International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 3, June 2023, pp. 3010~3018
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i3.pp3010-3018

An optimized deep learning model for optical character recognition applications

Sinan Q. Salih[1], Ahmed L. Khalaf[1], Nuha Sami Mohsin[2], Saadya Fahad Jabbar[2]
[1]Department of Communication Engineering Technology Department, College of Information Technology, Imam Ja'afar Al-Sadiq University, Baghdad, Iraq
[2]College of Education, Ibn Rushed for Human Science, University of Baghdad, Baghdad, Iraq

Article Info

Article history:

Received Jul 14, 2022
Revised Oct 5, 2022
Accepted Dec 2, 2022

Keywords:

Chaotic maps
Convolutional neural network
Training algorithm
Computational intelligence
Black hole algorithm

ABSTRACT

The convolutional neural networks (CNN) are among the most utilized neural networks in various applications, including deep learning. In recent years, the continuing extension of CNN into increasingly complicated domains has made its training process more difficult. Thus, researchers adopted optimized hybrid algorithms to address this problem. In this work, a novel chaotic black hole algorithm-based approach was created for the training of CNN to optimize its performance via avoidance of entrapment in the local minima. The logistic chaotic map was used to initialize the population instead of using the uniform distribution. The proposed training algorithm was developed based on a specific benchmark problem for optical character recognition applications; the proposed method was evaluated for performance in terms of computational accuracy, convergence analysis, and cost.

Corresponding Author:

Sinan Q. Salih
Department of Communication Engineering Technology, College of Information Technology, Imam Ja'afar Al-Sadiq University
Baghdad, Iraq
Email: sinan.salih@sadiq.edu.iq

1. INTRODUCTION

Every simple artificial neural network (ANN) is basically made up of the input and output layers of neurons, but the necessity for an intermediate hidden layer in ANN gave rise to the concept of deep learning (DL) [1], [2]. So, DL can be considered a more complicated version of ANN that relies on the use of numerous layers with nonlinear processing units; most DL frameworks rely on the supervised or unsupervised data representations learning concept [2]. The year 1965 witnessed the introduction of the first deep feedforward multilayer perceptrons-based working algorithm [3] and since then, DL has improved and been adopted in many applications. DL is appliable in several areas, such as pattern recognition, neural networks, optimization, graphical modeling, artificial intelligence, and signal processing [4]–[6].

Recently, convolutional neural networks (CNN) was developed as a way of achieving a better accuracy during recognition tasks. However, one of CNN main problems is the difficult design of its architecture for specific task. Thus, several architectures of CNN have been developed, such as the LeNet architecture which was originally developed by LeCun [7], [8]. This architecture was implemented for optical character recognition (OCR) and character recognition in several documents. ConvNet is another CNN architecture that relies on the use of 7 layers where each layer has a specific role. Using the same architecture for several tasks has a poor chance of reaching optimal performance; consequently, distinct CNN architectures are built for specific tasks, which requires a lot of research work since there are many types of machine learning (ML) activities in the sectors [8]. CNN is a robust network nevertheless; it still has some parameters that need to be optimized; these include the learning parameters and the network configuration settings. The performance of CNN has been reported to be reliant on the proper tuning of its network configuration parameters [9], [10]. Theoretical basis has been thoroughly investigated to develop strategies for enhancing CNN parameters, thereby improving its overall performance. This demands a conceptual shift from visual features extraction to the optimization of network structure configuration [10]. The last three decades have seen wide utilization of metaheuristics and nature-inspired algorithms for solving various kinds of NP optimization problems; such metaheuristics include firefly algorithm (FA), grey wolf optimizer (GWO), particle swarm optimization (PSO), nomadic people optimizer (NPO), and teaching-learning based optimization (TLBO) [11]–[14]. These algorithms were developed to handle problems in different engineering fields, information security, and machine learning [15]–[22].

A novel nature inspired algorithm or the black hole algorithm (BHA) was recently developed based on inspiration from the behavior of the black hole (BH) as it pulls in its surrounding stars [23]. The concept of the BHA stemmed from the nature and interaction of the BH with its surrounding solar bodies; it considers a set of stars as the total number of potential solutions in a given iteration and each star can be pulled by the BH at a time to represents the best solution. The next iteration generates a new set of solutions by moving the surrounding stars toward the BH. Stars that are close to the BH at a pre-determined distance are engulfed by the BH and the random generation of other set of stars is immediately implemented. With this concept, the BHA can launch an exploration task in the unexplored areas of the solution space rather than searching the already explored areas. The ability of BHA to solve data clustering problems has been demonstrated; its optimization performance was reportedly better than those of other meta-heuristics [24]. Recently, BH algorithm was adopted for CNN training [25]. The authors demonstrated that the exploration of BH algorithm slow down the searching process. Herein, novel Chaotic BHA-based training algorithm (CBH-CNN) for the training of the CNNs was developed and evaluated. The role of BHA in this approach is to establish the optimal CNN parameters. The exploration of BH was improved by initializing the population using logistic chaotic map, rather than the uniform distribution method. The performance of the proposed approach was evaluated based on specific criteria, such as accuracy and calculated errors. The modified national institute of standards and technology (MNIST) dataset, an approved dataset of handwritten digital images, was used as a reference. This dataset has digital images of 28×28 pixels size and contains 70,000 images. where 60,000 images were employed as the train set while the other 10,000 images were served as the testing data set. The rest of this article is arranged as follows: section 2 covers the basics of the CNN and BH models while section 3 covers the explanation of the proposed CBH-CNN algorithm. In section 4, the analysis results of the proposed CBH-CNN algorithm was presented while section 5 is the conclusion aspect of the work.

2. THEORETICAL BACKGROUND

2.1. Black hole algorithm

As aforementioned, Black hole algorithm (BHA) is based mostly on the idea of a region of space with much mass focus on it in a manner that nothing that comes close to it can escape its gravitational pull. Whatever is pulled into the BH is eternally lost. The BHA has two major aspects which are the migration and the re-initialization of stars that had crossed the event horizon around the BH. The BHA relies on the following working principle: first, the $N + 1$ stars, $x_i \in R^D, i = 1, \ldots, N + 1$ (where N=population size) are randomly initialized in the search space prior to the evaluation of their fitness. The candidate star with the best evaluation function is considered the black hole x_{BH} and being that the BH is static, it maintains its position until another star with a better solution is found. N represents the number of candidate stars searching for the optimum. The movement of each star towards BH in each generation can be calculated using this relation:

$$x_i(t+1) = x_i(t) + rand \times \left(x_{BH} - x_i(t)\right) \; i = 1.2. \cdots \; N, \tag{1}$$

where rand is a randomly generated number in the range between 0 and 1. Any star in the BHA that its distance to the BH is less than the event horizon will disappear. The event horizon has a radius (R) given as:

$$R = \frac{f_{BH}}{\sum_{i=1}^{N} f_i}, \tag{2}$$

where f_i and f_{BH} represent the fitness values of the BH and the i^{th} star, respectively. While N represents the number of stars (individual solutions). If the distance between a BH and an individual solution is less than R, then the individual solution will collapse for another individual will be created and randomly distributed in the solution space. The BHA is easy and simple to implement because it is a parameter-less algorithm. The

BHA can converge to the global optimum in all the runs and cannot be trapped in local optimum like some other heuristics [23] [24] [26]. In this work, the reasons behind using BHA to train CNN is due to its simple structure, easy implementation, and its parameter-less structure.

2.2. The basics of CNNs

The concept of DL was introduced by Kelly [27] in 1960 based on the idea of an output becoming the input in the next iteration. Basically, the CNN is a form of DL network that comprised of several convolution layers, the pooling layers (for the reduction of the size of the received input), the ReLU layers that enhances the outputs non-linearity, and the Fully Connected Layer that control output range using the softmax activation function. The CNN exploits the advantages of the relationship between the neighboring neurons. The 3 basic concepts of CNN are the local receptive fields, pooling, and shared weights [28], [29]. Figure 1 depicts a typical CNN architecture.

- Local receptive fields (LRF): The input region is small in size and therefore consist of only neuron found in the first 1st hidden layer. The LRF is passed across the input images to produce a different hidden neuron that is comprised of the whole first hidden layer.
- Shared weights and biases: The role of this feature is to detects the presence of similar features from different input image locations; the feature map is designated to the map between the input and hidden layers, while the weights are regarded as the shared weights and biases. The filter or kernel to be used is also determined by the shared weights and biases. Image recognition is performed using more than one feature map for the detection of several features; hence, different feature maps can co promise the convolutional layer.
- Pooling layers: This is the layer next to the convolutional layers ad its role is to analyze the incoming info from the convolution layer. The common pooling procedures commonly used are Max–pooling and L2 pooling approaches; Max-pooling involves maximum neuron activation while L2 pooling relies on the square root of the sum of the squares of the activations in the 2×2 region. These three concepts collectively form a complete CNN architecture as they are the major components of the connection layer.

Before introducing one or more fully connected layers, several convolutions, activation function, and pooling stages are merged first. A loss function is adopted by the model output for the evaluation of the performance in terms of the differences the actual image label and the CNN output. The training of the CNN is aimed at reducing the loss function, and this is achieved using the stochastic gradient descent [7] which is an optimization strategy that first uses the weight of each edge in the network for the estimation of the loss function gradient before updating the weights using the computed gradient [27].

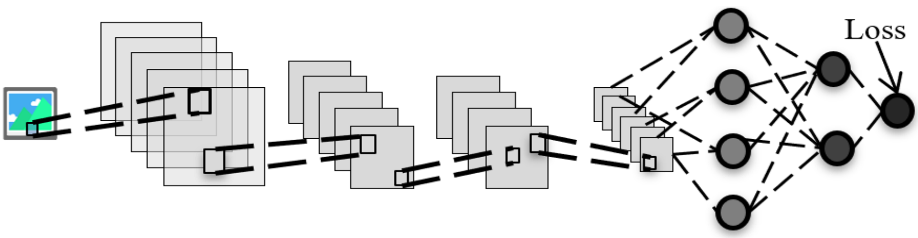

Figure 1. The typical architecture of a CNN model

3. THE PROPOSED TRAINING ALGORITHM

This paper presented the development of a training algorithm for CNN; hence, the major aim here is to determine the best values for the parameters of CNN. The BHA was used in this work as the training algorithm for the two reasons earlier stated. Figure 2 depicts the schematic of the proposed CBH-CNN which illustrates the main steps of the CBH-CNN. However, there are three main steps that need to be detailed as presented in the subsequent subsections.

3.1. Encoding the stars

The two basic parts of a CNN are the feature extraction component and the classification component; feature extraction (FE) as a process requires the input of several convolution layers, activation function, and Max-pooling. The classifier is mostly made up of fully connected NN layers. The weights set of the NN in the proposed system is considered the structure of each wolf; hence, the approach utilizes a vector of real values that contains all the NN weights. Figure 3 is a depiction of a star for the training of NNs.

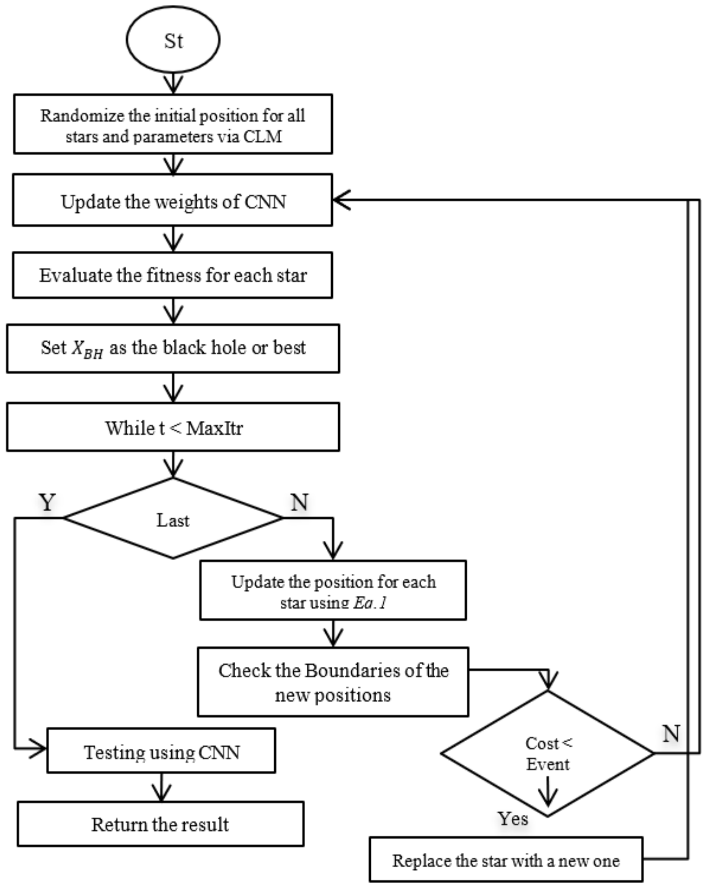

Figure 2. CBH-CNN

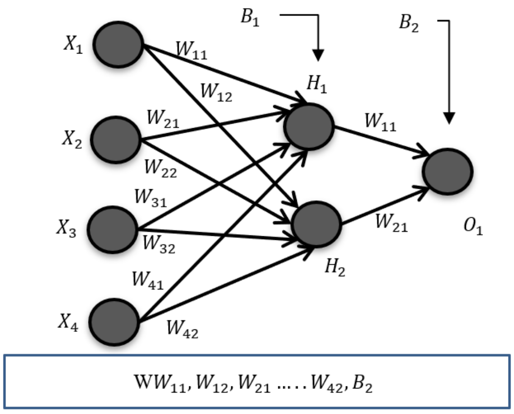

Figure 3. The encoding of each solution in the neural network in CNN [25]

3.2. Stars initialization

To initialize the population, the population size is first set up, followed by random creation of stars until the expected population size is reached. As stated earlier, the chaotic logistic map is used for initializing the stars in the search space, as in (3):

$$x_{i+1} = (Upper - Lower) \times [(1 - X_i) \times \mu] + Lower \tag{3}$$

where x_i is the current position of a star, x_{i+1} is the next position generated via logistic map, UB and LB is values of upper bound and lower bound, respectively, and μ=a controlling parameter, in range [0, 4].

3.3. Evaluation function

The cost function of any classification problem is the classification error which can be estimated thus:

$$Err = \frac{FP+FN}{FP+FN+TP+TN} \tag{4}$$

where *FP* is false positive, *FN* is false negative, *TP* is true positive, and *TN* is true negative; these metrics are computed based on the confusion matrix. The actual classification error value is determined by the number of incorrectly classified samples; this implies that the CNN is used to classify the samples based on the values of each star. The pseudocode of the proposed CBH-CNN is given in Figure 4.

CBH-CNN Algorithm

Input: Dataset, $Iterations, No.of\ Stars, Max\ Bound, Min\ Bound$
Output: Best Star
Procedure:
  Generate the initial positions for all stars X in via (3)
  Update the weights and biases of CNN using each star in the population
  Calculate the evaluation value of each star via (4)
  Determine the black hole star which is the current best solution
  For $itr = 1 \rightarrow Iterations$
    For $i = 1 \rightarrow No.ofStars$
      Calculate the new position for $star_i$ $using$ (1)
      Check the boundaries for the new position of $star_i$
      Set the weights and biases of CNN using the new generated position
      Calculate the evaluation value of each $star_i$ via (4)
    End For
    Determine the event horizon using (2)
    For $i = 1 \rightarrow No.ofStars$
      If $Star_i$ crosses the event horizon *Then*
        Replace the $Star_i$ with a new generated star via (3)
      End If
    End For
    Determine the black hole star which is the current best solution
  Next i
  Return final black hole solution

Figure 4. The main steps of CBH-CNN

4. RESULTS AND DISCUSSION

In this study, we have evaluated our enhanced algorithm based on the same experiment done by the dataset used was sourced from the database of the Modified National Institute of Standards and Technology (MNIST); the dataset comprised of 70,000 images. The dataset was divided into two parts; the first part comprised 60,000 images (considered the training set) which were scanned images of the handwriting of 250 people of which 50% of the people are employees of the US Census Bureau while the remaining 50% are images captured from the handwriting of high school students. These images were stored in the grayscale mode and were sized 28×28 pixels. The remaining 10,000 images in the dataset were used as the testing dataset; they were collected from another set of 250 persons for the sake of comparison) [29]. A population size of 30 stars was used in this study while the maximum number of epochs is 20. The new approach was benchmarked against the standard CNN, deep belief network (DBN), convolutional neural network-based simulating annealing (CNN-SA) [30], and CNN-based PSO in terms of performance [31], convolutional neural network based standard black hole (BH-CNN) [25] as shown in Table 1; the comparison was in terms of the classification accuracy based only on five epochs.

All used models attained almost similar performances, as shown in Table 1, while the proposed CBH-CNN performed better based on the considered metric. The performance of CNN-PSO algorithm was acceptable but not up to the performance of the new CBH-CNN. The PSO algorithm has three control parameters (cognitive parameter c_1, social parameters c_2, and inertia weight w) and these parameters must be optimally tunned (this is another optimization problem). Hence, the simple structure of BHA, which requires no parameter tuning, makes it a simpler ML model with less complexity compared to other models. Figure 5 and Figure 6 demonstrated the performance accuracy and error rate of the evaluated models, accordingly. The comparison of the adopted models based on the execution time is presented in Figure 7. From Figure 7, it can be noticed that DBN requires less execution time owing to its training process that relies on a rapid contrastive divergence. The proposed CBH-CNN shows longer execution time as compared to DBN and standard CNN due to the additional iterations required by the BHA, and the need to calculate the event horizon.

Table 1. The comparison of the performance of the new CBH-CNN some existing models

Models	Epochs				
	Epoch=1	*Epoch=2*	*Epoch=3*	*Epoch=4*	*Epoch=5*
CNN	88.87	92.25	93.9	94.81	95.68
DBN	87.46	89.72	90.64	91.14	92.93
CNN-SA	89.18	92.38	94.2	95.19	96.04
CNN-PSO	89.52	92.38	93.91	95.08	96.31
BH-CNN	89.91	92.38	93.91	95.08	96.31
CBH-CNN	89.93	93.12	94.98	95.62	96.96

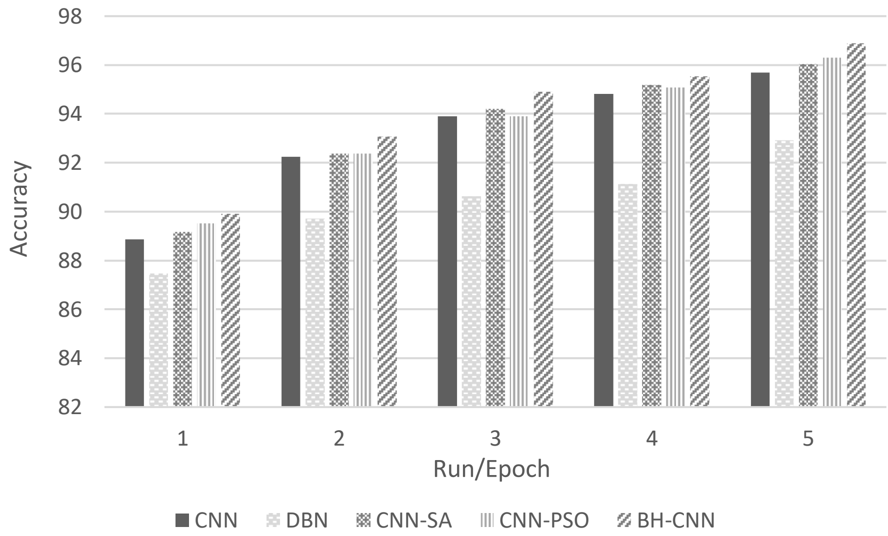

Figure 5. The classification accuracy

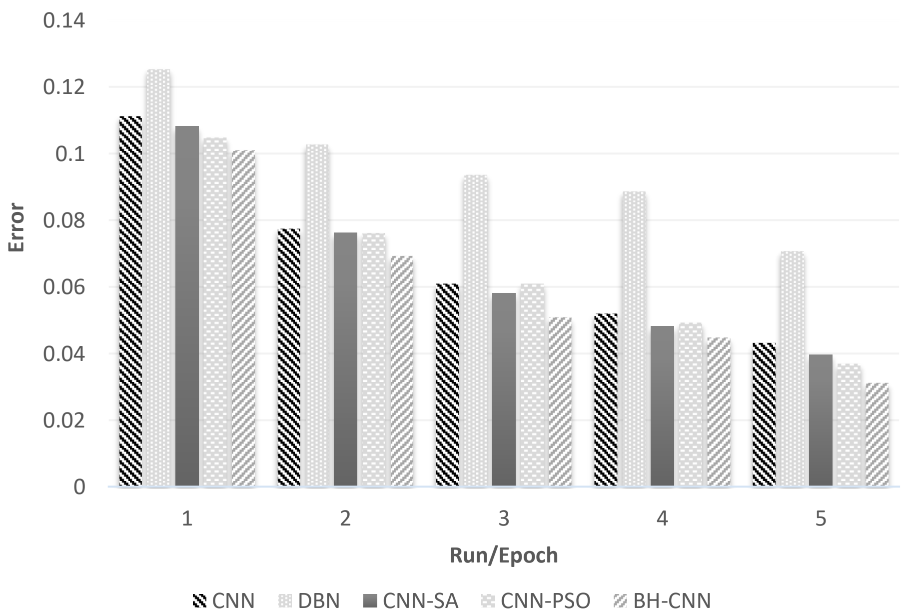

Figure 6. The error rates

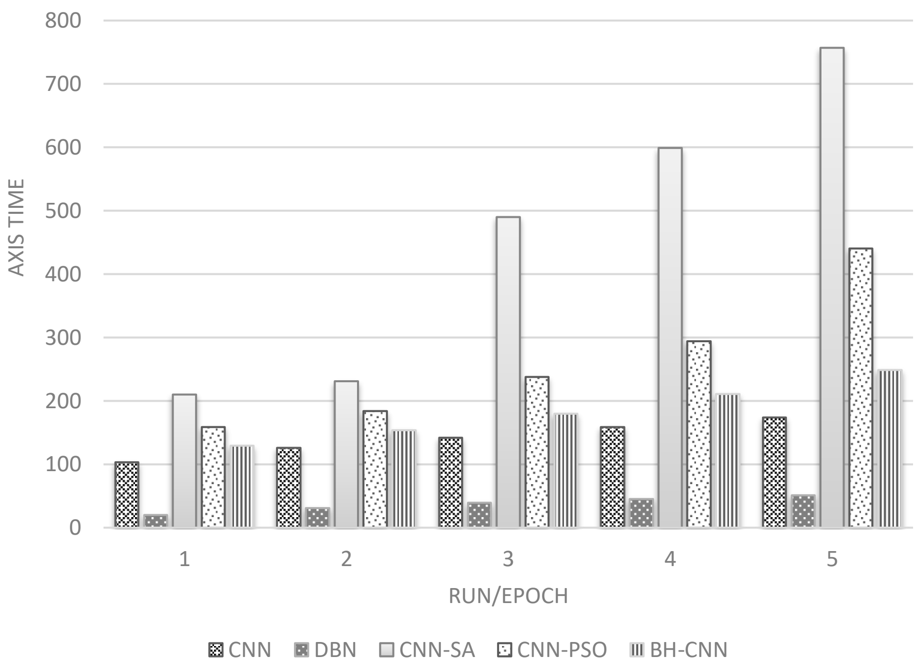

Figure 7. The execution times

5. CONCLUSION

Deep learning models have received the most attention in the field of machine learning because they can recognize the patterns in most classification problems. The CNN is among the well-known DL models and its main structure consist of two parts-feature extraction and neural network. This paper focused on the training of the second part of the CNN using CBHA, a novel nature-inspired algorithm. The proposed model outperformed the rest of the DL models on the MNIST dataset using only five epochs. The outcome of this study showed that the BHA is a suitable alternative for the training of CNN algorithms.

REFERENCES

[1] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006, doi: 10.1162/neco.2006.18.7.1527.

[2] N. B. Karayiannis and G. W. Mi, "Growing radial basis neural networks: merging supervised and unsupervised learning with network growth techniques," *IEEE Transactions on Neural Networks*, vol. 8, no. 6, pp. 1492–1506, Nov. 1997, doi: 10.1109/72.641471.

[3] A. G. Ivakhnenko and V. G. Lapa, "Cybernetic predicting devices," 1966.

[4] T. Dhieb, W. Ouarda, H. Boubaker, and A. M. Alimi, "Deep neural network for online writer identification using beta-elliptic model," in *2016 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2016, pp. 1863–1870, doi: 10.1109/IJCNN.2016.7727426.

[5] L. Haddad, T. M. Hamdani, W. Ouarda, A. M. Alimi, and A. Abraham, "An adaptation module with dynamic radial basis function neural network using significance concept for writer adaptation," *Journal of Information Assurance*, vol. 12, no. 1, 2017.

[6] 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, doi: 10.1109/5.726791.

[7] Y. LeCun and M. Ranzato, "Deep learning tutorial," in *Tutorials in international conference on machine learning (ICML'13)*, 2013, pp. 1–29.

[8] B. Wang, Y. Sun, B. Xue, and M. Zhang, "Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, Jul. 2018, pp. 1–8, doi: 10.1109/CEC.2018.8477735.

[9] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *2009 IEEE 12th International Conference on Computer Vision*, Sep. 2009, pp. 2146–2153, doi: 10.1109/ICCV.2009.5459469.

[10] T. Yamasaki, T. Honma, and K. Aizawa, "Efficient optimization of convolutional neural networks using particle swarm optimization," in *2017 IEEE Third International Conference on Multimedia Big Data (BigMM)*, Apr. 2017, pp. 70–73, doi: 10.1109/BigMM.2017.69.

[11] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the 6th International Symposium*, 1995, pp. 39–43, doi: 10.1109/MHS.1995.494215.

[12] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, vol. 5792, pp. 169–178, doi: 10.1007/978-3-642-04944-6_14.

[13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.
[14] R. V Rao, V. J. Savsani, and D. P. Vakharia, “Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems,” *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, Mar. 2011, doi: 10.1016/j.cad.2010.12.015.
[15] S. Q. Salih, A. A. Alsewari, and Z. M. Yaseen, “Pressure vessel design simulation,” in *Proceedings of the 2019 8th International Conference on Software and Computer Applications*, Feb. 2019, pp. 120–124, doi: 10.1145/3316615.3316643.
[16] S. Q. Salih *et al.*, “Integrative stochastic model standardization with genetic algorithm for rainfall pattern forecasting in tropical and semi-arid environments,” *Hydrological Sciences Journal*, vol. 65, no. 7, pp. 1145–1157, May 2020, doi: 10.1080/02626667.2020.1734813.
[17] U. Beyaztas, S. Q. Salih, K.-W. Chau, N. Al-Ansari, and Z. M. Yaseen, “Construction of functional data analysis modeling strategy for global solar radiation prediction: application of cross-station paradigm,” *Engineering Applications of Computational Fluid Mechanics*, vol. 13, no. 1, pp. 1165–1181, Jan. 2019, doi: 10.1080/19942060.2019.1676314.
[18] A. H. Zahid *et al.*, “A novel construction of dynamic s-box with high nonlinearity using heuristic evolution,” *IEEE Access*, vol. 9, pp. 67797–67812, 2021, doi: 10.1109/ACCESS.2021.3077194.
[19] K. Z. Zamli, A. Kader, F. Din, and H. S. Alhadawi, “Selective chaotic maps tiki-taka algorithm for the s-box generation and optimization,” *Neural Computing and Applications*, vol. 33, no. 23, pp. 16641–16658, Dec. 2021, doi: 10.1007/s00521-021-06260-8.
[20] S. Q. Salih, A. A. Alsewari, B. Al-Khateeb, and M. F. Zolkipli, “Novel multi-swarm approach for balancing exploration and exploitation in particle swarm optimization,” in *In Proceesdings of 3rd International Conference of Reliable Information and Communication Technology 2018 (IRICT 2018)*, 2018, pp. 196–206, doi: 10.1007/978-3-319-99007-1_19.
[21] H. Tao *et al.*, “A newly developed integrative bio-inspired artificial intelligence model for wind speed prediction,” *IEEE Access*, vol. 8, pp. 83347–83358, 2020, doi: 10.1109/ACCESS.2020.2990439.
[22] H. S. Alhadawi, S. Q. Salih, and Y. D. Salman, “Chaotic particle swarm optimization based on meeting room approach for designing bijective S-boxes,” in *Proceedings of International Conference on Emerging Technologies and Intelligent Systems*, 2022, pp. 331–341, doi: 10.1007/978-3-030-85990-9_28.
[23] A. Hatamlou, “Black hole: a new heuristic optimization approach for data clustering,” *Information Sciences*, vol. 222, pp. 175–184, Feb. 2013, doi: 10.1016/j.ins.2012.08.023.
[24] S. Kumar, D. Datta, and S. K. Singh, “Black hole algorithm and its applications,” in *Computational Intelligence Applications in Modeling and Control*, Springer International Publishing, 2015, pp. 147–170, doi: 10.1007/978-3-319-11017-2_7.
[25] S. Q. Salih, “A new training method based on black hole algorithm for convolutional neural network,” *Journal of Southwest Jiaotong University*, vol. 54, no. 3, p. 1, Jun. 2019, doi: 10.35741/issn.0258-2724.54.3.22.
[26] A. P. Piotrowski, J. J. Napiorkowski, and P. M. Rowinski, “How novel is the ‘novel’ black hole optimization approach?,” *Information Sciences*, vol. 267, pp. 191–200, May 2014, doi: 10.1016/j.ins.2014.01.026.
[27] H. J. Kelley, “Gradient theory of optimal flight paths,” *ARS Journal*, vol. 30, no. 10, pp. 947–954, Oct. 1960, doi: 10.2514/8.5282.
[28] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, “Towards better analysis of deep convolutional neural networks,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 91–100, Jan. 2017, doi: 10.1109/TVCG.2016.2598831.
[29] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, “Understanding neural networks through deep visualization,” Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.06579
[30] L. M. R. Rere, M. I. Fanany, and A. M. Arymurthy, “Simulated annealing algorithm for deep learning,” *Procedia Computer Science*, vol. 72, pp. 137–144, 2015, doi: 10.1016/j.procs.2015.12.114.
[31] A. R. Syulistyo, D. M. Jati Purnomo, M. F. Rachmadi, and A. Wibowo, “Particle swarm optimization (PSO) for training optimization on convolutional neural network (CNN),” *Jurnal of Computer Sciences and Information*, vol. 9, no. 1, Feb. 2016, doi: 10.21609/jiki.v9i1.366.

BIOGRAPHIES OF AUTHORS

Sinan Q. Salih received the B.Sc. degree in information systems from the University of Anbar, Al-Anbar, Iraq, in 2010, the M.Sc. degree in computer sciences from Universiti Tenaga National (UNITEN), Malaysia, in 2012, and the Ph.D. degree in soft modeling and intelligent systems from Universiti Malaysia Pahang (UMP). His current research interests include optimization algorithms, nature-inspired metaheuristics, machine learning, and feature selection problem for real world problems. He can be contacted at email: sinan.salih@sadiq.edu.iq.

Ahmed L. Khalaf received his B.Sc Eng. degree (Control and Systems Engineering) from University of Technology, Iraq (2001) and M.Sc. Eng, degree (Computer Engineering) from Middle Technical University, Iraq (2008). He did his PhD research at Universiti Putra Malaysia, Malaysia, (2018) in the area of optical sensor based on nanomaterials for chemical sensing applications. Currently, he is a senior lecturer at the Department of Computer Engineering Techniques, Al-Ma’moon University College. His main research interests are fiber optics sensors, optical chemical sensors, nanomaterials, and computer engineering. He can be contacted at email: ahmed.l.khalaf@sadiq.edu.iq.

Nuha Sami Mohsin was born in Baghdad, Iraq. She received her M.Sc. and degree in Computer Science in 2005 and 2021 from University of Technology and University of Information and Communication Technology/Institute of Informatics for Postgraduate Studies respectively. She is currently a doctor teacher in information technology unit at collage of education Ibn Rushed/University of Baghdad. Her research interests include optimization, machine learning, artificial intelligence, algorithm optimization, software engineering, copyright and watermark protection and information security. She can be contacted at email: nuha.sami@ircoedu.uobaghdad.edu.iq.

Saadya Fahad Jabbar was born in Baghdad, Iraq. She received her B.Sc. and M.Sc. degree in Computer Science in 2004 and 2015 from University of Baghdad and University of Al-Mustansiriyah respectively. She is currently a senior lecturer in information technology unit at collage of education Ibn Rushed/University of Baghdad. Her research interests include optimization, machine learning and natural language processing. She can be contacted at email: saadya.fahad@ircoedu.uobaghdad.edu.iq.