

Research Article

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

Sara Khosravi  and Abdolah Chalechale 

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

Correspondence should be addressed to Abdolah Chalechale; 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 Abdolah 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 5×5 to extract key information from the input data (the input grayscale image size is 32×32) for creating each feature map. Every neuron of Conv1 accepts achieved inputs from a 5×5 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 5×5 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 2×2 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 5×5 kernel for each feature map and 16 feature maps. The succeeding layer (Pooling4) is a downsampling layer with a 2×2 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 5×5 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]	32×32
Conv1	28×28	6	156	122304
Pooling2	14×14	6	12	5880
Conv3	10×10	16	1516	151600
Pooling4	5×5	16	32	2000
Conv5	1×1	120	...	48120
FC6	1×1	84	10164	...

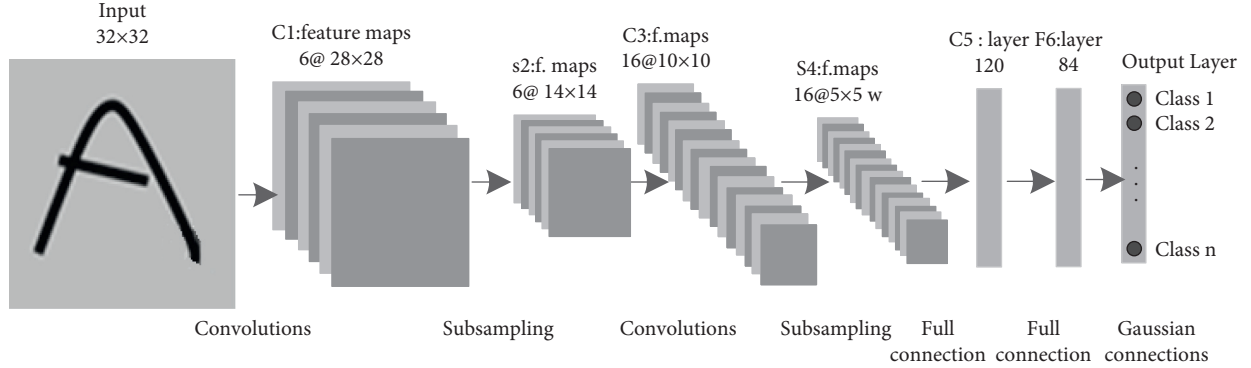


FIGURE 1: Design of LeNet-5 pipeline [51].

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_{chimp} indicates the position vector of a chimp, X_{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

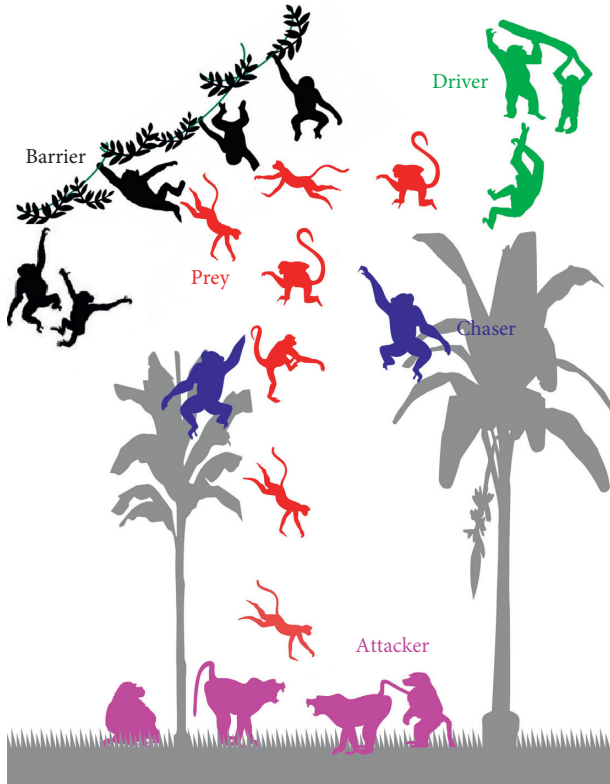


FIGURE 2: Demonstration of the hunting strategy [57].

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 θ_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)

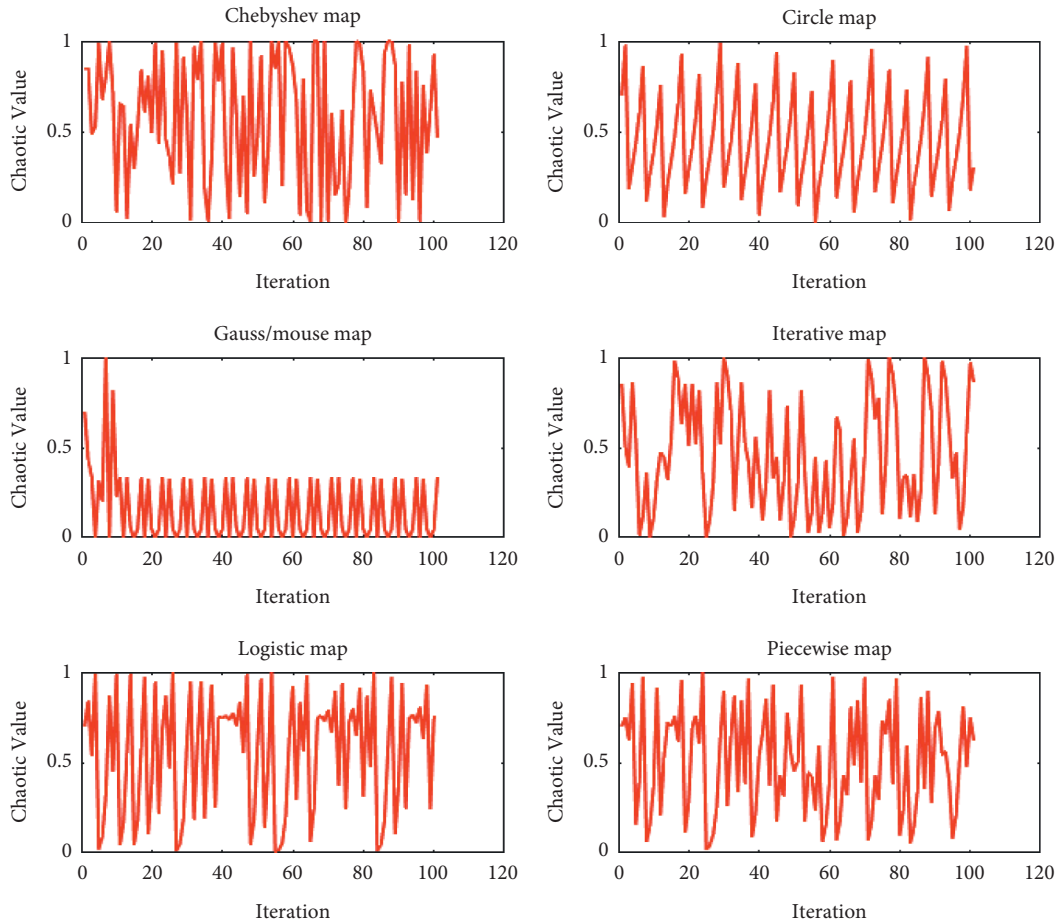


FIGURE 3: The chaotic map diagrams.

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

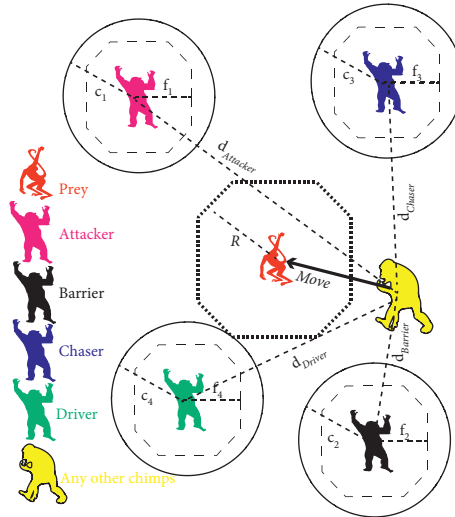


FIGURE 4: The procedure of position update [57].

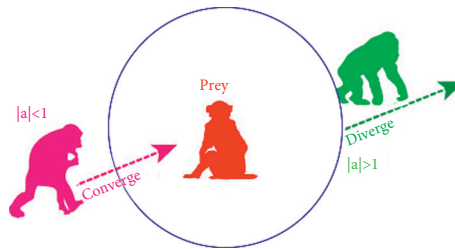


FIGURE 5: Position updating mechanism of chimps and effects of $|a|$ on it [57].

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

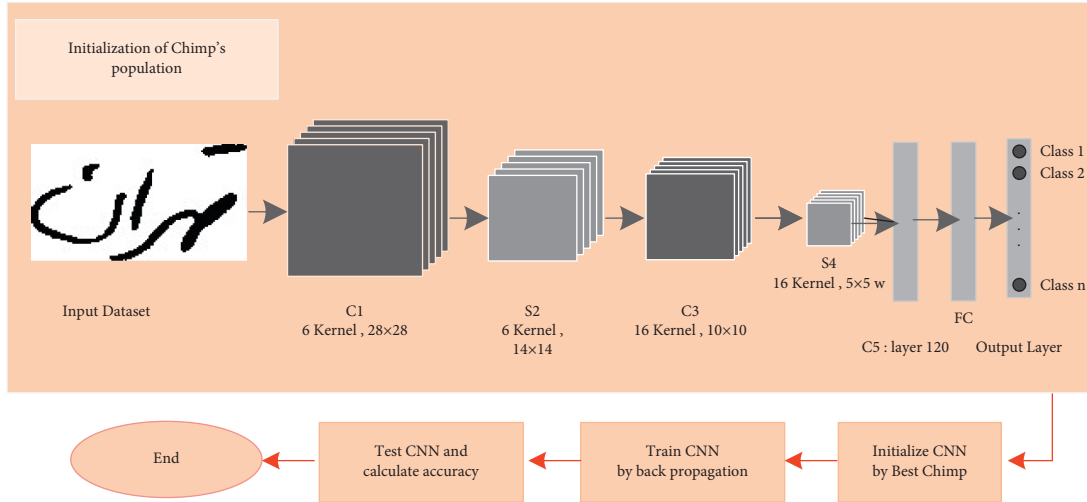


FIGURE 6: Handwritten recognition using CNN and ChOA.

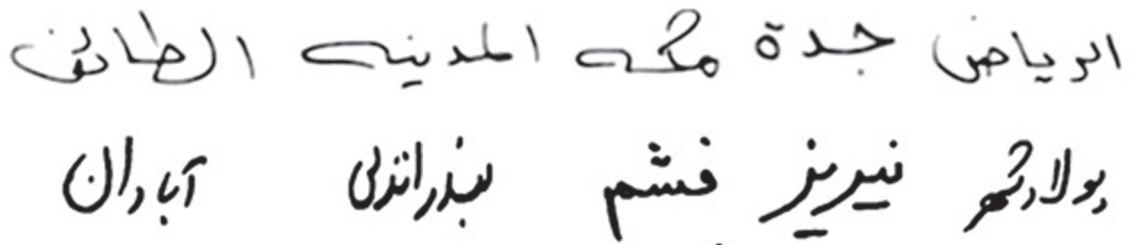


FIGURE 7: Some samples from the Arabic (first row) and the Persian (second row) datasets.

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.

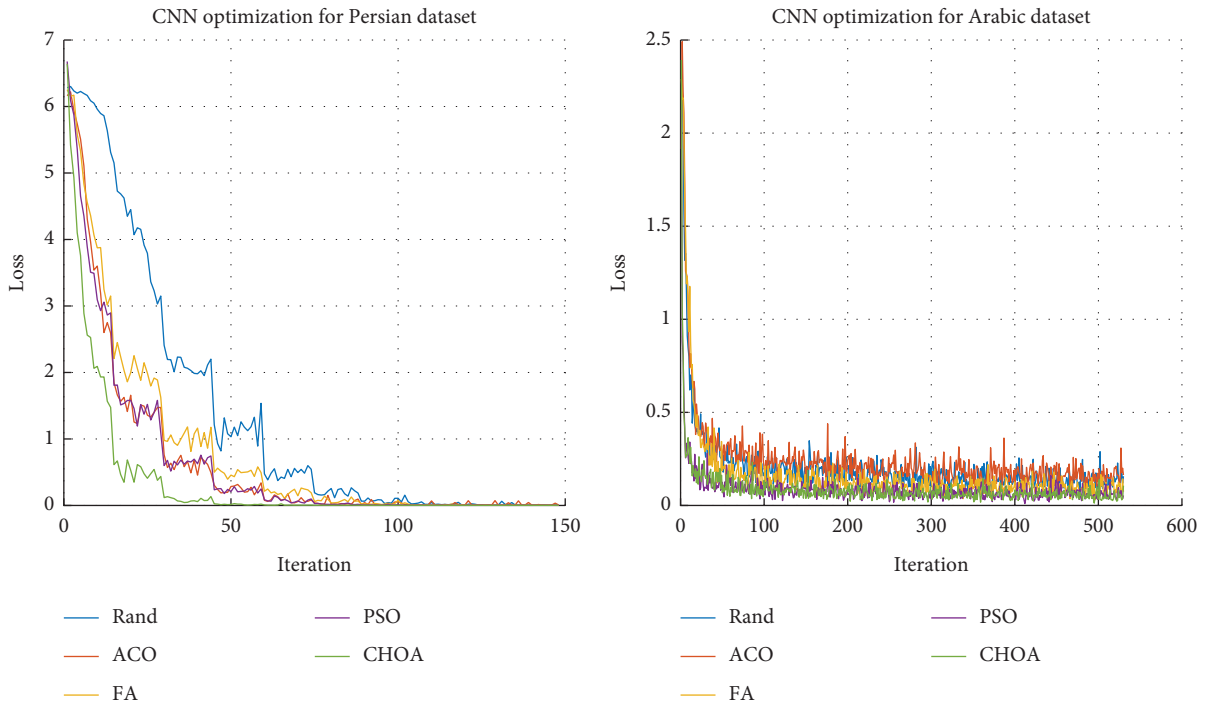


FIGURE 8: Error of classification employing two datasets and four optimization techniques for LeNet-5 for all iterations.

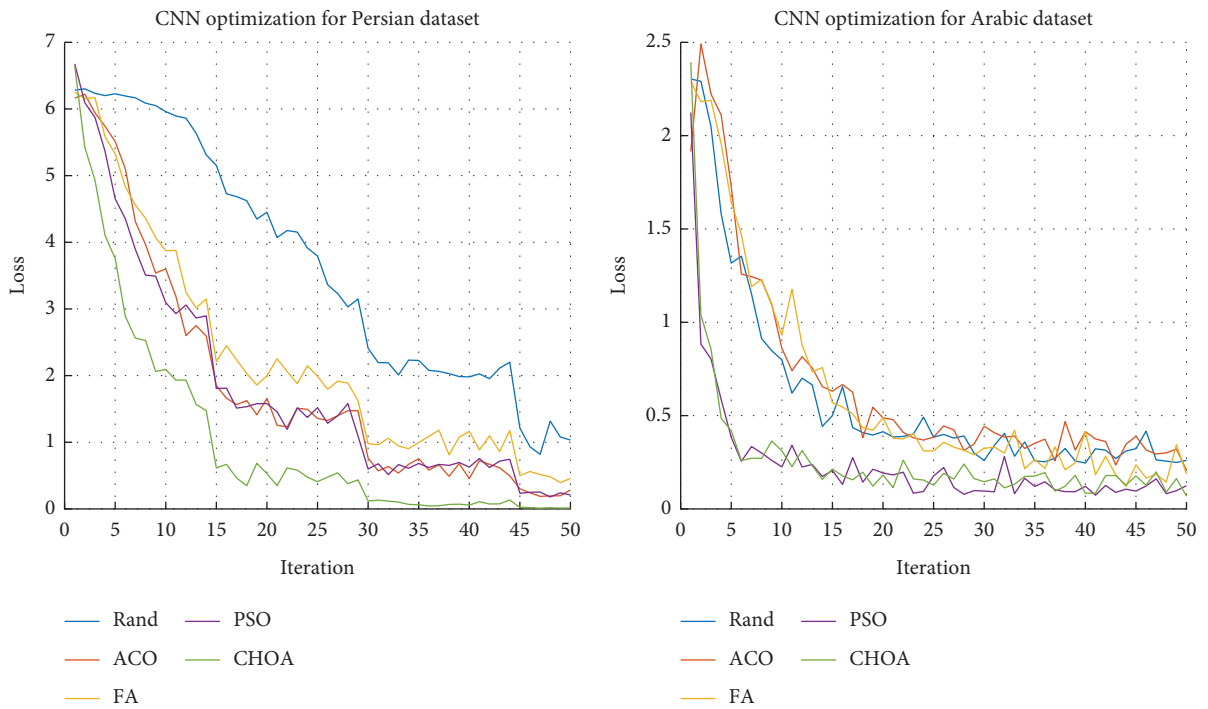


FIGURE 9: Error of classification employing two datasets and four optimization techniques for LeNet-5 for the first fifty iterations.

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
	ChOA	GD	99.17
ENIT/IFN	ChOA	—	69.84
	Random	GD	91.06
	ChOA	GD	97.36

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	99.17

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	97.36

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.