



A framework of genetic algorithm-based CNN on multi-access edge computing for automated detection of COVID-19

Md Rafiul Hassan¹ · Walaa N. Ismail² · Ahmad Chowdhury³ · Sharara Hossain⁴ · Shamsul Huda⁵ · Mohammad Mehedi Hassan⁶ 

Accepted: 17 November 2021 / Published online: 21 January 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021, corrected publication 2023

Abstract

This paper designs and develops a computational intelligence-based framework using convolutional neural network (CNN) and genetic algorithm (GA) to detect COVID-19 cases. The framework utilizes a multi-access edge computing technology such that end-user can access available resources as well the CNN on the cloud. Early detection of COVID-19 can improve treatment and mitigate transmission. During peaks of infection, hospitals worldwide have suffered from heavy patient loads, bed shortages, inadequate testing kits and short-staffing problems. Due to the time-consuming nature of the standard RT-PCR test, the lack of expert radiologists, and evaluation issues relating to poor quality images, patients with severe conditions are sometimes unable to receive timely treatment. It is thus recommended to incorporate computational intelligence methodologies, which provides highly accurate detection in a matter of minutes, alongside traditional testing as an emergency measure. CNN has achieved extraordinary performance in numerous computational intelligence tasks. However, finding a systematic, automatic and optimal set of hyperparameters for building an efficient CNN for complex tasks remains challenging. Moreover, due to advancement of technology, data are collected at sparse location and hence accumulation of data from such a diverse sparse location poses a challenge. In this article, we propose a framework of computational intelligence-based algorithm that utilize the recent 5G mobile technology of multi-access edge computing along with a new CNN-model for automatic COVID-19 detection using raw chest X-ray images. This algorithm suggests that anyone having a 5G device (e.g., 5G mobile phone) should be able to use the CNN-based automatic COVID-19 detection tool. As part of the proposed automated model, the model introduces a novel CNN structure with the genetic algorithm (GA) for hyperparameter tuning. One such combination of GA and CNN is new in the application of COVID-19 detection/classification. The experimental results show that the developed framework could classify COVID-19

Extended author information available on the last page of the article

X-ray images with 98.48% accuracy which is higher than any of the performances achieved by other studies.

Keywords Classification · CNN · COVID-19 · Genetic Algorithm · Multi-access edge

1 Introduction

An automatic detection of Coronavirus disease 2019 (COVID-19) can be developed through using the modern computational intelligence techniques and resources available on the high-performance computing facilities, e.g., cloud. The advent of convolutional neural network (CNN), a variant computational intelligence (CI) technique, has made the task of feature extraction from images and image analysis efficient. Moreover, the availability of high-performance computing (HPC) facilities, e.g., distributed edges on cloud, can help us to access COVID-19 data scattered at distant locations. Coronavirus disease 2019 (COVID-19) is a highly contagious viral disease. It is caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The first case of COVID-19 was identified in Wuhan, China, in December 2019, which eventually led to the ongoing pandemic [13, 33]. The World Health Organization declared the outbreak a Public Health Emergency of International Concern (PHEIC) in January 2020 and a pandemic in March 2020. COVID-19 symptoms include cough, fever or chills, shortness of breath or difficulty breathing, muscle or body aches, sore throat, temporary loss of taste or smell, diarrhea, headache, new fatigue, nausea or vomiting, and congestion or runny nose [25]. In more severe cases, the infection can cause pneumonia, the Middle East respiratory syndrome coronavirus (MERS-CoV), severe acute respiratory syndrome, kidney failure, and even death [19]. Diagnosis and prognosis of COVID-19 usually takes hours. Design and development of an automated system for rapid diagnosis/prognosis of COVID-19 will help to control spread of such epidemic. In this paper, we propose an automated model that uses genetic algorithm (GA) in developing a novel and effective CNN referred to as CGFor-Covid, which can assist experts in diagnosing COVID-19 from chest X-ray images through utilizing the state-of-art multi-access edge-computing 5G access network.

It is estimated that within the first ten months of the COVID-19 pandemic, suspected cases surpassed 10% of the global population (WHO/AP). As of December 25, 2020, 78,194,947 confirmed cases of COVID-19, including 1,736,752 deaths were reported to the WHO from around the world [8]. Traditionally, COVID-19 is diagnosed through Reverse Transcriptase Polymerase Chain (RT-PCR) testing, which has a sensitivity of 37.71% [9]. This test is expensive and takes several hours to diagnose the outcome [7, 8]. Chest radiography and computed tomography (CT) are also key tools in the diagnosis of lung diseases [22]. Using X-ray radiography is cheap, readily available, and the devices can be cleaned easily.

In this study, we propose a framework that utilizes the recent 5G technology that utilizes the multi-access edge computing facilities because, these technologies help

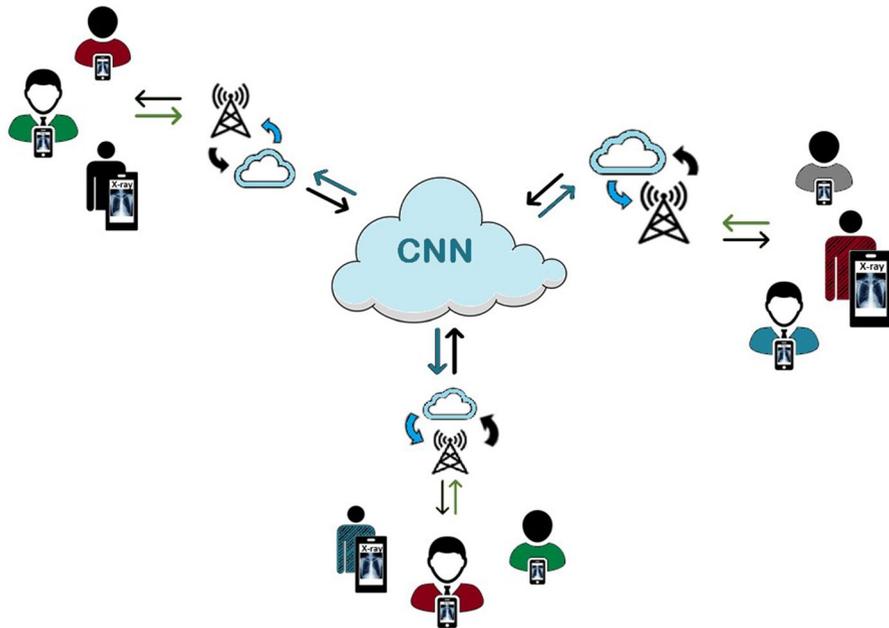


Fig. 1 Proposed framework for automated detection of COVID-19 through using multi-access edge computing and CNN

the end-user to use available resources on the cloud with convenience. Figure 1 shows the proposed framework. As shown in the framework, data are accumulated at the central cloud from sparse location through a number of 5G edge and access network. Note that, the edge framework is used once the CNN is trained and optimized using a large training dataset. The trained CNN is thus stored in the central cloud from where every end-user can access the CNN through 5G edges. The 5G access network would help in accessing massive data volume with high throughput while the 5G edge would help in executing the low-latency applications, in this framework, the CNN. The idea is to allow access of the CNN by end-user at hand held 5G devices (e.g., smart phone or tablet/iPAD). If end-user can provide the COVID-19 X-ray image, the CNN in the central cloud will be used to classify the image to either COVID-19 positive or negative case. The success of the proposed framework not only depends on the low-latency high-bandwidth multiple-access edge computing but also the classification accuracy of the CNN. Hence, we also design and develop an automated computational tool based on Artificial Intelligence (AI) techniques to diagnose COVID-19 using X-rays accurately. The subsequent sections of this article puts focus toward developing one such new AI techniques so that the developed technique can be used in the proposed framework as shown in Fig. 1.

In recent years, the adoption of deep learning (DL), a newly developed AI method, has about revolutionary changes in AI research and applications. Convolutional neural network (CNN), a variant of DL, is a widely used deep learning framework in biomedical image classification. A CNN algorithm can be used to extract

features from blocks of convolutional and backpropagation approaches, which in turn utilizes one or more pooling layers or fully connected layers. The extracted features are then mapped into classification using a classifier. The performance of the CNN depends on the initial parameters (e.g., the structure of the CNN, the size of the convolution kernel matrix, which is known as the kernel, and so on). These initial parameters are pre-selected by the user for a given problem. Many studies have used CNN for COVID-19 detection by analyzing chest X-ray images. For example, Tulin Ozturk et al. [20] proposed a fully automated DarkCovidNet model (a CNN model) for COVID-19 classification and achieved an accuracy of 98.08%. However, it is unclear how the authors chose the CNN structure. One of the challenges associated with the use of CNN is to identify a suitable structure for a given problem so that the best classification performance can be achieved. In our study, we propose and develop a deep CNN model in which a genetic algorithm (GA) is used to identify a suitable structure for efficiently classifying X-ray images for COVID-19 detection. The approach of combining GA with neural network architecture was inspired by previous studies that have investigated hyperparameter optimization for neural networks [5, 14, 17]. GA has been used to solve optimization problems in various fields, including autonomous crack detection [4], electromagnetics [1], synthesis of antenna patterns [16], computer vision and speech processing [36]. To the best of author's knowledge, the technique has not yet been applied to COVID-19 detection. Therefore, the proposed model uses this technique for the first time to classify X-ray images for COVID-19 detection.

In summary, the contributions of the current study are as follows:

1. Proposing a framework through utilizing the state-of-art multi-access edge computing, 5G access network and automated CNN to allow end-users access to the CNN for automatic diagnosis of chest X-ray images.
2. Developing a novel and effective CNN model optimized using a GA for COVID (CGForCovid), which can assist experts in diagnosing COVID-19 from chest X-ray images.
3. Using a GA algorithm to accelerate the CNN architecture design by selecting the optimal parameters for the CNN architectures, thereby enhancing model performance by reducing the search space and decreasing the computational complexity.
4. Identifying the specific layer of a multilayer CNN structure that provides features that enhance classification. The features extracted at each layer of the CNN are fed into the classifier individually and classification performance is analyzed.
5. Conducting extensive experiments and comparing the results with the classification performances of systems reported in other recent studies.

2 Related work

The world has experienced significant social and economic disruption since the beginning of the COVID-19 pandemic. These problems can be solved, or at least curbed through quick and reliable COVID-19 detection by deploying AI tools ([6,

11, 34]). Many studies have attempted to detect COVID-19 cases automatically using computational techniques. Given that this study focuses on detecting COVID-19 from X-ray images, this section reviews existing studies that used X-ray images for COVID-19 case identification.

A comparative study was undertaken by the authors in [8] of RT-PCR and chest CT during the peak of the Italian epidemic, showing high sensitivity and specificity scores for detection using chest CT images. Tulin Ozturk et al. [20] proposed a fully automated DarkCovidNet model with a binary classification accuracy of 98.08%. In the first few months of the pandemic, medical imaging data were scarce, and so to tackle this issue, Abdul Waheed et al. [29] used generative adversarial network (GAN) to generate new data and CNN for detection. The authors' COVID-GAN model displayed an improved accuracy of 95%, where training on the original data yielded 85% accuracy. COVID-NET, a deep CNN model using open access data, has also been proposed [30]. A framework consisting of seven deep learning architectures such as VGG19, ResNetV2, InceptionV3 and MobileNetV2, known as COVIDX-Net yielded an F1-score of up to 0.91 [12]. A rather novel algorithm referred to as Capsnet, which was designed for image classification, has been used in Convolutional Capsnet [28], which gives 97.24% accuracy on binary classification. A transfer learning-based model nCOVnet achieved 88% overall accuracy and 97.62% true positive rate [21]. Existing CNN architecture design algorithms require extensive work to manually design the CNN architecture, which often show slow performance for the complete model design. However, for users who may lack domain knowledge on CNNs, it is necessary to have powerful design that can offer an automatic way to tune model structure. GAs are inspired by the way biological development works, which involves selecting the population, and outputting more than one solution without being stuck to a local optimum [23, 32]. Therefore, GA can perform efficient optimization operations for selecting the optimal hyperparameter for deep neural network-based models. The researchers also implemented different optimization models for the selection of CNN hyperparameters using GA.

In NSANet [27], a new genetic CNN method was discussed for encoding the CNN architecture in multiple phases to replace the convolutional layers to build the final CNN. For each phase, the model architecture is gradually developed as a small unit known as a "cell" in Genetic CNN, and afterward, multiple building blocks of CNNs are ordered and encoded. The order of such building blocks is manually adjusted based on the first and last building blocks, and a binary string encoding method is used to encode the blocks connections. Setting all of these parameters is figured out manually with no techniques for acceleration having been designed, and this limits the applicability of the proposed model for complex datasets such as x-ray images. A population-based training (PBT) algorithm for utilizing GA for neural network model hyperparameter selection was presented in [14]. PBT serves as a practical way to augment the standard training of neural network models using adaptive schedules. The proposed algorithm led to an outstanding improvement in the performance of neural network-based models, including hierarchical reinforcement learning, deep reinforcement learning, GANs, and machine translation. In [37], the authors proposed two hybrid models for the prediction and classification of HGB-anemia, nutritional anemia, deficiency anemia, and folate deficiency anemia. The

model worked by initiating the GA for the purpose of selecting the optimal hyperparameters for the stacked autoencoder (SAE) and CNN. The system achieved a classification accuracy of 98.50%. In [27], GA was used to automate the CNN architecture design method. The authors targeted their work toward inexperienced users who lack CNN domain knowledge, specifically due to the value of their algorithm in automatically selecting the optimal CNN architecture to address the image classification process with 96.78% accuracy. Bakhshi et al. [3] proposed fast-CNN, a rapid and automatic CNN building architecture for image classification. Fast-CNN uses GA to define suitable CNN optimization parameters, including the learning rate, number of layers, momentum, number of feature maps, and weight decay factor. Similarly, in [35], the authors sought to solve the same problem using a variable length GA. The algorithm automatically searches for a CNN's optimal hyperparameters considering the expected depth increases of the CNN model and search space, which grows the number of hyperparameters selection time.

Based on these recent studies, and also inspired by the successful optimization of CNN hyperparameters using GA, this study proposes the application of a GA-initiated automatic CNN architecture design for COVID-19 X-ray image classification.

3 Overview of the method

In this study, we propose and develop the so-called CGForCovid model, a CNN structure for COVID-19 classification that uses GA to optimize the hyperparameters and determine suitable kernel sizes for the CNN structure. A kernel is a matrix that acts like a filter and moves over the input data, performing the dot product with the sub-region of input image. Figure 2 shows a schematic diagram of the proposed CGForCovid model. As shown in the figure, the model receives the COVID-19 X-ray image dataset as input, and after preprocessing the dataset, two datasets (training/test sets) are prepared for the experiment evaluation. Afterward, the training

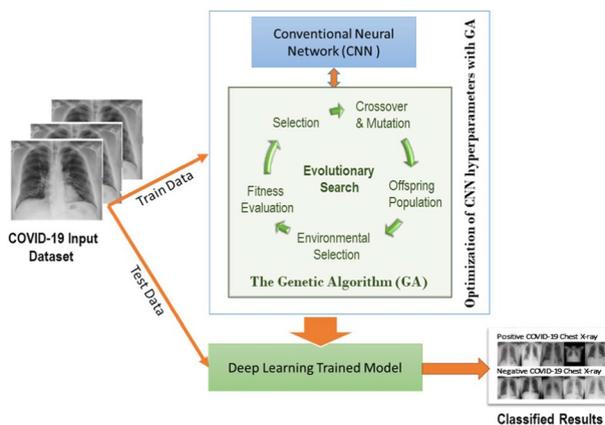


Fig. 2 The proposed CGForCovid model

dataset is used as input to the CNN algorithm. The GA is used to select the hyperparameter values, thereby optimizing the CNN model. In the optimization process, GA starts with a range of values for kernel sizes as initial population, crossover and mutation, and the offspring population. GA is used to select the best and most significant combinations of the CNN model hyperparameters (i.e., kernel sizes). Finally, the achieved CNN model is used to classify patient health status. Figure 3 shows one such CNN architecture that can be used for COVID-19 X-ray images (RGB images of size $224 \times 224 \times 3$) classification. The following subsections describe how GA is used to optimize an initially chosen CNN structure.

Before describing the combined GA-CNN model, let us first introduce the GA and CNN in brief.

3.1 The genetic algorithm

GA is a variant of the popular evolutionary algorithm (EA). The difference between GA and EA is that in the latter, the chromosomes represent real numbers, while in GA chromosomes represent binary numbers. EA is the most feasible solution that is widely used for many optimization problems. The concept of EA is inspired by Darwinism, a biological evolutionary theory of nature that posits the survival of the fittest. This makes it significantly different compared to other search engines as it allows sampling through large search effectively. In EA/GA, an initial population is randomly generated as candidate solutions followed by three operations: selection, crossover, and mutation. Each individual solution is represented as a chromosome that consists of a set of strings, each with their own fitness value. Fitness value is a score (e.g., classification accuracy of a model) that determines how satisfactory the solution is. A population is the set of current sets of solutions (chromosomes) from which new sets of solutions are to be generated. To select the parent string from the current population, the probability of each string in the population of the

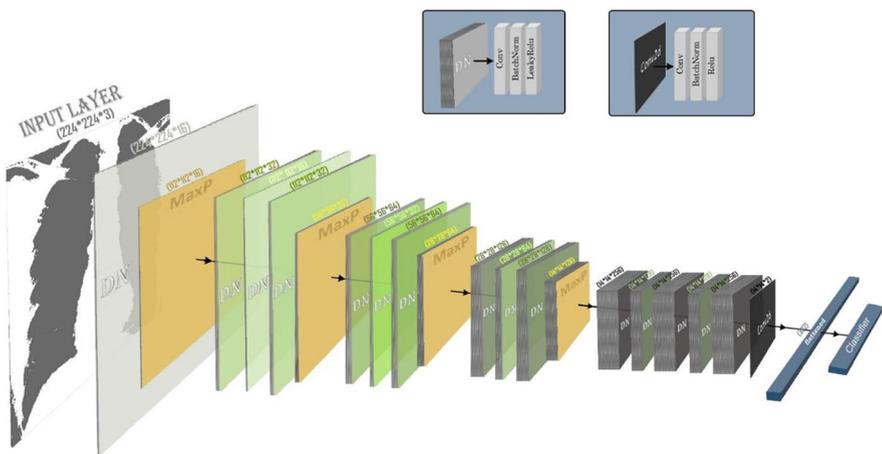


Fig. 3 An architecture of the CNN used in CGForCovid Model

current generation is determined considering the fitness value. Crossover in GA generates new generations by mating parent solutions (chromosomes) by crossing over the strings at some point to produce offspring. Mutation introduces variation into the population with a time interval by changing certain bit values in the string (chromosome). These operations are executed iteratively to create new generations until a stopping criterion is reached. The objective of EA/GA is either to minimize or maximize the fitness value based on survival of the fittest criterion.

3.2 Convolutional neural network

A typical CNN structure has one or more convolutional layer (CNN kernel) that extracts the significant feature set from the input dataset with its own filters. There is one or more pooling layer used in the CNN. The pooling layer decreases the size of the intermediate features without losing significant information from the feature set. The last layer of the CNN is typically used to classify data using the extracted features. By consolidating the layers, a CNN model is constructed, and using a training algorithm and dataset, the internal encoding parameters of the CNN are adjusted to classify/predict COVID-19 cases. There is no known approach available for selecting the best CNN structure for a problem. While attempting to use CNN for COVID-19 detection, we found that CNN kernel size affects classification performance. Therefore, optimization is important because poor classification accuracy may not necessarily be due to noisy data or algorithms suffering from weak learning; it also results from a poor combination of parametric values.

3.3 Combining GA with CNN

In our proposed CGForCovid model, chest X-ray images are used as input to the model. For each individual image input X and kernel K , the following convolution operation is used,

$$(X * K)(i, j) = \sum_m \sum_n K(m, n)X(i - m, j - n) \quad (1)$$

where $*$ represents the discrete convolution operation. Here, the kernel matrix K is slid over the input matrix L ($size(L) > size(M)$) to extract features through the convolution operation. After the convolution, the achieved features are further rectified using a nonlinear activation function. Leaky rectified linear unit (Leaky ReLu) has recently grown in popularity as an activation function because it does not change the size of the input and does not suffer from the vanishing gradient problem. In CGForCovid, Leaky ReLu is used as the rectifier at each convolutional layer. Equation 2 explains the calculation used in the Leaky ReLu function.

$$\text{Leaky ReLu}(x) = \begin{cases} 0.01x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (2)$$

The GA used in CGForCovid has a single objective because only one fitness value/score (i.e., classification accuracy for the training/validation dataset) is maximized using the standard GA operators of selection, crossover, and mutation. The chromosome of the GA is a potential solution/kernel size of the CNN for COVID-19 classification. As such the GA should provide a suitable kernel size for the CNN structure to maximize COVID-19 classification accuracy. Algorithm 1 shows the procedure used to identify suitable kernel size in pseudo-code.

Algorithm 1 Genetic Algorithm to identify suitable kernel size for a predefined CNN structure

Input: COVID-19 image training dataset, CNN_{para} : A set of predefined CNN building blocks (e.g., total number of layers), PN : the population size, M : mutation percentage, GN : Number of the maximal generation.

Output: HyperPara: the suitable CNN kernel sizes of each of the convolution layers for COVID-19 images classification.

```

1: Define  $t=0$  // the generation number
2: for Each Population  $P$  from Generation  $G_t$  do
3:   divide the training data into training data and validation data (10% of the training data)
4:    $k=0$  // the population number
5:   while  $k \neq PN$  do
6:     Generate kernel following the  $C$  and use the kernel to generate CNN model
7:     Obtain classification accuracy for validation data using the CNN model and other parameters  $CNN_{para}$ 
8:     Select the best Chromosomes from each of the populations  $P_0, P_1, \dots, P_{PN}$ 
9:     Do cross-over among the selected Chromosomes and generate new Population
10:    Generate new generation from each of the new generated Population
11:    if the classification accuracy is not improved for several generations then
12:      Do mutation operation to the chromosomes of new generation with a probability of  $M$ 
13: Assign HeperPara = the best Chromosome from the last generation
14: return HyperPara

```

As described in the algorithm, it starts to build the CNN model with a defined population size and maximal generation number. The GA algorithm begins to work iteratively by searching for suitable optimization parameters (kernel size) for the CNN architecture to classify whether the patient is COVID-19-positive or not from the collected image dataset. To encode the CNN model, a set of iterations are conducted to randomly initialize a population with the predefined maximal population size PN ; that is to say each generation consists of PN populations. In the process of evolution, the fitness of each individual chromosome in each population is calculated. Once the best chromosomes are identified based on the fitness score, a new population is generated from the selected chromosomes through the crossover operation. This process of evolution continues up to a maximum generation, and the best chromosome in the last generation is regarded as the suitable kernel size for the CNN structure for COVID-19 detection.

4 Experimental setup and results

4.1 Experimental setup

The proposed model utilizes GA for optimizing CNN structure. Following list of parameters for GA were used to conduct the experiment:

Max Iteration: 500
Generation=10
Population size:100
Mutation probability:0.01
Elite ratio: 1
Crossover probability: 0.5
Parents portion: 1
Crossover type: uniform

16-256 filters for each convolution layer were used in the proposed CGForCovid model. Each convolutional layer in the CNN was followed with a softmax layer, which is a fully connected dense neural network classifier, to observe accuracy performance before building up the next block of convolutional layers. To downsize the input X-ray images, the maxpool method was used in all the pooling operations. The maxpool method works by taking the maximum pixel value determined by its filter. The final CNN architecture was evaluated using fivefold cross-validation for different classifiers (several classifiers were used at the last layer of the CNN), including Dense (traditional fully connected neural network), random forest, and decision tree. The model thus classifies the input X-ray images either as COVID-19 positive or COVID-19 negative cases.

To identify the layer of the CNN that provided the optimal features, a classifier was used at the end of each layer to classify the data using extracted features by the respective CNN layer. To determine which classifier in combination with the extracted features provided better classification performance, three different classifiers were tested: fully connected neural network classifier with the random forest classifier and decision tree. The performance of each stage/layer of the CNN architecture was investigated by training the models with categorical cross-entropy as the loss function, a learning rate of 0.0001 for the Adam optimizer, and using a fully connected neural network as a classifying layer.

To overcome the overfitting problem, 20% of the training X-ray data were used for validation and 80% for training the CNN model. Additionally, to test robustness of the CGForCovid model, fivefold cross-validation was used. The parameters of the models used in the study are shown in Tables 2 and 3. The Keras platform written in Python was used to conduct all experiments. In the study, all implementations were executed on a system with a 64-bit Intel Core I5 processor, and 12 GB of RAM.

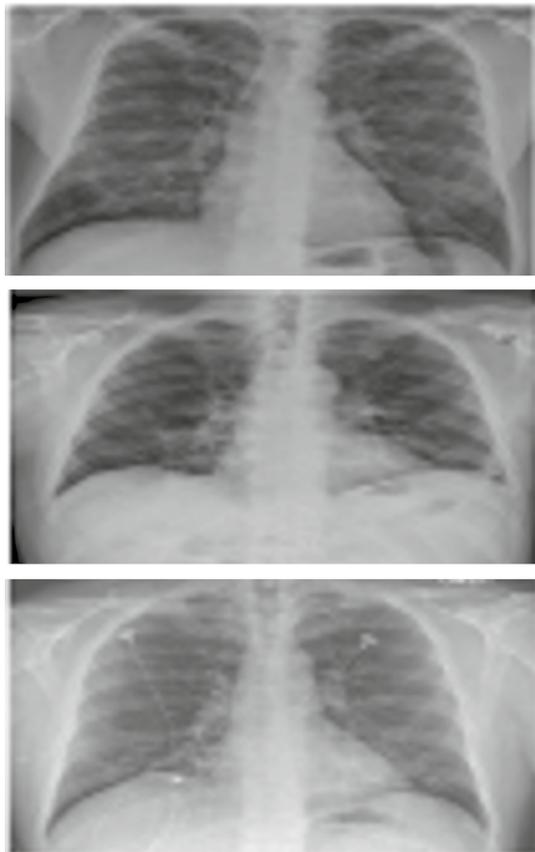
4.2 Dataset

X-ray images were obtained from a dataset developed by Joseph Paul Cohen and Paul Morrison and Lan Dao [15]. This dataset contains chest X-ray and CT scan images of COVID-19, MERS, SARS, and ARDS. The dataset includes frontal and lateral view imagery and metadata, including the time since first symptoms, survival status, and location. We opted to use X-ray images for our study, of which there are 420 COVID-19 positive cases and 505 COVID-19 negative cases in our X-ray dataset. Figure 4 shows example X-ray images of COVID-19 positive cases, and Fig. 5 illustrates X-ray images of COVID-19 negative cases from the dataset.

4.3 Performance metrics

For all the simulations, a binary classifier is initialized to predicts all the data instances with four possible outcomes: TP (true positive), TN (true negative), FP (false positive), and FN (false negative).

Fig. 4 COVID-19-positive X-ray images



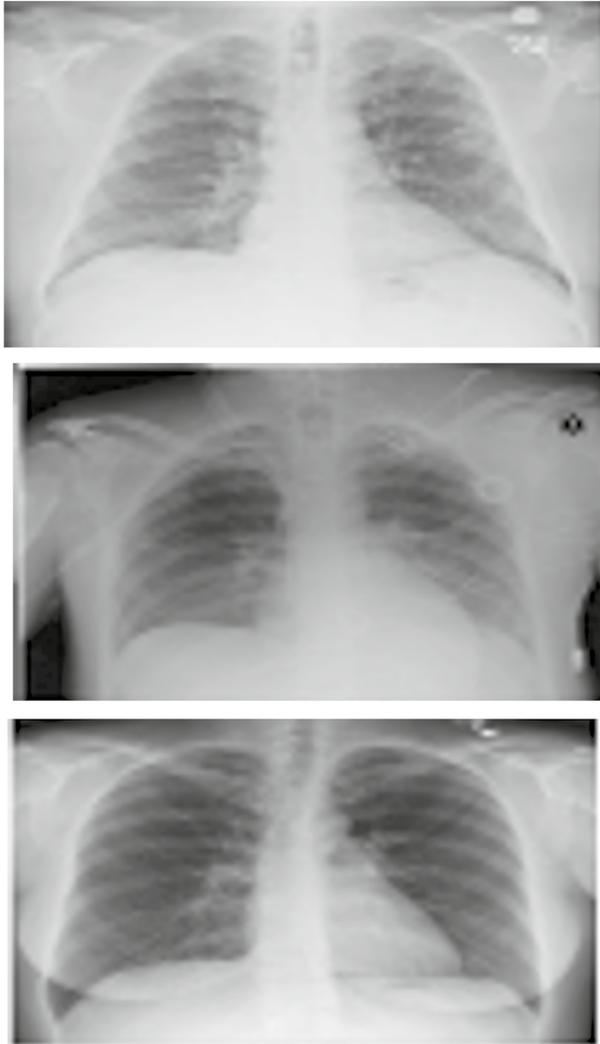


Fig. 5 COVID-19-negative X-ray images

- TP: correct positive prediction
- FP: incorrect positive prediction
- TN: correct negative prediction
- FN: incorrect negative prediction

Accuracy is calculated as the total number of two correct predictions (TP + TN) divided by the total number of a dataset (P + N).

- Accuracy = $(tp+tn)/(tp+tn+fp+fn)$

Sensitivity (recall or true positive rate) is calculated as the number of correct positive predictions (TP) divided by the total number of positives (P).

- Sensitivity (recall or true positive rate) = $tp/(tp+fn)$

Specificity (true negative rate) is calculated as the number of correct negative predictions (TN) divided by the total number of negatives (N).

- Specificity (true negative rate) = $tn/(tn+fp)$

Precision (positive predictive value) is calculated as the number of correct positive predictions (TP) divided by the total number of positive predictions (TP + FP).

- Precision (positive predictive value) = $tp/(tp+fp)$

F1 score is the harmonic mean of the precision and recall.

- $F1=(2*precision*recall)/(precision+recall)$

A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system by plotting the true positive rate against the false positive rate at various threshold settings. When using normalized units, the area under the curve (AUC) of the ROC plot is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one [10].

4.4 Experimental results of the proposed model

In this study, a combination of state-of-art CI technologies: GA and CNN (CGForCovid) was proposed to classify the COVID-19 X-ray image dataset. Table 1 lists the detailed parameters achieved by using the proposed CGForCovid model for the selected dataset. The proposed model uses 16 convolution layers to classify COVID-19 X-ray images. Compared to another CNN known as Darknet-19, that model used 19 convolutional layers to classify COVID-19 X-ray images. Moreover, the proposed model can be used for any COVID-19 dataset because the GA identifies a suitable CNN structure if the training dataset changes. In contrast, Darknet-19 is not adaptive under a change of training dataset.

Table 2 reports the experimental results for the dataset discussed in Section 2. To identify the layer of the CNN that can provide the best feature set for effectively identifying COVID-19 cases, features extracted from each block (see column 1 in the table for block numbers) are individually fed into the respective classifiers to classify the data. As the table shows, block number 7 provided the best features that assisted in classifying the COVID-19 cases, achieving 98.91% accuracy. This experimental results demonstrate that the optimization of a CNN using GA can deliver COVID-19 classification with high accuracy, as well as a 98.907 AUC score.

Table 1 The experimental hyperparameters of the CGForCovid model

Layer #	Block #	Layer(type)	Output Shape Param #
1	1	conv2d	(None, 224, 224, 16)
2		batch_normalization	(None, 224, 224, 16)
3		LeakyReLU	(None, 224, 224, 16)
4		MaxPooling2D	(None, 112, 112, 16)
5	2	conv2d	(None, 112, 112, 32)
6		batch_normalization	(None, 112, 112, 32)
7		LeakyReLU	(None, 112, 112, 32)
8		conv2d	(None, 112, 112, 16)
9		batch_normalization	(None, 112, 112, 16)
10		LeakyReLU	(None, 112, 112, 16)
11		conv2d	(None, 112, 112, 32)
12		batch_normalization	(None, 112, 112, 32)
13		LeakyReLU	(None, 112, 112, 32)
14		MaxPooling2D	(None, 56, 56, 32)
15	3	conv2d	(None, 56, 56, 64)
16		batch_normalization	(None, 56, 56, 64)
17		LeakyReLU	(None, 56, 56, 64)
18		conv2d	(None, 56, 56, 32)
19		batch_normalization	(None, 56, 56, 32)
20		LeakyReLU	(None, 56, 56, 32)
21		conv2d	(None, 56, 56, 64)
22		batch_normalization	(None, 56, 56, 64)
23		LeakyReLU	(None, 56, 56, 64)
24		MaxPooling2D	(None, 28, 28, 64)
25	4	conv2d	(None, 28, 28, 128)
26		batch_normalization	(None, 28, 28, 128)
27		LeakyReLU	(None, 28, 28, 128)
28		conv2d	(None, 28, 28, 64)
29		batch_normalization	(None, 28, 28, 64)
30		LeakyReLU	(None, 28, 28, 64)
31		conv2d	(None, 28, 28, 128)
32		batch_normalization	(None, 28, 28, 128)
33		LeakyReLU	(None, 28, 28, 128)
34		MaxPooling2D	(None, 14, 14, 128)
35	5	conv2d	(None, 14, 14, 256)
36		batch_normalization	(None, 14, 14, 256)
37		LeakyReLU	(None, 14, 14, 256)
38		conv2d	(None, 14, 14, 128)
39		batch_normalization	(None, 14, 14, 128)
40		LeakyReLU	(None, 14, 14, 128)
41		conv2d	(None, 14, 14, 256)
42		batch_normalization	(None, 14, 14, 256)
43		LeakyReLU	(None, 14, 14, 256)

Table 1 (continued)

Layer #	Block #	Layer(type)	Output Shape Param #
44	6	conv2d	(None, 14, 14, 128)
45		batch_normalization	(None, 14, 14, 128)
46		LeakyReLU	(None, 14, 14, 128)
47	7	conv2d	(None, 14, 14, 256)
48		batch_normalization	(None, 14, 14, 256)
49		LeakyReLU	(None, 14, 14, 256)
50		conv2d	(None, 14, 14, 2)
51		batch_normalization	(None, 14, 14, 2)
52		flatten (Flatten)	(None, 392)
53		dense (Dense)	(None, 2)

Moreover, using different classifiers at the last level of the model, we sought to identify the optimally performing classifier. As the table shows, all three classifiers, namely neural network, decision tree, and random forest, achieved the same level of classification performance for COVID X-ray image classification. This demonstrates that the combined use of GA and CNN can generate features that are significant for COVID-19 classification.

After achieving an impressive classification performance for the training and test data (80% of the data were randomly selected to train the model, while the remaining 20% were used to test the model), a fivefold cross-validation scheme was used to achieve a generalized performance of the proposed CGForCovid model. The graphs

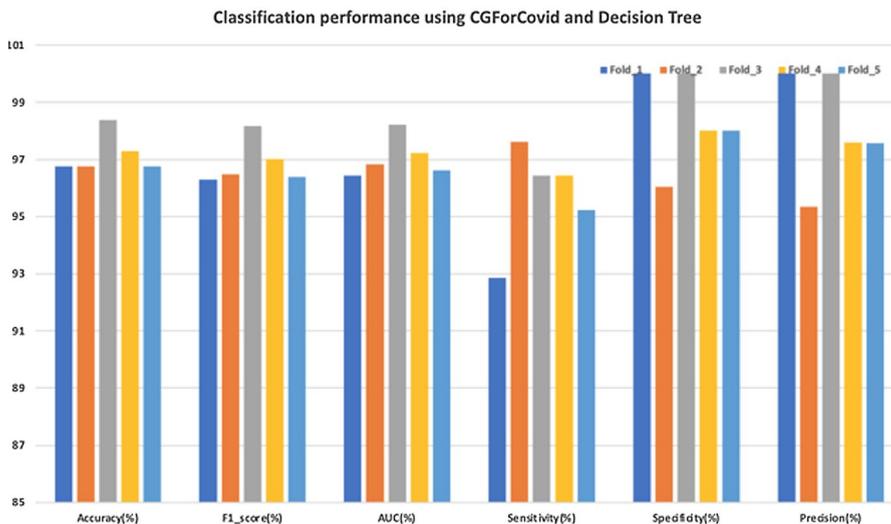


Fig. 6 Classification performances for each of the fivefolds using CGForCovid along with a decision tree at the last layer as classifier

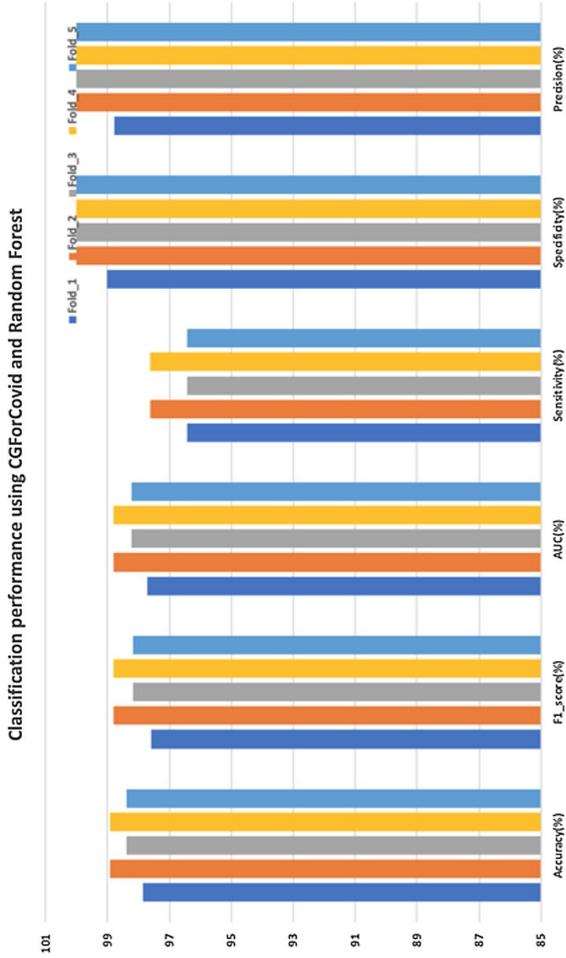


Fig. 7 Classification performances for each of the fivefolds using CGForCovid along with a random forest at the last layer as classifier

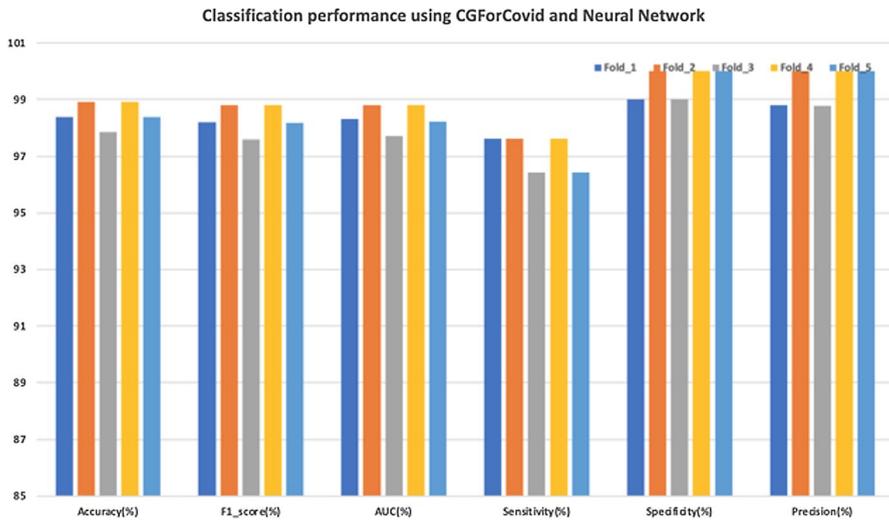


Fig. 8 Classification performances for each of the fivefolds using CGForCovid along with a neural network (i.e., softmax) at the last layer as classifier

in Figures 6, 7 and 8 show the performance variation in terms of different performance metrics (e.g., accuracy, F1-score, AUC, sensitivity, specificity, and precision) across different folds. Figures 6, 7, and 8 show that the performance variation is insignificant if the classifiers are varied that are used after having features extracted using the combination of GA and CNN. One such performance signifies that the GA-optimized CNN can extract influential features from the X-ray images. Table 3 shows the variation in classification performance among different classifiers (i.e., those that were used as the last layer of the CNN) for fivefold cross-validation. As shown in the table, the classifier neural networks can achieve a better performance in terms of accuracy (98.49 ± 0.45), F1-score (98.31 ± 0.50), AUC (98.37 ± 0.46), and Sensitivity (97.14 ± 0.65), while Random Forest can provide better Specificity (99.80 ± 0.44) and Precision scores (99.76 ± 0.55). Although neural networks can improve performance, the performance differences among these classifiers are negligible (especially between neural network and random forest). Moreover, the small standard deviation (e.g., for neural network the standard deviation for accuracy is 0.45) across the fivefold cross-validation scores indicates that the proposed model not only provide a very impressive classification performance for COVID-19 identification but also is a steady system.

Figure 9 shows the normalized confusion matrices of the models. The row of the matrix represents the instances in the actual class, while the column represents the instances in a predicted class. As the confusion matrix shows, no non-COVID cases were classified as COVID cases and only 12 out of 408 patients were misclassified as non-COVID cases. Certainly, such high classification accuracy would encourage physicians/radiologists/professionals to use the proposed automated COVID classifier. Figure 10 shows the ROC curve for the fivefold cross-validation test scheme. As

Fig. 9 Confusion Matrix for CGForCovid model

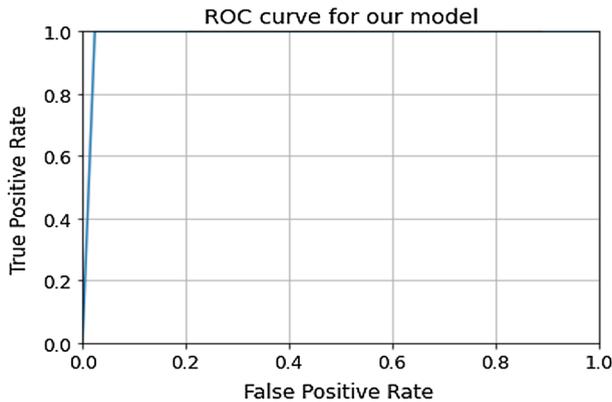
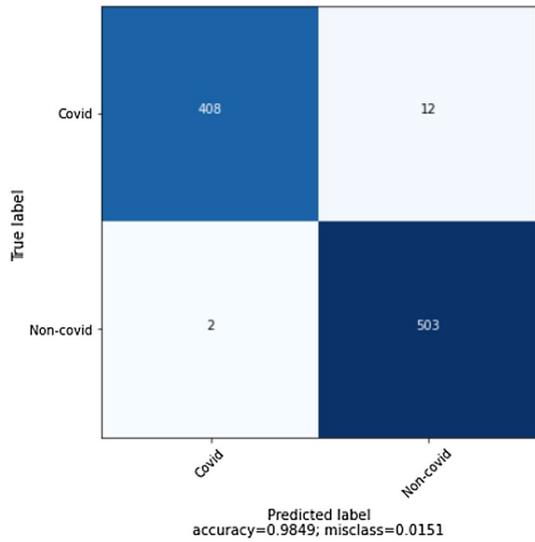


Fig. 10 CGForCovid model ROC Curve

usual, the curve is close to the point (0,1.0), which reveals the effectiveness of the proposed classifier.

5 Comparison with existing work

Table 4 reports the COVID-19 classification performances achieved in several previous studies. As shown in the table, [2] used transfer learning with VGG-19 feature extraction model for input images, achieving an accuracy of 93.48%. Our proposed model’s performance is 5% better than that of the model of [2]. Similarly, our model is at least 6.08% better than that of Wang and Wong [30], 3.1% better than that of

Table 2 Results of CGForCovid model

Features extracted from CNN block number	Classifier	Accuracy (%)	F1_score (%)	AUC (%)	Sensitivity (%)	Specificity (%)	Precision (%)
2	Neural network	96.7568	96.2963	96.4286	92.85714	100	100
	Decision tree (entropy)	80.5405	78.8235	80.475	79.7619	81.1881	77.9070
	Random forest	92.4324	91.25	91.9672	86.9048	97.0297	96.0526
3	Neural network	97.2973	96.9697	97.124	95.2381	99.0099	98.7654
	Decision tree (entropy)	75.1351	71.25	74.5226	67.85714	81.18812	75
	Random forest	93.5135	92.4050	92.9573	86.9047	99.0099	98.64865
4	Neural network	97.8378	97.5609	97.619	95.2381	100	100
	Decision tree (entropy)	84.3243	81.0457	83.4394	73.8095	93.0693	89.8551
	Random forest	95.1351	94.4099	94.743	90.4762	99.0099	98.7013
5	Neural network	97.8378	97.6190	97.8194	97.61905	98.0198	97.6191
	Decision tree (entropy)	97.8378	97.5609	97.619	95.2381	100	100
	Random forest	98.3784	98.1818	98.2143	96.4286	100	100
6	Neural network	97.8378	97.6191	97.8194	97.6191	98.0198	97.6191
	Decision tree (entropy)	97.8378	97.5904	97.7192	96.4286	99.0099	98.7805
	Random forest	97.8378	97.5609	97.6190	95.2381	100	100
7	Neural network	98.9189	98.7952	98.809524	97.619	100	100
	Decision tree (entropy)	98.9189	98.8095	98.9097	98.8095	99.0099	98.8095
	Random forest	98.9189	98.7952	98.8095	97.6191	100	100

Table 3 Classification results comparison with different classifiers at the last layer

Classifier	Accuracy (%)	F1_score (%)	AUC(%)	Sensitivity (%)	Specificity (%)	Precision (%)
	(Mean \pm Stdev)					
CGForCovid along with decision tree	97.19 \pm 0.71	96.87 \pm 0.79	97.07 \pm 0.71	95.71 \pm 1.81	98.42 \pm 1.66	98.10 \pm 1.96
CGForCovid along with random forest	98.49 \pm 0.45	98.31 \pm 0.51	98.35 \pm 0.46	96.91 \pm 0.65	99.80 \pm 0.44	99.76 \pm 0.55
CGForCovid along with neural networks	98.49 \pm 0.45	98.31 \pm 0.50	98.37 \pm 0.46	97.14 \pm 0.65	99.60 \pm 0.54	99.52 \pm 0.66

Table 4 Comparison with other existing studies

Model	Number of Cases	Method Used	Accuracy (%)
[21] *	224 COVID-19 (+) 700 Pneumonia 504 Healthy	VGG-19	93.48
[30]*	53 COVID-19 (+) 5526 COVID-19 (-) 8066 Healthy	COVID-Net	92.4
[24]*	25 COVID-19 (+) 25 COVID-19 (-)	ResNet50+ SVM	95.38
[12]*	25 COVID-19 (+) 25 Normal (-)	COVIDX-Net	90.0
[18]*	50 COVID-19 (+) 50 COVID-19 (-)	Deep CNN ResNet-	50.98
[26]*	777 COVID-19 (+) 708 Healthy	DRE-Net	86
[31]*	195 COVID-19 (+) 258 COVID-19 (-)	M-Inception	82.9
[39]*	313 COVID-19 (+) 229 COVID-19 (-)	UNet + 3D Deep Network	90.8
[38]*	219 COVID-19 (+) 224 Viral pneumonia 175 Healthy	ResNet + Location Attention	86.7
DarkCovidNet by [20]*	125 COVID-19 (+) 500 No-Findings	DarkCovidNet	98.08
DarkCovidNet by [20]	420 COVID-19 (+) 500 No-Findings	DarkCovidNet	96.54
Proposed Study	420 COVID-19 (+) 500 No-Findings	GCForCOVID	98.48

Table 5 List of parameters to design and develop DarkCovidNet-19 Model

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool			
Softmax			

Sethy and Behra [24], and 8.48% better than that of Hemdan et al. [12]. As the table reports, outperformed all others that have recently been proposed in terms of classification performance. It should be noted that all the classification accuracies with ‘*’ symbols are reported by the respective studies. Hence, the comparison might not be fair. For a fair comparison, the same dataset and data distribution among the fivefolds should be used. In our experiment, we implemented the same deep learning neural network that was used in DarkCovidNet [20]. Table 5 lists the parameters used to design and develop DarkCovidNet. The penultimate row in Table 4 shows the classification performance achieved using DarkCovidNet [20]. As the last two rows in the table indicate, the proposed model outperformed DarkCovidNet [20] by 1.94%. This performance improvement serves as empirical proof that the proposed model provides a better deep neural network structure compared to the structure of DarkCovidNet [20] for classifying COVID-19 cases.

6 Conclusion and future directions

Monitoring and tracking patient health status is a critical target in the ongoing COVID-19 pandemic. The advent of modern and efficient computational intelligence techniques like deep learning neural networks and high-performance distributed computing facilities can aid in automatically tracking and detecting COVID-19 cases. This study proposes a new model for accurately classify and diagnosing COVID-19 using state-of the art deep learning models, especially GAs and CNNs. The proposed CGForCovid model will help to reduce clinician workload significantly during the pandemic and will work as a decision support system. In the future, the model can be extended to help rehabilitate affected patients in a timely manner. In this regard, we plan to implement the proposed framework as a protocol and develop a Raspberry Pi-based device to be deployed at a number of hospitals.

Acknowledgements This work was supported by King Saud University, Riyadh, Saudi Arabia, through Researchers Supporting Project number RSP-2021/18 and MEIF-SCI FY2021 grant of University of Maine System (UMS), USA.

Declarations

Ethical approval It does not apply.

Conflicts of interest The authors declare that they have no conflict of interest.

Informed consent It does not apply.

References

1. Akada T, Fujimori K (2021) Designing microwave circuits using genetic algorithms accelerated by convolutional neural networks. In: 2020 50th European Microwave Conference (EuMC), IEEE, pp 61–64
2. Apostolopoulos ID, Mpesiana TA (2020) Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Phys Eng Sci Med* 43(2):635–640
3. Bakhshi A, Noman N, Chen Z, Zamani M, Chalup S (2019) Fast automatic optimisation of CNN architectures for image classification using genetic algorithm. In: 2019 IEEE congress on evolutionary computation (CEC), IEEE, pp 1283–1290
4. Blanco R, Malagón P, Cilla JJ, Moya JM (2018) Multiclass network attack classifier using cnn tuned with genetic algorithms. 2018 28th International symposium on power and timing modeling, Optimization and Simulation (PATMOS), IEEE, pp 177–182
5. Cabada RZ, Rangel HR, Estrada MLB, Lopez HMC (2020) Hyperparameter optimization in CNN for learning-centered emotion recognition for intelligent tutoring systems. *Soft Comput* 24(10):7593–7602
6. Chatterjee R, Maitra T, Islam SH, Hassan MM, Alamri A, Fortino G (2019) A novel machine learning based feature selection for motor imagery eeg signal classification in internet of medical things environment. *Future Gener Comput Syst* 98:419–434
7. Article published on News18 (2020) Know the result time of COVID-19 rt-pcr test and why it takes long. <https://www.news18.com/news/india/know-the-result-time-of-covid-19-rt-pcr-test-and-why-it-takes-long-3142547.html>

8. Falaschi Z, Danna PS, Arioli R, Pasché A, Zagaria D, Percivale I, Tricca S, Barini M, Aquilini F, Andreoni S et al (2020) Chest CT accuracy in diagnosing COVID-19 during the peak of the Italian epidemic: A retrospective correlation with RT-PCR testing and analysis of discordant cases. *European journal of radiology* 130(109):192
9. Fang Y, Zhang H, Xie J, Lin M, Ying L, Pang P, Ji W (2020) Sensitivity of chest CT for COVID-19: comparison to RT-PCR. *Radiology* 296(2):E115–E117
10. Fawcett T (2006) An introduction to ROC analysis. *Pattern Recognit Letters* 27(8):861–874
11. Hassan MR, Hassan MM, Altaf M, Yeasar MS, Hossain MI, Fatema K, Shaharin R, Ahmed AF (2021) B5g-enabled distributed artificial intelligence on edges for COVID-19 pandemic outbreak prediction. *IEEE Netw* 35(3):48–55
12. 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:200311055*
13. Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y, Zhang L, Fan G, Xu J, Gu X et al (2020) Clinical features of patients infected with 2019 novel coronavirus in Wuhan, china. *The lancet* 395(10223):497–506
14. Jaderberg M, Dalibard V, Osindero S, Czarnecki WM, Donahue J, Razavi A, Vinyals O, Green T, Dunning I, Simonyan K, et al. (2017) Population based training of neural networks. *arXiv preprint arXiv:171109846*
15. Joseph Paul Cohen LD Paul Morrison (2020) Covid-19 image data collection. *arXiv preprint arXiv:200311597*
16. Lei S, Hu H, Chen B, Lin Z, Tian J, Yang W, Tang P, Qiu X (2021) Analytical scannable-shaped beam pattern synthesis via superposition principle. *IET Microw, Antennas Propag* 15(6):600–605
17. Loussaief S, Abdelkrim A (2018) Convolutional neural network hyper-parameters optimization based on genetic algorithms. *Int J Adv Comput Sci Appl* 9(10):252–266
18. 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:200310849*
19. Ng MY, Lee EY, Yang J, Yang F, Li X, Wang H, Lui MMs, Lo CSY (2020) Imaging profile of the COVID-19 infection: radiologic findings and literature review. *Radiol Cardiothorac Imagin* 2(1):e200034
20. 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(103):792
21. Panwar H, Gupta P, Siddiqui MK, Morales-Menendez R, Singh V (2020) Application of deep learning for fast detection of COVID-19 in X-rays using nCOVnet. *Chaos, Solitons Fractals* 138(109):944
22. Ranney ML, Griffith V, Jha AK (2020) Critical supply shortages—the need for ventilators and personal protective equipment during the COVID-19 pandemic. *New Engl J Med* 382(18):e41
23. Sajja PS (2021) Examples and applications on genetic algorithms. In: *Illustrated Computational Intelligence*, Springer, pp 155–189
24. Sethy PK, Behera SK, Ratha PK, Biswas P (2020) Detection of coronavirus disease (COVID-19) based on deep features and support vector machine. *Int J Math, Eng Manag Sci*
25. Singhal T (2020) A review of coronavirus disease-2019 (COVID-19). *The Indian J Pediatr* 87(4):281–286
26. Song Y, Zheng S, Li L, Zhang X, Zhang X, Huang Z, Chen J, Wang R, Zhao H, Zha Y et al (2021) Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with ct images. *IEEE/ACM Trans Comput Biol Bioinform*
27. Sun Y, Xue B, Zhang M, Yen GG, Lv J (2020) Automatically designing CNN architectures using the genetic algorithm for image classification. *IEEE Trans Cybern* 50(9):3840–3854
28. Toraman S, Alakus TB, Turkoglu I (2020) Convolutional capsnet: A novel artificial neural network approach to detect covid-19 disease from x-ray images using capsule networks. *Chaos, Solitons Fractals* 140(110):122
29. Waheed A, Goyal M, Gupta D, Khanna A, Al-Turjman F, Pinheiro PR (2020) Covidgan: data augmentation using auxiliary classifier gan for improved COVID-19 detection. *IEEE Access* 8:91916–91923
30. Wang L, Lin ZQ, Wong A (2020) Covid-net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest x-ray images. *Sci Rep* 10(1):1–12

31. Wang S, Kang B, Ma J, Zeng X, Xiao M, Guo J, Cai M, Yang J, Li Y, Meng X, et al. (2021) A deep learning algorithm using CT images to screen for corona virus disease (COVID-19). *Euro Radiol* pp 1–9
32. Wang Z, Sobey A (2020) A comparative review between genetic algorithm use in composite optimisation and the state-of-the-art in evolutionary computation. *Composite Struct* 233(111):739
33. Wu F, Zhao S, Yu B, Chen YM, Wang W, Song ZG, Hu Y, Tao ZW, Tian JH, Pei YY et al (2020) A new coronavirus associated with human respiratory disease in china. *Nature* 579(7798):265–269
34. Wu JMT, Tsai MH, Huang YZ, Islam SH, Hassan MM, Alelaiwi A, Fortino G (2019) Applying an ensemble convolutional neural network with savitzky-golay filter to construct a phonocardiogram prediction model. *Appl Soft Comput* 78:29–40
35. Xiao X, Yan M, Basodi S, Ji C, Pan Y (2020) Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm. *arXiv preprint arXiv:200612703*
36. Xiao X, Huang H, Wang W (2021) Underwater wireless sensor networks: An energy-efficient clustering routing protocol based on data fusion and genetic algorithms. *Appl Sci* 11(1):312
37. Xie L, Yuille A (2017) Genetic CNN. In: *Proceedings of 2017 IEEE International Conference on Computer Vision, Venice, Italy, IEEE*, pp 1388–1397
38. Xu X, Jiang X, Ma C, Du P, Li X, Lv S, Yu L, Ni Q, Chen Y, Su J et al (2020) A deep learning system to screen novel coronavirus disease 2019 pneumonia. *Engineering* 6(10):1122–1129
39. Zheng C, Deng X, Fu Q, Zhou Q, Feng J, Ma H, Liu W, Wang X (2020) Deep learning-based detection for COVID-19 from chest CT using weak label. *MedRxiv*

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

Authors and Affiliations

Md Rafiul Hassan¹ · Walaa N. Ismail² · Ahmad Chowdhury³ · Sharara Hossain⁴ · Shamsul Huda⁵ · Mohammad Mehedi Hassan⁶ 

✉ Mohammad Mehedi Hassan
mmhassan@ksu.edu.sa

Md Rafiul Hassan
md.hassan@maine.edu

Walaa N. Ismail
wlaanagy@gmail.com

Ahmad Chowdhury
ahmadchowdhury@iut-dhaka.edu

Sharara Hossain
shararhossain@iut-dhaka.edu

Shamsul Huda
shamsul.huda@deakin.edu.au

¹ College of Arts and Sciences, University of Maine at Presque Isle, Presque Isle ME04769, USA

² Faculty of Computers and Information, Minia University, Minia 61519, Egypt

³ Imagine Consulting Services LLC, Dayton, NJ, USA

⁴ Simply Retrofits, Ontario, Canada

⁵ School of Information Technology, Deakin University, Melbourne, Australia

⁶ Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia