Media, Berlin, Germany, 2012.
- [45] K.-I. Itoh, "A method for predicting chaotic time-series with outliers," *Electronics and Communications in Japan*, vol. 78, no. 5, pp. 44–53, 1995.
- [46] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, Cambridge University Press, Cambridge, England, 2004.