

## Research Article

# Study on Convolutional Neural Network to Detect COVID-19 from Chest X-Rays

Azher Uddin <sup>1</sup>, Bayazid Talukder <sup>1</sup>, Mohammad Monirujjaman Khan <sup>1</sup>  
and Atef Zaguia <sup>2</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, North South University, Bashundhara, Dhaka 1229, Bangladesh

<sup>2</sup>Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia

Correspondence should be addressed to Mohammad Monirujjaman Khan; [monirujjaman.khan@northsouth.edu](mailto:monirujjaman.khan@northsouth.edu)

Received 10 July 2021; Revised 23 July 2021; Accepted 26 August 2021; Published 11 September 2021

Academic Editor: Dilbag Singh

Copyright © 2021 Azher Uddin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The world is facing a pandemic due to the coronavirus disease 2019 (COVID-19), named as per the World Health Organization. COVID-19 is caused by the virus called severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), which was initially discovered in late December 2019 in Wuhan, China. Later, the virus had spread throughout the world within a few months. COVID-19 has become a global health crisis because millions of people worldwide are affected by this fatal virus. Fever, dry cough, and gastrointestinal problems are the most common signs of COVID-19. The disease is highly contagious, and affected people can easily spread the virus to those with whom they have close contact. Thus, contact tracing is a suitable solution to prevent the virus from spreading. The method of identifying all persons with whom a COVID-19-affected patient has come into contact in the last 2 weeks is called contact tracing. This study presents an investigation of a convolutional neural network (CNN), which makes the test faster and more reliable, to detect COVID-19 from chest X-ray (CXR) images. Because there are many studies in this field, the designed model focuses on increasing the accuracy level and uses a transfer learning approach and a custom model. Pretrained deep CNN models, such as VGG16, InceptionV3, MobileNetV2, and ResNet50, have been used for deep feature extraction. The performance measurement in this study was based on classification accuracy. The results of this study indicate that deep learning can recognize SARS-CoV-2 from CXR images. The designed model provided 93% accuracy and 98% validation accuracy, and the pretrained customized models such as MobileNetV2 obtained 97% accuracy, InceptionV3 obtained 98%, and VGG16 obtained 98% accuracy, respectively. Among these models, InceptionV3 has recorded the highest accuracy.

## 1. Introduction

The current coronavirus disease 2019 (COVID-19) pandemic is very lamentable because the second wave seems to be more dangerous than the first wave. India is one of the most affected countries in the second wave of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The USA and Brazil are also two vulnerable countries as they have not recuperated from the first wave. On 26 April 2021, the total number of infected people in India was 360,960 and is increasing rapidly [1]. This is distressing for Bangladesh because of the close geographical location between these countries, and the Indian variant of SARS-CoV-2 is more dangerous than the other variants. The virus is spreading very fast and can be contracted at all ages, which can lead to

serious illness. As a highly contagious viral disease caused by SARS-CoV-2, COVID-19 has wreaked havoc on the world's demography, killing over 2.9 million people globally, making it the most significant global health epidemic since the 1918 influenza pandemic. Patients older than 60 years, as well as those with medical problems, should be considered at a higher risk of being infected by SARS-CoV-2 [2]. According to the estimates of the World Health Organization, there are approximately 167,011,807 COVID-19 cases worldwide [3]. When this virus attacks the human body, there may be two scenarios: mild and severe. At the onset of the coronavirus infection, one issue is certain: the virus has a negative effect on lung health. As a result, doctors advise patients to keep track of their oxygen levels with an oxygen meter so that any abnormalities can be detected and treated early [4].

Convolutional neural networks (CNNs) are appropriate for this type of problem [5].

The virus normally attacks the lungs in the human body and causes pneumonia in severe cases. Subsequently, it decreases the oxygen level instantly. Because this virus has no cure thus far, the only solution before a vaccine is to prevent the spread of the virus. Therefore, tests and trace is the only solution thus far. Normally, the polymerase chain reaction (PCR) test is widely used in medical science for testing. However, because the number of cases is increasing rapidly, it has become nearly impossible to perform enough tests through PCR, as it is time-consuming and costly. Therefore, an alternative testing is required so that infected people can be identified quickly and quarantined or isolated. To date, some deep learning approaches have been used to identify viruses. However, the results of these deep learning techniques are not sufficient to deal with a medical-related diagnosis system.

COVID-Net, a deep CNN architecture built from chest X-ray (CXR) images for the detection of COVID-19, was introduced in [6]. A research [7] was conducted to classify CXR images into three groups: a transfer learning-based CNN model was used for COVID-19, non-COVID-19, and regular pneumonia. The authors reported that the CNN-based computer-aided diagnosis (CAD) method yielded an overall precision of 94.5%. In [8], a model based on the auxiliary classifier generative adversarial network, called CovidGAN, was created. They added synthetic images created by the CovidGAN. The accuracy was increased to 95%. In [9], to minimise complexity and increase memory efficiency, the authors used iterative pruning. Combining modality-specific information transfer, iterative model pruning, and ensemble learning, they realised an enhanced prediction. In [10], the goal is to develop an automated deep transfer learning-based technique for detecting COVID-19 infection using the extreme version of the Inception model, with 95% accuracy in InceptionV3. The authors in [11] constructed DRE-Net and used ResNet50 as a pretraining model, which is based on MobileNetV2. To extract image details, they used feature pyramid networks. This study was based on image data augmentation. The authors in [12] used AlexNet and GoogLeNet, and two separate DCNNs were used to classify the images as having pulmonary manifestations: tuberculosis or healthy. In [13], five different types of pretrained models were used for the detection of coronavirus-infected patients using chest X-ray radiographs: InceptionV3, Inception-ResNetV2, ResNet152, ResNet50, and ResNet101. In [14], they proposed a classification model that was analysed by VGG16 with an accuracy of 95.9%, whereas in [15], the authors used MobileNetV2 for different data sets along with VGG19, where 97.40% accuracy was obtained in MobileNetV2. Further, fine-tuned ResNet50 has 92% accuracy in the study by Ismael and Şengür [16].

Most studies obtained an accuracy of approximately 90%. Conversely, the present paper study used some pretrained models, that is, ImageNetV2 with customization yielded 98% accuracy and validation accuracy of 97%; VGG16, accuracy of 98% and validation accuracy of 98%;

ResNet50, accuracy of 88% and validation accuracy of 91%; and InceptionV3, accuracy of 98% and validation accuracy of 99%. The designed custom CNN model obtained 97% accuracy and 97% validation accuracy. Clearly, the accuracy percentage of the models used in this study is higher than those of previous studies, making the models in the current study more reliable. Their robustness has been verified through multiple model comparisons, and the scheme can be drawn through the study analysis.

This paper describes a deep learning approach for identifying SARS-CoV-2-infected patients. In the classification, feature extraction in the CNN model can be achieved with high performance. Filter-based feature extraction is used in the CNN model, which can be effective for classification. CNNs can classify images with complex identities. A large number of weight parameters can be reduced using the CNN architecture. Considering these facts, this paper proposes different CNN architectures to detect COVID-19 [17], specifically using and CT scan images. In this study, CXR images have been used as a sample dataset because X-ray equipment is low cost and time-efficient, as well as small and available in almost every clinic. Therefore, fewer developing countries can benefit from this research. This system will help detect coronavirus from CXR images within the shortest possible time. One of the most common radiological tests is chest radiography. CXR analysis involves the detection and localization of thoracic illnesses. This will reduce the pressure on PCR testing, which is costly and time-consuming. False negatives was a common issue in PCR tests results, which is not helpful for the current situation. If we can develop a model with very high accuracy, false result problems can be resolved. If this test could be introduced, more people could be tested in a short time, and thus the spread could be decreased significantly.

The remainder of the paper is organised as follows. The materials and procedures are covered in Section 2, Section 3 presents the findings and analyses, and finally, in Section 4, the conclusions are presented.

## 2. Methods and Materials

The dataset was obtained from the open source Kaggle and GitHub and then merged to prepare a suitable dataset. The dataset contained CXR images of normal patients and patients with COVID-19. A CNN was used for feature extraction. The model has four Conv2D layers, three MaxPooling 2D layers, one flattened layer, two dense layers, and a rectified linear unit activation function. The final dense layer, softmax, was used as the activation function.

In this study, transfer learning is also used so that the accuracy of the designed model can be compared with that of the pretrained model. For pretrained models, MobileNetV2, VGG16, Resnet50, and InceptionV3 were used with some modifications in the final layers, and a head model from the base model. The customized final layers are average pooling, flattening, dense, and dropout. The CNN model is suitable for image feature extraction as it extracts the features of given images and learns and differentiates the images from these features.

**2.1. Materials and Tools.** Python was the ideal programming language for data analysis. Deep learning-based challenges are particularly effective with Python programming because of Python's large library access. To utilise a personal GPU for dataset preprocessing, Anaconda Navigator and Jupyter Notebook were used, as well as Google Colab, to handle large datasets and model training online. They were also used to save all data, code, and work so that it can be retrieved from any GPU using GitHub. Because GitHub has a tracking system for teamwork and code management, it is also suitable for teamwork.

**2.2. Dataset Description.** The dataset used included images of CXR of two classes. One class holds CXR images of COVID-19 patients, and the other holds CXR images of normal patients. These classes were divided into two subclasses. One of them is a training set, and the other is a validation set. The dataset contained 2541 images [18]. In this study, the dataset was split into training and test sets, with a percentage of 75% of the training data and 25% of the test data. Data augmentation has been used to increase the diversity of the data without collecting new data. Figures 1 and 2 show the CXR images of a COVID-19 patient and a normal patient, respectively.

Figure 1 shows a SARS-CoV-2-affected CXR image, and Figure 2 shows a normal CXR image. The height and width of the images in the dataset were initially different from each other. The shapes of the images were fixed in the model.

**2.3. Block Diagram.** In the block diagram of Figure 3, the input is given as a CXR image of a dataset, which has two subsections: COVID-19 patients and normal patients. Before fitting the model, this system underwent preprocessing, such as loading images of a particular size, splitting dataset, and data augmentation techniques. Fitting the model and fine-tuning provided a better accuracy. Plotting the confusion matrix, model loss, and model accuracy helped show how loss and accuracy change over time. Finally, in the output section, if the user provides an image as an input model, we can predict whether it is an image of a COVID-19 patient.

In the block diagram, the overall system is provided in the simplest manner. The decision part of this system is crucial and plays a vital role in this study. The decision is mainly based on the model, which is trained with a large amount of data that are extracted from CXR images.

**2.4. System Architecture.** The system architecture is an overview of the entire system. In this architecture, the input is a CXR image, and the output is a prediction of the image. In this case, it will predict whether the image is COVID-19 affected. The input shape is  $224 \times 224$ , and there are three channels. In the first two layers of the designed architecture, the filter size is 32 with padding, kernel size of 3, and activation function as ReLU. Thereafter, there is the first maxpooling layer, which has a pool size of 2 and strides of 2. The following layer is a flat layer that converts pooled

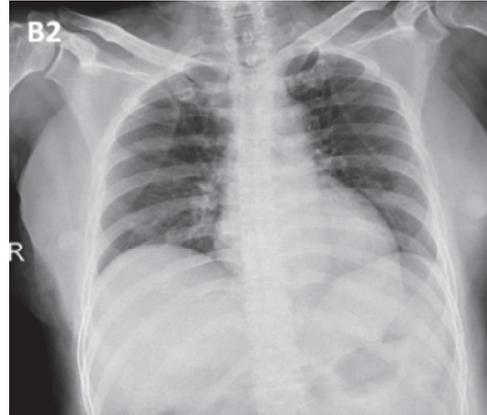


FIGURE 1: X-ray image of a COVID-19 patient.

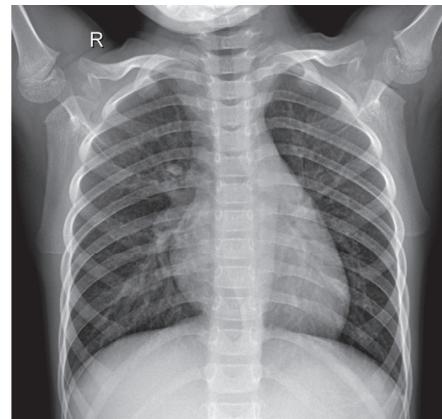


FIGURE 2: X-ray image of a normal patient.

features into a single column. Finally, two dense layers were formed. The first one has ReLU as an activation function, and the least dense layer's activation function is softmax. After preprocessing, the features enter the network. Figure 4 shows the bird's view of the architecture.

**2.4.1. Convolutional Layer.** The convolutional layer is the basic layer of the CNN. This is accountable for determining the design characteristics. The input picture is passed through a filter in this layer. The function map is obtained from the output of the same filters by convolution operation.

The multiplication of sets of weights with the input is performed by a convolution operation. A filter consists of a two-dimensional collection of weights multiplied by an array of input data. A dot product is a type of multiplication that is applied between a filter-sized patch of the input and the filter, which results in a single value. This product is applied between the filter-sized patch of the input and the filter. The filter is smaller than the input, and the same filter is used to multiply the input from different points. The filter is designed as a special technique to identify specific types of features as it systematically covers the entire image.

Assume that the NN input is  $V \in R^{A \times B}$ , where  $A$  denotes the number of features that indicate an input frequency band and  $B$  denotes the total number of input frequency bands.

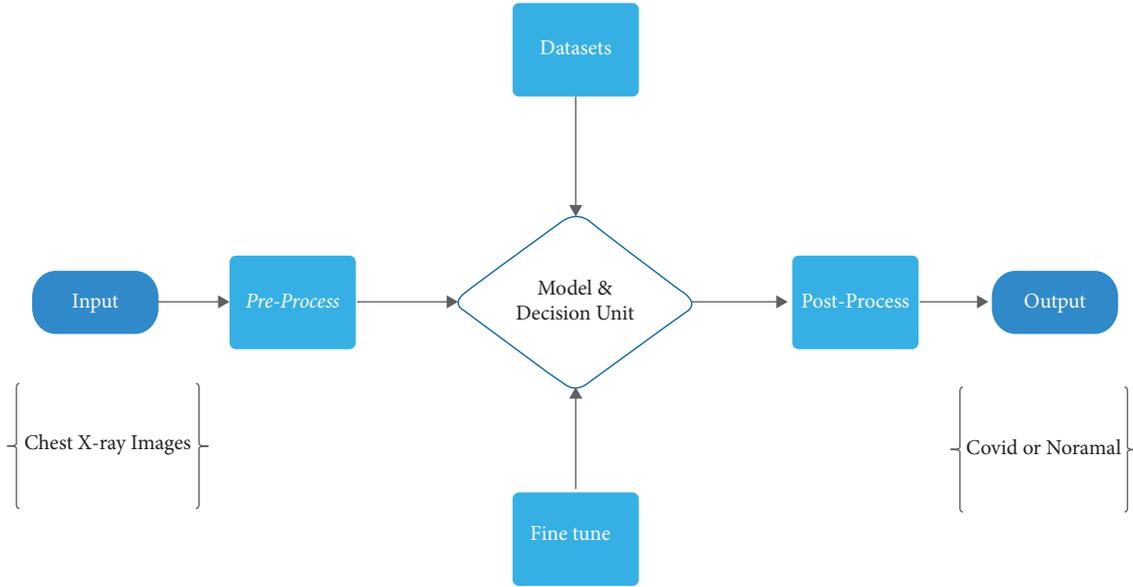


FIGURE 3: Block diagram of the system.

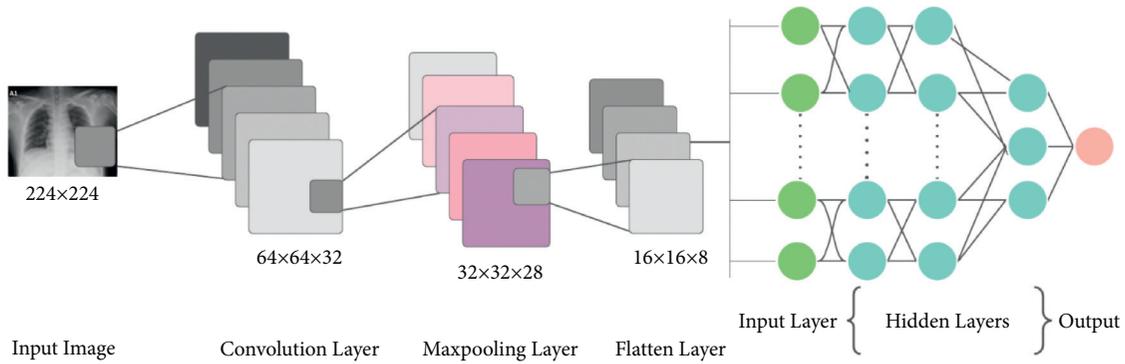


FIGURE 4: System architecture.

The size of the filter bank function vector is represented by  $B$  in the case of filter bank features. Assume that  $v = [v_1 \ v_2 \ \dots \ v_B]$ , where  $v_B$  denotes the function vector for band  $b$ . The activations of the convolution layer can be calculated as

$$h_{j,k} = \theta \left( \sum_{b=1}^s w_{b,j}^T v_{b+k-1} + a_j \right), \quad (1)$$

where  $h_{j,k}$  is the  $j$ th feature map's convolution layer output of the convolution layer band of  $k$ th,  $s$  indicates the filter scale,  $w_{b,j}$  indicates the weight vector for the  $j$ th filter's  $b$ th band,  $a_j$  is the  $j$ th feature map's bias, and  $\theta(x)$  represents the activation function [19].

**2.4.2. Pooling Layer.** The pooling layer summarizes the presence of features by facilitating the downsampling features. It is normally applied after a convolution layer and has some spatial invariance. Two popular pooling methods, average pooling and max pooling, summarize the average

presence of a function, and the most activated presence of a function [20].

In fact, the pooling layer deletes the unnecessary features from the images and makes the image iterate. In average pooling, the layer averages the value of its current view every time. When using maxpooling, the layer selects the maximum value from the filter's current view each time. By using the matrix size specified in each feature map, the maxpooling technique selects only the maximum value, resulting in reduced output neurons. Thus, the size of the image becomes very small, but the scenario remains the same. A pooling layer is important for reducing the number of feature maps and network parameters, and a dropout layer is used to prevent overfitting.

The activation of max pooling can be calculated as follows:

$$p_{j,m} = \max_{k=1}^r (h_{j,(m-1)(n+k)}), \quad (2)$$

where  $p_{j,m}$  is the performance of the pooling layer of the  $j$ th function map and the  $m$ th pooling layer band,  $n$  is the subsampling factor,  $r$  is the pooling scale which is the

number of bands to be pooled together, and  $n$  is the sub-sampling factor.

**2.4.3. Flattened Layer.** The flattened layer is used to convert data from the matrix into a one-dimensional array for use in the fully connected layer and to create a single one-dimensional feature that is both long and narrow. Flattening vectors are an option. Finally, it connects the single vector to the final classification model, which is also known as a fully connected layer [21]. All pixel data are given in one and connect with fully connected layers. Flattening and fully connected layers are the last few steps of the CNN. It is prepared for the next fully linked layer of picture categorization by converting it into a one-dimensional array.

**2.4.4. Fully Connected Layer.** CNNs rely mostly on fully connected layers, which have proven to be quite useful in computer vision image recognition and classification. Convolution and pooling are the starting levels of the CNN process, which breaks down the image into attributes and analyses them separately [22].

In a fully connected layer, each input is connected to all neurons, and the inputs are flattened. The ReLU activation function is commonly used as a fully connected layer. The softmax activation function was used to predict the output images in the last layer of the fully connected layer. The convolutional neural network architecture uses a fully connected layer. These are the last few layers and important layers of the convolutional neural network.

**2.4.5. Pretrained Models.** The scarcity of medical data or datasets is one of the greatest challenges for researchers in medical-related research, and data are one of the most crucial components of deep learning approaches. Data analysis and labelling are both costly and time consuming. Transfer learning provides the advantage of avoiding the requirement for large datasets. The calculations become lower and less costly. Transfer learning is a method in which the pretrained model, which is trained on a large dataset, is transferred to the new model that needs to be trained, including new data that are relatively smaller than required. For a certain task, this process initialized the training of the CNN with a small dataset, including a large-scale dataset that was already trained in the pretrained models [15].

Three CNN-based pretrained models were used to classify CXR images in this investigation. These applied models are MobileNet\_V2, VGG16, and InceptionV3. The CXR images were of two classes. One is normal, and the other is a SARS-CoV-2-infected patient. This study also used a transfer learning method, which can perform with inadequate data by using ImageNet data, and it is also efficient in training time. Figure 5 shows the symmetric system architecture of the transfer learning technique.

As shown in Figure 5, the system architecture has four main parts. The first part is the CXR images, and the second part involved loading a pretrained model. In the second part, three pretrained models are loaded. In the third part, the

loaded pretrained models were modified with the following layers, as shown in Figure 5. Finally, the output part is where the result will be published as COVID-19-infected and normal patients.

MobileNetV2 improves the cutting-edge performance of versatile models on numerous assignments and seat stamps across a range of model sizes. In every line of MobileNetV2, it works as a sequence of  $n$  repeated layers [23]. Depthwise separable is used in MobileNet, which factorizes the normal form into depthwise convolution. This implies a depth of  $1 \times 1$ , which is also known as a pointwise convolution [24]. InceptionV3 is another pretrained model that was used. It normally has a maximum number of pooling layers. VGG16 is also quite helpful because it can extract features at low levels with the help of a small kernel. For CXR images, a small-sized kernel can efficiently extract the features [25]. Because of the insufficient dataset, this study used VGG16 with appropriate layer addition for the final result [26].

### 3. Result and Analysis

After training the model with the train generator, validation generator, step per epoch = 8, and 10 epochs, our model provided 92% accuracy and 98% validation accuracy in the 10th epoch of our model. In the first few epochs, the accuracy of the training was quite low, starting at 55%, and after the 10th epoch, it changed to 92%. The validation accuracy started at 93% and ended at 0.9844 after the 10th epoch. VGG16 has a train accuracy of 98% and validation accuracy of 98%, where the loss of train was 4% and 6% of the validation loss. In ResNet50, this research found 88% training accuracy and 91% validation accuracy. The research found a 29% loss in training and 21% in validation loss. The histories of the accuracy and loss of the four models are given in Table 1.

**3.1. Model Accuracy.** From the plot of accuracy history, it can be observed that the train accuracy increased rapidly after every epoch. In the first epoch, the accuracy was 77%, and after every epoch, it increased. The validation accuracy of the model was 94% and also increased until the last epoch. From the plot of the model accuracy, it can be observed that an increasing line has been drawn for the training accuracy, and for test accuracy, a line which is around the region of 94%–98% accuracy all the time during the epoch. Figures 6(a) and 6(b) show the model accuracy and model loss, respectively.

From the plot of the model loss, it can be assumed that both the lines of training loss and test loss have decreased gradually. After the first epoch, the train loss was 45%, and after 10 epochs, it reached 7%. The validation loss was 16%, and after 10 epochs, it reached 9%. Figure 6 shows a plot of the model loss.

**3.2. Pretrained Model Accuracy and Loss.** In transfer learning, based on pretrained models, MobileNetV2 and the modified head model provided even smoother predictions. In the first epoch, the accuracy was 72%, and the validation

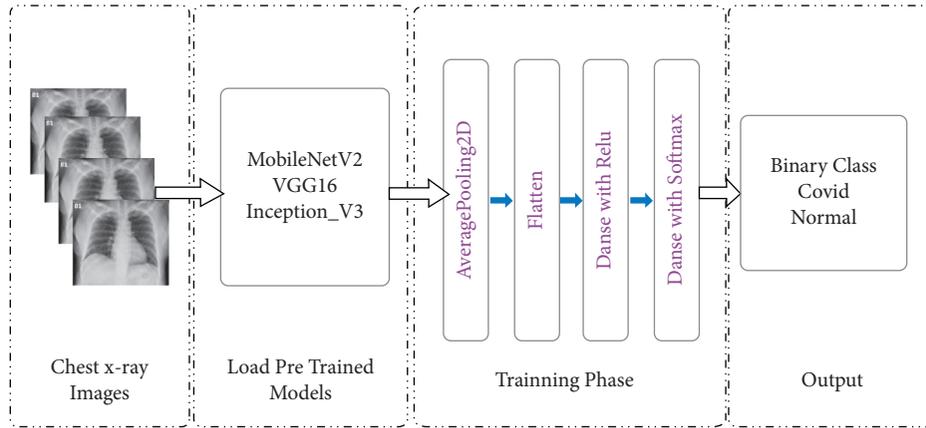


FIGURE 5: System architecture of the pretrained model.

TABLE 1: Histories of the accuracy and loss of the four models.

Model	Accuracy (%)	Validation accuracy (%)	Loss (%)	Validation loss (%)
Custom CNN	97	97	6	8
Modified MobileNetV2	98	97	5	6
VGG16	98	98	4	6
ResNet50	88	91	29	21
InceptionV3	98	98	5	5

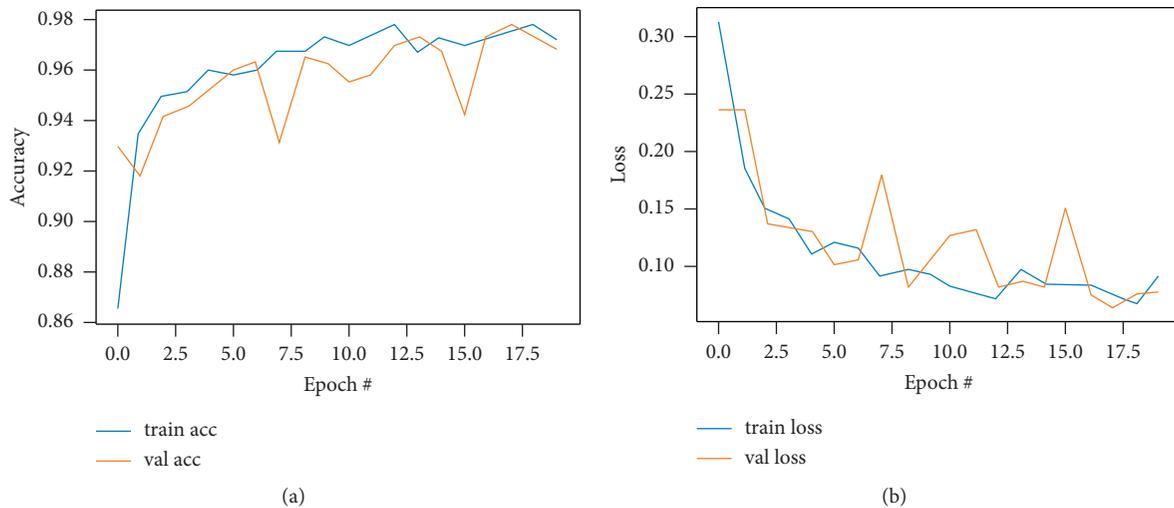


FIGURE 6: (a) Model accuracy and (b) model loss.

accuracy was 96%. After the 8th epoch, the accuracy increased to 98% of the training accuracy and 97% of the validation accuracy. From Figure 7, MobileNetV2 has smoother predictions. Early epochs of VGG16 did not provide satisfactory results. It had a higher margin of training loss. After some epochs, the training loss decreased, and the accuracy increased.

From Figure 8, the pretrained architecture of VGG16 shows that it has more training loss than MobileNetV2 in a few epochs, and after some epochs, it is similar to the result

given in the last few epochs. The accuracy of VGG16 is the same as in the last epoch, as in MobileNetV2. However, MobileNetV2 provided better results from the start of the training. It can be observed from the graph in Figure 9 that ResNet50 resulted in a large loss in the first few epochs and completed its epoch with a large training loss. Compared with other architectures presented in this study, it had a higher training loss and accuracy of 88%.

As shown in Figure 10, InceptionV3 also provided smooth training results. The accuracy was good compared to

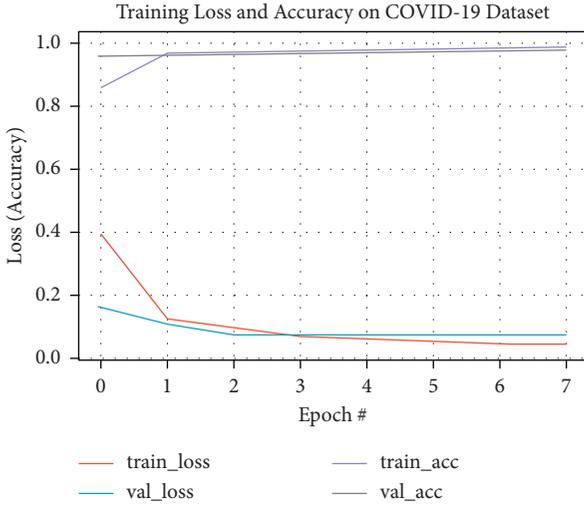


FIGURE 7: Model accuracy and loss (MobileNetV2).

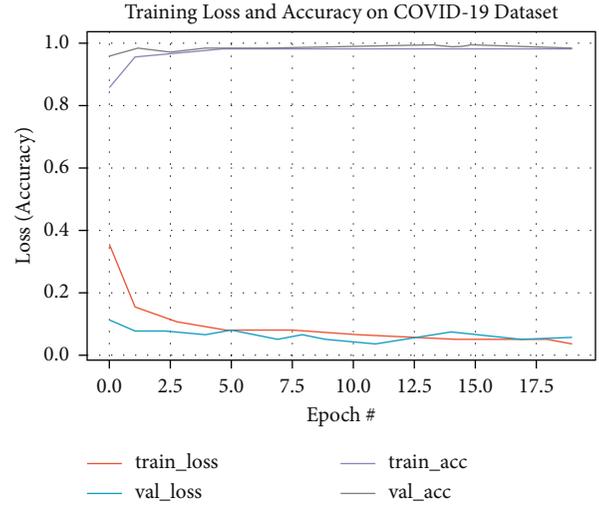


FIGURE 10: Accuracy and loss (InceptionV3).

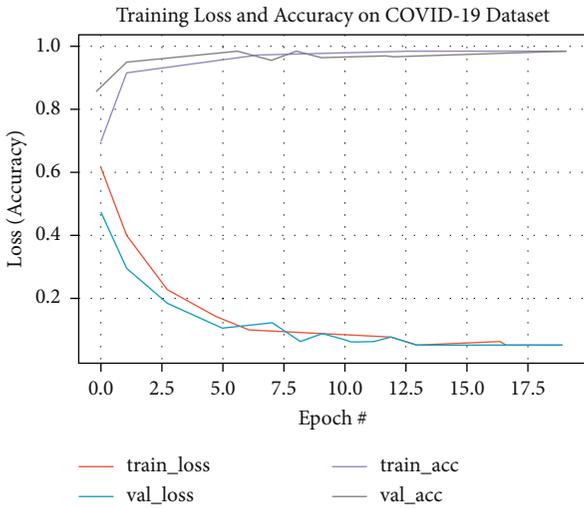


FIGURE 8: Model accuracy and loss (VGG16).

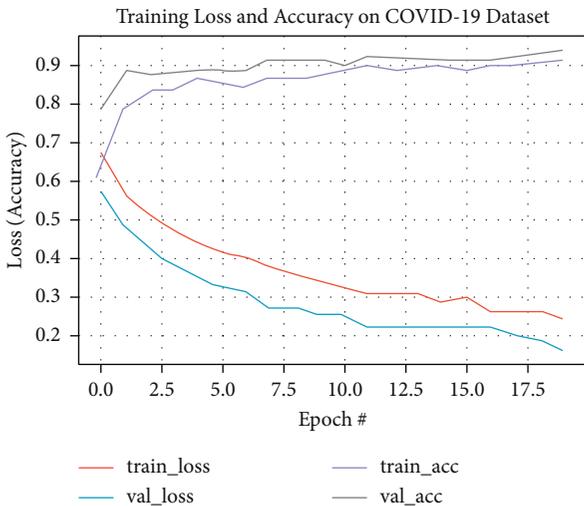


FIGURE 9: Model accuracy and loss (ResNet50).

others, and the training loss was less from the beginning. After a few epochs, it achieved higher accuracy and less training loss. It seems that it has properties similar to those of MobileNetV2. Among these architectures, MobileNetV2 and InceptionV3 provided better results compared with others in accuracy and training loss. After testing the model, COVID-19 and normal CXRs were also detected correctly.

3.3. *Confusion Matrix.* The systems plotted confusion matrix, with columns representing real values and rows representing the predicted values. In a classification model, the summary of the prediction results is known as the confusion matrix. In the confusion matrix, correct and incorrect predictions are summed and split down by class, and for  $n \times n$  matrices FP, FN, TP, and TN are calculated using the following equations [27]:

$$TP_i = a_{ii}, \tag{3}$$

$$FP_i = \sum_{j=1, j \neq i}^n a_{ji}, \tag{4}$$

$$FN_i = \sum_{j=1, j \neq i}^n a_{ij}, \tag{5}$$

$$TN_i = \sum_{j=1, j \neq i}^n \sum_{k=1, k \neq i}^n a_{jk}. \tag{6}$$

Figure 11 shows the confusion matrix.

Three terms are important in error analysis. These are predictions, data, and features. Prediction-based error analysis can be performed using a confusion matrix, where it can be visualised by the percentage of true positives, true negatives, false positives, and false negatives. Data size and nature are also important for the error analysis. Splitting the data accordingly for making trains and tests is also considerable for error analysis because the training and test sets

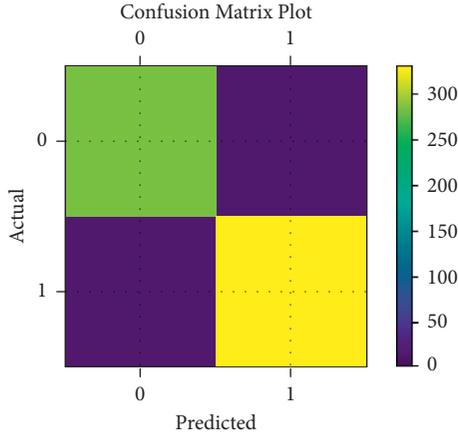


FIGURE 11: Confusion matrix.

may affect the result on a large scale. Features play a vital role in error analysis. Feature engineering and regularisation were also performed to reduce errors.

**3.4. Model Evaluation.** The performance analysis of the models is evaluated based on accuracy, precision, recall, and  $F1$ -score. The performance of the proposed model was assessed using the terms true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The rate of properly detecting the affected photographs from all images is referred to as recall, also called sensitivity. Precision is the opposite of recall. The  $F1$ -score is a combined measure of precision and recall, which shows how often the predicted value is accurate. It is also known as the harmonic mean of  $p$  and  $r$  in mathematics. These equations are given below.

Matrixes can be used to evaluate a system's performance, and after the development of the model, its performance. Accuracy is a measure of how well a model or system works (i.e., the number of times the model correctly predicted the actual outcome) and should be calculated. The mathematical formulas for determining the accuracy are expressed in the following equations [28]:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (7)$$

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{total number of examples}}. \quad (8)$$

The rate of successfully detecting the real value from a set of all values is recognised as recall, also called sensitivity. Recall can be determined using the following expression [29]:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (9)$$

The number of correct identifications is referred to as precision. The number of times the model's positive forecast was right can be calculated, and this is more related to the with the model's positive identification, using the following mathematical formula:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (10)$$

For both recall and precision, a single matrix can be used to summarize the classifier's performance, and the  $F1$ -score is a single matrix that characterizes precision and recall. It is also known as a harmonic means of precision and recall in other mathematical words. The  $F1$ -score is calculated using the following equation:

$$F1 - \text{score} = \frac{2pr}{p + r}. \quad (11)$$

In equations (3) and (4), TP stands for true positive, FP for false positive, and FN for false negative. The letters  $p$  and  $r$  in equation (10) represent precision and recall, respectively. Model evaluation of precision, recall, and  $F1$ -score of our custom CNN model, MobileNetV2, VGG16, and InceptionV3 is given in Table 2. It shows that InceptionV3 and MobileNetV2 have higher precision, recall, and  $F1$ -score values than the other models. InceptionV3 performs exceptionally well among other models, and its accuracy also yielded higher results, which are shown in Table 1.

**3.5. Model Test.** This study also included real testing by providing CXR images as input to the model. When the model is ready, a file is created with an hdf5 extension, which is the actual model that has been created. For this study, four hdf5 files were created for four different types of models. Subsequently, a new notebook file was created as an ipynb extension for the test. Four models were included in the test file, and individual CXR images were provided as input. In Figure 12, the result of the test shows the prediction of whether it is a CXR image of a COVID-19 patient.

In Figure 12, SARS-CoV-2-affected CXR image was provided as an input to the model. Then, the model provided its output, wherein the given CXR image was a COVID-19 patient (i.e., coronavirus infected).

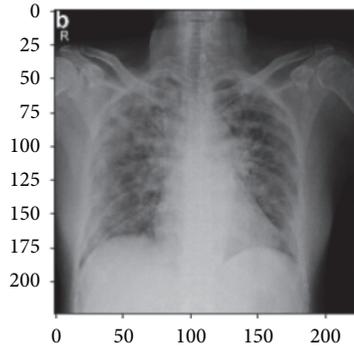
After this test, another CXR image was provided as an input to the model, and the image was normal. Figure 13 shows the results of the normal CXR image.

Figure 13 shows a normal output, and this image was of a normal person, and the model successfully predicted the condition of the patient. Not only were these two tests performed for that study, but all of the models were tested. All models passed the test with actual results.

**3.6. Model Comparison.** The pretrained models of this study (i.e., VGG16, InceptionV3, and MobileNetv2) were compared with some previous models. Compared with the models in the referenced studies, InceptionV3, MobileNetV2, and VGG16 in this study provided better results, accuracy, and efficiency. In the pretrained models, the accuracy increased to a significant level. Compared with those in previous studies [6, 7], the proposed custom CNN has provided better accuracy, precision, and recall. Table 3 shows the comparison of different models and for different datasets.

TABLE 2: Model evaluation.

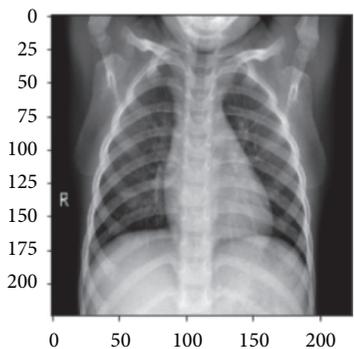
Model	State	Precision	Recall	F1-score
Custom CNN	COVID-19	1.00	0.94	0.97
Custom CNN	Normal	0.95	1.00	0.97
MobileNetV2	COVID-19	0.99	0.96	0.98
MobileNetV2	Normal	0.97	0.99	0.98
VGG16	COVID-19	1.00	0.95	0.97
VGG16	Normal	0.96	1.00	0.98
InceptionV3	COVID-19	1.00	0.98	0.99
InceptionV3	Normal	0.98	1.00	0.99



```
img1 = preprocess_input (img1)
img1 = np.expand_dims (img1, axis = 0)
pred = mdl.predict (img1)
print ("Given sample is" : ,dct [np.argmax (pred)])

Given sample is : Covid
```

FIGURE 12: Screenshot of affected image result.



```
[4] img = preprocess_input (img)
img = np.expand_dims (img, axis = 0)
pred = mdl.predict (img)
print ("Given sample is" : ,dct [np.argmax (pred)])

Given sample is : Normal
```

FIGURE 13: Screenshot of normal image result.

In Table 3, all of the models have given excellent accuracy except ResNet50. In Table 3, the study ResNet50 in [16] achieved higher accuracy than that in this study. From these pretrained models, InceptionV3 and MobileNetV2 have provided smooth results from the beginning and have

TABLE 3: Result comparison.

Reference	Model name	Accuracy (%)	Accuracy in this study (%)
In study [10]	InceptionV3	95	98
In study [14]	VGG16	95.9	98
In study [15]	MobileNetV2	97.4	98
In study [16]	ResNet50	92.5	88
In study [6]	Custom CNN	93	97
In study [7]	Custom CNN	94.5	97

almost the highest accuracy among the other studies. Compared with other articles, InceptionV3 and MobileNetV2 have provided smooth accuracy per epoch, which can be observed in this study. VGG16 also provided 98% accuracy, and the referenced article achieved an accuracy of 95.9%. The referenced articles of InceptionV3 and MobileNetV2 achieved 95% and 97.4% accuracy, respectively. However, this study found 98% and 98% accuracy in InceptionV3 and MobileNetV2, respectively. Regarding the precision, recall, and F1-scores, InceptionV3 clearly performs best among these pretrained and custom models.

#### 4. Conclusion

In this study, CNN models were presented, namely, a full custom CNN, pretrained MobileNetV2, VGG16, and InceptionV3, which are modified in the final layers. The models used in this study obtained almost the same accuracy. The dataset contains 2542 SARS-CoV-2-affected and normal CXR images. The accuracy of the pretrained model was 98%, whereas that of the customized CNN model was 97%. Further work will be performed on a larger dataset and with other pretrained models. These models obtained excellent results in the dataset. MobileNetV2 and VGG16 performed well on these datasets. Classification and feature extraction performed well, and the model checks provided the correct results. These models can detect SARS-CoV-2 using a simple CXR image in the shortest possible time. X-ray technology is currently available and is cost friendly. Thus, it can be a quite efficient method for detecting COVID-19-affected patients. To test and trace the virus, this method way is quick and has no risk of waiting to test for COVID-19 and spreading the virus during that time.

This innovation will greatly change the medical sector. Using this technique, COVID-19 patients can be quickly identified, which may contribute to addressing the current pandemic situation. Chest radiography is comparably safe for obtaining a sample than from the nose of a patient. In the future, this type of technique will help diagnosis. Several deep learning techniques can be used to optimise the parameters to create a robust model, which can help mankind. The metaheuristic-based deep COVID-19 model could also be a good technique to be explored in the future [30]. Some more transfer learning-based models can be added in further development to compare the accuracy and optimisation of parameters [31], as well as a large dataset of normal and COVID-19 patients. The results can be observed by changing the ratio of training and testing data, and further

comparative analysis can be performed. Analysing risk and survival would be an effective study for further development [32]. In the current scenario, as the volume of patients is quite high, deep learning-based COVID-19 detection systems can be helpful. For more accurate results, different types of CNN architectures were introduced. This technology will inspire future generations to address this unwanted situation. In the future, a large number of SARS-CoV-2-affected patients' CXR images can be included in the dataset and training, which can be used as an excellent observation. Other networks of CNN models can be trained to determine the accuracy values and compare them in the context of accuracy, precision, recall, and F1-score.

### Data Availability

The data used to support the findings of this study are freely available at <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>.

### Conflicts of Interest

The authors would like to confirm that there are no conflicts of interest regarding the study.

### Acknowledgments

The authors are thankful for the support from Taif University Researchers Supporting Project (TURSP-2020/114), Taif University, Taif, Saudi Arabia.

### References

- [1] K. Thiagarajan, "Why is India having a covid-19 surge?" *BMJ*, vol. 373, pp. 1–3, 2021.
- [2] M. Cascella, M. Rajnik, A. Aleem, S. C. Dulebohn, and R. D. Napoli, *Features, Evaluation, and Treatment of Coronavirus (COVID-19)*, StatPearls, Treasure Island, FL, USA, 2021, <https://www.ncbi.nlm.nih.gov/books/NBK55477>.
- [3] WHO Coronavirus (COVID-19) Dashboard, WHO, 2021, <https://covid19.who.int/>.
- [4] K. Sharma, *Coronavirus: Distressed Breathing, Lung Involvement in COVID? Here's What Doctors Want You to Know—Times of India*, The Times of India, Bombay, India, 2021, <https://timesofindia.indiatimes.com/life-style/health-fitness/health-news/coronavirus-distressed-breathing-lung-involvement-in-covid-heres-what-doctors-want-you-to-know/articleshow/82195455.cms>.
- [5] R. Sethi, M. Mehrotra, and D. Sethi, "Deep learning based diagnosis recommendation for COVID-19 using chest X-rays images," in *Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 1–4, Coimbatore, India, July 2020.
- [6] L. Wang, Z. Q. Lin, and A. Wong, "COVID-net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images," *Scientific Reports*, vol. 10, no. 1, pp. 1–4, 2020.
- [7] M. Heidary, S. Mirniaharikandehi, and A. Khuzani, "Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms," *International Journal of Medical Informatics*, vol. 144, no. 2, pp. 1–3, 2020.
- [8] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro, "CovidGAN: data augmentation using auxiliary classifier GAN for improved covid-19 detection," *IEEE Access*, vol. 8, pp. 91916–91923, 2020.
- [9] S. Rajaraman, J. Siegelman, P. O. Alderson, L. S. Folio, L. R. Folio, and S. K. Antani, "Iteratively pruned deep learning ensembles for COVID-19 detection in chest X-rays," *IEEE Access*, vol. 8, pp. 115041–115050, 2020.
- [10] N. N. Das, N. Kumar, M. Kaur, V. Kumar, and D. Singh, "Automated deep transfer learning-based approach for detection of COVID-19 infection in chest X-rays," *IRBM*, pp. 2–4. In press, 2020.
- [11] S. Ying, S. Zheng, L. Li et al., "Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, pp. 1–2, 2020.
- [12] P. Lakhani and B. Sundaram, "Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks," *Radiology*, vol. 284, no. 2, pp. 574–582, 2017.
- [13] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks," *Pattern Analysis & Applications*, vol. 24, no. 3, pp. 1207–1220, 2021.
- [14] A. Shelke, M. Inamdar, V. Shah et al., "Chest X-ray classification using deep learning for automated COVID-19 screening," *SN Computer Science*, vol. 2, no. 4, 2021.
- [15] I. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," *Physical and Engineering Sciences in Medicine*, vol. 43, no. 2, pp. 635–640, 2020.
- [16] A. M. Ismael and A. Şengür, "Deep learning approaches for COVID-19 detection based on chest X-ray images," *Expert Systems with Applications*, vol. 164, Article ID 114054, 2021.
- [17] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," in *Proceedings of the 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 844–848, Singapore, December 2014.
- [18] T. Rahaman, *COVID-19 Radiography DatabaseKaggle*, San Francisco, CA, USA, 2020, <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>.
- [19] O. A. Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition," *Indian Science Congress Association*, vol. 11, pp. 73–75, 2013.
- [20] J. Brownlee, "A gentle introduction to pooling layers for convolutional neural networks," *Machine Learning Mastery*, 2021, <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>.
- [21] J. Jeong, "The most intuitive and easiest guide for CNN," *Medium*, 2021.
- [22] S. Saha, "A comprehensive guide to convolutional neural networks — the ELI5 way," *Medium*, 2021.
- [23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and C. L. Chieh, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, Salt Lake City, UT, USA, June 2018.
- [24] A. G. Howard, M. Zhu, B. Chen et al., "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017, <https://arxiv.org/abs/1704.04861>.

- [25] C. Sitaula and M. B. Hossain, "Attention-based VGG-16 model for COVID-19 chest X-ray image classification," *Applied Intelligence*, vol. 51, no. 5, pp. 2850–2863, 2020.
- [26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, <https://arxiv.org/abs/1409.1556v4>.
- [27] M. S. Junayed, A. A. Jeny, S. T. Atik et al., "AcneNet—a deep CNN based classification approach for acne classes," in *Proceedings of the 2019 12th International Conference on Information & Communication Technology and System (ICTS)*, pp. 203–208, Surabaya, Indonesia, July 2019.
- [28] P. D. Ailab, "Evaluation: from precision, recall and  $F$ -measure to ROC, informedness, markedness& correlation," *Machine Learning Technologies*, vol. 2, pp. 37–63, 2011.
- [29] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and  $f$ -score, with implication for evaluation," *Lecture Notes in Computer Science*, Springer, Berlin, Germany, pp. 345–359, 2005.
- [30] M. Kaur, V. Kumar, V. Yadav, D. Singh, N. Kumar, and N. N. Das, "Metaheuristic-based deep COVID-19 screening model from chest X-ray images," *Journal of Healthcare Engineering*, vol. 2021, Article ID 8829829, 9 pages, 2021.
- [31] D. Singh, V. Kumar, V. Yadav, and M. Kaur, "Deep neural network-based screening model for COVID-19-infected patients using chest X-ray images," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 35, no. 3, Article ID 2151004, 2020.
- [32] N. Gianchandani, A. Jaiswal, D. Singh, V. Kumar, and M. Kaur, "Rapid COVID-19 diagnosis using ensemble deep transfer learning models from chest radiographic images," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2020.