Applied Intelligence (2021) 51:1351–1366
https://doi.org/10.1007/s10489-020-01904-z

OptCoNet: an optimized convolutional neural network for an automatic diagnosis of COVID-19

Tripti Goel[1] · R. Murugan[1] · Seyedali Mirjalili[2] · Deba Kumar Chakrabartty[3]

Published online: 21 September 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

The quick spread of coronavirus disease (COVID-19) has become a global concern and affected more than 15 million confirmed patients as of July 2020. To combat this spread, clinical imaging, for example, X-ray images, can be utilized for diagnosis. Automatic identification software tools are essential to facilitate the screening of COVID-19 using X-ray images. This paper aims to classify COVID-19, normal, and pneumonia patients from chest X-ray images. As such, an Optimized Convolutional Neural network (OptCoNet) is proposed in this work for the automatic diagnosis of COVID-19. The proposed OptCoNet architecture is composed of optimized feature extraction and classification components. The Grey Wolf Optimizer (GWO) algorithm is used to optimize the hyperparameters for training the CNN layers. The proposed model is tested and compared with different classification strategies utilizing an openly accessible dataset of COVID-19, normal, and pneumonia images. The presented optimized CNN model provides accuracy, sensitivity, specificity, precision, and F1 score values of 97.78%, 97.75%, 96.25%, 92.88%, and 95.25%, respectively, which are better than those of state-of-the-art models. This proposed CNN model can help in the automatic screening of COVID-19 patients and decrease the burden on medicinal services frameworks.

Keywords Automatic diagnosis · Coronavirus · COVID-19 · Convolutional neural network · Grey wolf optimizer · Stochastic gradient descent

1 Introduction

Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) is a novel virus that is enveloped with a large, single-stranded RNA genome. It emerged in Wuhan, China in December 2019 and caused the greatest pandemic of the millennium [1]. According to the World Health Organization (WHO) report, the total number of people infected by the disease as of 27 July 2020 is 16,114,449 with 646,641 deaths. The typical symptoms of the disease are fever, breathlessness, cough, fatigue, and loss of taste and smell [2]. The standard method for diagnosing COVID-19 is reverse transcription-polymerase chain reaction from a nasopharyngeal swab. Even though the continuous polymerase chain reaction examination of the sputum has the best quality for detecting COVID-19, the time required to confirm COVID-19 in infected patients can be high given the elevated false positive results of the examination [3]. Therefore, clinical imaging modalities, for example, chest X-ray (CXR), can play an important role in diagnosing individuals where there is a high doubt of infection according to symptoms and hazard factors, with the exception of pregnant women and children outside of emergency situations [4]. CXR images have been explored in the ongoing pandemic for detecting COVID-19.

Chest computed tomography (CT) imaging may likewise aid in the diagnosis of COVID-19; however, current guidelines do not suggest utilizing CT imaging for routine screening. Hence radiologists recommend CXR for the diagnosis of COVID-19. The advantage of X-ray machines is that most radiological laboratories and hospitals are capable of acquiring 2-dimensional projection images of the patient's chest. In general, CXR images play a vital role for radiologists in perceiving the chest pathology and have been applied in the confirmation or recommendation of COVID-19 in infected

✉ R. Murugan
murugan.rmn@ece.nits.ac.in

[1] Department of Electronics and Communication Engineering, National Institute of Technology Silchar, Silchar, Assam 788010, India

[2] Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Fortitude Valley, Brisbane, QLD 4006, Australia

[3] Department of Radiology, Silchar Medical College and Hospital, Silchar, Assam 788014, India

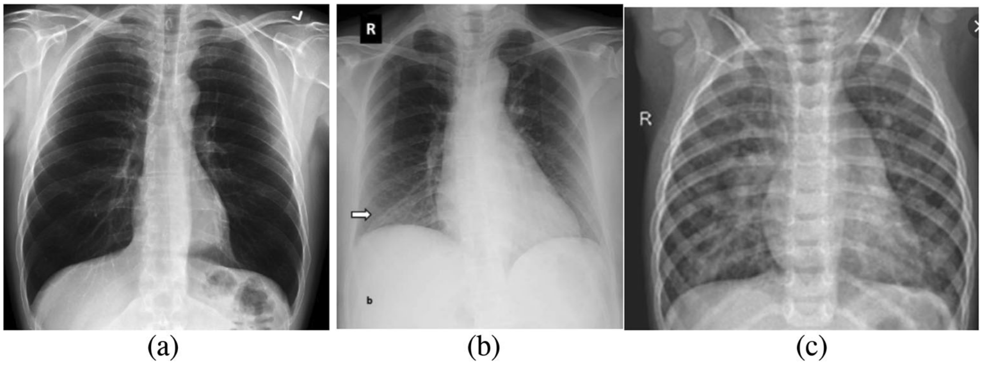


Fig. 1 Sample CXR images **a** Normal **b** COVID-19 **c** Pneumonia

patients [5]. Figure 1a, b, and c show sample CXR images of a normal, COVID-19, and pneumonia patient, respectively [6].

Computer-aided diagnosis (CAD) frameworks assist with a rapid, automatic diagnosis using graphical processing units (GPUs) by processing medical images. To the best of our knowledge, deep learning (DL) architecture has been used in many CAD frameworks and in numerous medical imaging applications, such as for COVID-19 [7]. In recent years, the convolutional neural network (CNN) has yielded the most promising results in classifying radiological images. CNNs are DL algorithms and have been used in many applications, including image classification. These advantages motivated our attempt to propose a CNN algorithm for COVID-19 diagnosis in this paper.

The hyperparameters of CNNs have an important influence on the network's performance, as they directly control the training process. The selection of appropriate hyperparameters plays a vital role in the training of the CNN network. For example, if the learning rate is too low, the network may lose important details in the data. By contrast, if the learning rate is too high, it may lead the model to converge too quickly. Therefore, there is a need to optimize the hyperparameters of CNNs for proper training and optimum performance results.

The novelty of this work is twofold. First, we optimized the hyperparameters of a CNN using the Grey Wolf Optimization algorithm according to CXR images. In addition, the proposed method avoids the overfitting of input images and real-time images with better performance metrics.

The details of the contributions are as follows:

I. Hyperparameters of the CNN are optimized using Grey Wolf Optimization to determine the best accurate results in diagnosing COVID-19, normal, and pneumonia patients from CXR images.
II. Grey Wolf Optimization is compared with other optimization algorithms for tuning the CNN's hyperparameters.
III. A CNN model is proposed for the automatic diagnosis of COVID-19 using CXR images with an accuracy of 97.78%.

The rest of this paper is arranged as follows: Related works are introduced in Section 2. The theoretical and mathematical fundamentals are given in Section 3. The material and methodology are presented in Section 4. The experiments are given in Section 5. Finally, the conclusion is presented in Section 6.

2 Related works

Recently, AI-based techniques for identifying and tracing COVID-19 have been popular in the global attempt to end the pandemic. This section briefly covers ongoing attempts at determining a COVID-19 diagnosis utilizing CXR images. Since uncovering the possibility of using CXR images in recognizing COVID-19 and the shortcomings of manual detection, there have been a number of investigations attempting to create automatic COVID-19 classification frameworks,

Table 1 Dataset used in this study

Dataset	Images category	No. of images	Reference
Chest imaging	COVID-19	134	[21]
COVID-19	COVID-19	64	[22]
Covid-chestxray	COVID-19	646	[23]
Figure 1 COVID-19 Chest X-ray	COVID-19	55	[24]
Provincial peoples hospital	COVID-19	1	[25]
Kaggle	Normal	900	[26]
Kaggle	Pneumonia	900	[26]

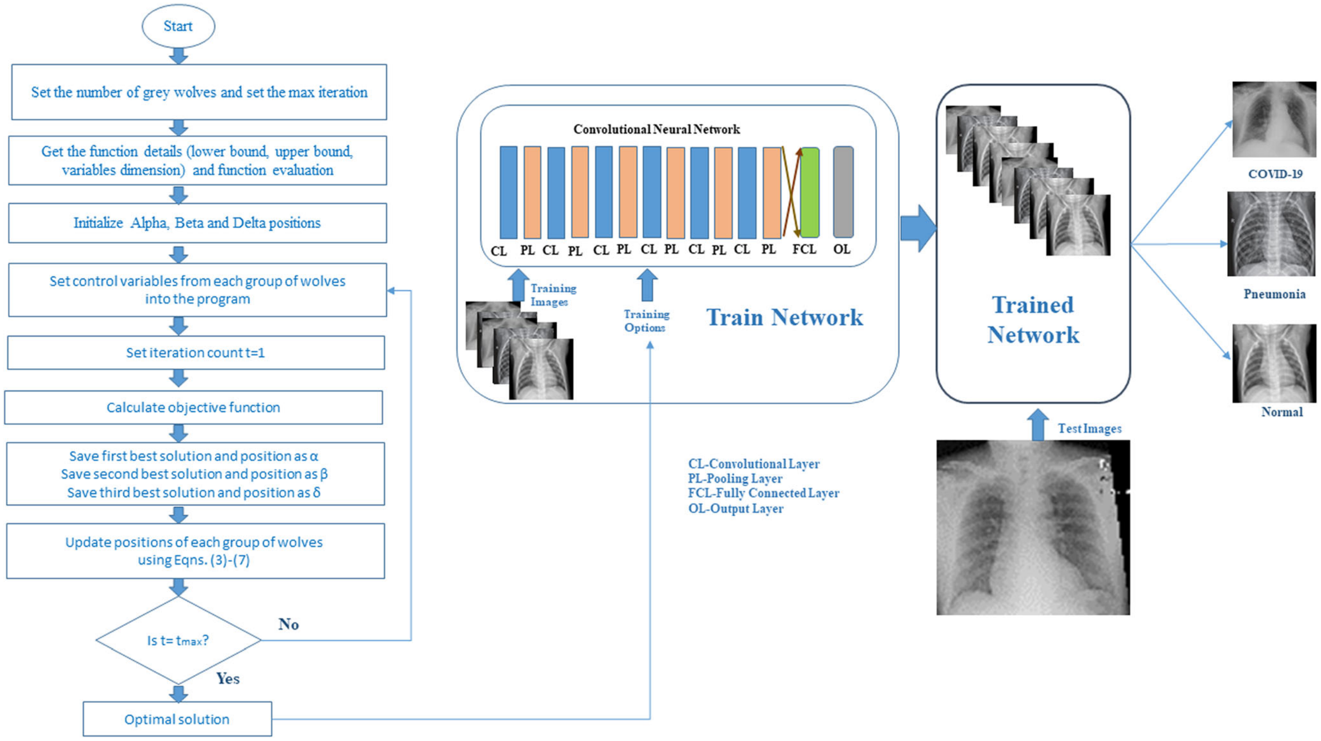

Start
Set the number of grey wolves and set the max iteration
Get the function details (lower bound, upper bound, variables dimension) and function evaluation
Initialize Alpha, Beta and Delta positions
Set control variables from each group of wolves into the program
Set iteration count t=1
Calculate objective function
Save first best solution and position as α
Save second best solution and position as β
Save third best solution and position as δ
Update positions of each group of wolves using Eqns. (3)-(7)
Is t= tmax?
No
Yes
Optimal solution
Convolutional Neural Network
CL PL CL PL CL PL CL PL CL PL CL PL FCL OL
Training Images
Training Options
Train Network
Trained Network
Test Images
COVID-19
Pneumonia
Normal
CL-Convolutional Layer
PL-Pooling Layer
FCL-Fully Connected Layer
OL-Output Layer

Fig. 2 Flow diagram of grey wolf optimized CNN for COVID-19 diagnosis

mainly utilizing CNNs. Narin et al. [8] introduced a study of three distinctive CNN models, including InceptionV3, ResNet50, and Inception-ResNetV2, that were proposed for the recognition of COVID-19 utilizing CXR images.

Hemdan et al. [9] presented COVIDX-Net, which incorporates seven unique designs of deep convolutional neural network (DCNN) including an improved VGG19 and the second form of Google MobileNet. Each DCNN model can inspect the normalized forces in CXR images to portray the patient status as either negative or positive COVID-19. The DenseNet and VGG19 models were capable of automatically detecting COVID-19. Khan et al. [10] proposed CoroNet, a DCNN model, to perform image processing on X-ray images and classify them positively or negatively. Li et al. [11] demonstrated a mobile-based lightweight DL network architecture, namely, COVID-MobileXpert.

Maghdid et al. [12] proposed an AI model for radiologists and health experts to analyze COVID-19 cases quickly and accurately. This involved constructing an extensive dataset of CXR and CT images from various sources and developed a necessary yet compelling COVID-19 identification procedure utilizing DL and TL techniques. Mahdy et al. [13] introduced a strategy for identifying COVID-19-infected persons utilizing CXR images. Multilevel thresholding and a support vector machine (SVM) were introduced to achieve high accuracy with images of the infected lungs of COVID-19 patients. Rehman et al. [14] utilized pretrained information to improve demonstrative execution using TL methods and performed a comparative analysis with different CNN structures.

Abbas et al. [15] validated and adapted their recently created Decompose, Transfer, and Compose (DeTraC) CNN for the diagnosis of COVID-19 with CXR images. DeTraC can manage any anomalies in the image dataset by researching its class limits utilizing a class disintegration system. The TL was used for tweaking the parameters.

Table 2 Architecture of proposed CNN

Layer	Type	Filter size	No .of filters	stride
Input	224x224x3			
Conv_1	CL + BN + ReLu	3 × 3	64	1 × 1
MPL_1		2 × 2		2 × 2
Conv_2	CL + BN + ReLu	3 × 3	64	1 × 1
MPL_2		2 × 2		2 × 2
Conv_3	CL + BN + ReLu	3 × 3	32	1 × 1
MPL_3		2 × 2		2 × 2
Conv_4	CL + BN + ReLu	3 × 3	16	1 × 1
MPL_5		2 × 2		2 × 2
Conv_5	CL + BN + ReLu	3 × 3	8	1 × 1
Fully connected	Output size =3			
Output	Classification layer	Soft-max		

BN Batch Normalization Layer

Table 3 Training options using GWO optimization

Training algorithm	Momentum	Initial learning rate	Maximum epoch	Validation frequency	L2Regularization
SGD	0.6	0.015	10	30	1.0000e-04

Afshar et al. [16] introduced an elective demonstrating system dependent on capsule networks (rather than CNNs), alluded to as the COVID-CAPS, which was equipped to work with small datasets vitally because of the abrupt and fast development of COVID-19. Apostolopoulos et al. (2020) [17] depicted a method to assess the performance of best-in-class CNN models proposed over recent years for clinical image classification. More specifically, a methodology called TL was employed in their work.

Apostolopoulos et al. (2020) [18] additionally introduced the best-in-class CNN called Mobile Net. This was utilized and developed without any preparation to research the significance of the extricated features for the characterization of COVID-19. Farook et al. [19] built open source datasets for COVID-19. They introduced an exact CNN structure for separating COVID-19 from pneumonia cases. This work utilized the best training techniques, including dynamic resizing, TL, and discriminative learning, to train the CNN quickly and accurately using DL frameworks.

All the techniques on DL–based COVID-19 screening have so far used CNNs, which, despite their incredible image handling capability, are unable to identify the unique relations between image examples. Because of this disadvantage, CNNs cannot perceive items as similar when one has been rotated or subject to another sort of change. As a result, CNNs require large datasets, including all possible changes to which images can be subjected. Nonetheless, in clinical imaging settings, including COVID-19 screening, large datasets are most certainly not adequately available. The proposed Optimized Convolutional Neural Network (OptCoNet) is an alternative model capable of screening COVID-19 without a vast dataset since it captures spatial information coming from instances and object parts using potential changes in the existence of the objects.

3 Grey wolf optimizer

The Grey Wolf Optimizer (GWO) [20], a recently developed swarm intelligence (SI) algorithm, has proven to be a reliable optimization algorithm compared to conventional evolutionary and swarm-based algorithms. The gray wolf belongs to the Canidae family. It is considered a high-level predator and dwells at the top of the food chain. They live in a pack that comprises 5–12 wolves on the whole. In gatherings, an exact predominance order is maintained. The pack is driven by alphas and trailed by betas, the subordinate wolves who are mindful to help the alpha in maintaining the dynamics of the pack.

The beta wolf strengthens the alpha's orders all through the pack and offers input to the alpha. In the interim, the lowest rung among the gray wolves is the omega, who generally assumes the scapegoat's job. They are the last wolves allowed to eat from the prey. On the off chance that a wolf is not alpha, beta, or omega, the individual in question is known as a delta. Delta wolves act as scouts, sentinels, seniors, trackers, and guardians.

The motivation for proposing the GWO algorithm for COVID-19 diagnosis is twofold. First, it is a very competitive optimization algorithm. It has been applied to various research fields, such as feature selection, economic load dispatch problems, and flow scheduling problems. In addition, the GWO algorithm benefits from avoiding high local optima, which leads to avoidance of overlapping features in the problem of feature selection.

The main mechanisms of GWO, including social pecking order, following, circling, and attacking prey, are introduced as follows.

In the GWO algorithm, the fittest solution can be represented as the alpha (α), with the second and third-best

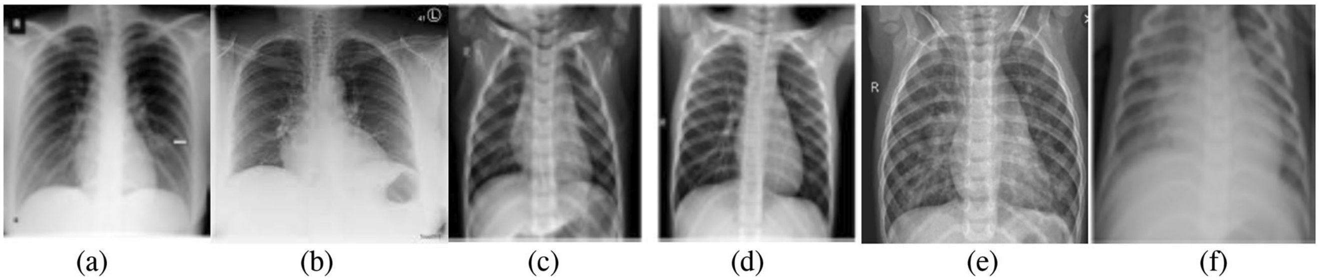


Fig. 3 Sample training images **a-b** COVID-19 **c-d** Normal **e-f** Pneumonia

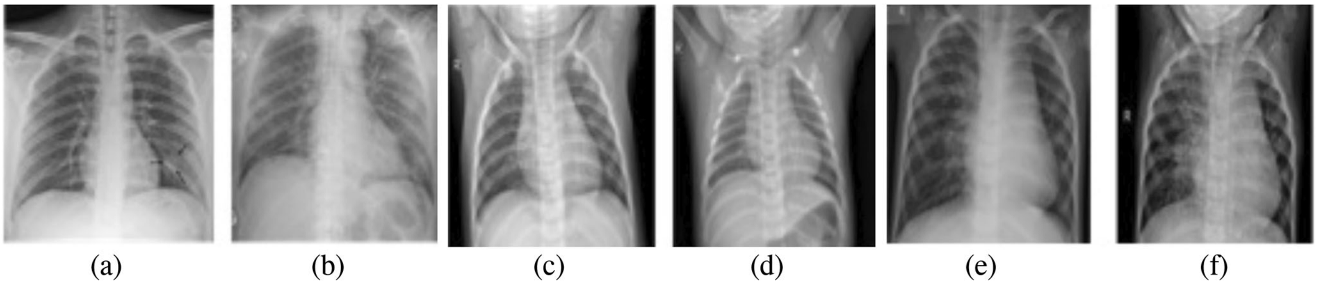


Fig. 4 Sample testing images **a**-**b** COVID-19 **c**-**d** Normal **e**-**f** Pneumonia

solutions represented as beta (β) and delta (δ), respectively. The remainder of the up-and-coming solutions are considered omegas (ω). GWO uses these simple principles to rank the solutions in each iteration and update their positions.

When wolves hunt, they will, in general, enclose their prey. The following mathematical equations delineate this encircling behavior

$$\mathrm{D} = |\mathrm{C.X_p(t)}-\mathrm{X(t)}| \tag{1}$$

$$\mathrm{X(t+1)} = \mathrm{X_p}-\mathrm{A.D} \tag{2}$$

where D denotes distance, t represents the present iteration, A and C are coefficients, X_p is the prey's location, and X denotes the position of a gray wolf.

The coefficients A and C are calculated as follows:

$$A = 2ar1-a \tag{3}$$

$$C = 2r2 \tag{4}$$

where r_1 and r_2 are random values between 0 and 1 and 'a' is a decreasing parameter.

As mentioned above, the three best solutions (alpha, beta, and delta) are updated first, and then, the other search agents (omega wolves) update their positions, all using the following equations:

$$\mathrm{D\alpha} = |\mathrm{C1X\alpha}-\mathrm{X}|, \mathrm{D\beta} = |\mathrm{C1X\beta}-\mathrm{X}|, \mathrm{D\delta} = |\mathrm{C1X\alpha}-\mathrm{X}| \tag{5}$$

$$X1 = X\alpha-A1.(D\alpha), X2 = X\beta-A1.(D\beta), X1 = X\delta-A1.(D\delta) \tag{6}$$

$$\mathrm{X(t+1)} = \frac{X_{1+} + X_{2+}X_3}{3} \tag{7}$$

To summarize, the optimization approach for GWO commence with making an arbitrary populace of grey wolves, which can be called applicants of solution. During the recreation, alpha, beta, and delta wolves gauge the conceivable situation of the prey. The parameter of 'a' in Eq. (3) can be demonstrated as investigation and abuse forms by diminishing the incentive from 2 to 0. Up-and-comer arrangements have separated from the prey if |A| > 1 and converged toward the prey |A| < 1. The GWO algorithm iteratively updates and evaluation candidate solutions until an end condition is met.

To summarize, the optimization approach for GWO begins with making an arbitrary populace of gray wolves, which can be called applicants of solution. During recreation, alpha, beta, and delta wolves gauge the conceivable situations of the prey. The parameter 'a' in Eq. (3) can represent investigation or abuse by decreasing the value from 2 to 0. Up-and-comer solutions indicate distancing from the prey if |A| > 1 and convergence toward the prey if |A| < 1. The GWO algorithm iteratively updates and evaluates candidate solutions until an end condition is met.

Table 4 Performance measures for COVID-19

Parameters	Formula
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity or recall	$\frac{TP}{TP+FN}$
Specificity	$\frac{TN}{TN+FP}$
Precision	$\frac{TP}{TP+FP}$
F1 Score	$2\,\frac{\text{Precision x Recall}}{\text{Precision+Recall}}$
True Positive Rate	$\frac{TP}{TP+FN}$
False Positive Rate	$\frac{FP}{FP+TN}$

4 Materials and methodology

4.1 Dataset

CXR images of COVID-19, normal, and pneumonia patients collected from publicly available repositories [21–26] were

Table 5 Experimental results of the proposed network model

Accuracy	Sensitivity	Specificity	Precision	F1 Score
97.78%	97.75%	96.25%	92.88%	95.25

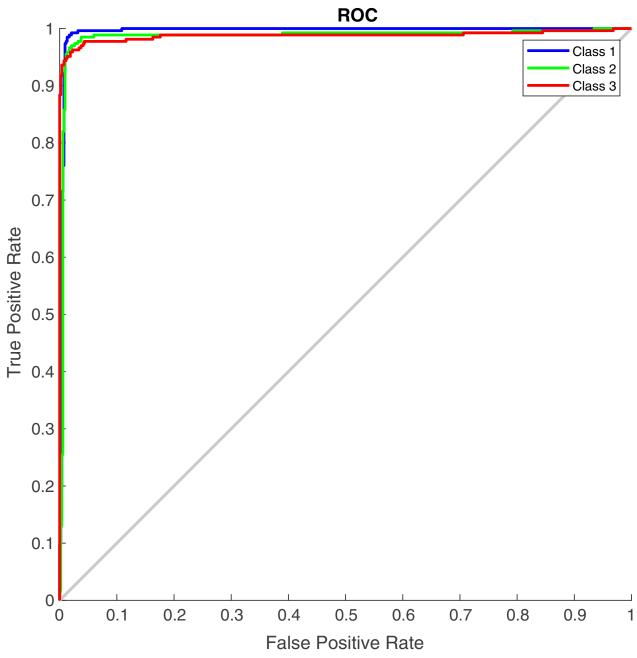


Fig. 5 Generated ROC curves of the proposed GWO optimized CNN (Class 1-COVID-19, Class 2- Normal, Class 3-Pneumonia)

used in this work. These repositories yielded a total of 2700 images, out of which 900 were COVID-19 images. The resolution of all images was set to 224x224x3 pixel size. The details of the images are stated in Table 1.

4.2 Workflow

A CNN is a type of DL model that extends the capability of the artificial neural network (ANN) by adding more layers to its architecture. CNNs have excellent feature extraction, pattern recognition, and classification performance of the raw input data without any preprocessing. The CNN architecture is divided into two parts: feature extraction and classification. Feature extraction is performed using several layers, including a convolutional layer (CL) followed by a max-pooling layer (MPL). Classification consists of a fully connected layer (FCL) and a classification layer to classify the input features into a particular class. CLs and FCLs are composed of weights and biases, which should be tuned using a gradient descent training algorithm. The training algorithm should also include many hyperparameters, which strongly reflect the CNN model's performance. These parameters are the training algorithm, momentum, learning rate, number of epochs, validation frequency, and L2Regularization. The current work is focused on optimizing these hyperparameters for training the CNN to yield optimal performance results. Although hyperparameters are very important for producing better performance, testing each hyperparameter is computationally expensive. Therefore, GWO is proposed in this architecture to optimize the hyperparameters for training the layers of the CNN. Figure 2 shows the flow diagram of the proposed network of hyperparameter optimization for a CNN using GWO. The description of the proposed model is discussed as follows.

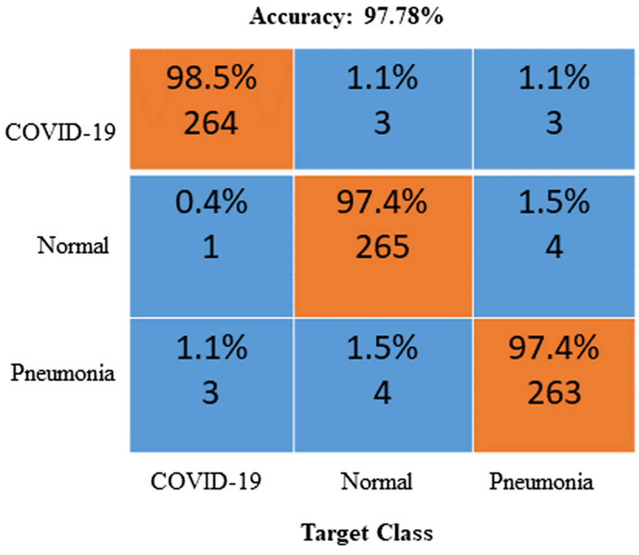


Fig. 6 Generated confusion matrix from the GWO optimized CNN

4.3 Convolutional neural network

The CNN architecture is composed of an input layer, CL, MPL, FCL, and output layer. The CL layer of the CNN is responsible for extracting features from the input images using several convolutional filters. These convolutional filters perform the convolutional operation at every offset of the input image. The CL contains weights that should be optimized using gradient descent training, which adjusts the parameters of the CL. The features extracted from the CL are mapped into feature space using a nonlinear rectified linear unit (ReLU) activation function. A batch normalization layer (BNL) is used between the CL and ReLU to normalize the gradients and activations through the network. The PL is used to reduce the dimensions of the feature maps obtained from the CL and retain the most relevant information of the image. Pooling methods include max-pooling and average pooling. There are no weights or biases in the PL of the CNN to train. The last layer is a fully connected classifier layer that classifies the extracted features from the CL and MPL into a particular class.

CNN training involves adjusting the parameters of the convolutional kernels and hidden neurons in the fully connected classification layer. Generally, CNNs utilize stochastic gradient descent (SGD) training to tune the CL and fully connected layer parameters. SGD minimizes the cost function by updating the weights of the network in the backward direction.

The disadvantage of using SGD is that it contains many hyperparameters that influence the network's performance. The next section explains the algorithm used to optimize these

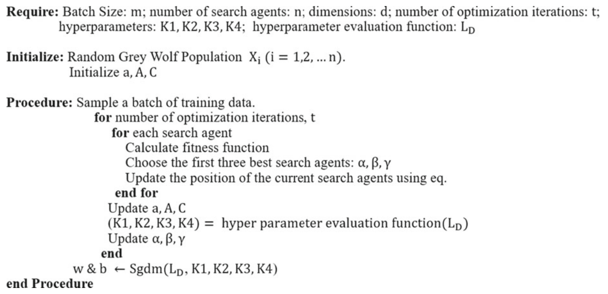
Table 6 Parameters of GWO

Batch Size	Number search Agent's	Dimensions	No. of optimizations iterations	Lower bound	Upper bound	Hyperparameter evaluation function
32	30	4	30	[0.5, 0.01, 5, 1.0000e-04]	[1, 0.5, 15, 5.0000e-04]	Error rate = (FP + FN)/ (TP + TN + FP + FN)

hyperparameters to achieve the best network performance. Details of the layers of the proposed CNN are given in Table 2.

4.4 Hyperparameter optimization using GWO

In this work, the hyperparameters are optimized using GWO. Hyperparameters play a crucial role in determining the accuracy and convergence of the CNN. Selecting the network's hyperparameters is essential and depends on the application for which the CNN is used. The learning rate, number of epochs, momentum, and regularization coefficient are the most common CNN training hyperparameters. The learning rate controls the gradient descent algorithm's speed, and the momentum controls the influence of the update of previous weights on the update of current weights. The number of epochs determines the number of times the learning algorithm will update the network parameters according to the training dataset. Regularization overcomes the issue of overfitting in the network. Therefore, to address all these settings, optimizing these hyperparameters is required to help the network yield the most accurate results. The GWO algorithm for the optimization of hyperparameters in SGD training is given in Algorithm (1).



```
Require: Batch Size: m; number of search agents: n; dimensions: d; number of optimization iterations: t;
         hyperparameters: K1, K2, K3, K4;  hyperparameter evaluation function: $L_D$

Initialize: Random Grey Wolf Population  $X_i$ $(i = 1,2, \dots n)$.
            Initialize a, A, C

Procedure: Sample a batch of training data.
           for number of optimization iterations, t
              for each search agent
                 Calculate fitness function
                 Choose the first three best search agents: $\alpha, \beta, \gamma$
                 Update the position of the current search agents using eq.
              end for
             Update a, A, C
             (K1, K2, K3, K4) =  hyper parameter evaluation function($L_D$)
             Update $\alpha, \beta, \gamma$
           end
      w & b $\leftarrow$ Sgdm($L_D$, K1, K2, K3, K4)
end Procedure
```

5 Experiments

To benchmark the performance of the proposed GWO-based CNN for COVID-19, normal, and pneumonia images, tests were conducted for classification using different optimization techniques.

5.1 Implementation details

The algorithm was implemented in MATLAB 2020a and executed using Windows 10 Pro with a 64 GB RAM Nvidia GPU. The proposed architecture was tested for both normal, phenomena, and COVID-19 CXR images using publicly available datasets.

5.1.1 Training

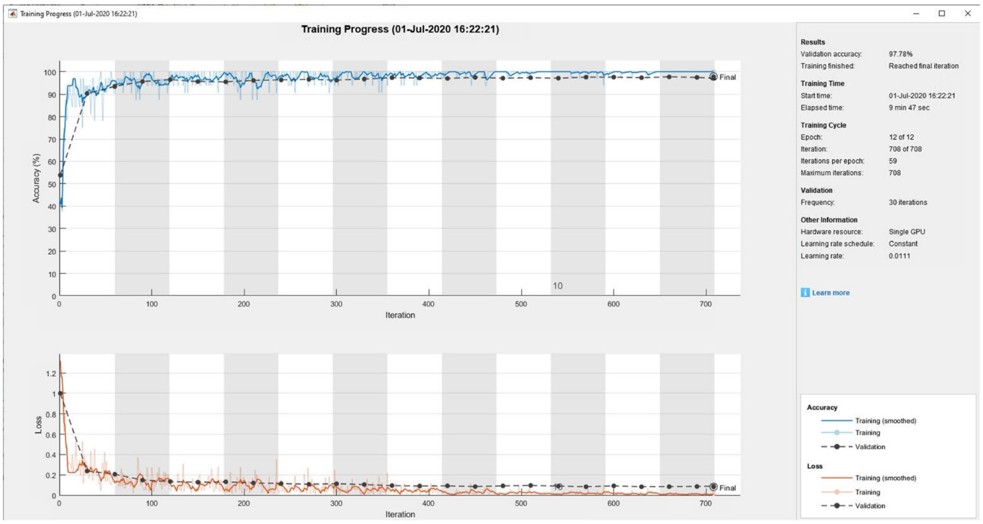
For the diagnosis of COVID-19 from CXR images, a six CL CNN is used. The A total of 70% of the data are used for training, and the remaining 30% of the data are used for testing the network. The proposed architecture classifies images into three categories: normal, pneumonia, and COVID-19. The dataset is partitioned randomly into training and test sets. All the images are resized to 224x224x3 using data augmentation and converted to color images. A six convolutional layer architecture is used to classify the images, and parameters are tuned using SGD training, whose hyperparameters are optimized using GWO at the time of training. The optimized hyperparameters obtained using GWO optimization are given in Table 3, and sample training images are provided in Fig. 3.

5.1.2 Testing

First, the images are resized to 224x224x3 using data augmentation and training the proposed network. After that, the test images are given as input to the trained CNN, where all the parameters of the CLs and FCLs are already optimized. Then, the CNN first extracts the images features and classifies them into the appropriate class using FCL and soft-max classifiers. The sample testing images are given in Fig. 4.

5.2 Performance indicators and evaluation metrics

This subsection presents the proposed methodology's performance in classifying the images into COVID-19, normal, and pneumonia. The proposed method is validated using the performance metrics of accuracy, sensitivity, specificity, precision, and F1-score, and receiver operating characteristic (ROC) analysis will be used for verifying the results from actual cases. The performance metrics equations are summarized in Table 4.where TP indicate True Positive; TN indicate



Results
Validation accuracy: 97.78%
Training finished: Reached final iteration

Training Time
Start time: 01-Jul-2020 16:22:21
Elapsed time: 9 min 47 sec

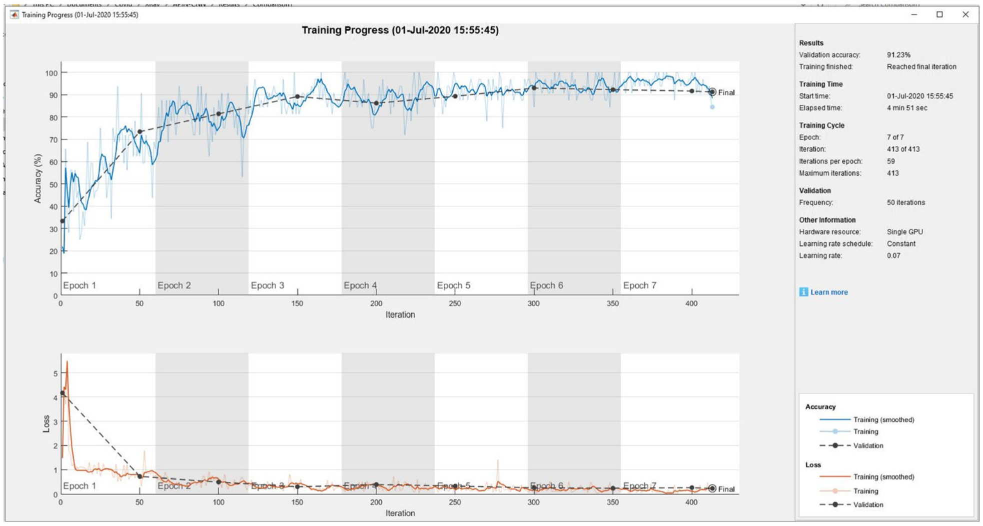
Training Cycle
Epoch: 12 of 12
Iteration: 708 of 708
Iterations per epoch: 59
Maximum iterations: 708

Validation
Frequency: 30 iterations

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.0111

Learn more

(a)



Results
Validation accuracy: 91.23%
Training finished: Reached final iteration

Training Time
Start time: 01-Jul-2020 15:55:45
Elapsed time: 4 min 51 sec

Training Cycle
Epoch: 7 of 7
Iteration: 413 of 413
Iterations per epoch: 59
Maximum iterations: 413

Validation
Frequency: 50 iterations

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.07

Learn more

(b)

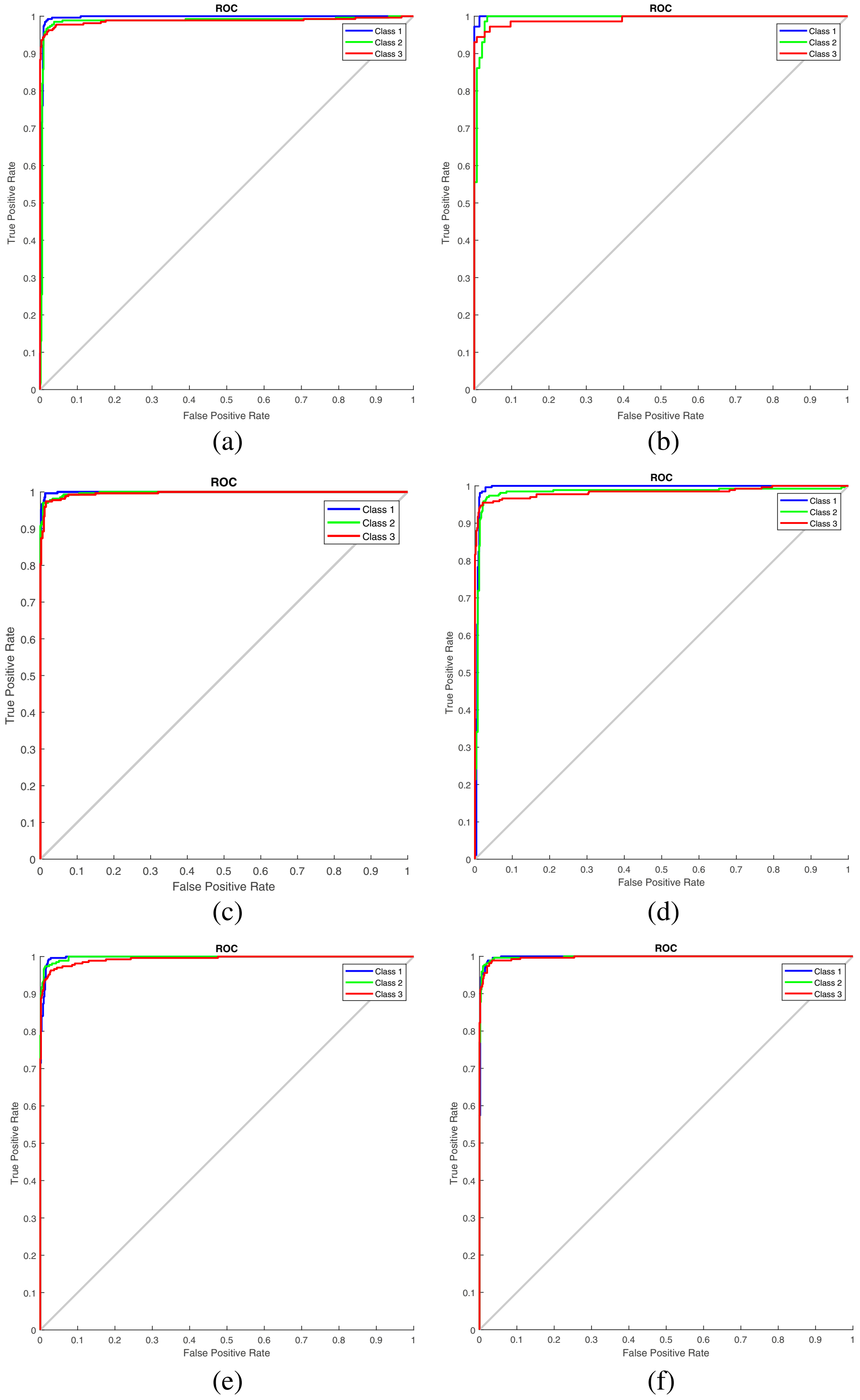
Fig. 7 Training progress for the (a) GWO-optimized (a) Nonoptimized CNNs

True Negative; FP indicate False Positive; FN indicate False Negative.

Accuracy evaluates the ability of the classifier to differentiate between COVID-19 and non–COVID-19 cases. A TP is where the model adequately predicts a positive case. Therefore, a TN is where the model viably predicts a negative instance. An FP is where the model erroneously predicts a positive case, and an FN is where the model mistakenly predicts a negative situation. Sensitivity gauges the proportion of correctly classified positive

Table 7 Performance analysis with other optimization methods

Method	Accuracy	Sensitivity	Specificity	Precision	F1 Score
Nonoptimization	91.23	93.33	90.19	82.62	87.64
GA	96.05	98.61	95.83	92.21	96.39
Pattern search	95.31	99.13	93.15	87.91	93.40
SA	95.93	95.93	92.17	95.93	95.93
PSO	95.76	97.75	94.76	90.31	93.88
WOA	96.41	95.93	96.67	93.50	94.70
Proposed optimization	97.78	97.75	96.25	92.88	95.25



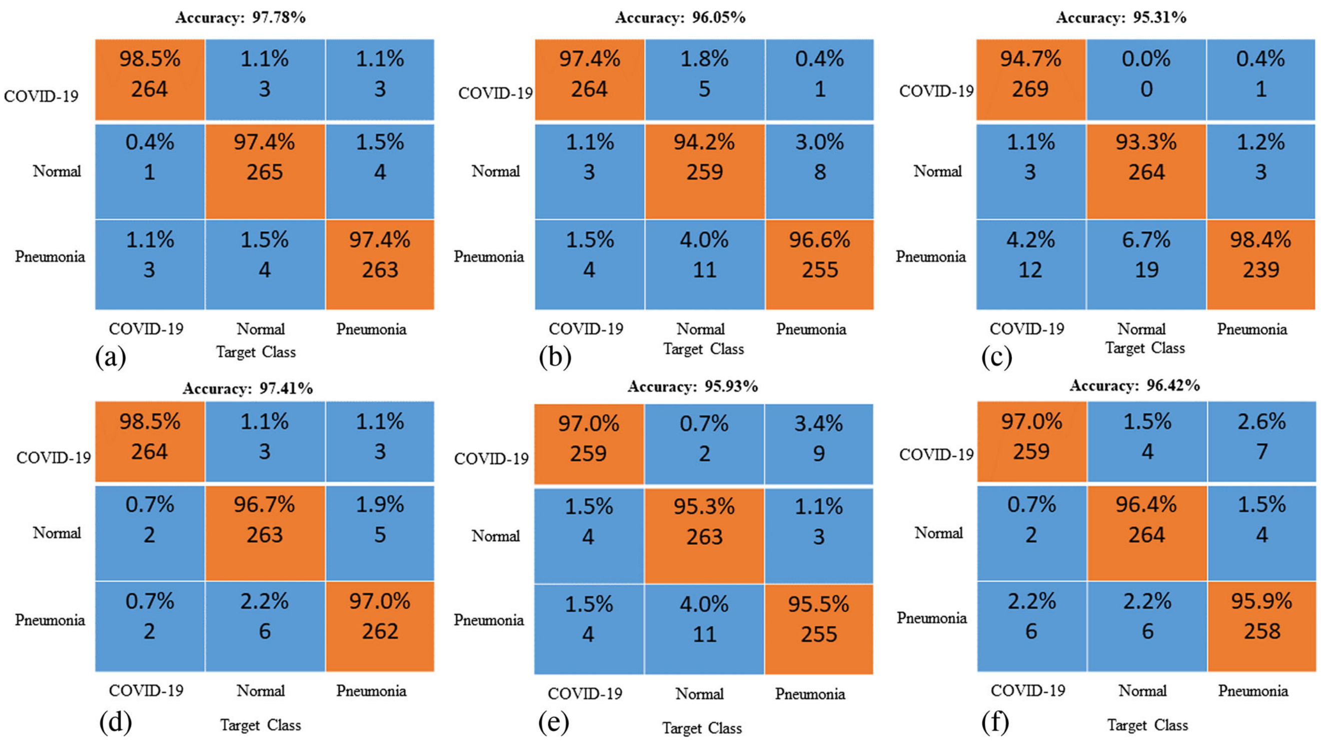
(a) (b)

(c) (d)

(e) (f)

Fig. 8 ROC curves of the **a** GWO **b** GA **c** PS **d** PSO **e** SA and **f** WOA-optimized CNNs (Class 1-COVID-19, Class 2- Normal, Class 3-Pneumonia)

instances. Specificity is a measure of the correctly classified negative instances. Precision measures the fraction of relevant cases among the retrieved cases and is also known as the positive predictive value. The F1-score measures a test's accuracy and is defined as the weighted harmonic mean of the test's precision and recall. The



Accuracy: 97.78%

	COVID-19	Normal	Pneumonia
COVID-19	98.5% 264	1.1% 3	1.1% 3
Normal	0.4% 1	97.4% 265	1.5% 4
Pneumonia	1.1% 3	1.5% 4	97.4% 263

Target Class

(a)

Accuracy: 96.05%

	COVID-19	Normal	Pneumonia
COVID-19	97.4% 264	1.8% 5	0.4% 1
Normal	1.1% 3	94.2% 259	3.0% 8
Pneumonia	1.5% 4	4.0% 11	96.6% 255

Target Class

(b)

Accuracy: 95.31%

	COVID-19	Normal	Pneumonia
COVID-19	94.7% 269	0.0% 0	0.4% 1
Normal	1.1% 3	93.3% 264	1.2% 3
Pneumonia	4.2% 12	6.7% 19	98.4% 239

Target Class

(c)

Accuracy: 97.41%

	COVID-19	Normal	Pneumonia
COVID-19	98.5% 264	1.1% 3	1.1% 3
Normal	0.7% 2	96.7% 263	1.9% 5
Pneumonia	0.7% 2	2.2% 6	97.0% 262

Target Class

(d)

Accuracy: 95.93%

	COVID-19	Normal	Pneumonia
COVID-19	97.0% 259	0.7% 2	3.4% 9
Normal	1.5% 4	95.3% 263	1.1% 3
Pneumonia	1.5% 4	4.0% 11	95.5% 255

Target Class

(e)

Accuracy: 96.42%

	COVID-19	Normal	Pneumonia
COVID-19	97.0% 259	1.5% 4	2.6% 7
Normal	0.7% 2	96.4% 264	1.5% 4
Pneumonia	2.2% 6	2.2% 6	95.9% 258

Target Class

(f)

Fig. 9 Confusion matrixes of the **a** GWO **b** GA **c** Pattern search **d** PSO, **e** Stimulated annealing, and **f** WOA-optimized CNNs

receiver operating characteristic (ROC) curve is the characteristic representation of the classification method performance executed for all values. This curve is drawn as the relationship between 1-specificity and sensitivity.

5.3 Experimental results

In this paper, CXR images are used to diagnose COVID-19 from normal and pneumonia-infected persons. X-ray imaging is a noninvasive technique for diagnosis and is also available in most hospitals. Using the trained proposed optimized CNN, testing of the images will take less than 5 s. The proposed network results are given in Table 5, which shows that an accuracy of 97.78% is achieved.

5.3.1 Receiver operating characteristics (ROC) and confusion matrix

The ROC curve (in Fig. 5) and confusion matrix (in Fig. 6) were generated for the proposed method to analyze the

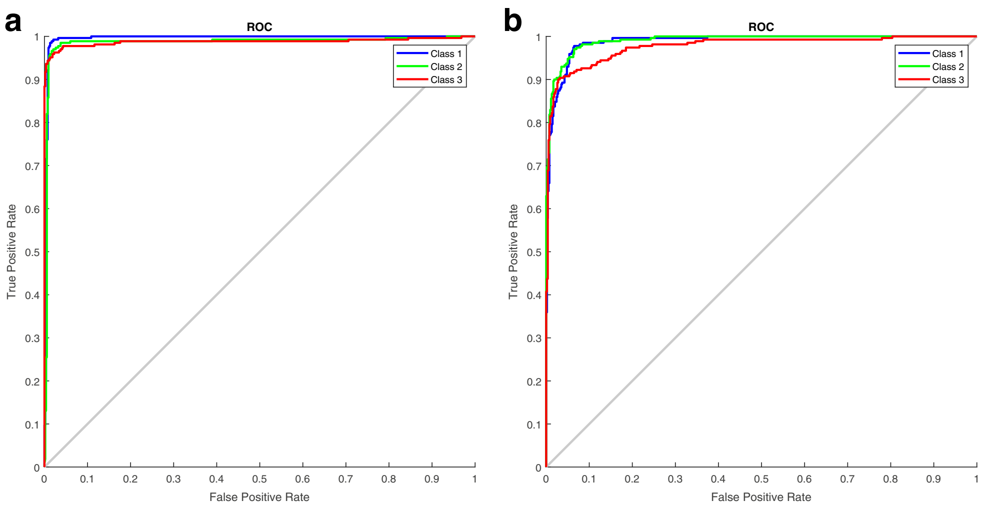
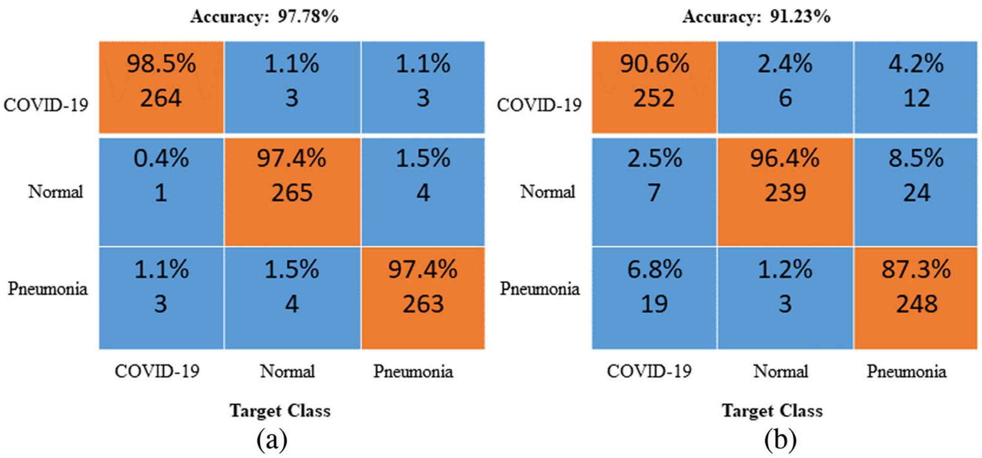


Fig. 10 Generated ROC curves for the **a** GWO-optimized CNN and **b** Nonoptimized CNNs (Class 1-COVID-19, Class 2-Normal, Class 3-Pneumonia)



Accuracy: 97.78%

	COVID-19	Normal	Pneumonia
COVID-19	98.5% 264	1.1% 3	1.1% 3
Normal	0.4% 1	97.4% 265	1.5% 4
Pneumonia	1.1% 3	1.5% 4	97.4% 263

Target Class

(a)

Accuracy: 91.23%

	COVID-19	Normal	Pneumonia
COVID-19	90.6% 252	2.4% 6	4.2% 12
Normal	2.5% 7	96.4% 239	8.5% 24
Pneumonia	6.8% 19	1.2% 3	87.3% 248

Target Class

(b)

Fig. 11 Confusion matrixes for the **a** GWO-optimized CNN and **b** Nonoptimized CNN

classification effectiveness. Following application of the proposed methodology for 2700 chest X-ray images, 98.5% were classified as COVID-19, 97.4% were classified as normal images and 97.4% were classified as pneumonia images. The proposed method has high accuracy for all three cases. Hence, this OptCoNet can be used for automatic screening of COVID-19. This OptCoNet will substantially help radiologists by overcoming the burden on the healthcare system.

5.3.2 Training progress of the proposed OptCoNet

As the selection of hyperparameters in training the CNN plays an important role, this subsection presents the results of an experiment comparing the training progress and loss functions between the GWO-optimized and nonoptimized CNNs. The training progress of the optimized and nonoptimized CNNs is shown in Fig. 7a and b, respectively. The proposed OptCoNet achieves better accuracy and minimum loss in all epochs. The training parameters of the optimized GWO are described in Table 6.

5.4 Performance analysis

In this work, five standard optimization techniques other than GWO are used in the classification of images. These are Genetic Algorithm (GA) [27], Pattern Search (PS) [28], Particle Swarm Optimization (PSO) [29], Simulated Annealing (SA) [30], and Whale Optimization Algorithm (WOA) [31]. For training the DL networks, 70% of the data are used; the remaining 30% of the data are used for testing. A comparison of all these networks in terms of the metrics accuracy, specificity, sensitivity, precision, and F1-score is given in Table 7. Comparisons in terms of ROC curves and confusion matrixes are illustrated in Figs. 8 and 9. The results of these comparisons indicates that the CNN optimized by GWO provides the best accuracy. Therefore, the proposed network

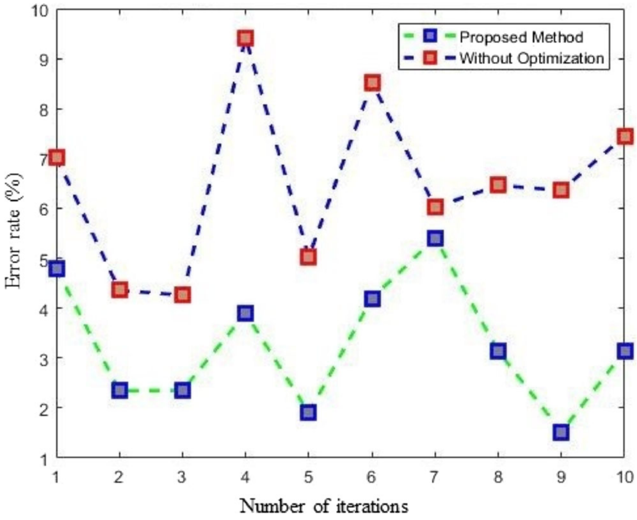


Fig. 12 Comparison with cross-validation

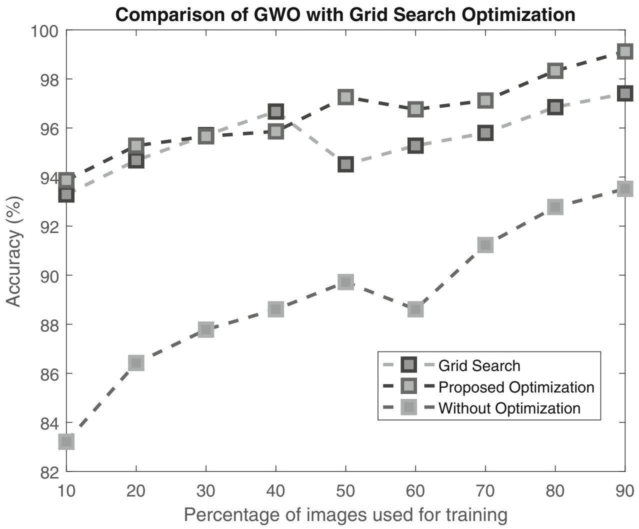


Fig. 13 Comparison of GWO with GSS

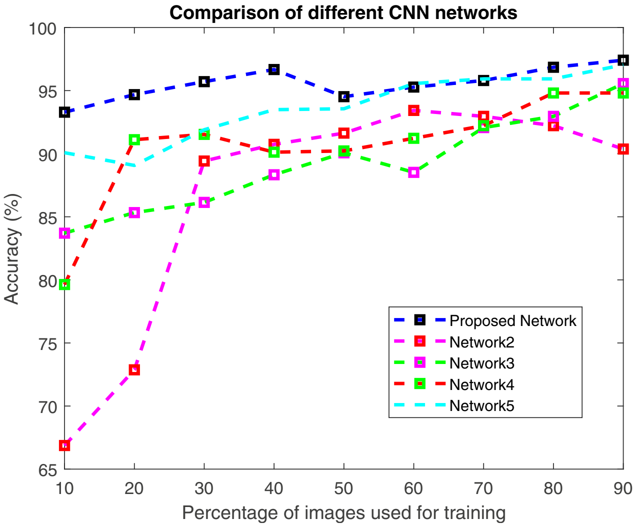


Fig. 14 Comparison with different CNNs

can be reliably used to diagnose COVID-19 in real-time applications.

5.5 Comparative analysis

This subsection presents a comparative analysis for the proposed methodology in classifying the images into COVID-19, normal, and pneumonia. Comparisons are made with other state-of-the-art methods using the aforementioned performance metrics. A discussion and limitations of the proposed work are given at the end of this subsection as well.

5.5.1 Comparison with nonoptimized CNN

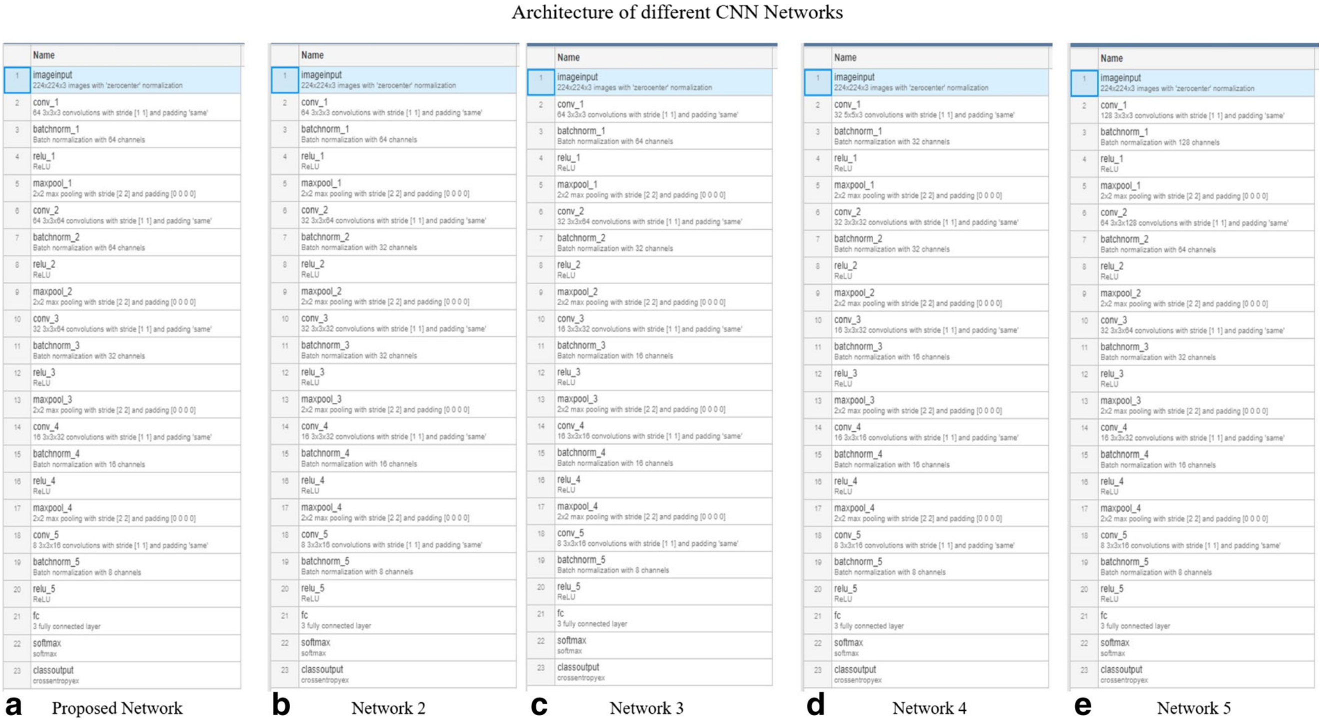
Figures 10 and 11 show the performance comparison in terms of the ROC curves and confusion matrixes between the optimized and nonoptimized CNNs. From Figs. 10 and 11, the optimized CNN yields a better accuracy than the nonoptimized CNN.

5.5.2 Comparison using cross-validation

Cross-validation (CV) is an important tool in predicting network performance by splitting the data k times into training and testing sets. In the present work, *k* is set to 10. Therefore, the X-ray data of each class are split into ten subsets. For every iteration of the CV, one subset from the *k* subsets is used for testing, and the remaining *k-1* subsets are used for training the network. Then, the error rate is calculated k times for the proposed optimized nonoptimized CNN. A comparison of the CV results is shown in Fig. 12.

5.5.3 Comparison with grid search optimization

The conventional method for optimizing the hyperparameters of a neural network is the grid search strategy (GSS) [32]. In the GSS, a subset of each hyperparameter space is manually


Architecture of different CNN Networks

a Proposed Network

	Name
1	imageinput 224x224x3 images with 'zerocenter' normalization
2	conv_1 64 3x3x3 convolutions with stride [1 1] and padding 'same'
3	batchnorm_1 Batch normalization with 64 channels
4	relu_1 ReLU
5	maxpool_1 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
6	conv_2 64 3x3x64 convolutions with stride [1 1] and padding 'same'
7	batchnorm_2 Batch normalization with 64 channels
8	relu_2 ReLU
9	maxpool_2 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
10	conv_3 32 3x3x64 convolutions with stride [1 1] and padding 'same'
11	batchnorm_3 Batch normalization with 32 channels
12	relu_3 ReLU
13	maxpool_3 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
14	conv_4 16 3x3x32 convolutions with stride [1 1] and padding 'same'
15	batchnorm_4 Batch normalization with 16 channels
16	relu_4 ReLU
17	maxpool_4 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
18	conv_5 8 3x3x16 convolutions with stride [1 1] and padding 'same'
19	batchnorm_5 Batch normalization with 8 channels
20	relu_5 ReLU
21	fc 3 fully connected layer
22	softmax softmax
23	classoutput crossentropyex

b Network 2

	Name
1	imageinput 224x224x3 images with 'zerocenter' normalization
2	conv_1 64 3x3x3 convolutions with stride [1 1] and padding 'same'
3	batchnorm_1 Batch normalization with 64 channels
4	relu_1 ReLU
5	maxpool_1 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
6	conv_2 32 3x3x64 convolutions with stride [1 1] and padding 'same'
7	batchnorm_2 Batch normalization with 32 channels
8	relu_2 ReLU
9	maxpool_2 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
10	conv_3 32 3x3x32 convolutions with stride [1 1] and padding 'same'
11	batchnorm_3 Batch normalization with 32 channels
12	relu_3 ReLU
13	maxpool_3 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
14	conv_4 16 3x3x32 convolutions with stride [1 1] and padding 'same'
15	batchnorm_4 Batch normalization with 16 channels
16	relu_4 ReLU
17	maxpool_4 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
18	conv_5 8 3x3x16 convolutions with stride [1 1] and padding 'same'
19	batchnorm_5 Batch normalization with 8 channels
20	relu_5 ReLU
21	fc 3 fully connected layer
22	softmax softmax
23	classoutput crossentropyex

c Network 3

	Name
1	imageinput 224x224x3 images with 'zerocenter' normalization
2	conv_1 64 3x3x3 convolutions with stride [1 1] and padding 'same'
3	batchnorm_1 Batch normalization with 64 channels
4	relu_1 ReLU
5	maxpool_1 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
6	conv_2 32 3x3x64 convolutions with stride [1 1] and padding 'same'
7	batchnorm_2 Batch normalization with 32 channels
8	relu_2 ReLU
9	maxpool_2 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
10	conv_3 16 3x3x32 convolutions with stride [1 1] and padding 'same'
11	batchnorm_3 Batch normalization with 16 channels
12	relu_3 ReLU
13	maxpool_3 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
14	conv_4 16 3x3x16 convolutions with stride [1 1] and padding 'same'
15	batchnorm_4 Batch normalization with 16 channels
16	relu_4 ReLU
17	maxpool_4 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
18	conv_5 8 3x3x16 convolutions with stride [1 1] and padding 'same'
19	batchnorm_5 Batch normalization with 8 channels
20	relu_5 ReLU
21	fc 3 fully connected layer
22	softmax softmax
23	classoutput crossentropyex

d Network 4

	Name
1	imageinput 224x224x3 images with 'zerocenter' normalization
2	conv_1 32 5x5x3 convolutions with stride [1 1] and padding 'same'
3	batchnorm_1 Batch normalization with 32 channels
4	relu_1 ReLU
5	maxpool_1 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
6	conv_2 32 3x3x32 convolutions with stride [1 1] and padding 'same'
7	batchnorm_2 Batch normalization with 32 channels
8	relu_2 ReLU
9	maxpool_2 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
10	conv_3 16 3x3x32 convolutions with stride [1 1] and padding 'same'
11	batchnorm_3 Batch normalization with 16 channels
12	relu_3 ReLU
13	maxpool_3 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
14	conv_4 16 3x3x16 convolutions with stride [1 1] and padding 'same'
15	batchnorm_4 Batch normalization with 16 channels
16	relu_4 ReLU
17	maxpool_4 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
18	conv_5 8 3x3x16 convolutions with stride [1 1] and padding 'same'
19	batchnorm_5 Batch normalization with 8 channels
20	relu_5 ReLU
21	fc 3 fully connected layer
22	softmax softmax
23	classoutput crossentropyex

e Network 5

	Name
1	imageinput 224x224x3 images with 'zerocenter' normalization
2	conv_1 128 3x3x3 convolutions with stride [1 1] and padding 'same'
3	batchnorm_1 Batch normalization with 128 channels
4	relu_1 ReLU
5	maxpool_1 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
6	conv_2 64 3x3x128 convolutions with stride [1 1] and padding 'same'
7	batchnorm_2 Batch normalization with 64 channels
8	relu_2 ReLU
9	maxpool_2 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
10	conv_3 32 3x3x64 convolutions with stride [1 1] and padding 'same'
11	batchnorm_3 Batch normalization with 32 channels
12	relu_3 ReLU
13	maxpool_3 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
14	conv_4 16 3x3x32 convolutions with stride [1 1] and padding 'same'
15	batchnorm_4 Batch normalization with 16 channels
16	relu_4 ReLU
17	maxpool_4 2x2 max pooling with stride [2 2] and padding [0 0 0 0]
18	conv_5 8 3x3x16 convolutions with stride [1 1] and padding 'same'
19	batchnorm_5 Batch normalization with 8 channels
20	relu_5 ReLU
21	fc 3 fully connected layer
22	softmax softmax
23	classoutput crossentropyex

Fig. 15 Architecture information of different CNNs **a** OptCoNet **b** Network2 **c** Network3 **d** Network4 **e** Network5

Table 8 Comparison of the results with state of the art CNN methods using X-ray images

Reference	Task	No of images	Method	Accuracy	Sensitivity	Specificity	Precision	F1 Score
Narin et al. [8]	COVID-19	50	ResNet-50	98	–	–	–	–
	Normal	50						
Hemdan et al. [9]	COVID-19(+)	90	COVIDX-Net	90	–	–	–	–
	Normal	25						
Khan et al. [10]	COVID-19	284	CoroNet	89.5	97	100	–	–
Maghdid et al. [12]	COVID-19	85	AlexNet	94.1	–	–	–	–
Razzak et al. [14]	COVID-19	200	DL	98.75	–	–	–	–
Abbas et al. [15]	COVID-19	105	DCNN	95.12	97.91	91.87	–	93.36
Afshar et al. [16]	COVID-19	1668	COVID-CAPS	95.7	90	95.8	–	–
Farook et al. [19]	COVID-19	68	COVID-Net	96.23	–	–	–	–
Ghoshal et al. [33]	COVID-19	70	CNN	92.9	–	–	–	–
Wang et al. [34]	COVID-19	45	CNN	83.5	–	–	–	–
	Bac.Pneu*	931						
	Vir.Pneu#	660						
Zhang et al. [35]	COVID-19	70	ResNet	–	96.6	70.7	–	–
Ioannis et al. [36]	COVID-19	224	VGG-19	93.48				
	Pneumonia	700						
	Normal	504						
Sethy et al. [37]	COVID-19 (+)	25	ResNet-50 + SVM	95.38				
	COVID-19(−)	25						
Ozturk et al. [38]	COVID-19(+)	125	DarkCovidNet	98.8	–	–	–	–
	No findings	500						
	COVID-19(+)	125		87.02	–	–	–	–
	Pneumonia	500						
	No findings	500						
Proposed	COVID-19	1000	OptCoNet	97.78	97.75	96.25	92.88	95.25
	Normal	900						
	Pneumonia	900						

*Bacterial Pneumonia, # Virus Pneumonia

specified and evaluated for a particular performance metric. The GSS evaluates the network performance by making a grid of all possible candidates within the hyperparameter space. The combination of hyperparameters that yields the best performance metric will be selected as the optimized hyperparameter values. The main drawback of the GSS is the exponential increase in the number of evaluations as a new parameter is added. Therefore, with only four hyperparameters with five candidates for each, the number of iterations is 758, making this optimization impractical.

On the other hand, GWO uses random values for the parameters and stops according to prespecified stopping criteria such as maximum time, number of parameters, or performance goal. In this way, it avoids overfitting the data and is practical for real-time problems with good performance results. The accuracy comparison of GWO with GSS is shown in Fig. 13.

5.5.4 Comparison with different CNN architectures

An experiment was conducted to compare the proposed OptCoNet architecture with other nonoptimized CNN architectures, namely, network2, network3, network4, and network5. The proposed OptCoNet provides better accuracy than the other CNN architectures. The results and the developed architectures are found in Figs. 14 and 15, respectively.

5.5.5 Comparisons of the results with state-of-the-art CNN methods

For further analysis of the results, the performance metrics, such as accuracy, sensitivity, specificity, precision, and F1-score, are also compared between the proposed CNN and pretrained DL networks. The acquired performance parameters for the proposed CNN are better than those of other state-of-the-art approaches, as shown in Table 8. The advantage of the proposed method is that no settings are required while training the images. Consequently, no tuning is needed for different databases, as opposed to other model-based methodologies, as in [8, 14]. In this way, the proposed approach can successfully deal with any concealed databases with no particular parameter tuning. The specialists in [8, 14, 32] accomplished a marginally poorer accuracy with their DLs compared to the proposed OptCoNet because of the absence of primary data in the images. Likewise, the proposed method is compared with recently published works in terms of the accuracy and F1-score. The proposed approach produced better results than other state-of-the-art techniques.

5.6 Discussion

Given how COVID-19 has infected millions of people worldwide, there is a need for quick and accurate diagnosis of the disease. The proposed network can discriminate COVID-19 patients from normal and pneumonia patients using X-ray images. X-ray image diagnosis is a noninvasive, cost-effective technique that is available in almost all hospitals. The proposed optimized CNN network was trained using 1890 images and then tested on 810 images. It provides an accuracy of 97.78%, which is the best accuracy achieved on X-ray images to date with more number of images. The optimized CNN was compared with state-of-the-art optimization techniques and four other CNN to show its efficiency. The advantages of the proposed optimized CNN include its ability to directly give diagnostics results from X-ray images without the need of a radiologist. In this way, it can help clinicians test patients with accurate results quickly. There is no need to preprocess the test images; the data augmentation step will resize all the test images to 224 × 224 pixels. Finally, the CNN hyperparameters are optimized using the GWO algorithm on X-ray images to avoid the problem of overfitting and mode collapse and give the best performance results.

Similar to any other new methods, the proposed method also has some limitations. One of the limitations is that the proposed OptCoNet does not encode the position orientation of the input images, so some preprocessing steps are required. In addition, OptCoNet is not spatially invariant to the input images. Finally, GWO has a number of controlling parameters that need to be carefully chosen if applied to problems with an extremely large number of features and parameters.

6 Conclusions

This work proposes OptCoNet, which depends on an optimized CNN structure for the automatic diagnosis of COVID-19, normal, and pneumonia patients from X-ray images. This system comprises a number of convolutional, batch normalization, and pooling layers. The acquired outcomes show that the OptCoNet performed well, even with a low number of trainable parameters. The proposed OptCoNet produced better performance metrics such as accuracy, sensitivity, specificity, precision, and F1-score than current state-of-the-art methods. As increasingly more COVID-19 cases are being recognized all around the globe, larger datasets are being created. Future studies will focus on advancing and altering the architecture of the OptCoNet and fusing newly accessible datasets to test the network.

Acknowledgments The authors acknowledge the authors of Chest imaging, COVID-19, Covid-chestxray, Figure 1 COVID-19 Chest X-ray, Provincial peoples hospital and Kaggle datasets for making publicly online. The authors also acknowledge the medical imaging laboratory of Computer Science and Engineering, National Institute of Technology Silchar, Assam for providing the necessary facilities to carry out this work.

Compliance with ethical standards

Conflict of interest The authors declare no conflict of interest.

Ethical standard This article does not contain any studies with human participants performed by any of the authors. This article does not contain any studies with animals performed by any of the authors. This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent There is no individual participant included in the study.

Code availability The source code of the OptCoNet are publicly available at https://github.com/biomedicallabecenitsilchar/optconet

References

1. Calvo C, López-Hortelano MG, De Carlos Vicente JC, Martínez JLV, De Trabajo de la Asociación G (2020) Recommendations on the clinical management of the COVID-19 infection by the «new coronavirus» SARS-CoV2. Spanish Paediatric Association working group. An Pediatría (English Edition) 92(4):241.e1–241.e11
2. COVID-19 Coronavirus Pandemic (20120, February). Worldometers, Retrieved July 23, 2020, from https://www.worldometers.info/coronavirus/?utm_campaign=homeAdvegas1?
3. Butt C, Gill J, Chun D, Babu BA (2020) Deep learning system to screen coronavirus disease 2019 pneumonia. Appl Intell p.1. https://doi.org/10.1007/s10489-020-01714-3
4. Liu H, Liu F, Li J, Zhang T, Wang D, Lan W (2020) Clinical and CT imaging features of the COVID-19 pneumonia: focus on pregnant women and children. J Inf Secur 80:e7–e13
5. Bai Y, Yao L, Wei T, Tian F, Jin DY, Chen L, Wang M (2020) Presumed asymptomatic carrier transmission of COVID-19. JAMA 323(14):1406–1407
6. Sohrabi C, Alsafi Z, O'Neill N, Khan M, Kerwan A, Al-Jabir A, Agha R (2020) World Health Organization declares global emergency: a review of the 2019 novel coronavirus (COVID-19). Int J Surg 76:71–76

7. Kang H, Xia L, Yan F, Wan Z, Shi F, Yuan H, Jiang H, Wu D, Sui H, Zhang C, Shen D (2020) Diagnosis of coronavirus disease 2019 (COVID-19) with structured latent multi-view representation learning. IEEE transactions on medical Imaging. Preprint may 5 2020
8. Narin A, Kaya C, Pamuk Z (2020) Automatic detection of coronavirus disease (COVID-19) using x-ray images and deep convolutional neural networks. arXiv preprint arXiv:2003.10849
9. Hemdan EED, Shouman MA, Karar ME (2020) COVIDX-NET: a framework of deep learning classifiers to diagnose COVID-19 in x-ray images. arXiv preprint arXiv:2003.11055
10. Khan AI, Shah JL, Bhat M (2020) CORONET: a deep neural network for detection and diagnosis of COVID-19 from chest X-ray images. arXiv preprint arXiv:2004.04931
11. Li L, Qin L, Xu Z, Yin Y, Wang X, Kong B, Cao K (2020) Artificial intelligence distinguishes covid-19 from community acquired pneumonia on chest CT. Radiology, 200905
12. Maghdid HS, Asaad AT, Ghafoor K, Sadiq AS, Khan MK (2020) Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning algorithms. arXiv preprint arXiv:2004.00038
13. Hassanien AE, Mahdy LN, Ezzat KA, Elmousalami HH, Ella HA (2020) Automatic X-ray COVID-19 lung image classification system based on multi-level Thresholding and support vector machine. medRxiv
14. Razzak I, Naz S, Rehman A, Khan A, Zaib A (2020) Improving coronavirus (COVID-19) diagnosis using deep transfer learning. medRxiv
15. Abbas A, Abdelsamea MM, Gaber MM (2020) Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. arXiv preprint arXiv:2003.13815
16. Afshar P., Heidarian, S., Naderkhani, F., Oikonomou, A., Plataniotis, K. N., & Mohammadi, A. (2020). COVID-CAPS: a capsule network-based framework for identification of COVID-19 cases from X-ray images. arXiv preprint arXiv:2004.02696
17. Apostolopoulos ID, Mpesiana TA (2020) Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. Australas Phys Eng Sci Med 43: 635–640
18. Apostolopoulos I, Aznaouridis S, Tzani M (2020) Extracting possibly representative COVID-19 biomarkers from X-ray images with deep learning approach and image data related to pulmonary diseases. J Med Biol Eng 14:1–8
19. Farooq M, Hafeez A (2020) COVID-ResNet: a deep learning framework for screening of COVID19 from radiographs. arXiv preprint arXiv:2003.14395
20. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
21. Chest Imaging (2020, March). This is a thread of COVID-19 CXR (all SARS-CoV-2 PCR+) from my hospital (Spain), version 1. Retrieved may 06, 2020, from https://twitter.com/ChestImaging/status/1243928581983670272
22. SIRM COVID-19 Database (2020, May) Italian Society of Medical and Interventional Radiology COVID-19 dataset, Version 1. Retrieved may 05, 2020, from https://www.sirm.org/category/senza-categoria/covid-19/
23. Cohen, J. P., Morrison, P., Dao, L., Roth, K., Duong, T. Q., & Ghassemi, M. (2020). COVID-19 image data collection: prospective predictions are the future. arXiv preprint arXiv:2006.11988
24. Wang L, Wong A, Lin ZQ, Lee J, McInnis P, Chung A, Ross M, van Berlo B, Ebadi A. “FIgure 1 COVID-19 Chest X-ray Dataset Initiative”, version 1, Retrieved may 05, 2020, from https://github.com/agchung/Figure1-COVID-chestxray-dataset
25. Kong W, Agarwal PP (2020) Chest imaging appearance of COVID-19 infection. Radiology 2(1):1–22
26. Mooney, P. (2018, March). Chest X-ray images (pneumonia), version 2. Retrieved May 05, 2020, from https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia/metadata
27. Holland JH (1992) Genetic algorithms. Sci Am 267(1):66–72
28. Hooke R, Jeeves TA (1961) Direct search solution of numerical and statistical problems. J ACM 8(2):212–229
29. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks 1942–1948
30. Van Laarhoven PJ, Aarts EH (1987) Simulated annealing. In simulated annealing: theory and applications. Springer, Dordrecht, pp 7–15
31. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
32. Lerman PM (1980) Fitting segmented regression models by grid search. J R Stat Soc: Ser C: Appl Stat 29(1):77–84
33. Ghoshal B, Tucker A (2020) Estimating uncertainty and interpretability in deep learning for coronavirus (COVID-19) detection. arXiv preprint arXiv:2003.10769
34. Wang L, Wong A (2020) COVID-net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest radiography images. arXiv preprint arXiv:2003.09871
35. 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
36. Apostolopoulos ID, Tzani B (2020) COVID-19: Automatic Detection from X-Ray Images Utilizing Transfer Learning with Convolutional Neural Networks, arXiv:2003.11617
37. Sethy PK, Behera SK (2020) Detection of coronavirus disease (COVID-19) based on deep features. Preprints, 2020030300, p.2020
38. Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR (2020) Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput Biol Med 121:1–11

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Tripti Goel obtained his Bachelor of Engineering (Hons) from Maharishi Dayanand University in 2004. She obtained her MTech in 2008 from Chottu Ram State College of Engineering, Haryana and Ph.D in 2017 from BPS Mahilla Vishwavidyalaya, Haryana. She joined Bhagwan Mahaveer Institute of Engineering and Technology, Haryana as Lecturer in Auguat 2005. After completing her M. Tech. She joined Guru Premsukh Memorial College of Engg. as lecturer in 2009 and became an Senior Lecturer in 2012. She joined at NIT, Delhi as an Assistant Professor in July, 2015. After that she Joined National Brain Research Center, Gurugram as Research Scientist in February, 2018. She joined NIT, Silchar as an Assistant Professor in June 2018. Her research interest which includes Medical Image Processing, Machine Learning, Deep Learning, Pattern Recognition, Neuroimaging.

R Murugan received his B.E. degree in Electronics and Communication Engineering, and M.E. degree in Embedded System Technologies from Anna University, Chennai, Tamilnadu, in 2005, and 2010 respectively. He received his Ph.D. degree from Information and Communication Engineering, Centre for Research, Anna University, Chennai, Tamilnadu, India. He is working as an Assistant Professor, in the Department of Electronics and Communication Engineering, National Institute of Technology Silchar since 15th June 2018. He published more than 26 journal publications, 22 conference proceedings, 2 books, 9 book chapters and 4 patents in his credit. His area of interest which includes Bio-medical signal and image processing, medical imaging, Retinal image analysis, computer vision, pattern recognition, machine learning and deep learning.

Seyedali Mirjalili is an Associate Professor at Torrens University Center for Artificial Intelligence Research and Optimization and internationally recognized for his advances in nature-inspired Artificial Intelligence (AI) techniques. He received his B.Sc. degree in Computer Engineering (software) from Yazd University, M.Sc. degree in Computer Science from Universiti Teknologi Malaysia (UTM), and Ph.D. in Computer Science from Griffith University. He was a member of Soft Computing Research Group (SCRG) at UTM. His research interests include Robust Optimization, Multi-objective Optimization, Swarm Intelligence, Evolutionary Algorithms, and Artificial Neural Networks. He is working on the application of multi-objective and robust metaheuristic optimization techniques. He is the author of more than 150 publications including five books, 100 journal articles, 20 conference papers, and 30 book chapters. With over 15,000 citations and H-index of 45, he is one of the most influential AI researchers in the world. From Google Scholar metrics, he is globally the most cited researcher in Optimization using AI techniques, which is his main area of expertise. He has been the keynote speaker of several international conferences and is serving as an associate editor of top AI journals including Neurocomputing, Applied Soft Computing, Advances in Engineering Software, Applied Intelligence, IEEE Access, and the Journal of Algorithms.

Deba Kumar Chakrabartty received the MBBS degree from Gauhati Medical College, India, in 1998 and the M.D. degree in Radio Diagnosis from Gauhati Medical College, India. He Joined as Registrar on Silchar Medical College, Assam, India in 1995. He Promoted to Assistant Professor and Associate Professor in 1995 and 2004 respectively on Silchar Medical College, Assam, India. Presently he working as Professor and Head of Radiology Department of Silchar Medical College and Hospital. He served various administrative responsibilities which includes PG thesis examiner from January 2014 and PG External Examiner on 2019 to North Bengal Medical College etc.