

# A Transfer Learning-Enabled Optimized Extreme Deep Learning Paradigm for Diagnosis of COVID-19

Ahmed Reda\*, Sherif Barakat and Amira Rezk

Department of Information System, Faculty of Computer and Information Science, Mansoura University,  
Mansoura, Egypt

\*Corresponding Author: Ahmed Reda. Email: ahmedreda17@mans.edu.eg

Received: 25 April 2021; Accepted: 26 May 2021

**Abstract:** Many respiratory infections around the world have been caused by coronaviruses. COVID-19 is one of the most serious coronaviruses due to its rapid spread between people and the lowest survival rate. There is a high need for computer-assisted diagnostics (CAD) in the area of artificial intelligence to help doctors and radiologists identify COVID-19 patients in cloud systems. Machine learning (ML) has been used to examine chest X-ray frames. In this paper, a new transfer learning-based optimized extreme deep learning paradigm is proposed to identify the chest X-ray picture into three classes, a pneumonia patient, a COVID-19 patient, or a normal person. First, three different pre-trained Convolutional Neural Network (CNN) models (resnet18, resnet25, densenet201) are employed for deep feature extraction. Second, each feature vector is passed through the binary Butterfly optimization algorithm (bBOA) to reduce the redundant features and extract the most representative ones, and enhance the performance of the CNN models. These selective features are then passed to an improved Extreme learning machine (ELM) using a BOA to classify the chest X-ray images. The proposed paradigm achieves a 99.48% accuracy in detecting covid-19 cases.

**Keywords:** Butterfly optimization algorithm (BOA); covid-19; chest X-ray images; convolutional neural network (CNN); extreme learning machine (ELM); feature selection

## 1 Introduction

The novel outbreak of coronavirus has come from Wuhan, China, and since January 2020, it has spread extensively across the globe [1]. Health practitioners worldwide are conducting extensive studies to find an effective treatment for the disease. The virus will cause people with compromised immune systems to die [2]. COVID-19 is spread primarily through direct touch from person to person. As the second wave of COVID-19 begins and the number of infected individuals grows, the need for fast diagnostic methods increases. These methods prevent the rapid spread of this virus by detecting infected people as soon as possible and putting them in isolation. The



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

traditional way to diagnose infection is Polymerase chain reaction (PCR), but this method takes a lot of time and is more costly.

Chest X-Ray (CXR) is an effective therapeutic addition that plays an important part in the preliminary inquiry of lung abnormalities [3]. CT scans and chest rays can detect the disease, especially without the existence of symptoms. It is a time-consuming procedure to read the X-ray and CT scans of many patients manually. Computer-Aided Diagnosis (CAD) are systems that assist doctors in the risk identification of medical images. A CAD system has been developed to assist radiologists in diagnosing COVID-19.

Deep learning (DL) has proved its promising performance to diagnose different diseases using computer-aided detection models [4,5]. DL, an artificial intelligence (AI) subfield, facilitates the development of end-to-end models to produce promising results using input data without manual extraction of features [6,7]. Many disease detection systems have effectively applied deep learning strategies such as arrhythmia discovery [8,9], skin cancer classification [10,11], brain disease classification [12], chest X-ray pneumonia detection [13], fundus image segmentation [14].

The features should be extracted from CXR images before utilizing a detection system. The key objective of extracted features is to obtain a feature set from the CXR images used in the classification phase to predict the diseases [15] correctly. Various feature extraction techniques have been utilized, such as texture features, Gabor features, co-occurrence matrix features, and wavelet transform-based features [16]. Convolutional Neural Network (CNN) is the robust method that is most widely used for feature extraction and learning. Big data is required to train a CNN from scratch. But it is challenging to acquire a large amount of labeled medical images. In this case, the pre-trained CNN models trained on an extensive database like ImageNet can be utilized [17]. Pre-trained deep networks were successfully utilized in diagnosing prostate cancer [18], brain diseases [19], leukemia [20], to name a few. Pre-trained CNNs have also successfully recognized the COVID-19 [21,22]. The extracted feature set may carry redundant or irrelevant information [23]. Therefore, it is required to eliminate irrelevant information before the classification phase [24]. The most relevant feature set obtains accurate classification results for the COVID-19 disease with the minimum time consumption. Different ML techniques have been utilized in literature to recognize COVID-19 patients and include support vector machine (SVM), and several others.

The extreme learning machine (ELM) [25] is widely used in recognition systems because of its fast-learning capability and adequate generalization [26]. Generally, in the basic ELM model, the random initialization of the input weights and hidden biases can put the ELM solution models closest to local minima [27]. The generalization ability of ELM can be enhanced by combining this model with other techniques [28,29]. In some studies, researchers have successfully optimized the ELM using nature-inspired algorithms. Mohapatra et al. [30] introduced the ELM model optimized by cuckoo search to organize medical datasets. Satapathy et al. [31] suggested another optimized ELM by firefly algorithm. The hybrid firefly-ELM model was then applied to a photovoltaic interactive microgrid for stability analysis. Researchers in [32] enhanced the ELM using a whale optimization algorithm and then utilized it to evaluate the ageing degree of the insulated gate bipolar transistor. The comparison of optimized ELM and singleton ELM proved that the optimized ELM produces reasonable recognition.

Butterfly optimization algorithm (BOA) [33] is a new meta-heuristic swarm intelligence technique motivated by the food foraging butterflies' behavior. Based on random exploration and improvement by exploitation, the BOA can solve complex problems. Authors in [33] have observed that mostly BOA perform better on the unimodal and multimodal benchmark functions. The

exploitation and high convergence rate are the exceptional strength of BOA, based on elitism and employed random movement. The appealing behaviors of BOA allow researchers to implement BOA in several other applications (e.g., function selection dependent on wrapping algorithms). A binary Butterfly optimization algorithm (bBOA) is presented by Arora et al. [34] to resolve the feature selection issues. The bBOA selects robust features that enhance the classification accuracy. The bBOA also possesses a good convergence rate and produces the optimal solution.

This study presents an end-to-end ML-based system to identify COVID-19 from CXR scans automatically. Pre-trained CNN models, which are Resnet18 [35], Resnet50 [35], densenet201 [36], were applied to CXR images for extracting the discriminative features. The bBOA algorithm was then used to pick the most informative features from the collected deep features. These features, then, were combined and were applied as the input to an optimized ELM model (ELM-BOA). Briefly, this study demonstrates that combining the deep characteristics extracted from the common levels of different CNN architectures enhances the efficiency of the classification process. The contributions of this study are recapitulated as follow:

- The proposed framework uses several pre-trained CNNs for feature extraction.
- The framework uses a butterfly optimization algorithm (BOA) for feature selection and optimizing the ELM model for classification.
- Evaluate and equate the output of the proposed methodology with state-of-the-art approaches.

The remainder of the paper is broken down into this structure. Section 2 describes the literature review of previous studies. Methods are explained in Section 3. Section 4 describes the proposed model. The obtained results and their detailed analysis are discussed in Section 5. In Section 6, the paper is concluded.

## 2 Literature Review

Recently, various methods have been developed for the identification of COVID-19 patients from their X-ray and CT images. These methods utilized computer vision (CV) based machine learning (ML) algorithms. Apostolopoulos et al. [37] executed transfer learning on X-ray images dataset using different Convolutional Neural Network (CNN) models. They evaluate their technique on two datasets. The datasets include confirming COVID-19, bacterial pneumonia, and normal cases. The maximum accuracy achieved on MobileNetv2 is 96.78% on 2-class and 94.72% on 3-class. Researchers in [38], introduced a novel DL model (COVNet) for the detection of COVID-19 disease from chest CT frames. This model achieved sensitivity and specificity of 90% and 96%, respectively. Ghoshal et al. [39] conducted a study to investigate uncertainty in DL models to identify COVID-19 in chest X-ray images. They estimated the uncertainty by performing transfer learning on the Bayesian DL classifier.

Narin et al. [40] performed experiments on three CNN models including ResNet50, Inception-ResNetV2, and InceptionV3 to identify the COVID-19 patients from chest X-ray images. According to their experiments, ResNet50 produced the highest performance with 98% classification accuracy. In [41], a DL-based COVIDX-Net was proposed for automatic detection of COVID-19 in X-ray images. This COVIDX-Net model is based on seven different CNN models including VGG19, DenseNet201, InceptionV3, ResNetV2, InceptionResNetV2, Xception, and MobileNetV2. This technique achieved a 0.91 F1-score on VGG19 and DenseNet201. Wang et al. [42] proposed a DL-based method to predict the COVID-19 disease in CT images. They fine-tuned the modified

Inception architecture and extract the CNN features for classification. This method achieved an accuracy of 82.9% and AUC 0.90.

A new artificial neural network based CapsNet [43] was introduced for coronavirus detection in chest X-Ray image dataset. Researchers tested their network on binary and multi-class classification. The achieved recognition accuracy on binary class, and multi-class classification is 97.24%, and 84.22% respectively. A dual-branch combination network (DCN) [44] was introduced to identify and risk identification the COVID-19 in chest CT scans. First, researchers segment the lesion area to gain more accurate results in the classification phase. The DCN model achieved 96.74% accuracy. Horry et al. [45] utilized the TL approach for COVID-19 determination in Ultrasound, X-Ray, and CT frames. To perform TL, they selected the VGG-19 network and make appropriate changes in the parameters to fine-tune the model. The capabilities of the presented technique were measured using precision and achieved 100%, 86%, 84% precision rate for Ultrasound, X-Ray, and CT scans, respectively.

COVIDiagnosis-Net [46] is a deep learning model based on the SqueezeNet with Bayesian optimization. This developed model was trained and estimated on a dataset that contains a little number of X-ray images for COVID-19 cases and produced a 100% accuracy for detection of COVID-19 class and overall accuracy of 98.26%. Rahimzadeh et al. [47] merged the features produced by two pre-trained networks: Xception and ResNet50 V2. But this merge made the model's size up to 560 MB, which is unsuitable for practical implementation for real-time detection. The model was trained using 3783 CXR images and tested on 11302 CXR images. The model achieved an overall accuracy of 99.5 and 91.4 in the COVID-19 class. The authors compared the results obtained from a model that used concatenated features, and the results produced by models that use features come from a single CNN. Moreover, they proved that using concatenated features achieves better results.

Shaban et al. [48] achieved a high accuracy of 96% by exploiting the advantages of genetic algorithm in feature selection and using an enhanced K-nearest neighbor as a classifier. This model is tested on chest CT images for identification of infected and non-infected people and achieved results better than other recent models. A medical model of COVID-19 detection was proposed by Nour et al. [49] to support clinical applications. It is constructed on deep learning and Bayesian optimization. The CNN model automatically extracts features, which are often processed using various machine learning methods, including KNN, SVM, and Decision Tree. Data augmentation was applied to increase COVID-19 class samples. The proposed system's efficiency was assessed using 70% and 30% of the data set for training and testing, respectively. The introduced technique obtained an accuracy of 97.14%.

Deep learning models have gained significant importance in detecting patients infected by a coronavirus from the chest X-ray images. The detection model for COVID-19 can be enhanced by using deep learning models as feature extractors or by fine-tuning deep learning models. Thus, our work's primary goal is to develop a deep learning-based methodology for the recognition of COVID-19 affected patients.

### 3 Methods and Methodologies

#### 3.1 Convolutional Neural Network (CNN)

Deep learning techniques help extract meaningful features in some data types, such as images and videos. In Medical research, a Convolutional Neural network (CNN) is significantly utilized in extracting these features from a large volume of medical images such as X-ray images and

computerized tomography (CT) images. Besides, it achieves high accuracy and lowers computational cost, which provides generous support in improving health community research [50]. “Deep” refers to the large numbers of layers in the network. This type of architecture helps these networks to find complex features while the simple networks cannot. CNN’s primary fundamental is to gain local features at beginning layers and merge them at last layers to form more complex features [51].

Transfer learning is an efficient technique that takes advantage of a formerly learned model’s knowledge to solve another probably related job by demand minimal re-training or fine-tuning [52]. Deep learning algorithms require two essential requirements to work effectively: a massive amount of labeled data and mighty computing power. Forming a large-scale and high-quality dataset is very difficult and complicated [53]. While providing a powerful device to implement deep learning techniques require Equipped laboratories or large fund [54]. Hence, Deep Transfer Learning (DTL) attempts to solve this issue.

The first shape of transfer learning is using the pre-trained CNN as a fixed feature extractor [55]. This approach preserves the primary architecture and all learned weights. After CNN extracts features, it is inserted into a new network to perform the classification tasks. In the second and more complex shape called fine-tuning [56], some particular modifications are applied to the pre-trained CNN to achieve better results. These modifications involve architecture adjustment and parameter tuning. Besides, some New parameters are inserted into the network and demand training on a relatively significant amount of data to be more beneficial.

### 3.2 *Extreme Learning Machine (ELM)*

ELM is a learning model for a single hidden layer feed-forward neural network (SLFN). Also, it is classified as a neural network with random weights (NNRW). It was presented by Haung et al. [57]. The ELM was introduced to avoid over-fitting problems and reduce the training time. ELM is a learning model for a single hidden layer feed-forward neural network (SLFN). Also, it is classified as a neural network with random weights (NNRW). It was presented by Haung et al. [57]. The ELM was introduced to avoid over-fitting problems and reduce the training time. NNRW is preferable than the traditional artificial neural networks (ANN), because ANN has several drawbacks due to its training mechanism (error backpropagation) and the number of hidden layers, such as slow convergence, time-consuming, and local minima problems [58]. The ELM was introduced to avoid the over-fitting problems, reduce the training time, and solve all problems that traditional ANN have. This method randomly initializes the connection weights between the input and hidden layers and hidden biases. Then output weights are computed to connect the hidden layer with the output layer. This training mechanism allows ELM to be faster than traditional ANN. To determine the impact of the random parameters on ELM, Cao et al. [59] organized an experimental framework and studied the relationship between parameters (e.g., the number of neurons in the hidden layer, the threshold randomization range between the hidden nodes, the randomization range of the weights between the input layer and hidden layer, and the activation function types) and the optimal performance of ELM. They found that all the parameters mentioned above performs a dominant role in the stability of the model.

Let for  $Z$  distinct samples, the training data is  $\{(p_i, q_i)\}$ , where  $p_i = [p_{i1}, p_{i2}, \dots, p_{im}]^S \in \mathbb{R}^m$  is the input data,  $q_i = [q_{i1}, q_{i2}, \dots, q_{in}]^S \in \mathbb{R}^n$  is the output data, and  $i = 1, 2, \dots, Y$ ,  $j = 1, 2, \dots, Z$ .

For the activation function  $g(p)$ , the ELM model can be given as

$$\sum_{i=1}^Y \gamma_i g(\alpha_i \cdot p_j + b_i) = t_j \quad (1)$$

where  $\alpha_i = [\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{im}]$  is the input weight,  $b_i$  denotes the bias for the  $i^{th}$  hidden neuron, the output weight is  $\gamma_i = [\gamma_{i1}, \gamma_{i2}, \dots, \gamma_{im}]$ , and the network output is represented by  $t_j = [t_{1j}, t_{2j}, \dots, t_{nj}]^S$ . The equation for the ELM model can be simplified and represented in matrix form as

$$D\gamma = T' \quad (2)$$

where  $\gamma = [\gamma_1, \gamma_2, \dots, \gamma_Y]^S$ ,  $T = [t_1, t_2, \dots, t_Z]$ , and  $T'$  denotes the transpose of matrix  $T$ .  $D$  denotes the hidden layer output matrix of and can be given as

$$D = \begin{bmatrix} g(\alpha_1 \cdot p_1 + b_1) & \cdots & g(\alpha_Y \cdot p_1 + b_Y) \\ \vdots & \cdots & \vdots \\ g(\alpha_1 \cdot p_Z + b_1) & \cdots & g(\alpha_Y \cdot p_Z + b_Y) \end{bmatrix}_{Z \times Y} \quad (3)$$

$\hat{\gamma}$  can be determined using the Moore-Penrose (MP) inverse function as given below.

$$\hat{\gamma} = D^+ T' \quad (4)$$

### 3.3 Butterfly Optimization Algorithm (BOA)

Butterfly optimization algorithm (BOA) [33] is a new meta-heuristic swarm intelligence technique motivated by the food foraging butterflies' behavior. Based on random exploration and improvement by exploitation, the BOA can solve complex problems.

In the BOA, a butterfly has its unique fragrance. Mathematically, the fragrance of the butterfly can be defined as follow:

$$pf_i = cI^a \quad (5)$$

where  $pf_i$  denotes the perceived magnitude of fragrance, which is the intensity of fragrance of the  $i^{th}$  butterfly, detected by other butterflies,  $I$  represents the fragrance intensity,  $c$  represents the sensor modality, and the power exponent  $a$  accounts for the varying degree of absorption and dependent on modality. The BOA, consists of three phases as follow:

Phase 1: Initialization phase. This phase consists of three steps:

- Step 1: The values are assigned for the algorithm's parameters.
- Step 2: The fitness function and its solution space are defined
- Step 3: An initial population of butterflies is generated.

Phase 2: Iteration phase. In this phase, the search is performed by the algorithm with the artificial butterflies generated in the initialization phase. This phase consists of the following steps:

- Step 1: Each butterfly produces fragrance at its position and can be calculated by Eq. (5).
- Step 2: The fitness value of each butterfly in the search space is computed.

- Step 3: In the current iteration, the best butterfly/solution  $X^*$  is found among all the solutions.
- Step 4: Initialize random number  $r$ , where  $r \in [0, 1]$ .
- Step 5: The position of butterflies is updated. There are two critical strategies for updating butterflies' position, i.e., global search strategy and local search strategy. It must be a balance between the global and the local search procedures.
  - Global search algorithm: If the butterfly can sense the fragrance of the fittest butterfly/solution  $X^*$ , it moves toward that solution  $X^*$  which is given in Eq. (6)

$$x_i^{t+1} = x_i^t + (r^2 \times X^* - x_i^t) \times f_i \quad (6)$$

where the  $i$ th butterfly has the solution vector  $x_i$  in iteration  $t$  and denoted by  $x_i^t$ . The fragrance of  $i$ th butterfly is denoted by  $f_i$  and  $r$  is a random value within the range  $[0, 1]$ .

- Local search algorithm: In this process, the butterflies fail to feel the scent of the other butterflies, they randomly change their position in the search space. This process can be defined as follow.

$$x_i^{t+1} = x_i^t + (r^2 \times x_j^t - x_k^t) \times f_i \quad (7)$$

where  $x_j^t$  and  $x_k^t$  are  $j$ th and  $k$ th butterflies from the search space.

Phase 3: Termination phase. The iteration phase continues until the stopping criteria (Ex. maximum number of iterations, a specific value of error rate, etc) are reached.

## 4 Proposed Framework

Briefly, the proposed framework is done in four stages as shown in Fig. 1. First, the chosen and collected dataset crosses the pre-processing stage to make it suitable as input for the CNN network. Then, the pre-processed images are fed to the next stage to compute features from each input image. After that, the feature set go through the feature selection stage to choose the most relevant features. Finally, the selected features are forwarded to the classification model to decide which class these features belong to.

### 4.1 Pre-processing

The prepared dataset must pass through some pre-processing steps before fed to CNN pre-trained models. First, images are resized to the pre-trained model acceptable size, which is  $224 \times 224$  pixels in this experiment. And then are transformed to RGB with 24-bit depth.

### 4.2 CNN Feature Extraction Using Transfer Learning

Resnet18, Resnet50, and Densenet201 used for feature extraction in this work. We extract the features from the fully connected layer with 1000 features of the pre-trained CNN models. The full feature set obtained from each pre-trained CNN model is size  $n \times 1000$ , where  $n$  denotes the total number of X-ray images in the dataset.

### 4.3 Binary Butterfly Optimization Algorithm (bBOA) for Feature Selection

The binary optimization problems perform feature selection between the binary values  $\{0, 1\}$  only. These feature optimization problems are multi-objective optimization problems. These

algorithms' fundamental goals are to be achieved: selecting the smaller number of features and obtaining the maximum recognition accuracy. The best solution is obtained when the optimization algorithm achieves the best performance results with few features.

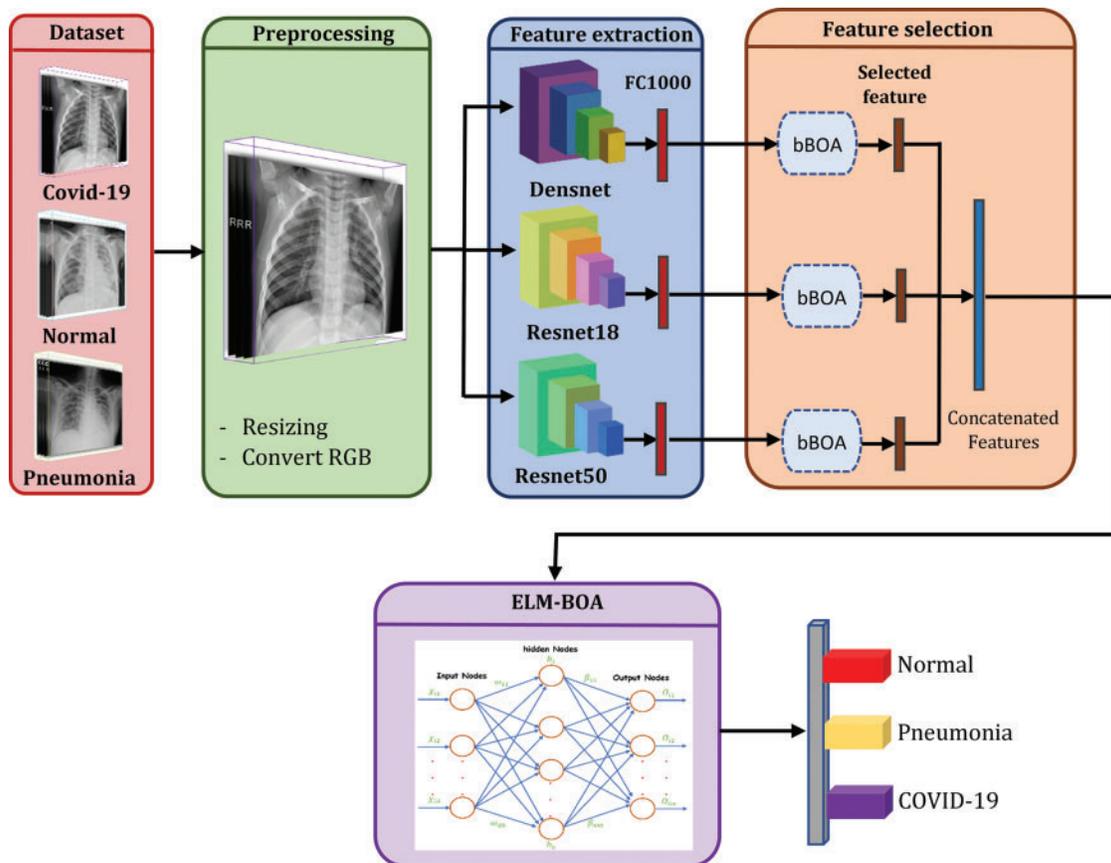


Figure 1: Proposed framework

In our proposed framework, the bBOA is utilized to select the most informative features to increase the recognition accuracy and reduces the computational time for Covid-19 prediction. Each feature subset in the bBOA is presented as a butterfly or solution. Each solution is specified as a single-dimensional vector, and the number of features in the dataset defined the dimension of the feature vector. The feature vector cells consist of two values, 1 or 0. The value 1 represents that the relevant features are selected, while the value 0 shows that the feature is not chosen.

Firstly, the butterflies are randomly initialized, and their fitness value can be computed by Eq. (5). The butterflies sense the fragrance, when a butterfly feel the fragrance of the best butterfly in the search space, the butterfly step towards the butterfly containing best traits, and this movement is given by Eq. (6). This phase is known as the global search. If the butterflies do not sense fragrance, this is known as the local search phase. Eq. (7) described the local search phase as random strides. After the local or global search phase, the butterflies change their positions and continuous solutions. The values of continuous solutions are converted into relative binary values. The transformation into binary values is carried out by implementing squashing of continuous

solutions. The squashing is performed utilizing a transfer function name Sigmoidal (S-shaped). The Sigmoidal function forcefully moves the butterflies in a binary search space. The Sigmoidal function is described in Eq. (8).

$$S\left(F_i^k(t)\right) = \frac{1}{1 + e^{-F_i^k(t)}} \quad (8)$$

where  $F_i^k$  representing the fragrance with a continuous value of  $i$ th butterfly in  $k$ th element during the iteration  $t$ . The transfer functions can fluently map the infinite input values to a finite output. The S-shaped transfer function produces the output continuously; therefore, a threshold is defined to obtain the binary-valued output. The applied threshold is given in Eq. (9) to obtain the binary solution.

$$x_i^k(t+1) = \begin{cases} 0 & \text{if } rand < S\left(F_i^k(t)\right) \\ 1 & \text{if } rand > S\left(F_i^k(t)\right) \end{cases} \quad (9)$$

where  $x_i^k(t)$  and  $F_i^k(t)$  designate the position and fragrance respectively, during the iteration  $t$  of  $i$ th butterfly in the  $k$ th element.

A fitness function is specified based on the KNN classifier to evaluate each solution. The fitness function measures the exactness to define the selected functions. Mathematically, the suggested fitness function is defined as.

$$Fitness = \alpha \times ERR(D) + \beta \frac{|R|}{|N|} \quad (10)$$

where  $ERR(D)$  represents the error rate (calculated using the KNN classifier),  $|N|$  represents the original feature set, and  $|R|$  shows the selected features. The parameters  $\alpha$  and  $\beta$  can be selected within the range  $[0, 1]$ .  $\alpha$  and  $\beta$  are the weights of error rate and the selection ratio respectively where  $\alpha$  is the complement of  $\beta$ .

In the proposed framework, the multi-CNN feature vectors after bBOA are combined to obtain a feature set. The moral behind the concatenation of feature sets is to exploit each CNN's capabilities in extracting useful features.

#### 4.4 Hybrid ELM-BOA Model for Classification

ELM model randomly initializes the input weights and hidden biases and systematically calculates the output weights using the MP inverse technique [23]. However, the assignment of random weights and biases, ELM has some limitations such as long training time and weak generalization ability. In this study, ELM is optimized to overcome these problems using the BOA, and a new hybrid model ELM-BOA is presented (Fig. 2). The BOA is mainly used to find the optimal set of weights and biases to improve the learning performance of ELM. In the ELM-BOA model's network structure consists of  $m$  inputs and  $K$  hidden nodes, the particle's length ( $L$ ) can be determined in Eq. (11).

$$L = m \times K + K \quad (11)$$

In this study, we select the minimization of the Root Mean Square Error (RMSE) value as a fitness function to assess each generated solution's performance by the BOA. The learning method of the ELM-BOA model is described as following:

- (1) A learning sample is set, including the input vector and the output vector.
- (2) The ELM-BOA neural network's topology is established as the number of inputs, hidden and output layer neurons are determined, and the activation function is selected.
- (3) Randomly generating the swarm of BOA butterflies. Each butterfly consists of the input weights and hidden biases, which are optimized and represent a candidate ELM. Here, the butterfly randomly initialized values for the elements within the range  $[-1, 1]$ .
- (4) Calculating the fitness value of each butterfly in the swarm. To evaluate the fitness value, the output weights of the ELM are generated using Eq. (4) after formulating the SLFN using the butterfly elements and computing the matrix  $D$  using Eq. (3).
- (5) Determining the best butterfly with the highest fitness  $X^*$ .
- (6) Applying the BOA algorithm to update the position of each butterfly using Eqs. (6) and (7).
- (7) The iteration is terminated when the algorithm is reached to the stopping criteria and output the best butterfly  $X^*$ . The ELM-BOA model is applied to test the obtained model's generalization performance; otherwise, this process again starts from Step 5.

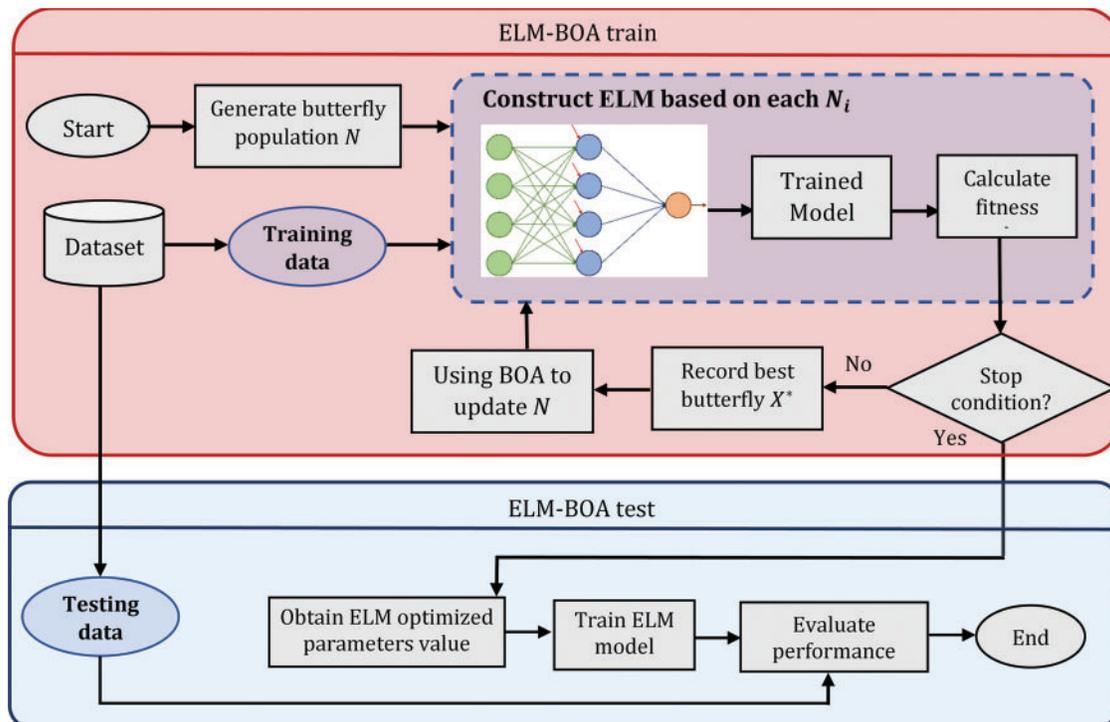


Figure 2: ELM-BOA model flowchart

## 5 Experiments

In this section, we present the used datasets, the parameter setting for methods, experimental design.

### 5.1 Datasets Description

In this experiment, the proposed model is trained and tested in a balanced CXR dataset composed of three different classes: normal, pneumonia, and COVID-19. Out of 3885 CXR images, there are 1295 COVID-19 images, 1295 normal images, and 1295 pneumonia images. Normal and pneumonia classes are extracted from one source: COVID-19 Radiography Database on the Kaggle website [60]. While the COVID-19 class, due to the disease's novelty, is obtained from various open-source image databases, it is organized as 800 already augmented images were collected from Alqudah et al. [61] and 495 images from Cohen et al. [62].

### 5.2 Parameter Settings

In this study, all experiments are carried out in MATLAB R2020a software on a PC with a 2.60 GHz CPU Intel (R) Core (TM) i7-4510U and 8 GB RAM, and Windows 10 (64 bit). The images dataset is split up into a 70:30 ratio, 70% for training and 30% for testing. The transfer learning approach was performed on the pre-trained deep networks to utilize them for a new task. The pre-trained CNN training parameters used in these experiments are presented in Tab. 1.

**Table 1:** CNN models parameters values

Software used	Model	Image size	Validation frequency	Max epochs	Mini batch	Learning rate
MATLAB	Resnet18 Resnet50 Densenet201	224*224*3	5	8	64	Le-4

The presented ELM-BOA performance is compared with SVM, KNN, and ELM. The parameter setting of ELM, GA, PSO, GWO, and BOA are shown in Tab. 2. The five algorithms all have 100 iterations. The average experimental results are obtained after running the algorithm 20 times.

**Table 2:** parameter settings

Algorithm	Parameter	values	Algorithm	Parameter	values
<b>ELM</b>	No of hidden layer neurons	30	<b>PSO</b>	C1 & C2	2
	Activation function	Sigmoid		Inertia weight	[0.2–0.8]
<b>GA</b>	Crossover rate	0.8	<b>GWO</b> <b>BOA</b>	$a$	[0, 2]
	Mutation rate	0.2		Power exponent ( $a$ )	0.1
	Selection	Roulette wheel		Sensory modality ( $c$ )	[0.01–0.25]
			Switch probability ( $p$ )	0.5	

### 5.3 Experiments Design

To evaluate the proposed automated COVID-19 detection and classification system, the following experiments were formulated:

- Experiment 1—The three CNN models were applied to the dataset to evaluate the classification performance.
- Experiment 2—The 1000 features from the FC1000 layer of each CNN model were extracted, and classification performance was evaluated using four machine learning classifiers and the proposed ELM-BOA model.
- Experiment 3—The BOA algorithm was applied to the feature set obtained from the FC1000 layer, combined different feature sets from CNN models, and evaluated performance.
- Experiment 4—The ELM-BOA classification performance was assessed and compared with the other optimized ELM model.

## 6 Results and Discussion

### 6.1 Performance Measures

The performance of the proposed COVID-19 detection and classification model is measured by computing the four major performance measures: true positives ( $TP$ ), false positives ( $FP$ ), true negatives ( $TN$ ), and false negatives ( $FN$ ). Following measures are utilized to evaluate the functioning of the proposed framework.

$Accuracy = \frac{TP + FN}{TP + TN + FP + FN}$	$Precision = \frac{TP}{TP + FP}$
$Sensitivity = \frac{TP}{TP + FN}$	$F-Score = \frac{2 \times TP}{2 \times TP + FP + FN}$
$Specificity = \frac{TN}{TN + FP}$	$FPR = \frac{FP}{TN + FP}$

### 6.2 Results

In this study, the evaluation of the proposed framework is demonstrated through four experiments. In the first experiment, the original three pre-trained CNN models were applied to classify the COVID-19 images. The features were extracted first from the dataset's images in each model, and then the SoftMax classifier was used to classify the image to a predefined class. [Tab. 3](#) presents the results for the experimental analysis. The best result was obtained using the descent201 model with an accuracy of 96.66 % and a training time of 670 min. As seen in [Tab. 3](#), the three CNN models' training and validation processes were completed with high accuracy; however, the training time was very high. In such a scheme, transfer learning can be used for feature extraction, leverage the power of CNNs, and reduce the computational costs, as illustrated in the next experiments.

In the second step, the dataset is processed by the three CNN models, and 1000 features were extracted from the FC1000 layer of each model. The results of the four machine learning models

and the proposed ELM-BOA model are shown in [Tab. 4](#). The proposed ELM-BOA classifier was superior to traditional ELM, SVM, and KNN machine learning methods. It was observed that the ELM-BOA model guarantees an enhancement in the automated detection task for COVID-19. Contrary to what was observed, the classification performance was significantly reduced when KNN and SVM performed the classification task. The best performance was obtained on the ELM-BOA classifier when that classifier fed with the Densenet201 features.

**Table 3:** Classification results for the three pre-trained CNN models and training time

	Accuracy (%)	Sensitivity (%)	Precision (%)	F1-Score (%)	Specificity (%)	FPR	Time (min)
Resnet18	96.05	96.18	96.10	96.13	98.01	0.0199	51
Resnet50	96.48	96.58	96.59	96.58	98.22	0.0178	288
Densenet201	96.66	96.72	96.77	96.74	98.30	0.0170	670

**Table 4:** Classification results by applying CNN models with four ML classifiers and ELM-BOA

Experiments		Classifier	Measures					FPR
CNN Models	# of feature (bBOA)		Accuracy	Sensitivity	Precision	F1-score	Specificity	
Resnet18	1000	SVM	95.7	95.80	95.80	95.80	97.83	0.0217
		KNN	93.9	94.03	94.13	94.07	96.90	0.0310
		ELM	95.79	95.90	95.98	95.93	97.85	0.0215
		ELM-BOA	97.08	97.14	97.16	97.14	98.52	0.0148
Resnet50	1000	SVM	96.65	96.75	96.74	96.68	98.33	0.0167
		KNN	93.91	94.00	94.18	94.08	96.90	0.0310
		ELM	95.88	95.97	95.79	95.91	97.95	0.0205
		ELM-BOA	96.82	96.87	96.92	96.89	98.39	0.0161
DenseNet201	1000	SVM	97.08	97.15	97.15	97.14	98.53	0.0147
		KNN	96.05	96.15	96.21	96.18	97.99	0.0201
		ELM	97.08	97.11	97.14	97.13	98.52	0.0148
		ELM-BOA	97.42	97.56	97.57	97.55	98.68	0.0132

In the third experiment, the bBOA method was utilized to select the most relevant features. As compared to the results in [Tabs. 4–6](#) show that the bBOA produces better classification results with fewer features. The best performance was ensured by combining the three deep CNN models, as shown in [Tab. 6](#). The 601 selected features achieved the best results. The maximum recognition rate was 99.48% computed by the ELM-BOA classifier.

In the fourth experiment, we used the original dataset to analyze the presented framework. 30% data was designated as the test data. By utilizing the bBOA technique, we combined the robust features extracted from the CNNs, and a new feature set is obtained with 601 features. The next process is the classification performed using the proposed ELM-BOA compared with ELM optimized with different metaheuristics algorithms such as Gray wolf optimization (GWO)

algorithm, Genetic algorithm (GA), and particle swarm optimization (PSO). The classification accuracy results are shown in Tab. 7. In this step, the best-achieved recognition rate is 99.48% on the ELM-BOA method with training time of 34 s.

**Table 5:** Classification results obtained by applying the bBOA method

Experiments		Classifier	Measures					
CNN Models	# of feature (bBOA)		Accuracy	Sensitivity	Precision	F1-score	Specificity	FPR
Resnet18	186	SVM	96.74	96.85	96.80	96.82	98.35	0.0165
		KNN	94.85	94.98	95.05	95.01	97.38	0.0262
		ELM	97.25	97.40	97.41	97.39	98.59	0.0141
		ELM-BOA	97.77	97.80	97.89	97.83	98.86	0.0114
Resnet50	240	SVM	97.08	97.14	97.16	97.14	98.52	0.0148
		KNN	94.59	94.68	94.85	94.75	97.25	0.0275
		ELM	97.08	97.16	97.11	97.13	98.53	0.0147
		ELM-BOA	97.77	97.83	97.85	97.83	98.87	0.0113
DenseNet201	175	SVM	97.25	97.32	97.37	97.33	98.61	0.0139
		KNN	96.22	96.31	96.38	96.34	98.08	0.0192
		ELM	97.34	97.37	97.34	97.40	98.65	0.0135
		ELM-BOA	98.11	98.14	98.20	98.16	99.04	0.0096

**Table 6:** The classification results obtained by combining features obtained from CNN models

Experiments		Classifier	Measures					
CNN Models	# of feature (bBOA)		Accuracy	Sensitivity	Precision	F1-score	Specificity	FPR
Resnet18 & Resnet50	426	SVM	97.77	97.78	97.83	97.80	98.87	0.0113
		KNN	95.02	95.10	95.22	95.16	97.47	0.0253
		ELM	96.47	96.82	96.91	96.83	98.34	0.0166
		ELM-BOA	98.28	98.38	98.37	98.35	99.13	0.0087
Resnet18 & DenseNet201	361	SVM	97.77	97.78	97.83	97.80	98.87	0.0113
		KNN	96.91	96.99	97.04	97.01	98.43	0.0157
		ELM	97.60	97.65	97.74	97.68	98.77	0.0123
		ELM-BOA	98.11	98.17	98.21	98.18	99.03	0.0097
Resnet50 & DenseNet201	415	SVM	97.85	97.90	97.94	97.91	98.91	0.0109
		KNN	95.11	95.17	95.36	95.25	97.51	0.0249
		ELM	97.42	97.53	97.53	97.53	98.68	0.0132
		ELM-BOA	98.03	98.11	98.12	98.12	98.99	0.0101
Resnet18 & Resnet50 & DenseNet201	601	SVM	97.94	98.00	98.03	98.01	98.95	0.0105
		KNN	95.36	95.44	95.59	95.50	97.64	0.0236
		ELM	97.94	98.02	98.04	98.01	98.95	0.0105
		ELM-BOA	<b>99.48</b>	99.49	99.46	99.47	99.75	0.0025

**Table 7:** ELM-BOA model performance compared with other optimized Elm models

Model	# No. of features	Accuracy	Time (s)
ELM-GA	601	97.52	51.6
ELM-PSO	601	98.11	48.5
ELM-GWO	601	98.02	59.3
ELM-BOA	601	99.48	34.42

### 6.3 Discussion

The COVID-19 has infected several people. But the large-scale labeled databases still not available. Different datasets are combined to perform the computational work on the automated COVID-19 detection model. Recently, researchers concentrated on chest X-ray images to develop the automated systems for the clinical assessment of COVID-19 disease. Several COVID-19 computational models have been introduced based on CNN models. As compared to traditional machine learning techniques, CNN-based models perform well in terms of efficiency and accuracy. These models extract robust features and produce good results in the classification phase. In this respect, our proposed technique consists of innovative components.

In this work, we exploited the advantages of the CNN models' end-to-end learning scheme. We extracted the deep features and performed transfer learning on Resnet18, Resnet50, and Densenet201 models. After the selection of robust features, the final feature vector fed to the ELM-BOA classifier for the recognition of disease. For a fair comparison, we used a balanced dataset with the same number of images for normal, pneumonia, and COVID-19. Besides, we utilized the feature selection method (bBOA) to obtain the robust feature set that yields better results compared with other studies [Tab. 8]. We also improve the time efficiency and classification accuracy. The best classification accuracy achieved is 99.48%. We conclude that the second shape of transfer learning used in our system outperforms the first shape (which was tested in the first experiment) in terms of efficiency.

**Table 8:** Comparison with some previous studies

Study	Methodology/technique	# of classes	Acc (%)
Apostolopoulos et al. [37]	MobileNet v2 model	3	93.48
Gao et al. [44]	Dual-branch combination network (DCN)	3	96.74
Ucar et al. [46]	COVIDiagnosis-Net	3	98.26
Rahimzadeh et al. [47]	Modified CNN: concatenation of Xception and ResNet50V2	3	99.5
Nour et al. [49]	CNN models: A novel CNN model, training from scratch strategy, deep feature extraction, Classifier: SVM	3	98.97
Proposed Model	CNN models: Concatenation of Resnet18, Resnet50 & Densenet201, Feature selection, optimized ELM model	3	99.48

## 7 Conclusion

This study's main objective is to establish an accurate and rapid AI-diagnostic method that can categorize patients into COVID-19 or regular or pneumonia. We use multi-class classification to decide whether the respiratory infections are caused by coronavirus or other viruses (pneumonia). Consequently, the hospital's workload will be reduced significantly. We aimed to have an equal number of images in each class in the dataset, which improved our proposed model's robustness and effectiveness. We based our model on multi-CNN, which concatenate deep features resulting from each network after passing a feature selection step using a butterfly optimization algorithm. Then, we used the optimized ELM model (ELM-BOA) as a classifier due to its ability to learn and modify weights, which led to a decrease in the error between actual and predicted output and achieves an accuracy of 99.48%. To prevent overfitting, 5-fold cross-validation is used. It is evident from experimental results that the proposed methodology outperforms competitive techniques, as shown in Tab. 7. The drawback of this approach is that when the patient is in a critical condition and is unable to attend X-ray scanning. In future work, we aim to develop our proposed model as a mobile application to increase reliability and availability.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] World Health Organization, "Novel Coronavirus (2019-nCoV) Situation Report-11. Novel Coronavirus (2019-nCoV): Situation report, 11 (who.int)," 2020. [Online]. Available: <https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200405-sitrep-76-covid-19.pdf> [Accessed 10 March 2020].
- [2] T. Lancet, "COVID-19: Too little, too late?," *Lancet (London, England)*, vol. 395, no. 10226, pp. 755, 2020.
- [3] T. B. Chandra and K. Verma, "Pneumonia detection on chest X-ray using machine learning paradigm," in *Proc. of 3rd Int. Conf. on Computer Vision and Image Processing*, Singapore, Springer, pp. 21–33, 2020.
- [4] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciampi *et al.*, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, no. 13, pp. 60–88, 2017.
- [5] J. Ker, L. Wang, J. Rao and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2017.
- [6] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [7] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [8] Ö. Yildirim, P. Pławiak, R.-S. Tan and U. R. Acharya, "Arrhythmia detection using deep convolutional neural network with long duration ECG signals," *Computers in Biology and Medicine*, vol. 102, no. 4, pp. 411–420, 2018.
- [9] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn *et al.*, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature Medicine*, vol. 25, no. 1, pp. 65–69, 2019.
- [10] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.
- [11] N. C. Codella, Q.-B. Nguyen, S. Pankanti, D. A. Gutman, B. Helba *et al.*, "Deep learning ensembles for melanoma recognition in dermoscopy images," *IBM Journal of Research and Development*, vol. 61, no. 4/5, pp. 5:1–5:15, 2017.

- [12] M. Talo, O. Yildirim, U. B. Baloglu, G. Aydin and U. R. Acharya, "Convolutional neural networks for multi-class brain disease detection using MRI images," *Computerized Medical Imaging and Graphics*, vol. 78, no. 1, pp. 101673, 2019.
- [13] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest X-rays with deep learning," arXiv preprint arXiv: 1711.05225, 2017.
- [14] J. H. Tan, H. Fujita, S. Sivaprasad, S. V. Bhandary, A. K. Rao *et al.*, "Automated segmentation of exudates, haemorrhages, microaneurysms using single convolutional neural network," *Information Sciences*, vol. 420, pp. 66–76, 2017.
- [15] F. Shariaty, S. Hosseinlou and V. Y. Rud, "Automatic lung segmentation method in computed tomography scans," *Journal of Physics: Conference Series*, vol. 1236, pp. 12028, 2019.
- [16] G. Mohan and M. M. Subashini, "MRI based medical image analysis: Survey on brain tumor grade classification," *Biomedical Signal Processing and Control*, vol. 39, pp. 139–161, 2018.
- [17] B. Abraham and M. S. Nair, "Computer-aided grading of prostate cancer from MRI images using convolutional neural networks," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 3, pp. 2015–2024, 2019.
- [18] B. Abraham and M. S. Nair, "Automated grading of prostate cancer using convolutional neural network and ordinal class classifier," *Informatics in Medicine Unlocked*, vol. 17, no. 1, pp. 100256, 2019.
- [19] S. P. Singh, L. Wang, S. Gupta, B. Gulyás and P. Padmanabhan, "Shallow 3D CNN for detecting acute brain hemorrhage from medical imaging sensors," *IEEE Sensors Journal*, 2020.
- [20] M. Doan, M. Case, D. Masic, H. Hennig, C. McQuin *et al.*, "Label-free leukemia monitoring by computer vision," *Cytometry Part A*, vol. 97, no. 4, pp. 407–414, 2020.
- [21] M. Toğaçar, B. Ergen and Z. Cömert, "COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches," *Computers in Biology and Medicine*, vol. 121, pp. 103805, 2020.
- [22] H. Mukherjee, S. Ghosh, A. Dhar, S. Obaidullah, K. Santosh *et al.*, "Shallow convolutional neural network for COVID-19 outbreak screening using chest X-rays," *Cognitive Computation*, pp. 1–14, 2020.
- [23] I. M. El-Hasnony, S. I. Barakat and R. R. Mostafa, "Optimized ANFIS model using hybrid meta-heuristic algorithms for Parkinson's disease prediction in IoT environment," *IEEE Access*, vol. 8, pp. 119252–119270, 2020.
- [24] I. M. El-Hasnony, S. I. Barakat, M. Elhoseny and R. R. Mostafa, "Improved feature selection model for big data analytics," *IEEE Access*, vol. 8, pp. 66989–67004, 2020.
- [25] G.-B. Huang, L. Chen and C. K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.
- [26] D. Cui, G.-B. Huang and T. Liu, "ELM based smile detection using distance vector," *Pattern Recognition*, vol. 79, no. 5, pp. 356–369, 2018.
- [27] J. Cao, Z. Lin and G.-B. Huang, "Self-adaptive evolutionary extreme learning machine," *Neural Processing Letters*, vol. 36, no. 3, pp. 285–305, 2012.
- [28] H. Liu, X. Mi and Y. Li, "An experimental investigation of three new hybrid wind speed forecasting models using multi-decomposing strategy and ELM algorithm," *Renewable Energy*, vol. 123, no. 1, pp. 694–705, 2018.
- [29] H. Zhu, E. C. Tsang and J. Zhu, "Training an extreme learning machine by localized generalization error model," *Soft Computing*, vol. 22, no. 11, pp. 3477–3485, 2018.
- [30] P. Mohapatra, S. Chakravarty and P. K. Dash, "An improved cuckoo search based extreme learning machine for medical data classification," *Swarm and Evolutionary Computation*, vol. 24, no. 4, pp. 25–49, 2015.
- [31] P. Satapathy, S. Dhar and P. Dash, "An evolutionary online sequential extreme learning machine for maximum power point tracking and control in multi-photovoltaic microgrid system," *Renewable Energy Focus*, vol. 21, no. 1, pp. 33–53, 2017.

- [32] L.-L. Li, J. Sun, M.-L. Tseng and Z.-G. Li, "Extreme learning machine optimized by whale optimization algorithm using insulated gate bipolar transistor module aging degree evaluation," *Expert Systems with Applications*, vol. 127, no. 1, pp. 58–67, 2019.
- [33] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Computing*, vol. 23, no. 3, pp. 715–734, 2019.
- [34] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Expert Systems with Applications*, vol. 116, no. 3, pp. 147–160, 2019.
- [35] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [36] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.
- [37] I. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," *Physical and Engineering Sciences in Medicine*, vol. 43, no. 2, pp. 635–640, 2020.
- [38] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang *et al.*, "Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT," *Radiology*, vol. 296, no. 2, pp. E65–E71, 2020.
- [39] B. Ghoshal and A. Tucker, "Estimating uncertainty and interpretability in deep learning for coronavirus (COVID-19) detection," arXiv preprint arXiv: 2020.10769, 2003.
- [40] A. Narin, C. Kaya and Z. Pamuk, "Automatic detection of coronavirus disease (covid-19) using X-ray images and deep convolutional neural networks," arXiv preprint arXiv: 2020.10849, 2003.
- [41] E. E.-D. Hemdan, M. A. Shouman and M. E. Karar, "Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in X-ray images," arXiv preprint arXiv: 2020.11055, 2003.
- [42] S. Wang, B. Kang, J. Ma, X. Zeng, M. Xiao *et al.*, "A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19)," *European Radiology*, 2021.
- [43] S. Toraman, T. B. Alakus and I. Turkoglu, "Convolutional capsnet: A novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks," *Chaos Solitons & Fractals*, vol. 140, no. 18, pp. 110122, 2020.
- [44] K. Gao, J. Su, Z. Jiang, L.-L. Zeng, Z. Feng *et al.*, "Dual-branch combination network (DCN): Towards accurate diagnosis and lesion segmentation of COVID-19 using CT images," *Medical Image Analysis*, vol. 67, pp. 101836, 2021.
- [45] M. J. Horry, S. Chakraborty, M. Paul, A. Ulhaq, B. Pradhan *et al.*, "COVID-19 detection through transfer learning using multimodal imaging data," *IEEE Access*, vol. 8, pp. 149808–149824, 2020.
- [46] F. Ucar and D. Korkmaz, "COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images," *Medical Hypotheses*, vol. 140, no. 1122–1131, pp. 109761, 2020.
- [47] M. Rahimzadeh and A. Attar, "A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2," *Informatics in Medicine Unlocked*, vol. 19, no. 6, pp. 100360, 2020.
- [48] W. M. Shaban, A. H. Rabie, A. I. Saleh and M. Abo-Elsoud, "A new COVID-19 Patients Detection Strategy (CPDS) based on hybrid feature selection and enhanced KNN classifier," *Knowledge-Based Systems*, vol. 205, no. 1, pp. 106270, 2020.
- [49] M. Nour, Z. Cömert and K. Polat, "A novel medical diagnosis model for COVID-19 infection detection based on deep features and Bayesian optimization," *Applied Soft Computing*, vol. 97, no. 5, pp. 106580, 2020.
- [50] J. Choe, S. M. Lee, K.-H. Do, G. Lee, J.-G. Lee *et al.*, "Deep learning-based image conversion of CT reconstruction kernels improves radiomics reproducibility for pulmonary nodules or masses," *Radiology*, vol. 292, no. 2, pp. 365–373, 2019.
- [51] M. A. Wani, F. A. Bhat, S. Afzal and A. I. Khan, *Advances in deep learning*. Berlin: Springer, 2020.
- [52] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

- [53] R. Altman, "Artificial intelligence (AI) systems for interpreting complex medical datasets," *Clinical Pharmacology & Therapeutics*, vol. 101, no. 5, pp. 585–586, 2017.
- [54] S. Mittal and S. Vaishay, "A survey of techniques for optimizing deep learning on GPUs," *Journal of Systems Architecture*, vol. 99, no. 4, pp. 101635, 2019.
- [55] S. Koitka and C. M. Friedrich, "Traditional feature engineering and deep learning approaches at medical classification task of ImageCLEF 2016," *CLEF (Working Notes)*, Portugal, pp. 304–317, 2016.
- [56] A. Kumar, J. Kim, D. Lyndon, M. Fulham and D. Feng, "An ensemble of fine-tuned convolutional neural networks for medical image classification," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 31–40, 2016.
- [57] G.-B. Huang, Q.-Y. Zhu and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [58] W. Cao, X. Wang, Z. Ming and J. Gao, "A review on neural networks with random weights," *Neurocomputing*, vol. 275, no. 5786, pp. 278–287, 2018.
- [59] W. Cao, J. Gao, Z. Ming and S. Cai, "Some tricks in parameter selection for extreme learning machine," in *IOP Conf. Series: Materials Science and Engineering*, pp. 12002, 2017.
- [60] M. E. H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir *et al.*, "Can AI help in screening Viral and COVID-19 pneumonia?," *IEEE Access*, vol. 8, pp. 132665–132676, 2020.
- [61] A. M. Alqudah and S. Qazan, "Augmented COVID-19 X-ray images dataset," *Mendeley Data*, V4, 2020. [Online]. Available <https://data.mendeley.com/datasets/2fxz4px6d8/4?fbclid=IwAR1Kn8H0GihLcOnxft-cRR-fsNgyTe8BcEjY0hF27j1-KgY33zXEgtPWztfw>.
- [62] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong *et al.*, "Image data collection: Prospective predictions are the future," arXiv: 2006.11988, 2006. [Online]. Available: <https://github.com/ieee8023/covid-chestxray-dataset>.