Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Expert Systems With Applications 201 (2022) 116942

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

X-ray image based COVID-19 detection using evolutionary deep learning approach

Seyed Mohammad Jafar Jalali [a,*], Milad Ahmadian [b], Sajad Ahmadian [c], Rachid Hedjam [d], Abbas Khosravi [a], Saeid Nahavandi [a]

[a] Institute for Intelligent Systems Research and Innovation, (IISRI), Deakin University, Geelong, Australia
[b] Department of Computer Engineering, Razi University, Kermanshah, Iran
[c] Faculty of Information Technology, Kermanshah University of Technology, Kermanshah, Iran
[d] Department of Computer science, Sultan Qaboos University, Muscat, Sultanate of Oman

ARTICLE INFO

Keywords:
COVID-19
Image classification
Coronavirus
Deep neuroevolution learning
Convolutional neural network
K-nearest neighbor classifier

ABSTRACT

Radiological methodologies, such as chest x-rays and CT, are widely employed to help diagnose and monitor COVID-19 disease. COVID-19 displays certain radiological patterns easily detectable by X-rays of the chest. Therefore, radiologists can investigate these patterns for detecting coronavirus disease. However, this task is time-consuming and needs lots of trial and error. One of the main solutions to resolve this issue is to apply intelligent techniques such as deep learning (DL) models to automatically analyze the chest X-rays. Nevertheless, fine-tuning of architecture and hyperparameters of DL models is a complex and time-consuming procedure. In this paper, we propose an effective method to detect COVID-19 disease by applying convolutional neural network (CNN) to the chest X-ray images. To improve the accuracy of the proposed method, the last Softmax CNN layer is replaced with a K-nearest neighbors (KNN) classifier which takes into account the agreement of the neighborhood labeling. Moreover, we develop a novel evolutionary algorithm by improving the basic version of competitive swarm optimizer. To this end, three powerful evolutionary operators: Cauchy Mutation (CM), Evolutionary Boundary Constraint Handling (EBCH), and tent chaotic map are incorporated into the search process of the proposed evolutionary algorithm to speed up its convergence and make an excellent balance between exploration and exploitation phases. Then, the proposed evolutionary algorithm is used to automatically achieve the optimal values of CNN's hyperparameters leading to a significant improvement in the classification accuracy of the proposed method. Comprehensive comparative results reveal that compared with current models in the literature, the proposed method performs significantly more efficient.

1. Introduction

At the end of 2019, in Wuhan, China, the first outbreak of novel coronavirus disease (COVID-19) occurred. The COVID-19 virus attacks and mutates rapidly in the lungs of a confirmed patient. The tainted lungs are inflamed and overflowing with fluid in such a situation. When we undertake CT-Scan or X-ray images of an infected person, the results reveal dark spots in the lungs named Ground Glass Opacity (Parekh et al., 2020). The spreading level of COVID-19 disease is significantly higher than its forecasting or detection speed according to its highly contagious existence (Ismael & Şengür, 2020). In the failure of an intelligent diagnostic tool, it is important to diagnose suspected COVID-19 patients quickly and accurately (Chakraborty & Mali, 2020; Panwar, Gupta, Siddiqui, Morales-Menendez, Bhardwaj et al., 2020). The current techniques available to diagnose this spreading pandemic are slightly accurate and time-consuming (Alamoodi et al., 2020; Shi et al., 2020). There are generally three major variations of COVID-19 testing procedures: Reverse Transcription Polymerase Chain Reaction (RT-PCR), Computed Tomography (CT), and Chest X-ray (CXR) (Yoo et al., 2020). RT-PCR is among these three recognized as time-consuming approach. CT-Scan can detect inflammation of the lungs in terms of location, shape, and scale. CXR offers a more accurate way of diagnosis for COVID-19 and a clear image of air sacs. Thus, we have chosen CXR images of the lungs for the experiments conducted in this work (Panwar, Gupta, Siddiqui, Morales-Menendez, Bhardwaj et al., 2020; Pereira et al., 2020).

Medical institutions produce and collect large volumes of data that provide immensely helpful patterns and information, which go way

* Corresponding author.
E-mail addresses: mohammadjj.it@gmail.com (S.M.J. Jalali), m.ahmadian@stu.razi.ac.ir (M. Ahmadian), s.ahmadian239@gmail.com (S. Ahmadian), rachid.hedjam@squ.edu.om (R. Hedjam), abbas.khosravi@deakin.edu.au (A. Khosravi), saeid.nahavandi@deakin.edu.au (S. Nahavandi).

https://doi.org/10.1016/j.eswa.2022.116942
Received 8 November 2020; Received in revised form 24 December 2021; Accepted 17 March 2022
Available online 30 March 2022

further than what traditional analytical methods can handle. It should be noted that the RT-PCR test is one of the expensive and time-consuming approaches for the identification of suspects of COVID-19 (Ahmadian, Jalali et al., 2021; Jalali, Ahmadian, Ahmadian et al., 2021; Kashir & Yaqinuddin, 2020). Therefore, a better approach could be found in which a combination of deep learning classifiers and clinical images provides rapid and reliable identification of the COVID-19 virus throughout CXR analysis of pulmonary images. Deep Learning is a machine learning category that facilitates computer systems to learn important data features automatically from a set of data gathered for different applications such as machine vision (Pouyanfar et al., 2018; Yadav & Jadhav, 2019) and recommender systems (Ahmadian, Ahmadian et al., 2021; Ahmadian et al., 2022; Ahmadian, Joorabloo et al., 2018; Ahmadian et al., 2020, 2018a, 2018b, 2014; Moradi et al., 2016; Rahmani et al., 2019; Tahmasebi et al., 2021). Convolutional neural network (CNN) is one of the popular deep learning models which has been widely used in different research area such as medial applications. CNN can take input in 2D or 3D images into consideration and make proper utilization of spatial and features information (Yuan et al., 2018).

Machine learning algorithms have been extensively employed in different real applications such as image classification. In Altan and Karasu (2020), an effective COVID-19 detection approach is introduced by integrating Curvelet transformation, chaotic salp swarm algorithm, and deep learning models. For this purpose, a feature matrix is constructed by applying Curvelet transformation on the chest X-ray images of patients. Moreover, the chaotic salp swarm algorithm is utilized to optimize the feature matrix. Then, the EfficientNet-B0 model is used as a deep learning approach to detect COVID-19 cases. Karasu et al. (2017) proposed a feature representation model based on KNN approach to improve the performance of image classification algorithms. To this end, a character segmentation method is employed to derive 35 different characters from the background of the gray scale images which can be classified using KNN. Then, a set of effective features are obtained using the histogram components of the derived characters. The authors showed that this representation of features is able to significantly improve the image classifiers' accuracy. A novel COVID-19 detection approach is introduced in Al-Itbi et al. (2022) in which the X-ray and CT images are utilized as the inputs. Particularly, a feature extraction method is employed based on the Scatter Wavelet Transform to derive the features of input images and then, the extracted features are used as the inputs of Dense Deep Neural Network to classify the images into COVID-19/Non-COVID-19 cases. Hu et al. (2022) proposed a novel framework for COVID-19 detection by integrating classification and regression tasks in a multi-task multi-modality support vector machine approach. Accordingly, a feature extraction method is applied to the chest CT images to obtain appropriate features for each segment of input image. Then, an effective model is considered based on data augmentation of over-sampling approach to address class imbalance problem. Finally, the proposed support vector machine model is utilized to detect COVID-19 cases. In Deb et al. (2022), four different deep CNN models: VGGNet, GoogleNet, DenseNet, and NASNet are integrated as an ensemble model to extract latent features from the chest X-ray images. Then, the extracted features of X-ray images are used as the inputs of a fully connected layer to make an effective COVID-19 disease detection approach. The authors have investigated the performance of their proposed method on two COVID-19 datasets where the results demonstrate that the proposed ensemble method outperforms single-mode classification models.

Although the deep neural network models have gained considerable accuracy in image classification applications, the design of these models has been mainly done manually, which is a time-consuming and complex process. It is worth mentioning that deep neural networks contain several hyperparameters which determining their values is a challenging task. Also, the efficiency of these models is strongly dependent on the values used for their hyperparameters. Therefore, developing effective methods to automatically determine the values of these hyperparameters is a critical issue to improve the performance of deep neural networks (Ahmadian & Khanteymoori, 2015; Jalali, Ahmadian et al., 2019, 2020; Jalali, Khosravi et al., 2019; Jalali, Osorio et al., 2021; Khodayar et al., 2021; Mousavirad et al., 2020). Such methods should automatically provide the optimal structure of deep neural networks leading to obtain the most accuracy in the shortest time. Deep Neuroevolution (DNE) is a remarkably effective and feasible approach to obtain automatically the architectures of deep neural networks using powerful evolutionary algorithms. The aim of this approach is to select the best values for hyperparameters leading to promote accuracy, minimize the network over-fitting, and promote consistency. However, evolutionary algorithms mainly suffer from low convergence speed and tapping into local optima. Moreover, making a balance between the exploration and exploitation phases of these algorithms plays a critical role in enhancing their ability of search process (Ahmadian, Jalali, Raziani et al., 2021; Jalali et al., 2022; Jalali, Ahmadian, Khodayar et al., 2021; Jalali, Ahmadian, Khosravi et al., 2021; Jalali, Hedjam et al., 2020; Jalali, Khodayar et al., 2021; Jalali, Khosravi, Kebria et al., 2019; Qazani et al., 2020).

In this paper, an effective image classification method is proposed based on deep CNN model to detect COVID-19 disease from chest X-ray images. In order to improve the classification accuracy of the proposed method, the last Softmax CNN layer is replaced with a KNN classifier to take into account the agreement of the neighborhood labeling. Moreover, an evolutionary algorithm is developed by improving the original version of competitive swarm optimizer (CSO) model (Cheng & Jin, 2014). To this end, three powerful evolutionary operators: Cauchy Mutation (CM), Evolutionary Boundary Constraint Handling (EBCH), and tent chaotic map are incorporated into the search process of CSO model to improve its search capabilities, increase its convergence speed, and make an excellent balance between the exploration and exploitation phases. Then, the improved evolutionary algorithm is employed to derive the optimal architecture of deep CNN model for classifying X-ray images whether or not they are affected by COVID-19 disease. Therefore, the proposed COVID-19 detection method takes much shorter time in designing of deep neural network compared to manual methods, and at the same time achieves higher classification accuracy. In summary, the principal contributions of this work are as follows:

- A novel image classification approach is developed by employing deep CNN model to identify patients whether or not they are infected with COVID-19 disease according to the chest X-ray images. In the proposed method, the last Softmax CNN layer is replaced with a KNN classifier to improve the classification accuracy by taking into account the agreement of the neighborhood labeling.
- An effective evolutionary algorithm is proposed by considering three powerful evolutionary operators including Cauchy mutation, evolutionary boundary constraint handling, and tent chaotic map in the search process of the original version of competitive swarm optimizer model. The main advantage of the proposed evolutionary algorithm is to make a balance between the exploration and exploitation phases which results in increasing convergence speed and reducing the probability of falling into local optima.
- The proposed evolutionary algorithm is applied to automatically achieve the optimal values of hyperparameters of CNN model leading to a significant improvement in the accuracy of classification method. Therefore, different from most of the previous methods, the proposed method does not need to perform a manual trial and error process to obtain the appropriate values of CNN's hyperparameters.
- Extensive experiments are carried out to assess the effectiveness of the proposed DNE algorithm. The results demonstrate that the proposed DNE method can detect COVID/NON-COVID cases more accurate and reliable than the other state-of-the-art models.

The remainder of this paper is structured accordingly. The literature review is discussed in Section 2. Section 3 outlines the entire proposed DNE classification method for COVID-19 images. Section 4 explains the experimental procedures. Finally, Section 5 sets out the conclusion remarks and future directions.

2. Literature review

Although the COVID-19 disease is a very new issue in the world, an enormous number of researches have been conducted to address different challenges related to this disease. Specifically, the researchers have recently investigated the utilization of deep classifiers to retrieve the features of the appearance of COVID-19 disease from different radiological images. A novel deep learning algorithm named nCOVnet is proposed by Panwar, Gupta, Siddiqui, Morales-Menendez, Singh (2020), which is based on the theory of data leakage for the rapid identification of the COVID-19 cases. The detection accuracy in their experiments was achieved in 88% which is not excellent as a classifier used for a very critical issue. Kumar et al. (2020) developed a model named DeQueezeNet, which categorizes X-ray images into two positive and negative categories in the course of detection for COVID-19 patients. With a 94% accuracy and 90% precision, the proposed model predicts the potential for the disease by pre-processing the X-ray images of positive COVID-19 and normal cases. In another work, Luján-García et al. (2020) used X-rays for COVID-19 detection and additional classification among patients who had or did not have COVID-19 infections. Another improvement for the classification of COVID-19 has been introduced in Ozturk et al. (2020) by presenting a CNN-based framework for binary and multi-class views of conditions such as COVID-19, proper pneumonia, and none of them. They reached 98% in binary classification accuracy and 87% in multi-class accuracy using 17 convolutional layers. Shan et al. (2020) developed a unique deep learning model to automatically classify affected lung regions of COVID-19 patients. Their technique has been evaluated for 300 coronavirus infected individuals with the obtained accuracy of 91%. The deployment of transfer learning in COVID-19 detection was outlined by Apostolopoulos and Mpesiana (2020) since there is very little data available. The authors have achieved the 96 percent accuracy, however, the experimental results show that the dataset was not very balanced. Thus, feature extraction and classification have been successful through transfer learning methodology.

In Desai et al. (2020), the authors investigated deep learning models and their contributions in interpreting medical images of COVID-19. This work claims that the main reason to apply deep learning models for medical images of COVID-19 is their successful in processing such images and extracting useful features from them. An open source network named COVID-Net is developed in Wang et al. (2020) to detect COVID-19 cases from chest X-ray images. The main advantage of this framework is its freely availability and the possibility of modification and improvement by other researchers. In addition to this framework, the authors introduced an open access dataset containing 13,975 chest X-ray images across 13,870 positive COVID-19 cases. In another work Mukherjee et al. (2020), the authors employed convolutional neural network to predict COVID-19 cases according to both CT scans and chest X-ray images. Using both CT and chest X-ray images together as the inputs of CNN model is the main advantage of this work which leads to improve the prediction accuracy. They reported an overall accuracy of 96.28% for their proposed approach. An effective approach based on deep neural networks is proposed in Sheykhivand et al. (2021) to automatically identify COVID-19 cases. To this end, four different classes including healthy, viral, bacterial, and COVID-19 are considered for the chest X-ray images in the used benchmark dataset. This proposed method uses an integration of Generative Adversarial Networks (GANs), deep transfer learning, and LSTM to make an effective classification model. The authors compared their model with other deep transfer learning approaches in terms of different evaluation metrics including accuracy, precision, sensitivity, and specificity. The results showed that their model has a promising performance. In Aviles-Rivero et al. (2022), an image classification method is proposed based on a deep learning model to diagnose COVID-19 disease from chest X-ray images. To this end, a graph diffusion model is developed using an optimization methodology to make pseudo-labels of input data for the inputs of deep neural network model. In Bhattacharyy et al. (2022), the chest X-ray images are utilized as the input data to make an effective approach for diagnosing COVID-19 disease. In particular, a segmentation method is developed based on the conditional generative adversarial network to extract the lung area from the chest X-ray images. Then, a deep neural network is employed to derive the latent features from the extracted lung areas. Finally, a number of machine learning methods have been applied to detect COVID-19 cases using the derived latent features. Kumar et al. (2022) developed a novel framework named SARS-Net in which the graph convolutional network and CNN model are integrated to automatically detect the presence of COVID-19 disease in chest X-ray images of patients. The authors compared the SARS-Net model with a number of existing classification approaches, and reported a significant accuracy for their proposed model in comparison to other state-of-the-art methods. In Gaur et al. (2022), a COVID-19 detection method is developed by preprocessing the CT scan images and identifying the positive and negative cases. For this purpose, the CT scan images are preprocessed by the empirical wavelet transformation to obtain their appropriate components of RGB channels. Then, a transfer learning methodology is utilized to classify the input CT scan images based on the obtained components.

By investigating the approaches reviewed in this section, it can be concluded that all of them utilize a static design of deep neural networks for developing their classification models. In other words, these approaches adjust the hyperparameters of deep neural networks manually without considering any heuristic. Whilst, the performance of deep neural networks is significantly dependent on the values selected for their hyperparameters. In addition, adjusting the values of these hyperparameters in a manual way is very time-consuming and might cause to reduce the efficiency of deep neural networks. Different from the previous works, we develop an automatic strategy in this paper to tune the hyperparameters of deep neural networks using an enhanced evolutionary algorithm. Therefore, the main advantage of the proposed method is to automatically determine the optimal values of the hyperparameters of deep neural networks leading to improve the performance of the proposed method in identifying COVID-19 cases.

3. The proposed DNE methodology

In this section, we describe the proposed X-ray image classification framework for detecting coronavirus (COVID-19) which is based on a deep Convolutional Neural Network (CNN) model optimized by an evolutionary algorithm. In order to configure the deep CNN hyperparameters and architecture, automatically, an efficient and scalable algorithm called Modified Competitive Swarm Optimizer (MCSO) has been developed. This allows the network architecture to be more consistent. Furthermore, in order to enhance the accuracy of the proposed image classification model, the latest CNN Softmax layer is substituted by a K-nearest neighbor classifier. Fig. 1 illustrates the overall procedure of the proposed DNE framework.

3.1. Convolutional Neural Network (CNN)

Deep learning is a technique using in machine learning, in which a deep architecture designs several linear and non-linear processor modules to represent high-level functionality in a given dataset. A wide range of applications currently employ several such deep learning methods including restricted Boltzmann machines (RBMs), deep belief networks (DBNs), auto-encoders, and deep convolutional neural networks (CNNs). CNN approaches have been widely common in computer

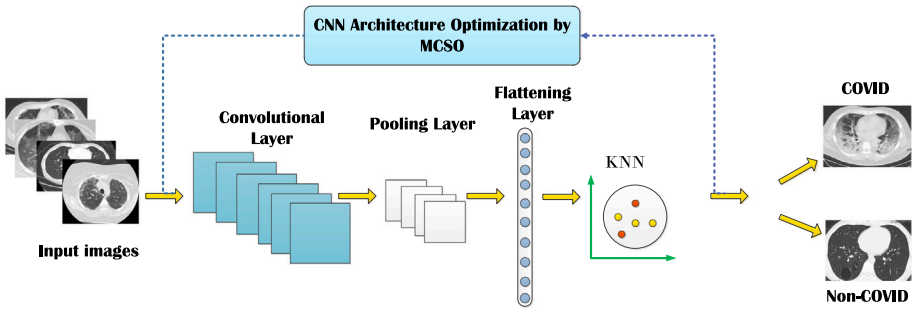


Fig. 1. Overall procedure of the proposed DNE framework.

vision as well as in the field of medical image processing in the past decades (Anwar et al., 2018).

Bio-inspired versions of multi-layered perceptron are named as CNNs. They generally associate visual features from input images effectively. Such deep networks address small image pixel spots named temporal information through using multiple layer neurons and the weights shared within each convolutional level (Mahendran & Vedaldi, 2016). Three architectural concepts incorporate CNNs to verify some level of non-linearity in scale, transition, and disruption (Zheng et al., 2018). The local interconnections between the patterns of the neurons in adjoining CNN layers are chosen to take as a subcategory of units in the previous layer which have amenable spatially adjacent areas to leverage the spacial local correlation in the hidden units of the current layer. Furthermore, any filter in the CNN is repeated across the entire field of vision. Such filters exchange weight and bias vectors for establishing a feature map. The shared weight gradient is the same as the common shared parameter gradients. A feature map is generated when convolution operations are conducted on subsections of the entire image (Tian et al., 2020).

Let X with $R^{A\times B}$ is used to determine the input in the convolutional layer and the input data size is defined as A and B. The output for the convolutional layer is then stated as follows:

$$C_n = f\left(\sum_{c=1}^{N} X_c^{l-1} * W_m^l + B_m^l\right), \tag{1}$$

where C_n represents the mth output of convolutional layer, and n refers to the number of outputs equal to the filter size. The number of channels is represented by N; the convolutional operator is defined as *, and X_c is the cth input data channel for the prior $(l-1)$th layer. For the present layer l, the width and height of the filter are indicated by W and H, respectively. W_m^l is the mth filter weight for the present lth layer, and B_m^l is considered as mth bias. Generally, activation function in CNN structures is represented by f which their most common functions are Sigmoid function, Rectified Linear Unit (ReLU), Leaky ReLU, and Tanh. The ReLU is the widely-used activation function which is defined in the following form:

$$f(x) = \max(0, x). \tag{2}$$

The pooling technique allows in reducing the algorithm complexity with the feature maps, prevents proper over-fitting after the last fully convolutional layer, and significantly reduces training parameters to shorten the training time of the model. The most widely recognized method of pooling is max-pooling, which mainly involves traversing functional maps with flexible steps and evolving optimal values to the traversed areas. After performing the traversing operation, the updated feature maps will eventually be extracted. This is the entire process of the CNNs in general.

3.2. K-nearest neighbor

K-nearest neighbor (KNN) algorithm is a supervised-learning classification approach founded on samples of the closest training set throughout the feature domain which is considered as one of the simplest algorithms in machine learning applications (Kim et al., 2012; Ning et al., 2019). This algorithm is trained by the collection of vectors and labels of the training objects (here images). The unknown data points are simply allocated to their closest neighbors' K labels throughout the classification phase. The primary benefit of KNN algorithm is to work efficiently in binary classes since its decision is centered on a small neighborhood of related objects, and therefore can contribute to acceptable accuracy.

Instead of Softmax classifier as the last layer of the CNN model, we replace it with KNN classifier providing significant improvement in model accuracy and help to minimize the possibility of over-fitting in the classification performance. The objects are typically classified by majority voting on the labels of the closest K neighbors. For $K = 1$, the entity is labeled basically as the nearest object class. If only two classes exist in the dataset, K parameter has to be an odd integer number. Upon transforming each image to a vector of real numbers, we used Euclidean distance feature in KNN as the most widely accepted distance mechanism. The Euclidean distance of KNN algorithm is formulated as following:

$$d(x_i, x_j) = \|x_i - x_j\|_2^2 = \left(\sum_{r=1}^{m} \left(x_{ir} - x_{jr}\right)^2\right)^{1/2}, \tag{3}$$

where x_i and x_j are the $m-$dimensional vectors of the ith and jth samples, and the index r indicates the rth real-valued feature of each sample.

3.3. Modified Competitive Swarm Optimizer (MCSO)

In order to solve large-scale optimization problems, Cheng and Jin (2014) introduced a robust and efficient particle swarm optimization (PSO) variant, known as competitive swarm optimizer (CSO). In CSO, the particles benefit from randomly chosen opponents and not from the strongest global or the individual position. The population of swarms is split randomly into two groups in each iteration and competitions between the particles in any group take place on a pair principle. The winner particle is directly transferred during every competition toward the next iteration, while the losing particle updates its position and velocity with a knowledge of the winning particle based on the following equations:

$$v_l^{t+1} = G_1^t v_l^t + G_2^t\left(x_w^t - x_l^t\right) + \lambda G_3^t\left(\bar{x}^t - x_l^t\right) \tag{4}$$

$$x_l^{t+1} = x_l^t + v_l^{t+1} \tag{5}$$

where t represents the iteration counter, G_1^t, G_2^t, G_3^t represent three $[0, 1]$ range vectors that are generated randomly, while x_w^t and x_l^t are respectively the winner and loser particles. $\bar{x}^t$ represents the mean swarm position in iteration t, and the effect of $\bar{x}^t$ is handled by λ.

In order to strength more the search capabilities of CSO algorithm, we introduce a three-phase modification on this evolutionary algorithm to make an efficient balance between exploration and exploitation

phases as well as the ability to escape from local minima. We name our novel evolutionary swarm intelligence algorithm as modified CSO (MCSO) in which the modifications we have applied over CSO are listed as follows:

-First modification:

Generally, during the beginning phases of the search procedure, the CSO algorithm converges increasingly. Nevertheless, once CSO hits global optima point, the convergence decreases significantly. We implement Cauchy Mutation (CM) evolutionary operator in CSO to set up a procedure that monitors and helps search agent improvements by escaping the local search space through spreading to proper positions. To this end, the following equation is used:

$$f(x) = \frac{1}{\pi}\frac{t}{x^2 + t^2}, \quad t > 0 \ \& - \infty < x < \infty \tag{6}$$

where t shows the scale element. The probability distribution function of the Cauchy component is shown by:

$$F(x) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{x}{t}\right) \tag{7}$$

The mutation operator disrupts the population of the search agents and enables them break of the local minima. The deployment of CM in CSO is defined as following:

$$W_j = \frac{\left(\sum_{i=1}^{P} x_{ij}\right)}{P} \tag{8}$$

where W_j represents the matrix of weights, x_{ij} is the jth position of the ith search agent and population size of the search agents is denoted by P. The Cauchy mutation operator is applied as follows:

$$x'_j = x_j + W_j \cdot A \tag{9}$$

where A is a random vector generated by Cauchy.

-Second modification:

In CSO algorithm, per each iteration, the position of search agents is updated continuously. After finalizing the updating procedure, the search agents exceeding the search space boundaries are considered worthless in order to find the best solutions. Therefore, the search agents must be redirected to the search field, accomplished by the evolutionary boundary constraint handling (EBCH) technique. Once we incorporate this technique into CSP, its exploration and exploitation adaptability is enhanced effectively. Thus, the relevant attributes of the best search agent positions substitute the search agents that contravene the search space boundary conditions. The EBCH updating process is given by the following formula:

$$f(o_i \to x_i) = \begin{cases} \xi \cdot \mathrm{lb}_i + (1-\xi)x_i^b & \text{if } o_i < \mathrm{lb}_i \\ \varrho \cdot \mathrm{ub}_i + (1-\varrho)x_i^b & \text{if } o_i > \mathrm{ub}_i \end{cases} \tag{10}$$

where o_i represents the out-of-bound search agent, ξ and ϱ are random variables in [0,1] interval, ub_i and lb_i denote to the upper and lower bounds of the search space, respectively, and x_i^b is considered as the ith best search agent.

-Third modification:

In order to improve CSO convergence speed, λ parameter plays a key role. In the iterative cycle of CSO, the evolutionary chaotic maps are used to strengthen the ability of utilizing and keeping the core search step harmonized. This concept motivates us to fine tune the λ parameter using a powerful chaotic map named tent map. The formula of tent map is given as following:

$$\lambda = \begin{cases} \frac{\lambda}{0.7} & \lambda < 0.7 \\ \frac{10}{3}(1-\lambda) & \lambda \geq 0.7 \end{cases} \tag{11}$$

In Fig. 2, the detailed framework of the proposed MCSO algorithm is summarized as a flowchart.

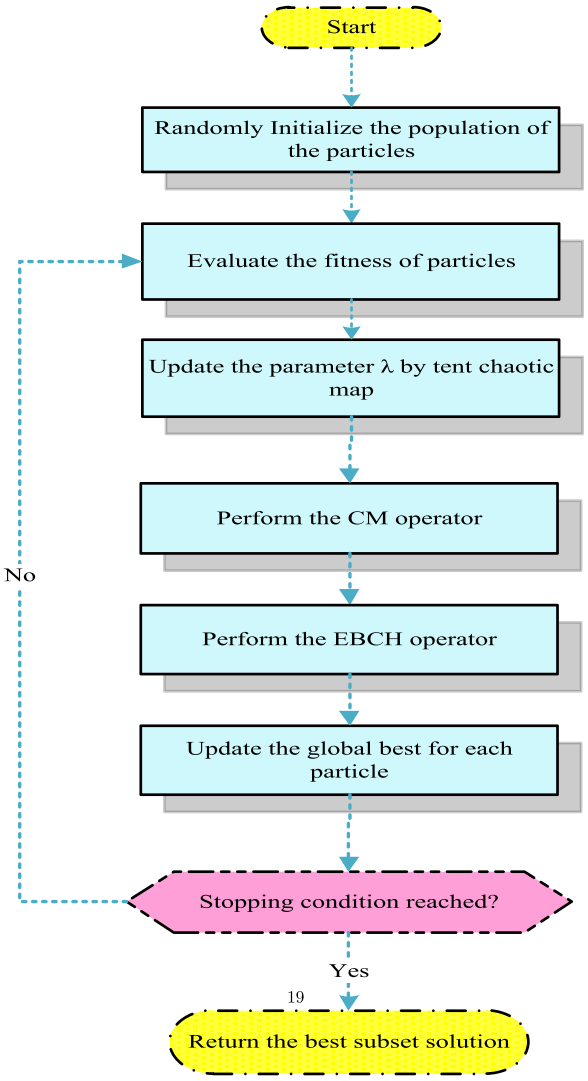


Fig. 2. The proposed MCSO evolutionary algorithm.

3.4. Proposed MCSO-CNN

Within this section, we introduce our novel proposed deep neuroevolution approach named as MCSO-CNN. In order to achieve optimal CNN hyperparameters that increase the accuracy of X-ray image classification, the proposed method is optimized using the MSCO algorithm. Two essential challenges should be addressed by each evolutionary optimization technique, namely, solution representation and fitness function calculation. On the other hand, eleven key hyperparameters of CNN model are highlighted in the proposed classification model to be optimized by the MCSO including convolution filter size, number of filters, number of convolutional layers, activation function type, dropout rate, maxpooling size, learning rate, momentum rate, optimizer type, number of epochs, and batch size. Therefore, any solution in MCSO can be represented as an eleven dimensional vector that corresponds each to one of the eleven hyperparameters of CNN model. Learning rate, momentum rate, and dropout rate are the continuous value hyperparameters that MCSO can achieve their optimum values. In addition, number of convolutional layers, number of filters, kernel size, batch size, maxpooling size, and number of epochs are other discrete hyperparameters. Optimizer type and activation function type hyperparameters are the categorial ones that are utilized in the optimization procedure by MCSO. While MCSO actively searches the space

for solutions, the values obtained optimally for the hyperparameters should be converted all into their corresponding discrete values. Thus, we design a cost-effective strategy to convert each real value into an integer value. In this regard, the continuous values in each hyperparameter are passed to a discrete search space as $D = [H_1, H_2, \dots, H_n]$. To formulate the discretization model, the following equations are taken into consideration:

$$\zeta = 1 + n \times C \tag{12}$$

$$\psi = min(\lfloor \zeta \rfloor, n) \tag{13}$$

where C represent a continues value in the $[0, 1]$ interval for exploration purpose in the continuous search space area, ζ is an operator responsible to map from C to $[1, n + 1]$, and ψ is another mapping operator from ζ to $[1, 2, 3, \dots, n]$. Each integer value belonging to the solution's continuous dimension can thereby be computed using the following formula:

$$X_{ij} = H_{\psi} \tag{14}$$

Using Eqs. (4) and (5), the proposed MCSO-CNN method starts randomly to initialize the population of n solutions. X_{ij}, where $i = 1..n$ and $j = 1..11$ is a 11-dimensional vector standing for the ith solution encoding the eleven hyperparameters of CNN model. A continuous update of current solutions after generation of the first population can provide new solutions. We use CM strategy to help the CSO in order to escape from local optima by Eq. (9). For improving the convergence of CSO, we incorporate the EBCH technique into the CSO using Eq. (10). Then, we tune the parameter λ with the tent chaotic map using Eq. (11) for providing a balance between exploration and exploitation steps. The whole process is repeated until the final threshold is met and the best solution is found as expected. The best CNN hyperparameter values can be adopted in this solution. We also require to specify a fitness function to find out the success of all the solutions that have been obtained by the MCSO-CNN model. To facilitate this, the input data from the COVID-19 dataset is divided into two distinct training and test sets. The training set optimizes CNN hyperparameters, while the test set evaluates the performance of the final COVID-19 detection model.

It should be noted that the CNN model is equipped with the hyperparameter values from any solution in MCSO. Therefore, the fitness function for the efficiency of the optimized CNN model for detection of COVID-19 can be interpreted. The images in the training set can be included in the training period as the inputs of the CNN model. The fitness value of solutions during optimization procedure is the accuracy of the CNN classifier in classification of input images. The accuracy factor denotes to the number of images classified correctly over all of the total input images in the dataset. Thus, the accuracy value can be formulated by using the following formula:

$$Accuracy = \frac{\#\text{ number of correctly classified images}}{\#\text{ total number of input images}} \tag{15}$$

Precisely, a solution with higher accuracy has a higher fitness value and likewise. The objective of the proposed approach is thereby to achieve a solution that contains the optimum values of CNN hyperparameters of the highest accuracy (i.e. highest fitness value). The optimized CNN model is then used to classify images in the test set. The overall steps of the MCSO-CNN method are outlined by Algorithm 1.

Algorithm 1 Pseudo-code of the proposed DNE COVID-19 detection framework (MCSO-CNN)

Input: **D**, labeled X-ray images dataset;
Output: Classified X-ray images (Normal or infected by COVID-19);
Begin algorithm:
Set the max number of iterations, Max_{it};
Split **D** into training set (Tr) and test set (Te);
Generate the initial population of P particles, $S = \{X_i, i = 1..P\}$;
t=1;
while ($t < Max_{ite}$) **do**
  Design P CNN models based on S;
  Calculate the fitness (accuracy) F_i of each particle X_i with Eq. (17) based on Tr;
  $gbest \leftarrow X_i$; X_i of the best fitness;
  for i=1 to $\lfloor P \rfloor$ (half of population) **do**
    Select randomly two particles, X_k and X_m;
    if $F_k > F_m$ **then**
      $X_w = X_k$; #X_w is the winner particle
      $X_l = X_m$; #X_l is the loser particle
    else
      $X_w = X_m$, $X_l = X_k$;
    end if
    Add X_w into new population;
    Remove X_k and X_m from the population;
    $i = i + 1$;
    for i=1 to $\lfloor P \rfloor$ (half of population) **do**
      Update velocity of loser as shown in Eq. (4);
      Update the parameter λ by tent chaotic map using Eq. (11);
      Update position of loser using Eq. (5);
      Apply CM operator using Eq. (9);
      Apply the EBCH operator using Eq. (10);
      Calculate the fitness of new loser, F_i;
      Move new loser into new population;
      Update $gbest$ if there is better solution;
      $i = i + 1$;
    end for
    Pass new population to next iteration;
  end for
  t=t+1;
end while
Fit the trained CNN model on the basis of the best particle $gbest$ hyperparameters and classify images based on KNN algorithm;
In the Te test set, classify the images with the best CNN model acquired;
Return the classified images as the optimal output;
End algorithm

4. Experimental setting, results and evaluation

4.1. Dataset description

The proposed DNE framework is evaluated and tested using X-ray images provided by Alqudah and Qazan (2020) which can be downloaded for free of use from Mendeley open source repository.[1] The dataset consists of 1824 images including 912 images for normal cases and the rest for COVID-19 affected cases. Fig. 3 shows three samples of normal cases and also three samples of affected cases. Since CNN requires input images to be the same size, we scaled all the images to 224 × 224 pixels.

4.2. Parameter setting

The proposed DNE framework uses eleven parameters that should be learned automatically in a deep evolutionary scheme. With respect to parameterization of the MCSO evolutionary model, we set the maximum number of iterations to 20 and population size to 30. For the sake of fairness, all the algorithms used in the comparison phase, including the one proposed, are executed 10 times, then the average accuracy measurement is reported. Table 1 defines all hyperparameters and their categorical or integer coding values in the proposed DNE algorithm. In the experiments, we compare the proposed method with other models based on three different scenarios. In the first scenario, the proposed method is compared with other evolutionary algorithms used to optimize the hyperparameters of CNN model. To have a fair comparison, we set the maximum number of iterations and population size for all the compared algorithms to exactly the values used for the proposed method. Other specific parameters related to different evolutionary algorithms are set according to the optimal values obtained through a greedy search process or by their suggested values in their related literature (Al-Betar et al., 2021; Yue & Zhang, 2021; Zhou et al., 2021). In the second scenario, the proposed method is compared with other supervised learning classifiers in which the values of their parameters are set based on the best values reported in their corresponding papers,

[1] https://data.mendeley.com/datasets/2fxz4px6d8/4.

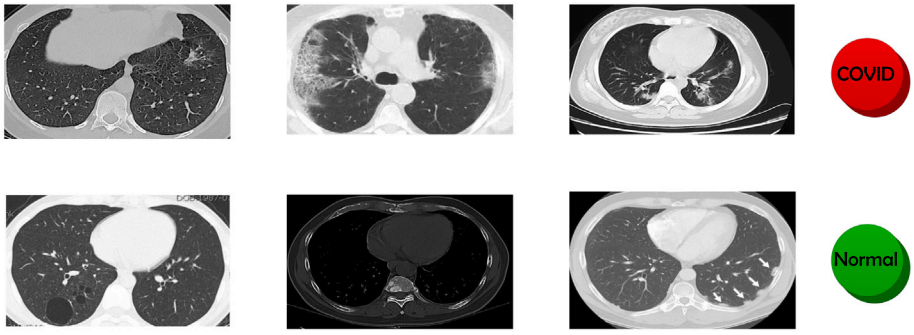


Fig. 3. Samples from the X-ray image dataset.

Table 1
Involved hyperparameters in the evolutionary algorithm and their corresponding values.

Symbol	Description	Values
K_s	Kernel size	[1,25]
N_f	# Of filters	[1500]
Opt	Optimizer type	[Adagrad, Adam, SGD, Adamax]
N_e	# Of epochs	[1600]
B_s	Batch size	[10,20, ….,600]
N_c	# Convolution layers	[1,2, …,15]
MP_s	Maxpooling size	[1,25]
D_r	Dropout rate	[0.2, 0.25, …,0.65]
Act	Activation function	[Sigmoid, ReLU, Hard Sigmoid, Tanh]
L_r	Learning rate	[0.001, 0.006, …, 0.1]
M_r	Momentum rate	[0.05, 0.1, …,0.95]

and also a greedy search process is performed to obtain the optimal values for these models. Finally, in the third scenario, the state-of-the-art image classification models are considered to compare with the proposed method. In this case, the values of parameters for the compared models are initialized based on the best values reported in their corresponding papers. For implementing the proposed method and other algorithms used in the experiments, we used TensorFlow which is a high-level platform developed in the Python programming language for machine learning techniques. Also, all algorithms are performed on a machine with one GeForce GTX 1080 Ti GPU and one 16 GB RAM.

4.3. Performance evaluation metrics

We employed standard performance metrics including accuracy, precision, recall, area under the curve (AUC), and F1-score to evaluate the prediction efficiency of the compared models. In these metrics, the number of positive samples correctly predicted is denoted by true positive (TP) while the number of positive samples wrongly predicted is represented by false negative (FN). On the other hand, the true negative (TN) is the accurately predicted number of negative samples, whereas a false positive (FP) is the wrong predicted number of negative samples. The equations related to the evaluation metrics are defined as follows:

- Precision is expressed as the ratio of samples correctly classified in truly positive patients which is obtained by:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{16}$$

- Recall shows the rate of total positive samples correctly classified by the algorithm which can be calculated as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{17}$$

- F1-score ($F1$) is the harmonic average of two other metrics precision (P) and recall (R) defined as follows:

$$F1 = \frac{2 \times P \times R}{P + R} \tag{18}$$

- Accuracy is defined as the rate of the corrected classified cases:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{19}$$

- AUC is described as the area under the ROC (receiver operating characteristic) curve showing the performance of classification/detection model. It is defined as follows:

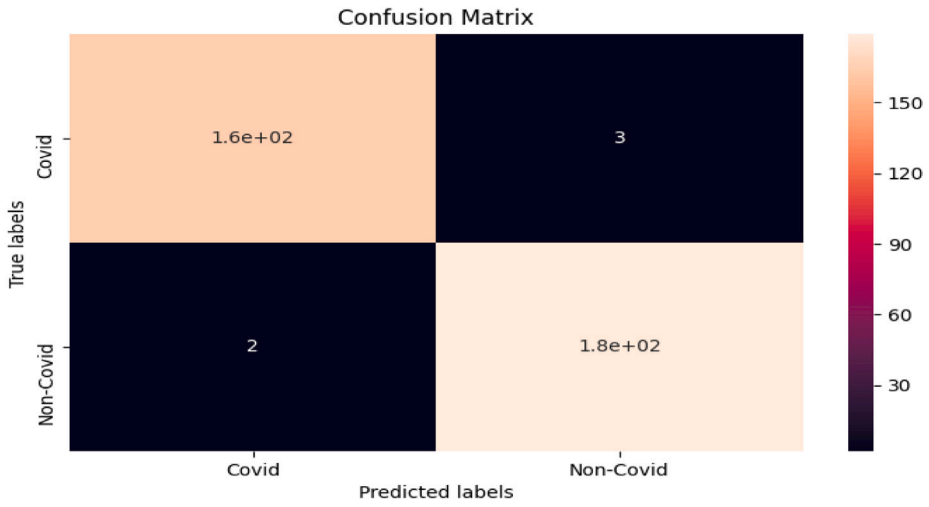
$$\text{AUC} = \int_0^1 \frac{\text{TP}}{P} \, \text{d} \, \frac{\text{FP}}{N} = \frac{1}{P.N} \int_0^1 \text{TP} \, \text{d} \, \text{FP} \tag{20}$$

4.4. Results and discussion

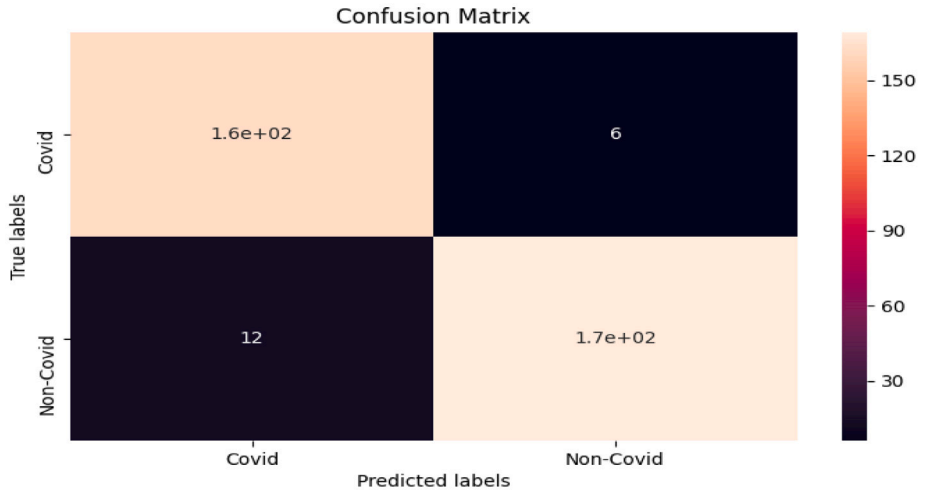
In this section, we perform an extensive performance comparison with different evolutionary algorithms, classifiers, and state-of-the-art deep learning architectural models to show the competency and efficiency of our proposed MCSO-CNN model.

4.4.1. Performance comparison based on different evolutionary algorithms

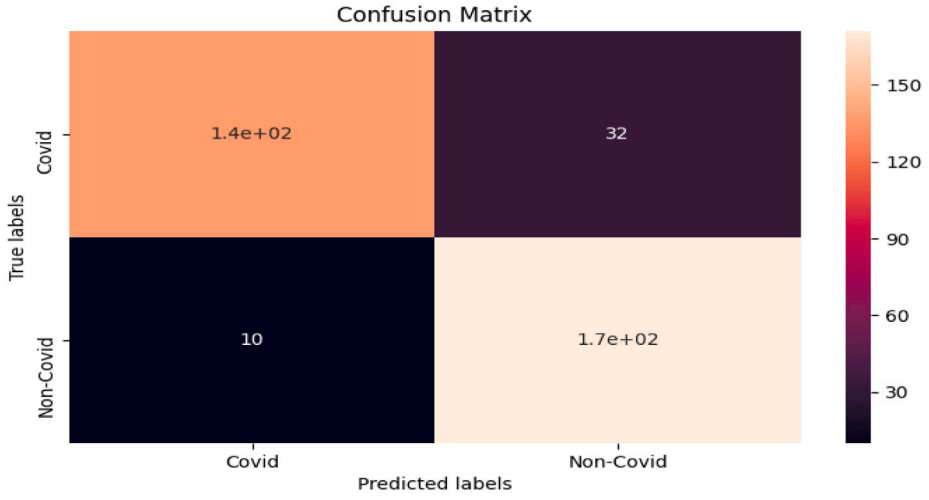
In order to demonstrate the impact of the proposed MSCO approach on the optimization of CNN, we compare it to different evolutionary algorithms including GA (genetic algorithm) (Jalali, Kebria et al., 2019), DE (differential evolution) (Deng et al., 2020), PSO (particle swarm optimization) (Tan et al., 2019), MFO (moth-flame optimization) (Li et al., 2016), WOA (whale optimization algorithm) (Mirjalili & Lewis, 2016), SSA (salp swarm algorithm) (Yang et al., 2019), HHO (Harris hawks optimization) (Heidari et al., 2019), GOA (Grasshopper optimization algorithm) (Saremi et al., 2017), and the original CSO. Table 2 reports the results obtained by the different EAs based on the evaluation metrics mentioned above. These results demonstrate that the proposed method outperforms all other algorithms on all the performance metrics. It can be shown that, based on AUC, the proposed method can improve the classification accuracy with an average of approximately 3 percent compared to the second best algorithm (i.e., HHO). Moreover, MSCO achieved the highest number of true classifications reflected in the recall and precision measures. One of the effective ways to evaluate the performance of classification models is the comparison of their confusion matrix. The confusion matrix divides the input samples into four subsets according to true/false predictions. Fig. 4 shows the confusion matrix of different EA models. As we can see from this figure, the proposed method presents a better confusion matrix than other models. This is due to the fact that the number of true classifications obtained by the proposed method is more than other classifiers. The box plots of different evolutionary algorithms based on the accuracy metric are shown in Fig. 5. These results reveal that the proposed method is more accurate than other models as the average accuracy of the proposed method is higher than other evolutionary algorithms.



(a) MCSO (Proposed)



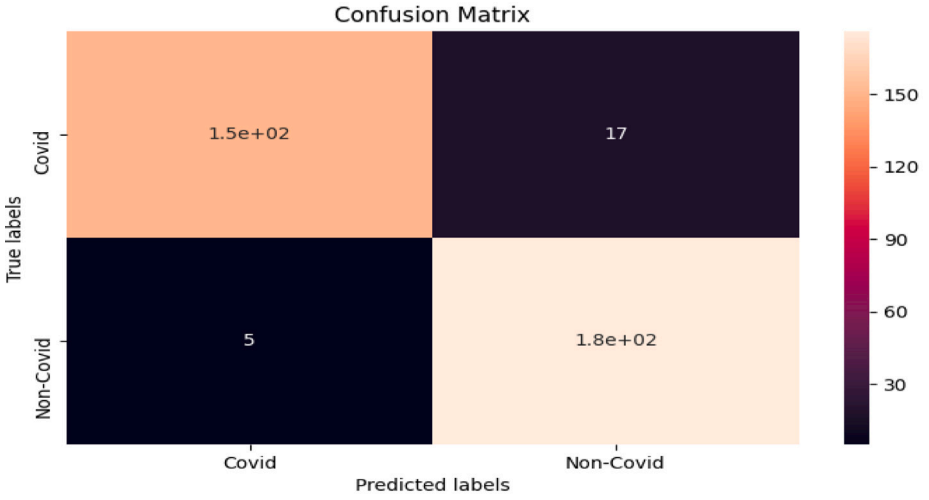
(b) GA



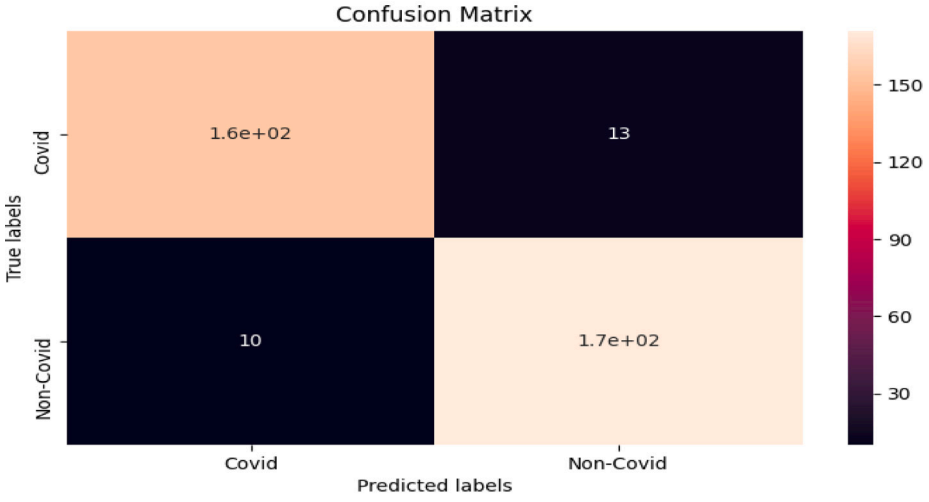
(c) DE

Fig. 4. Confusion matrices of different evolutionary algorithms.

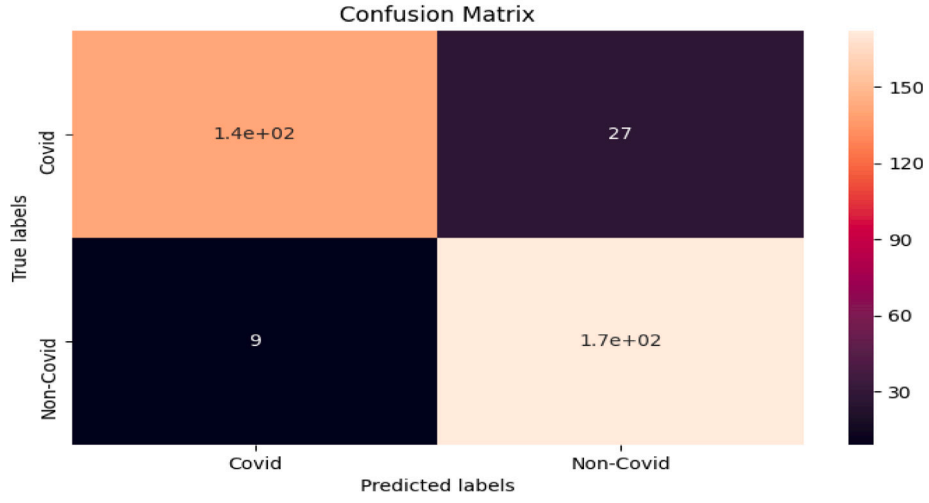
Moreover, this figure shows that the proposed MCSO model does not have an horizontal line on the top compared to other models. Therefore, it can be concluded that the proposed model has a better variance value than the other models. The convergence speed of the EA models is compared and the results are shown in Fig. 6. It can be seen from this figure that the convergence speed of the proposed method is better than other EA models. Moreover, the proposed method obtains the best accuracy among other EA models in different iterations.



(d) PSO



(e) MFO

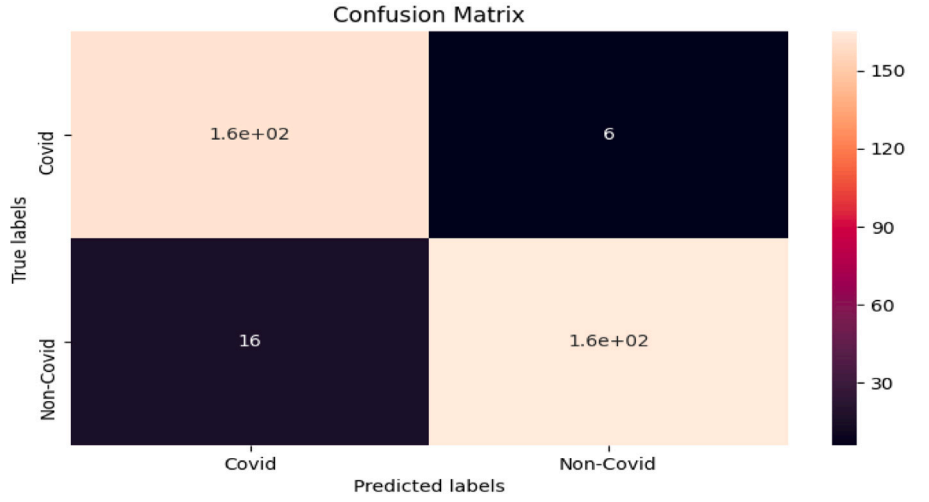


(f) WOA

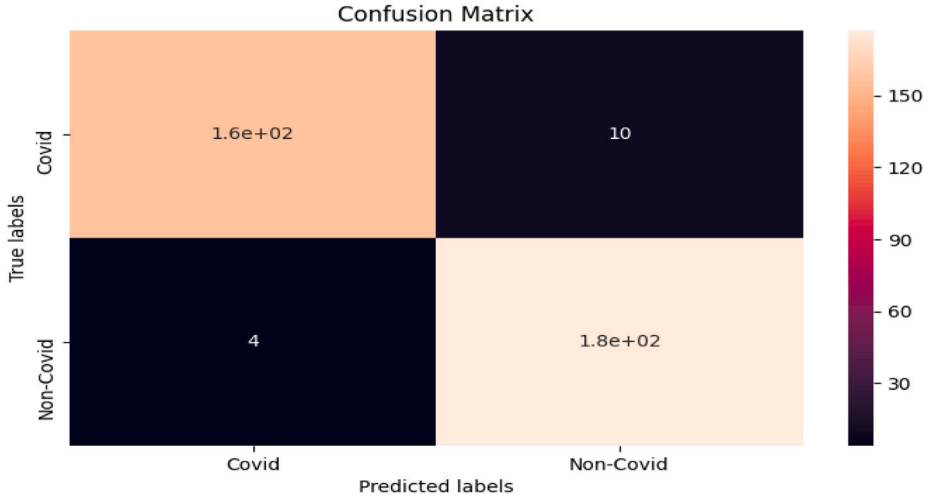
Fig. 4. (*continued*).

4.4.2. Performance comparison based on different supervised learning classifiers

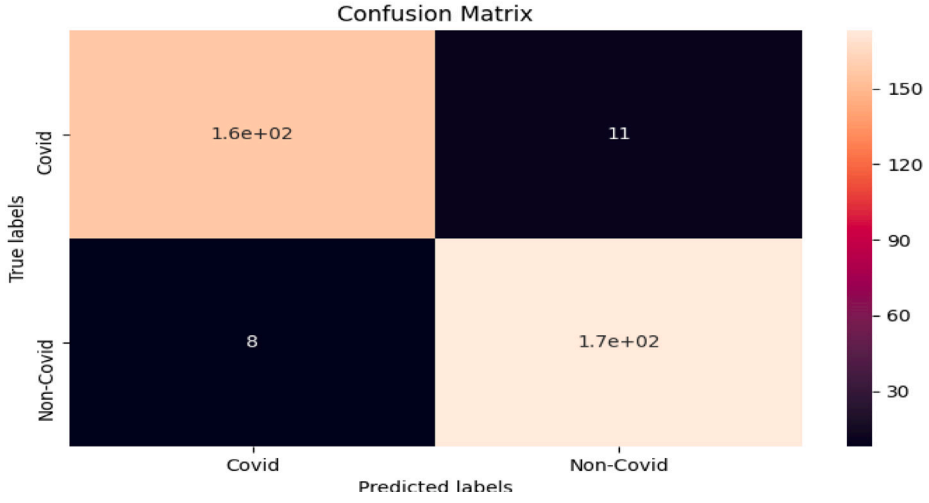
In this section, we consider the effect of different classification algorithms such as Decision Tree, Random Forest, LightGBM, AdaBoost, SoftMax, and Bagging on the proposed framework and compare them with the KNN algorithm. In the proposed method, we use the KNN algorithm to classify the input images into two classes including COVID-19/Non-COVID-19 cases. The results of experiments based on different



(g) SSA



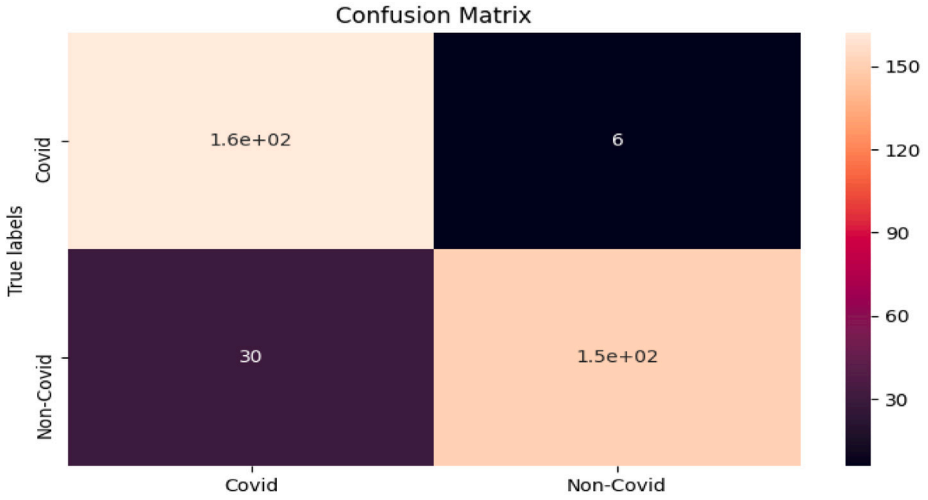
(h) HHO



(i) GOA

Fig. 4. (*continued*).

classification algorithms are reported in Table 3. These results reveal the superiority of the proposed method in comparison to other classification algorithms. The accuracy of the proposed method is 0.985673 while the Bagging algorithm obtains the accuracy of 0.919771 as the second best performer. Therefore, the proposed method can improve the classification accuracy by up to 7 percent. Moreover, the proposed method outperforms other models in terms of other metrics including precision, recall, F-measure, and AUC. Fig. 7 shows the confusion matrix of different classification algorithms. As we can see from this figure, the confusion matrix of the proposed method presents better result than



(j) CSO

Fig. 4. (*continued*).

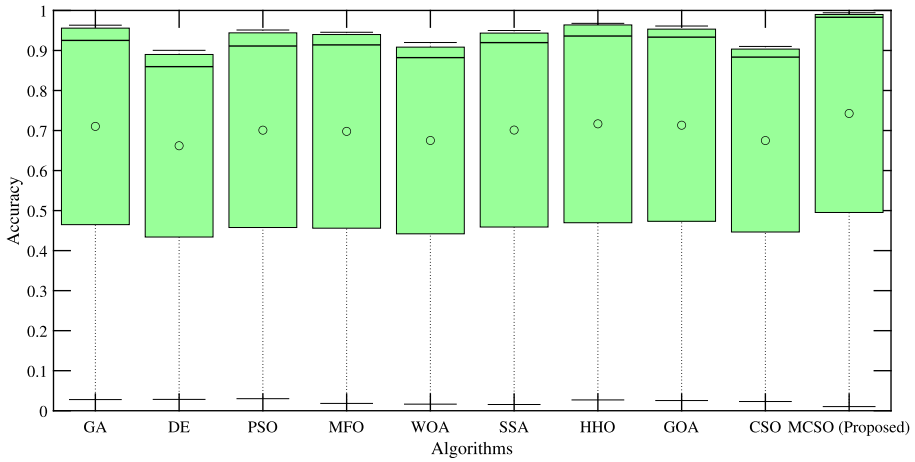


Fig. 5. Box plots of different evolutionary algorithms for accuracy metric as the fitness function.

Table 2
The results of experiments based on different evolutionary algorithms.

Metric		GA	DE	PSO	MFO	WOA	SSA	HHO	GOA	CSO	MCSO (Proposed)
Accuracy	AVG	0.948424	0.879656	0.936963	0.934118	0.896848	0.937718	0.959885	0.945559	0.897091	**0.985673**
	STD	0.027857	0.028322	0.030082	0.018272	0.016444	0.015544	0.026981	0.025433	0.022989	**0.010329**
	Best	0.963189	0.900377	0.951132	0.945489	0.919887	0.949899	0.967782	0.961089	0.910082	**0.994258**
	Worst	0.901882	0.839555	0.885334	0.893778	0.867442	0.902313	0.912287	0.921055	0.869988	**0.980227**
Precision	AVG	0.931034	0.931507	0.967949	0.939394	0.943184	0.910112	0.975309	0.951515	0.84375	**0.988024**
	STD	0.017291	0.016891	0.011838	0.013949	0.015623	0.012234	0.018996	0.026744	0.057239	**0.011017**
	Best	0.959888	0.953998	0.973553	0.945772	0.958669	0.922043	0.981544	0.968892	0.890665	**0.990881**
	Worst	0.925343	0.936886	0.946767	0.910093	0.924883	0.887877	0.959882	0.927751	0.800487	**0.982626**
Recall	AVG	0.964131	0.809524	0.898823	0.922619	0.839286	0.965233	0.940476	0.934524	0.968893	**0.982143**
	STD	0.042911	0.033578	0.063431	0.051967	0.049677	0.056444	0.019843	0.020729	0.019759	**0.019556**
	Best	0.986567	0.788992	0.945539	0.958882	0.886321	0.976621	0.948881	0.952881	0.978992	**0.989945**
	Worst	0.910533	0.846653	0.812322	0.865531	0.786928	0.870032	0.925521	0.911899	0.947721	**0.972774**
F-measure	AVG	0.947368	0.866242	0.932099	0.930931	0.886792	0.936416	0.957576	0.942943	0.893378	**0.985075**
	STD	0.035201	0.031878	0.029889	0.022213	0.027344	0.020992	0.024434	0.027666	0.028891	**0.018847**
	Best	0.969189	0.906675	0.957877	0.960219	0.907682	0.948977	0.970221	0.968891	0.919884	**0.993927**
	Worst	0.903888	0.824844	0.896763	0.911661	0.865433	0.918896	0.933833	0.928823	0.869066	**0.978863**
AUC	AVG	0.948994	0.877138	0.935593	0.933685	0.894781	0.937944	0.959188	0.945162	0.89927	**0.985547**
	STD	0.032367	0.026767	0.031198	0.019421	0.022322	0.017669	0.019006	0.021424	0.025589	**0.016383**
	Best	0.972081	0.903278	0.963122	0.945488	0.917066	0.945543	0.969928	0.965881	0.918553	**0.990394**
	Worst	0.888977	0.837655	0.890789	0.910886	0.865518	0.913908	0.940133	0.919998	0.877001	**0.978963**

Table 3
The results of experiments based on different supervised learning classifiers.

Metric		Decision Tree	Random Forest	LightGBM	AdaBoost	SoftMax	Bagging	KNN (Proposed)
Accuracy	AVG	0.914844	0.905444	0.899713	0.902579	0.914048	0.919771	**0.985673**
	STD	**0.005888**	0.006131	0.006254	0.007677	0.007147	0.006222	0.010329
	Best	0.919422	0.912554	0.905334	0.913183	0.920839	0.931064	**0.994258**
	Worst	0.897532	0.890411	0.883633	0.897712	0.901899	0.911441	**0.980227**
Precision	AVG	0.901163	0.929936	0.907975	0.852632	0.965577	0.897727	**0.988024**
	STD	0.012277	0.011899	0.014884	0.049899	0.037877	0.04402	**0.011017**
	Best	0.927478	0.939988	0.917888	0.913233	0.971322	0.956566	**0.990881**
	Worst	0.903536	0.921443	0.898666	0.811171	0.908788	0.853887	**0.982626**
Recall	AVG	0.922619	0.869048	0.880952	0.964008	0.857143	0.940476	**0.982143**
	STD	0.022968	0.024344	0.023534	**0.011433**	0.045936	0.015366	0.019556
	Best	0.944542	0.879231	0.911488	0.973134	0.918334	0.959189	**0.989945**
	Worst	0.890181	0.831066	0.867578	0.938788	0.819878	0.920066	**0.972774**
F-measure	AVG	0.911765	0.898462	0.89426	0.905028	0.905668	0.918605	**0.985075**
	STD	0.006989	0.007461	0.008644	0.006079	0.006256	**0.005587**	0.018847
	Best	0.933543	0.916466	0.909689	0.924157	0.921177	0.935525	**0.993927**
	Worst	0.909978	0.896366	0.880343	0.900758	0.901333	0.917578	**0.978863**
AUC	AVG	0.914348	0.904137	0.89904	0.904795	0.911997	0.920514	**0.985547**
	STD	0.005979	0.006365	0.006887	0.006989	0.006425	**0.005479**	0.016383
	Best	0.922432	0.916576	0.906966	0.912089	0.927816	0.935455	**0.990394**
	Worst	0.903879	0.899932	0.883588	0.891777	0.903133	0.912189	**0.978963**

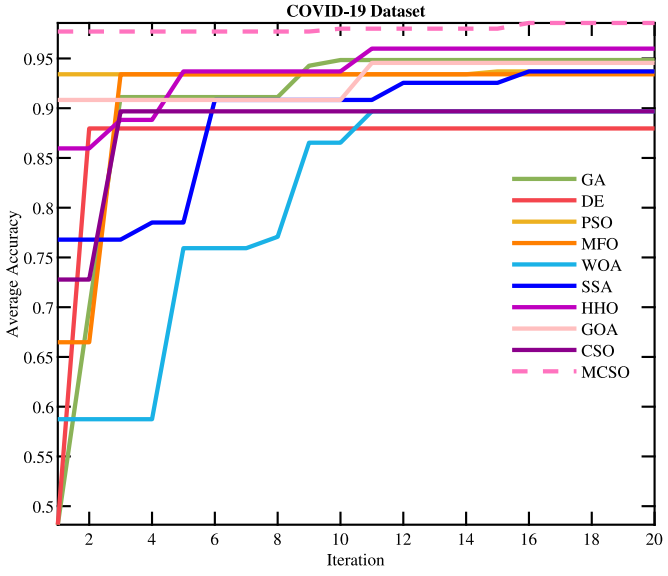


Fig. 6. Convergence curves of different evolutionary algorithms and the proposed MCSO algorithm.

the other models. It is worth noting that a confusion matrix with more true predicted samples can be considered as a better result. Therefore, it can be concluded that the proposed method based on KNN algorithm obtains better performance than the other classification algorithms. Fig. 8 shows the box plots of different classification algorithms based on the accuracy metric. It can be seen from this figure that the proposed method is the best performer among all classification algorithms as it obtains the best average accuracy. Therefore, it can be concluded that the KNN algorithm can provide better classification strategy in comparison to other models to classify the input COVID/Non-COVID images.

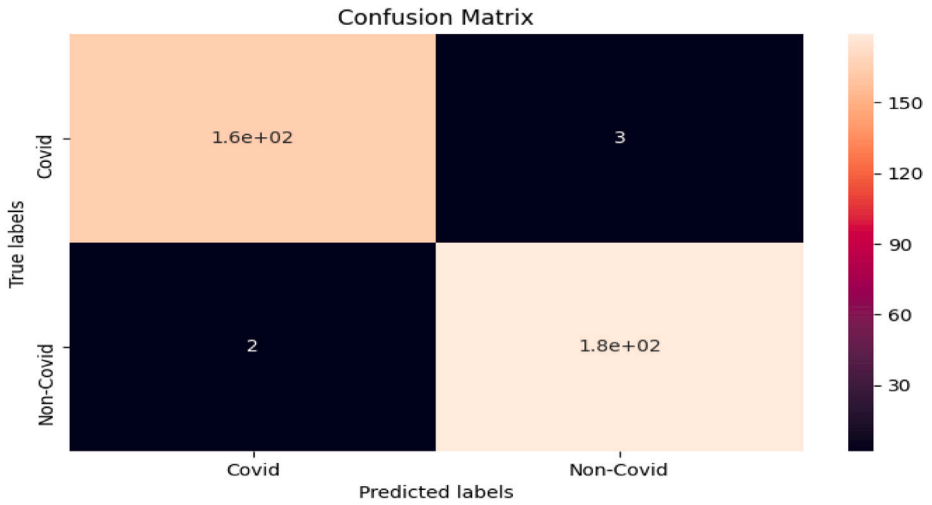
4.4.3. Performance comparison with the-state-of-the-art image classification models

The experiments in this section are carried out to compare the proposed method with the-state-of-the-art approaches related to the image classification. To this end, several pre-trained deep learning image classification approaches including DenseNet121, MobileNet, InceptionV3, XCeption, ResNet50, VGGNet19, and two well-known approaches in COVID-19 detection named DeCoVNet Brunese et al. (2020) and Zheng et al. (2020) are considered in the experiments. Experimental results for different image classification methods based on the used evaluation metrics are reported in Table 4. These results demonstrate that the proposed method can achieve better performance than the other image classification methods in terms of all evaluation metrics. For instance, the average accuracy of the proposed method is 0.985673 while the second best result is obtained by the XCeption method as 0.977077. It is obvious that the classification accuracy is a vital issue in real applications especially for the detection of coronavirus disease. Therefore, it can be concluded that the proposed method can open a promising research direction in the field of the coronavirus disease detection. The experiments are repeated and their results are shown in Fig. 9 as the confusion matrices of different image classification methods. As we can see from these results, the confusion matrix of the proposed method contains more true classified cases than the other models. Therefore, the proposed method significantly outperforms other image classification methods by providing a better model to detect COVID/Non-COVID cases. Moreover, Fig. 10 shows the box plots of different image classification methods based on the accuracy metric. As illustrated in this figure, the accuracy of the proposed method is higher than other compared methods. Thus, it can be inferred from these results that the ability of the proposed method in detecting COVID/Non-COVID cases is much better than other state-of-the-art image classification approaches.

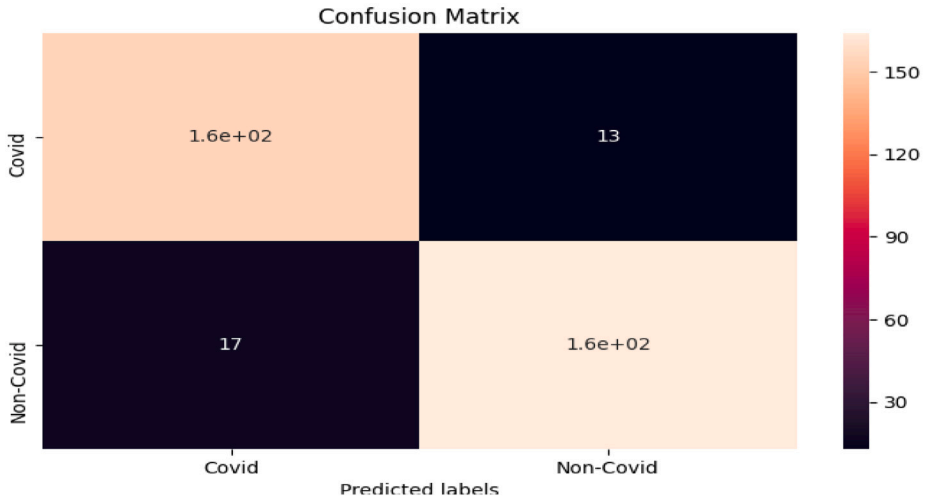
Fig. 11 shows the Violin plots representing the distributions of the hyperparameters evolved in the CNN architectures. It can be revealed that the range of evolved CNN hyperparameters values that has been selected by the MCSO approach incorporates to the minimum values of the hyperparameters range. This is valuable and important since these selected values show the higher capability of MCSO for having the best performance and lowest computational costs during the evolutionary process.

4.4.4. Statistical analysis

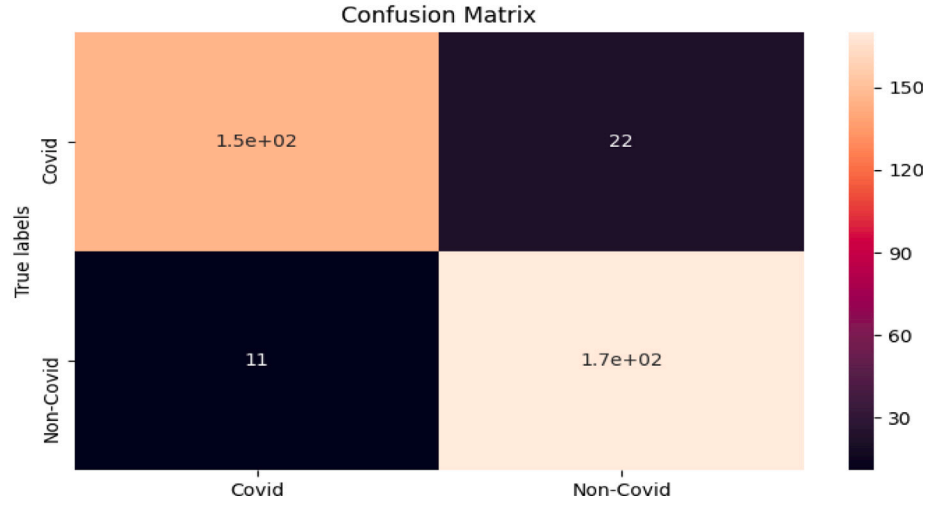
To infer the validity of the reported results, a ranking statistical test called the Friedman ranking test is employed to assess and analyze the performance of the proposed model and all benchmark algorithms. As can be seen in Table 5, our proposed deep MCSO-CNN model statistically achieves the first rank and is recognized as the most efficient algorithm among 23 benchmark models, followed by XCeption,



(a) KNN (Proposed)



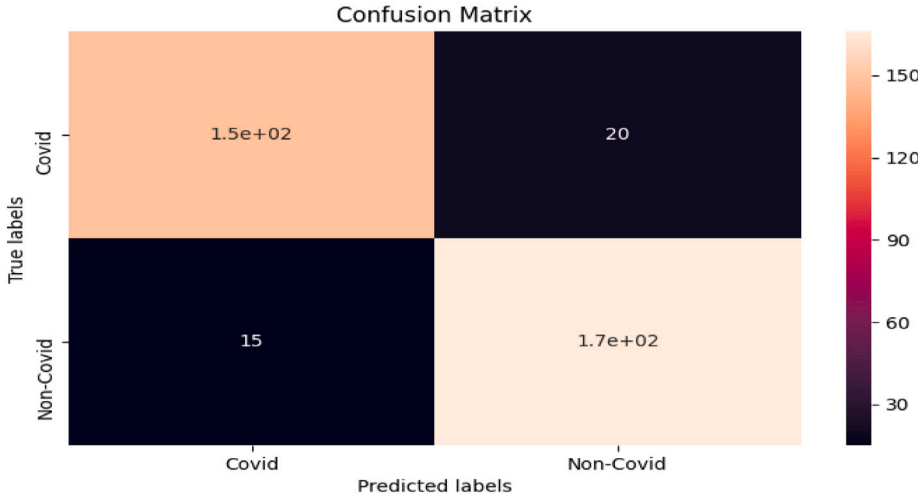
(b) Decision Tree



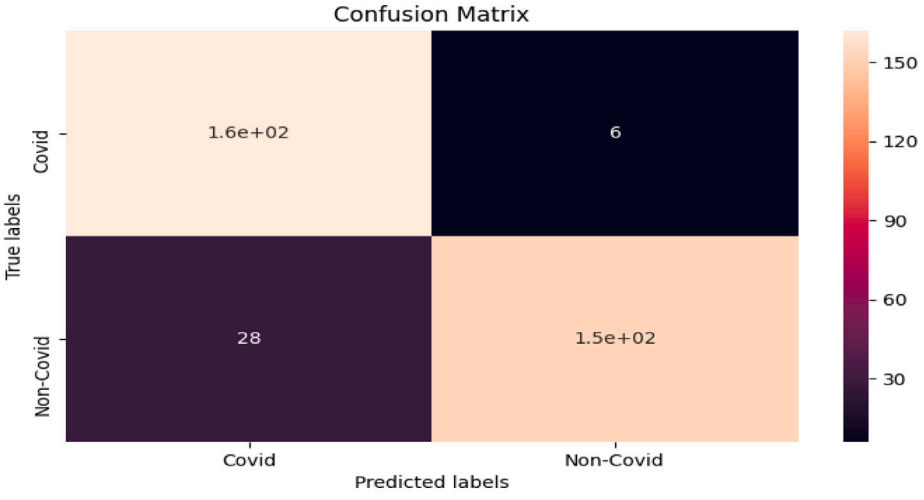
(c) Random Forest

Fig. 7. Confusion matrices of different classifiers.

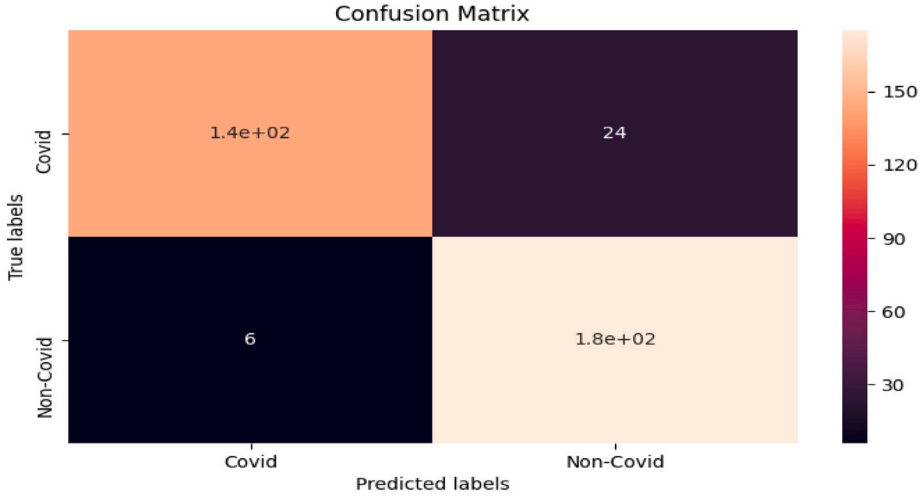
ResNet50, HHO, GA, and GOA, as the top five comparative models. The second (i.e., XCeption) and third (i.e., ResNet50) rank algorithms indicate the effectiveness of the pre-trained deep state-of-the-art algorithms in classifying the X-ray images for COVID-19 detection. We should also acknowledge that the proposed framework hybridized with other powerful evolutionary algorithms including HHO, GA, and GOA have achieved the optimal performance over the remaining deep learning benchmark algorithms.



(d) LightGBM



(e) AdaBoost

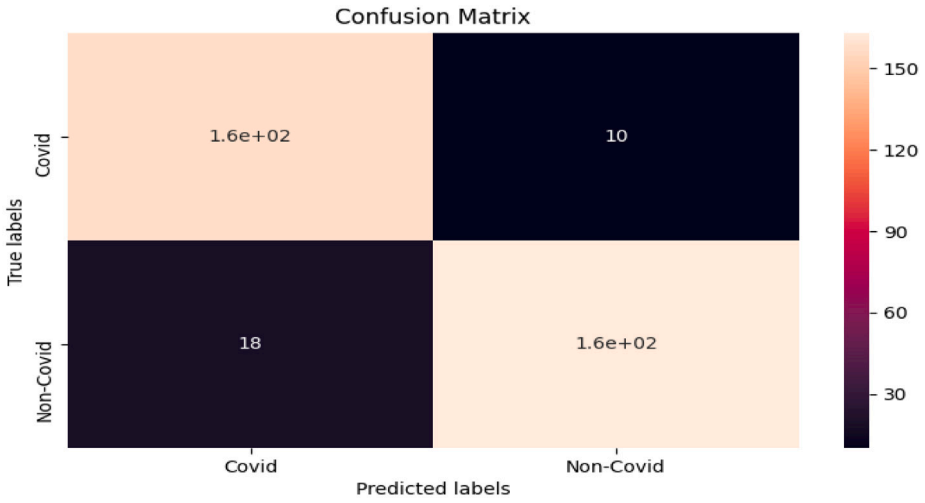


(f) SoftMax

Fig. 7. (*continued*).

4.5. Run-time analysis

In this section, we analyze the run-time of our proposed algorithm and other benchmarking methods for COVID-19 diagnosis. The time consuming results for all algorithms in terms of optimization, training, and testing times are tabulated in Table 6. As can be seen from this table, in terms of optimization time consumption, our proposed algorithm has the shortest time among other methods that have used the optimization method. Furthermore, among all methods, including those that have used optimization and non-optimization methods, our



(g) Bagging

Fig. 7. (*continued*).

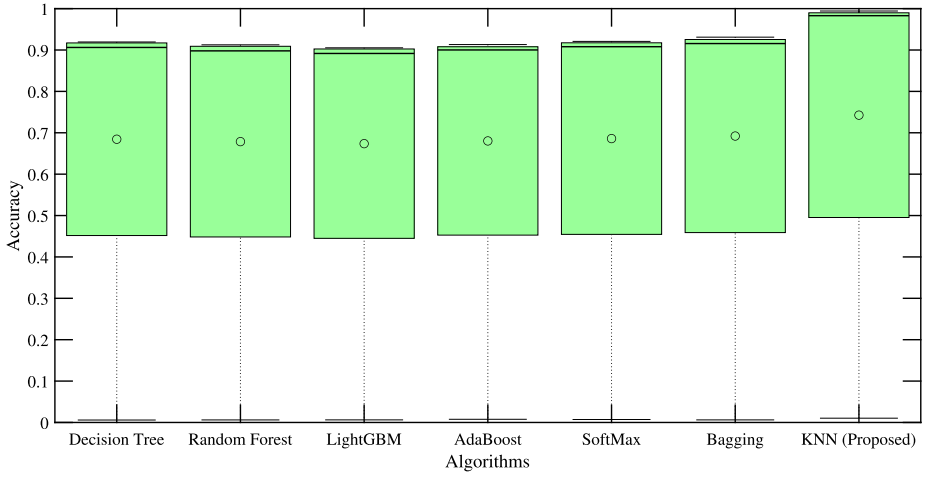
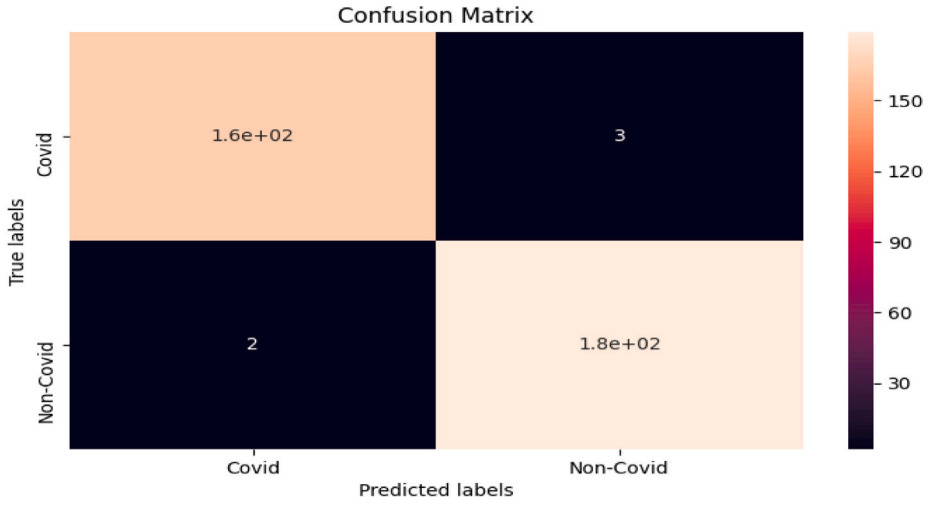


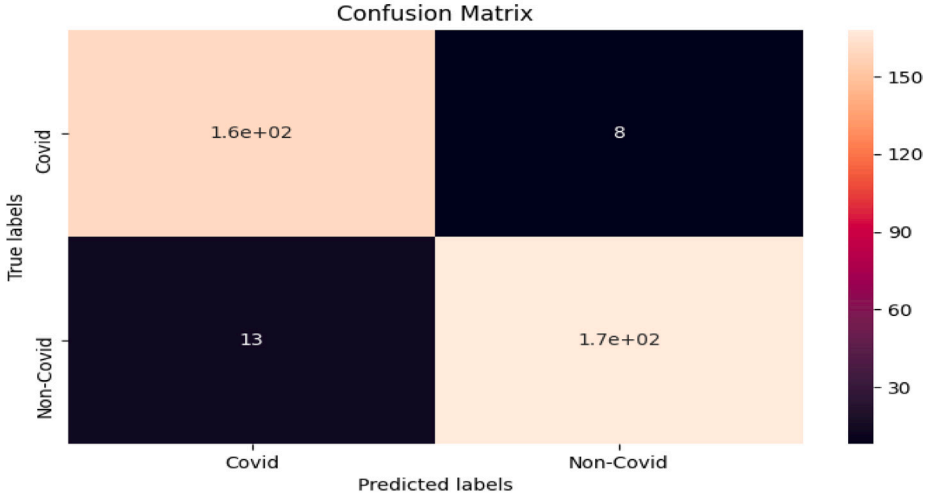
Fig. 8. Box plots of different classification algorithms for accuracy metric as the fitness function.

Table 4
Performance comparison of the proposed method with different state-of-the-art image classification approaches.

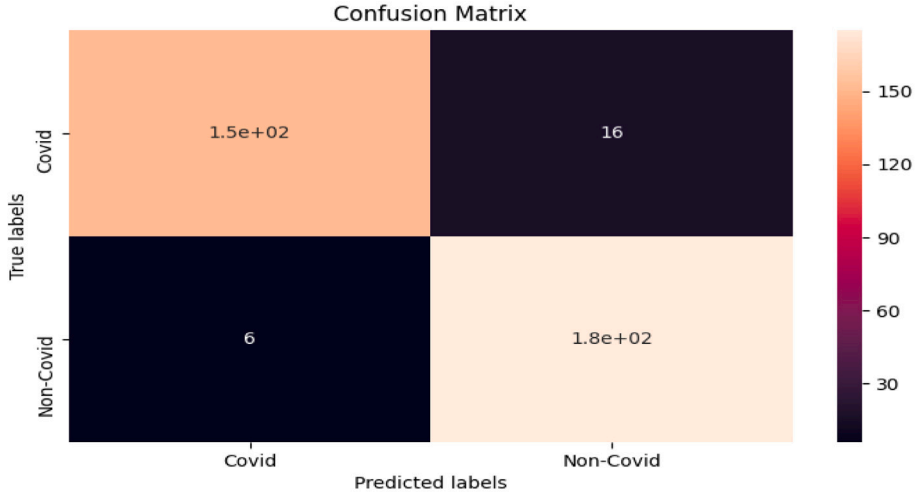
Metric		DenseNet121	MobileNet	InceptionV3	XCeption	ResNet50	VGGNet19	DeCoVNet	Brunese et al.	MCSO-CNN
Accuracy	AVG	0.939828	0.937745	0.934097	0.977077	0.974212	0.902666	0.925501	0.942693	**0.985673**
	STD	0.002357	**0.002002**	0.002155	0.011622	0.012113	0.034463	0.025306	0.016311	0.010329
	Best	0.947612	0.942991	0.940911	0.982833	0.981202	0.948821	0.945599	0.957781	**0.994258**
	Worst	0.929881	0.922881	0.916833	0.961198	0.957088	0.870924	0.899006	0.938876	**0.980227**
Precision	AVG	0.924855	0.962025	0.950311	0.987805	0.970414	0.838384	0.961039	0.974359	**0.988024**
	STD	0.015516	0.011909	0.018922	0.014481	0.018899	0.066718	0.014456	0.013367	**0.011017**
	Best	0.950888	0.978811	0.972033	0.990629	0.978884	0.903318	0.969938	0.979665	**0.990881**
	Worst	0.909442	0.937022	0.931992	0.970666	0.959993	0.782229	0.948994	0.968834	**0.982626**
Recall	AVG	0.952381	0.904762	0.910714	0.964286	0.976193	0.980095	0.880952	0.906772	**0.982143**
	STD	0.011828	0.021185	0.026811	0.009999	**0.008888**	0.009725	0.043198	0.039293	0.019556
	Best	0.969022	0.933966	0.930019	0.973811	0.981194	0.992322	0.934823	0.928282	**0.989945**
	Worst	0.916671	0.885917	0.876614	0.979948	0.971191	**0.981106**	0.835558	0.869949	0.972774
F-measure	AVG	0.938416	0.932515	0.930091	0.975904	0.973294	0.907104	0.919255	0.938272	**0.985075**
	STD	**0.003496**	0.003891	0.004029	0.016669	0.012279	0.031835	0.027657	0.022004	0.018847
	Best	0.945589	0.947112	0.941992	0.981885	0.978888	0.945433	0.940278	0.949773	**0.993927**
	Worst	0.931058	0.922516	0.921001	0.955322	0.955931	0.869028	0.882996	0.917005	**0.978863**
AUC	AVG	0.940279	0.935806	0.933258	0.976618	0.974283	0.905644	0.923902	0.941331	**0.985547**
	STD	**0.002902**	0.003322	0.003102	0.017744	0.022882	0.032918	0.025946	0.021966	0.016383
	Best	0.949881	0.942554	0.941921	0.986464	0.979669	0.938883	0.944888	0.958892	**0.990394**
	Worst	0.934155	0.929973	0.926767	0.955032	0.933992	0.869399	0.908873	0.922883	**0.978963**



(a) MCSO-CNN

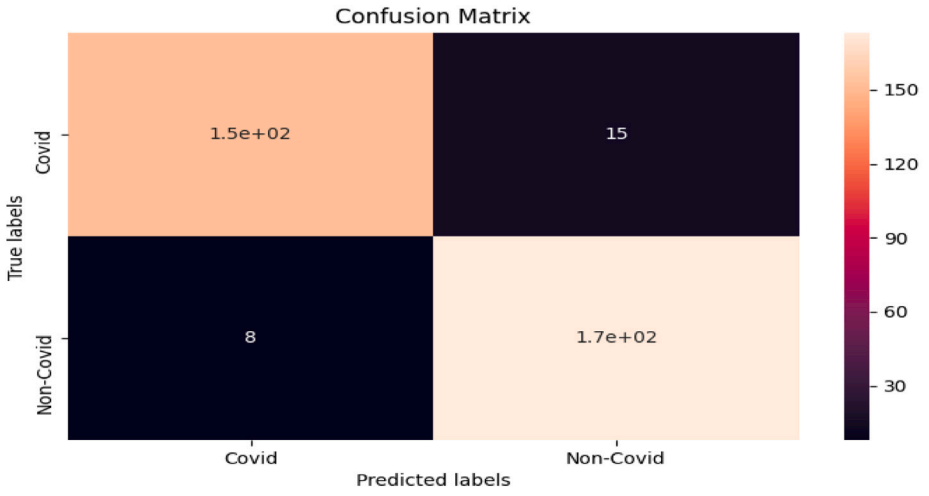


(b) DenseNet121

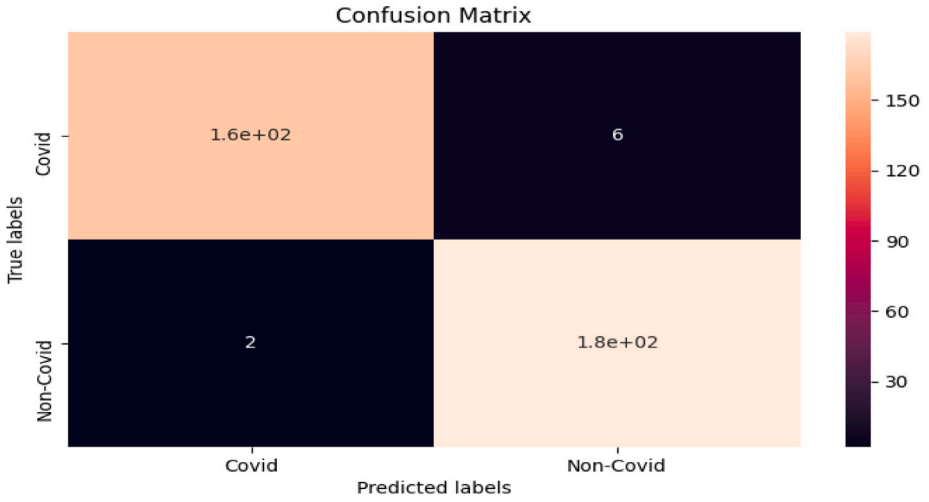


(c) MobileNet

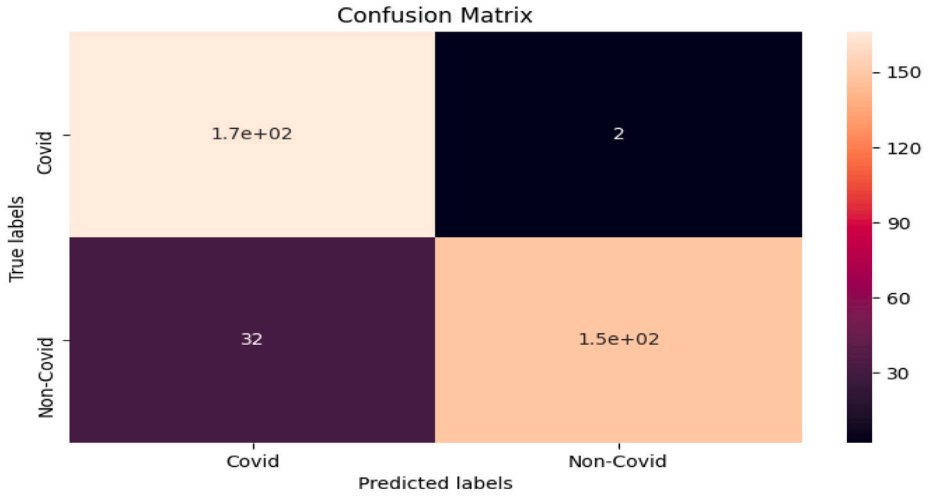
Fig. 9. Confusion matrices of different state-of-the-art deep learning architectures.



(d) InceptionV3

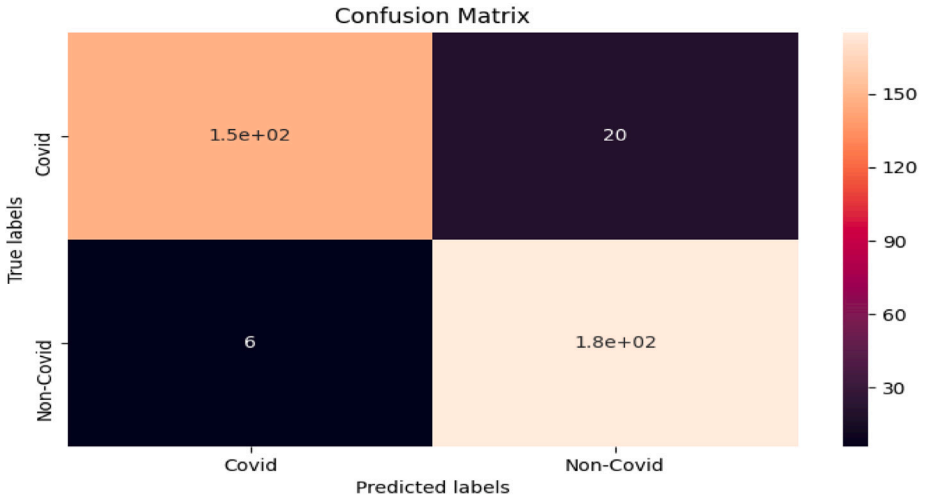


(e) XCeption

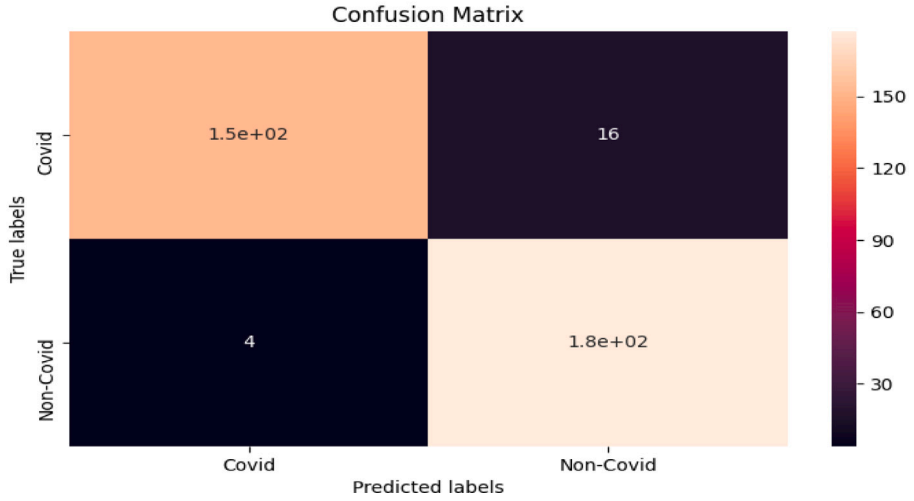


(f) VGGNet19

Fig. 9. (*continued*).



(g) DeCoVNet



(h) Brunese *et al.*

Fig. 9. (*continued*).

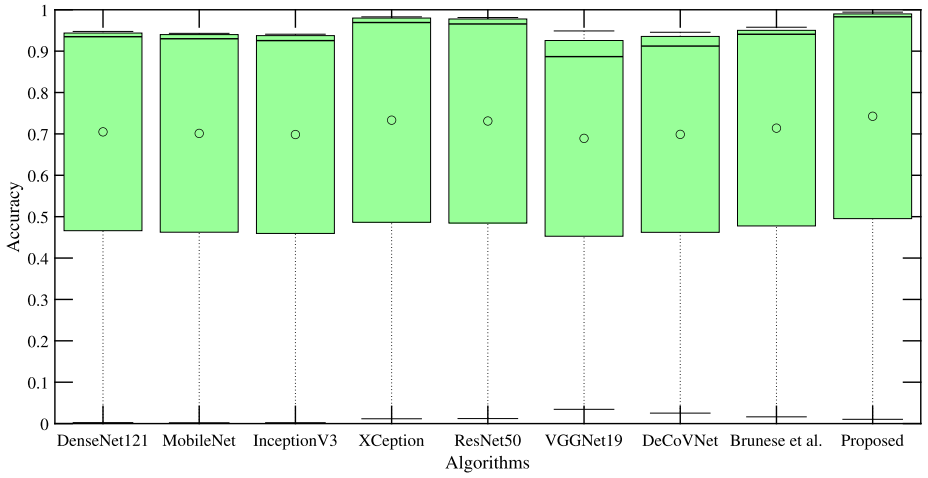


Fig. 10. Box plots of different state-of-the-art deep learning architectures for accuracy metric as the fitness function.

Table 5
The average results of the Friedman ranking test for the proposed and other benchmark models based on different classification performance metrics.

Model	DenseNet121	MobileNet	InceptionV3	XCeption	ResNet50	VGGNet19	DeCoVNet	Brunese et al.	GA	DE	PSO	MFO	WOA	SSA	HHO	GOA	CSO	Decision Tree	Random Forest	LightGBM	AdaBoost	SoftMax	Bagging	MCSO-CNN
ACC	8	9	13	2	3	19	14	7	5	24	11	12	23	10	4	6	22	16	18	21	20	17	15	**1**
Precision	17	8	11	2	5	24	9	4	15	14	6	13	12	18	3	10	23	20	16	19	22	7	21	**1**
Recall	9	17	15	6	3	2	19	16	7	24	18	13	23	5	10	12	4	13	21	19	8	22	10	**1**
F-measure	7	10	13	2	3	17	14	8	5	24	11	12	23	9	4	6	22	16	20	21	19	18	15	**1**
AUC	8	10	13	2	3	18	14	7	5	24	11	12	23	9	4	6	21	16	20	22	19	17	15	**1**
Summation	49	54	65	14	17	80	70	42	37	110	57	62	104	51	25	40	92	81	95	102	88	81	76	5
Average	9.8	10.8	13	2.8	3.4	16	14	8.4	7.4	22	11.4	12.4	20.8	10.2	5	8	18.4	16.2	19	20.4	17.6	16.2	15.2	1
Final Ranking	8	10	13	2	3	16	14	7	5	24	11	12	23	9	4	6	20	17	21	22	19	17	15	**1**

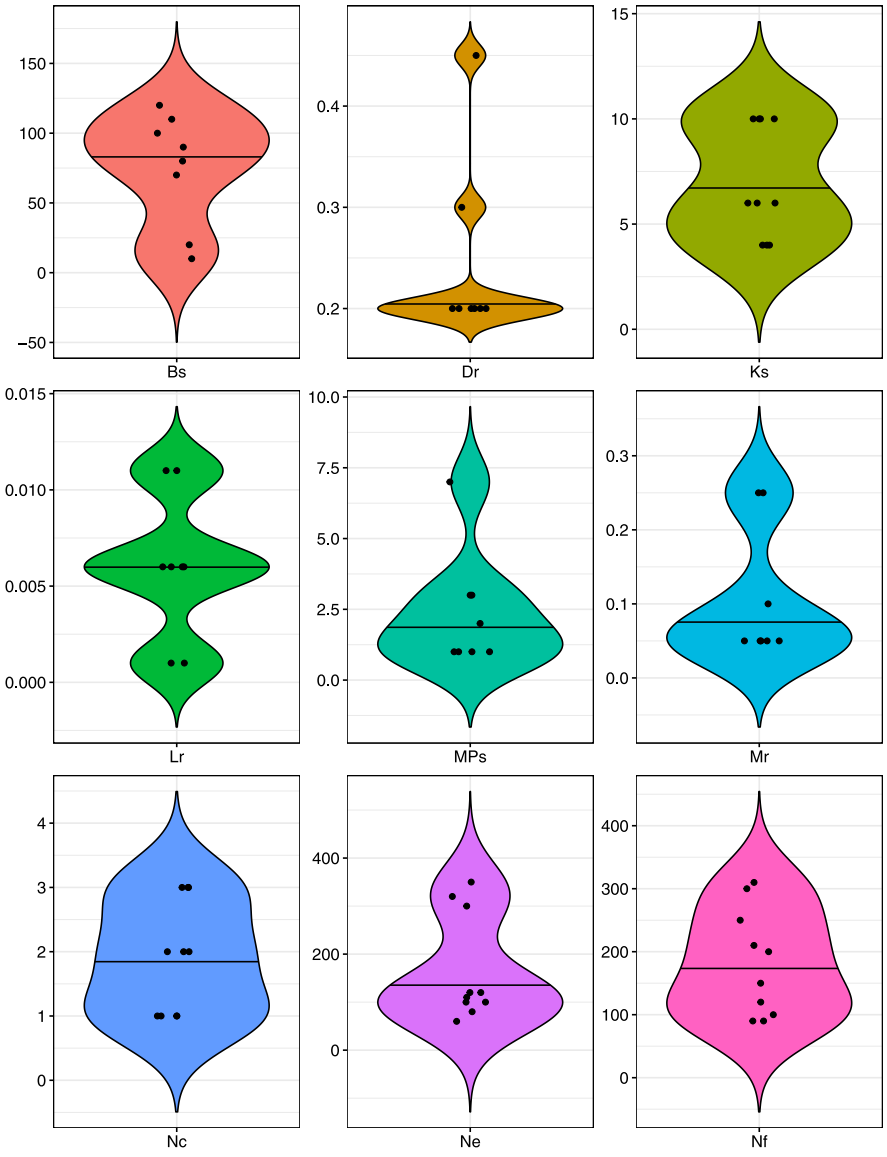


Fig. 11. Violin plots of the evolved hyperparameters utilized in the MCSO-CNN model.

proposed method has the lowest consumption time for training and testing time indicators.

4.6. Remarks

In this section, we provide some discussions about the insight of the main results and also the reasons why the tested results are achieved. It is worth mentioning that the contributions of the proposed method can be investigated in three different perspectives. First, we proposed a novel evolutionary algorithm by incorporating three evolutionary operators: Cauchy mutation, evolutionary boundary constraint handling, and tent chaotic map into the search process of the original version of competitive swarm optimizer model. Second, the Softmax layer of CNN model is replaced with a KNN classifier to improve the classification accuracy. Third, the proposed evolutionary algorithm is applied to automatically achieve the optimal values of hyperparameters of CNN model. In order to show the efficiency of these three contributions, several experiments have been conducted in this paper. For the first perspective, the performance of the proposed evolutionary algorithm is compared with a number of state-of-the-art evolutionary methods including GA, DE, PSO, MFO, WOA, SSA, HHO, GOA, and CSO. The results of these experiments are reported in Table 2 where it is shown that the proposed method outperforms other compared methods. The main reason to obtain such results is to consider the three evolutionary operators in the proposed evolutionary algorithm which leads to a significant improvement in the accuracy of classification method. In other words, the proposed evolutionary algorithm makes a balance between the exploration and exploitation phases which results in increasing convergence speed and reducing the probability of falling into local optima. For the second perspective, the proposed method is compared with different supervised learning classifiers to investigate the performance of the KNN classifier in comparison to other models including Decision Tree, Random Forest, LightGBM, AdaBoost, Softmax, and Bagging. The results of these experiments are reported in Table 3 where we can see that the KNN classifier performs better than other models. This is due to the fact that the KNN classifier takes into account the agreement of the neighborhood labeling which results in achieving accurate outputs in the last layer of CNN model. Finally, for the third perspective, the performance of the proposed classification method is compared with different deep learning based classifiers to investigate that how the deep CNN model optimized by the proposed evolutionary algorithm can outperform other compared models. To this end, the proposed method is compared with DenseNet121, MobileNet, InceptionV3, XCeption, ResNet50, VGGNet19, and DeCoVNet models.

Table 6
Run-time (in second) of the proposed model and other competitive algorithms for the COVID-19 dataset.

Model	Optimization time	Training time	Test time
DenseNet121	–	1863	357
MobileNet	–	1812	353
InceptionV3	–	1875	416
XCeption	–	1794	331
ResNet50	–	1783	345
VGGNet19	–	1631	337
DeCoVNet	–	1611	321
Brunese et al.	–	1523	311
GA	3830	1278	293
DE	3810	1263	296
PSO	3794	1231	287
MFO	3840	1539	304
WOA	3820	1381	298
SSA	3780	1216	286
HHO	3835	1372	291
GOA	3793	1201	271
CSO	3789	1123	263
Decision Tree	3642	1217	269
Random Forest	3621	1108	234
LightGBM	3597	1116	212
AdaBoost	3607	1173	225
SoftMax	3543	989	183
Bagging	3589	1009	195
MCSO-CNN	**3431**	**824**	**154**

The results of these experiments are shown in Table 4 where it can be seen that the proposed method is better than other models. Therefore, we can conclude that considering the KNN classifier in the last layer of CNN model and also using the proposed evolutionary algorithm to optimize the hyperparameters of CNN model significantly improve the classification accuracy.

In addition, to prove that the experimental results are statistically significant, we performed a ranking statistical test called the Friedman ranking test to theoretically evaluate and analyze the performance of the proposed model and all benchmark algorithms. The results of this statistical test reported in Table 5 prove that the experimental results are statistically significant. Also, the run time of the proposed method is compared with other methods and the results are shown in Table 6 based on optimization time, training time, and test time. These results demonstrate that the proposed classification method performs in a shorter time with respect to other compared models. Therefore, it can be concluded that the proposed method not only achieves more accurate classification results, it also performs faster than other compared methods.

5. Conclusion

Coronavirus disease (COVID-19) has become a very challenging issue in the world and made enormous problems in different countries. One of the main approaches to detect COVID-19 cases is to investigate chest X-ray images by physicians. Due to the high performance of deep neural networks in interpreting images, several COVID-19 detection approaches have already been developed. However, these approaches mainly ignore the tuning of hyperparameters of deep neural networks optimally and utilize a static way to manually set the values of these hyperparameters. To address this challenge, we proposed a novel image classification method in this paper to detect COVID-19 cases based on the chest X-rays images. Specifically, we utilized CNN model as a deep neural network to make a trained classification model based on the input samples. Moreover, to optimize the hyperparameters of the CNN model, we used an improved evolutionary algorithm to find optimal values leading to enhance the classification accuracy. In the improved evolutionary algorithm, three effective operators are incorporated to make a balance between the exploration and exploitation phases and speed up the algorithm's convergence. Therefore, different to other previous approaches, our proposed method can automatically find the optimal values of CNN's hyperparameters leading to improve the classification accuracy. Extensive experiments are performed based on a real-world dataset to assess the effectiveness of the proposed classification method. Experimental results showed the superiority of the proposed image classification method in comparison to other state-of-the-art models. In addition, we analyzed the run time of different algorithms where the results showed that the proposed method can perform in a faster way than the other models. As the future works, other evolutionary algorithms can be employed to obtain better results for the CNN model. Moreover, other deep neural network models can be developed to make a better representation of the input images and provide better performance for the detection of coronavirus disease.

CRediT authorship contribution statement

Seyed Mohammad Jafar Jalali: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Milad Ahmadian:** Methodology, Software, Writing – original draft, Writing – review & editing. **Sajad Ahmadian:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Rachid Hedjam:** Writing – original draft, Writing – review & editing, Project administration. **Abbas Khosravi:** Writing – original draft, Project administration. **Saeid Nahavandi:** Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Ahmadian, S., Ahmadian, M., & Jalili, M. (2021). A deep learning based trust- and tag-aware recommender system. *Neurocomputing*, http://dx.doi.org/10.1016/j.neucom.2021.11.064.

Ahmadian, S., Jalali, S. M. J., Islam, S. M. S., Khosravi, A., Fazli, E., & Nahavandi, S. (2021). A novel deep neuroevolution-based image classification method to diagnose coronavirus disease (COVID-19). *Computers in Biology and Medicine*, *139*, Article 104994.

Ahmadian, S., Jalali, S. M. J., Raziani, S., & Chalechale, A. (2021). An efficient cardiovascular disease detection model based on multilayer perceptron and moth-flame optimization. *Expert Systems*, http://dx.doi.org/10.1111/exsy.12914.

Ahmadian, S., Joorabloo, N., Jalili, M., & Ahmadian, M. (2022). Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach. *Expert Systems with Applications*, *187*, Article 115849.

Ahmadian, S., Joorabloo, N., Jalili, M., Meghdadi, M., Afsharchi, M., & Ren, Y. (2018). A temporal clustering approach for social recommender systems. In *IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)* (pp. 1139–1144). IEEE.

Ahmadian, S., Joorabloo, N., Jalili, M., Ren, Y., Meghdadi, M., & Afsharchi, M. (2020). A social recommender system based on reliable implicit relationships. *Knowledge-Based Systems*, *192*, Article 105371.

Ahmadian, S., & Khanteymoori, A. R. (2015). Training back propagation neural networks using asexual reproduction optimization. In *2015 7th conference on information and knowledge technology (IKT)* (pp. 1–6). IEEE.

Ahmadian, S., Meghdadi, M., & Afsharchi, M. (2018a). Incorporating reliable virtual ratings into social recommendation systems. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *48*(11), 4448–4469.

Ahmadian, S., Meghdadi, M., & Afsharchi, M. (2018b). A social recommendation method based on an adaptive neighbor selection mechanism. *Information Processing and Management*, *54*, 707–725.

Ahmadian, S., Moradi, P., & Akhlaghian, F. (2014). An improved model of trust-aware recommender systems using reliability measurements. In *2014 6th conference on information and knowledge technology (IKT)* (pp. 98–103). IEEE.

Al-Betar, M. A., Awadallah, M. A., Heidari, A. A., Chen, H., Al-khraisat, H., & Li, C. (2021). Survival exploration strategies for harris hawks optimizer. *Expert Systems with Applications*, *168*, Article 114243.

Al-Itbi, A. S., Alwahhab, S. A., & Ahmed Bahaaulddin, M. (2022). X-ray covid-19 detection based on scatterwavelet transform and dense deep neural network. *Computer Systems Science and Engineering*, *41*, 1255–1271.

Alamoodi, A., Zaidan, B., Zaidan, A., Albahri, O., Mohammed, K., & Malik, R. (2020). Sentiment analysis and its applications in fighting COVID-19 and infectious diseases: A systematic review. *Expert Systems with Applications*, *167*, Article 114155.

Alqudah, A. M., & Qazan, S. (2020). Augmented COVID-19 X-ray images dataset. In *Mendeley data, V4*. http://dx.doi.org/10.17632/2fxz4px6d8, 4.

Altan, A., & Karasu, S. (2020). Recognition of COVID-19 disease from X-ray images by hybrid model consisting of 2D curvelet transform, chaotic salp swarm algorithm and deep learning technique. *Chaos, Solitons & Fractals*, *140*, Article 110071.

Anwar, S. M., Majid, M., Qayyum, A., Awais, M., Alnowami, M., & Khan, M. K. (2018). Medical image analysis using convolutional neural networks: a review. *Journal of Medical Systems*, *42*(11), 226.

Apostolopoulos, I. D., & Mpesiana, T. A. (2020). Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*, *43*, 635—640.

Aviles-Rivero, A. I., Sellars, P., Schönlieb, C.-B., & Papadakis, N. (2022). GraphXCOVID: Explainable deep graph diffusion pseudo-Labelling for identifying COVID-19 on chest X-rays. *Pattern Recognition*, *122*, Article 108274.

Bhattacharyya, A., Bhaik, D., Kumar, S., Thakur, P., Sharma, R., & Pachori, R. B. (2022). A deep learning based approach for automatic detection of COVID-19 cases using chest X-ray images. *Biomedical Signal Processing and Control*, *71*, Article 103182.

Brunese, L., Mercaldo, F., Reginelli, A., & Santone, A. (2020). Explainable deep learning for pulmonary disease and coronavirus COVID-19 detection from X-rays. *Computer Methods and Programs in Biomedicine*, *196*, Article 105608.

Chakraborty, S., & Mali, K. (2020). Sufmofpa: A superpixel and meta-heuristic based fuzzy image segmentation approach to explicate COVID-19 radiological images. *Expert Systems with Applications*, *167*, Article 114142.

Cheng, R., & Jin, Y. (2014). A competitive swarm optimizer for large scale optimization. *IEEE Transactions on Cybernetics*, *45*(2), 191–204.

Deb, S. D., Jha, R. K., Jha, K., & Tripathi, P. S. (2022). A multi model ensemble based deep convolution neural network structure for detection of COVID19. *Biomedical Signal Processing and Control*, *71*, Article 103126.

Deng, W., Liu, H., Xu, J., Zhao, H., & Song, Y. (2020). An improved quantum-inspired differential evolution algorithm for deep belief network. *IEEE Transactions on Instrumentation and Measurement*, *69*, 7319–7327.

Desai, S. B., Pareek, A., & Lungren, M. P. (2020). Deep learning and its role in COVID-19 medical imaging. *Intelligence-Based Medicine*, *3*, Article 100013.

Gaur, P., Malaviya, V., Gupta, A., Bhatia, G., Pachori, R. B., & Sharma, D. (2022). Covid-19 disease identification from chest CT images using empirical wavelet transformation and transfer learning. *Biomedical Signal Processing and Control*, *71*, Article 103076.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, *97*, 849–872.

Hu, R., Gan, J., Zhu, X., Liu, T., & Shi, X. (2022). Multi-task multi-modality SVM for early COVID-19 diagnosis using chest CT data. *Information Processing & Management*, *59*, Article 102782.

Ismael, A. M., & Şengür, A. (2020). Deep learning approaches for COVID-19 detection based on chest X-ray images. *Expert Systems with Applications*, *164*, Article 114054.

Jalali, S. M. J., Ahmadian, M., Ahmadian, S., Khosravi, A., Alazab, M., & Nahavandi, S. (2021). An oppositional-Cauchy based GSK evolutionary algorithm with a novel deep ensemble reinforcement learning strategy for COVID-19 diagnosis. *Applied Soft Computing*, *111*, Article 107675.

Jalali, S. M. J., Ahmadian, S., Kavousi-Fard, A., Khosravi, A., & Nahavandi, S. (2022). Automated deep CNN-LSTM architecture design for solar irradiance forecasting. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *52*, 54–65.

Jalali, S. M. J., Ahmadian, S., Kebria, P. M., Khosravi, A., Lim, C. P., & Nahavandi, S. (2019). Evolving artificial neural networks using butterfly optimization algorithm for data classification. In *International conference on neural information processing* (pp. 596–607). Springer.

Jalali, S. M. J., Ahmadian, S., Khodayar, M., Khosravi, A., Ghasemi, V., Shafie-khah, M., & Catalão, J. P. S. (2021). Towards novel deep neuroevolution models: chaotic levy grasshopper optimization for short-term wind speed forecasting. *Engineering with Computers*, http://dx.doi.org/10.1007/s00366-021-01356-0.

Jalali, S. M. J., Ahmadian, S., Khosravi, A., Mirjalili, S., Mahmoudi, M. R., & Nahavandi, S. (2020). Neuroevolution-based autonomous robot navigation: a comparative study. *Cognitive Systems Research*, *62*, 35–43.

Jalali, S. M. J., Ahmadian, S., Khosravi, A., Shafie-khah, M., Nahavandi, S., & Catalão, J. P. S. (2021). A novel evolutionary-based deep convolutional neural network model for intelligent load forecasting. *IEEE Transactions on Industrial Informatics*, *17*, 8243–8253.

Jalali, S. M. J., Hedjam, R., Khosravi, A., Heidari, A. A., Mirjalili, S., & Nahavandi, S. (2020). Autonomous robot navigation using moth-flame-based neuroevolution. In *Evolutionary machine learning techniques* (pp. 67–83). Springer.

Jalali, S. M. J., Kebria, P. M., Khosravi, A., Saleh, K., Nahavandi, D., & Nahavandi, S. (2019). Optimal autonomous driving through deep imitation learning and neuroevolution. In *2019 IEEE international conference on systems, man and cybernetics (SMC)* (pp. 1215–1220). IEEE.

Jalali, S. M. J., Khodayar, M., Ahmadian, S., Shafie-khah, M., Khosravi, A., Islam, S. M. S., & Catalão, J. P. S. (2021). A new ensemble reinforcement learning strategy for solar irradiance forecasting using deep optimized convolutional neural network models. In *International conference on smart energy systems and technologies (SEST)* (pp. 1–6). IEEE.

Jalali, S. M. J., Khosravi, A., Alizadehsani, R., Salaken, S. M., Kebria, P. M., Puri, R., & Nahavandi, S. (2019). Parsimonious evolutionary-based model development for detecting artery disease. In *2019 IEEE international conference on industrial technology (ICIT)* (pp. 800–805). IEEE.

Jalali, S. M. J., Khosravi, A., Kebria, P. M., Hedjam, R., & Nahavandi, S. (2019). Autonomous robot navigation system using the evolutionary multi-verse optimizer algorithm. In *2019 IEEE international conference on systems, man and cybernetics (SMC)* (pp. 1221–1226). IEEE.

Jalali, S. M. J., Osorio, G. J., Ahmadian, S., Lotfi, M., Campos, V., Shafie-khah, M., & Catalao, J. P. (2021). A new hybrid deep neural architectural search based ensemble reinforcement learning strategy for wind power forecasting. *IEEE Transactions on Industry Applications*.

Karasu, S., Altan, A., Saraç, Z., & Hacioglu, R. (2017). Histogram based vehicle license plate recognition with KNN method. In *5th international conference on advanced technology & sciences (ICAT'17)* (pp. 1–4).

Kashir, J., & Yaqinuddin, A. (2020). Loop mediated isothermal amplification (LAMP) assays as a rapid diagnostic for COVID-19. *Medical Hypotheses*, *141*, Article 109786.

Khodayar, M., Khodayar, M. E., & Jalali, S. M. J. (2021). Deep learning for pattern recognition of photovoltaic energy generation. *The Electricity Journal*, *34*(1), Article 106882.

Kim, J., Kim, B., & Savarese, S. (2012). Comparing image classification methods: K-nearest-neighbor and support-vector-machines. In *Proceedings of the 6th WSEAS international conference on computer engineering and applications, and proceedings of the 2012 American conference on applied mathematics, Vol. 1001* (pp. 48109–2122).

Kumar, S., Mishra, S., & Singh, S. K. (2020). Deep transfer learning-based COVID-19 prediction using chest X-rays. *MedRxiv*, 1–14.

Kumar, A., Tripathi, A. R., Satapathy, S. C., & Zhang, Y.-D. (2022). Sars-net: COVID-19 detection from chest x-rays by combining graph convolutional network and convolutional neural network. *Pattern Recognition*, *122*, Article 108255.

Li, C., Li, S., & Liu, Y. (2016). A least squares support vector machine model optimized by moth-flame optimization algorithm for annual power load forecasting. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *45*(4), 1166–1178.

Luján-García, J. E., Yáñez-Márquez, C., Villuendas-Rey, Y., & Camacho-Nieto, O. (2020). A transfer learning method for pneumonia classification and visualization. *Applied Sciences*, *10*(8), 2908.

Mahendran, A., & Vedaldi, A. (2016). Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, *120*(3), 233–255.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51–67.

Moradi, P., Rezaimehr, F., Ahmadian, S., & Jalili, M. (2016). A trust-aware recommender algorithm based on users overlapping community structure. In *2016 sixteenth international conference on advances in ICT for emerging regions (ICTer)* (pp. 162–167). IEEE.

Mousavirad, S. J., Jalali, S. M. J., Ahmadian, S., Khosravi, A., Schaefer, G., & Nahavandi, S. (2020). Neural network training using a biogeography-based learning strategy. In *International conference on neural information processing* (pp. 147–155). Springer.

Mukherjee, H., Ghosh, S., Dhar, A., Obaidullah, S. M., Santosh, K. C., & Roy, K. (2020). Deep neural network to detect COVID-19: one architecture for both ct scans and chest X-rays. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *51*, 2777—2789.

Ning, Q., Ma, Z., & Zhao, X. (2019). Dforml (KNN)-pseaac: Detecting formylation sites from protein sequences using K-nearest neighbor algorithm via chou's 5-step rule and pseudo components. *Journal of Theoretical Biology*, *470*, 43–49.

Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O., & Acharya, U. R. (2020). Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Computers in Biology and Medicine*, *121*, Article 103792.

Panwar, H., Gupta, P., Siddiqui, M. K., Morales-Menendez, R., Bhardwaj, P., & Singh, V. (2020). A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-scan images. *Chaos, Solitons & Fractals*, *140*, Article 110190.

Panwar, H., Gupta, P., Siddiqui, M. K., Morales-Menendez, R., & Singh, V. (2020). Application of deep learning for fast detection of COVID-19 in X-Rays using ncovnet. *Chaos, Solitons & Fractals*, *138*, Article 109944.

Parekh, M., Donuru, A., Balasubramanya, R., & Kapur, S. (2020). Review of the chest CT differential diagnosis of ground-glass opacities in the COVID era. *Radiology*, *297*, Article 202504.

Pereira, R. M., Bertolini, D., Teixeira, L. O., Silla Jr, C. N., & Costa, Y. M. (2020). Covid-19 identification in chest X-ray images on flat and hierarchical classification scenarios. *Computer Methods and Programs in Biomedicine*, *194*, Article 105532.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., & Iyengar, S. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys*, *51*(5), 1–36.

Qazani, M. R. C., Jalali, S. M. J., Asadi, H., & Nahavandi, S. (2020). Optimising control and prediction horizons of a model predictive control-based motion cueing algorithm using butterfly optimization algorithm. In *2020 IEEE congress on evolutionary computation (CEC)* (pp. 1–8). IEEE.

Rahmani, H. A., Aliannejadi, M., Ahmadian, S., Baratchi, M., Afsharchi, M., & Crestani, F. (2019). Lglmf: local geographical based logistic matrix factorization model for POI recommendation. In *Asia information retrieval symposium* (pp. 66–78). Springer.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: theory and application. *Advances in Engineering Software*, *105*, 30–47.

Shan, F., Gao, Y., Wang, J., Shi, W., Shi, N., Han, M., & Shi, Y. (2020). Lung infection quantification of covid-19 in ct images with deep learning. (pp. 1–15). arXiv preprint arXiv:2003.04655.

Sheykhivand, S., Mousavi, Z., Mojtahedi, S., Yousefi Rezaii, T., Farzamnia, A., Meshgini, S., & Saad, I. (2021). Developing an efficient deep neural network for automatic detection of COVID-19 using chest X-ray images. *Alexandria Engineering Journal*, *60*(3), 2885–2903.

Shi, F., Wang, J., Shi, J., Wu, Z., Wang, Q., Tang, Z., & Shen, D. (2020). Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19. *IEEE Reviews in Biomedical Engineering*, *14*, 4–15.

Tahmasebi, F., Meghdadi, M., Ahmadian, S., & Valiallahi, K. (2021). A hybrid recommendation system based on profile expansion technique to alleviate cold start problem. *Multimedia Tools and Applications*, *80*(2), 2339–2354.

Tan, T. Y., Zhang, L., & Lim, C. P. (2019). Intelligent skin cancer diagnosis using improved particle swarm optimization and deep learning models. *Applied Soft Computing*, *84*, Article 105725.

Tian, C., Fei, L., Zheng, W., Xu, Y., Zuo, W., & Lin, C.-W. (2020). Deep learning on image denoising: An overview. *Neural Networks*, *131*, 251–275.

Wang, L., Qiu Lin, Z., & Wong, A. (2020). Covid-net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Scientific Reports*, *10*(19549), 1–12.

Yadav, S. S., & Jadhav, S. M. (2019). Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big Data*, *6*(1), 113.

Yang, B., Zhong, L., Zhang, X., Shu, H., Yu, T., Li, H., & Sun, L. (2019). Novel bio-inspired memetic salp swarm algorithm and application to mppt for PV systems considering partial shading condition. *Journal of Cleaner Production*, *215*, 1203–1222.

Yoo, S. H., Geng, H., Chiu, T. L., Yu, S. K., Cho, D. C., & Heo, J. (2020). Deep learning-based decision-tree classifier for COVID-19 diagnosis from chest X-ray imaging. *Frontiers in Medicine*, *7*, 427.

Yuan, Q., Zhang, Q., Li, J., Shen, H., & Zhang, L. (2018). Hyperspectral image denoising employing a spatial–spectral deep residual convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(2), 1205–1218.

Yue, S., & Zhang, H. (2021). A hybrid grasshopper optimization algorithm with bat algorithm for global optimization. *Multimedia Tools and Applications*, *80*(3), 3863–3884.

Zheng, C., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H., & Wang, X. (2020). Deep learning-based detection for COVID-19 from chest CT using weak label. *MedRxiv*, 1–12.

Zheng, Q., Yang, M., Yang, J., Zhang, Q., & Zhang, X. (2018). Improvement of generalization ability of deep CNN via implicit regularization in two-stage training process. *IEEE Access*, *6*, 15844–15869.

Zhou, W., Wang, P., Heidari, A. A., Wang, M., Zhao, X., & Chen, H. (2021). Multi-core sine cosine optimization: Methods and inclusive analysis. *Expert Systems with Applications*, *164*, Article 113974.