

Early prediction of COVID-19 using ensemble of transfer learning[☆]Pradeep Kumar Roy^a, Abhinav Kumar^{b,*}^a Department of Computer Science and Engineering, Indian Institute of Information Technology, Surat, Gujarat, India^b Department of Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India

ARTICLE INFO

Keywords:

Convolutional Neural Network
 Deep Learning
 Classification
 Transfer learning
 Ensemble learning
 IoMT

ABSTRACT

In the wake of the COVID-19 outbreak, automated disease detection has become a crucial part of medical science given the infectious nature of the coronavirus. This research aims to introduce a deep ensemble framework of transfer learning models for early prediction of COVID-19 from the respective chest X-ray images of the patients. The dataset used in this research was taken from the Kaggle repository having two classes—COVID-19 Positive and COVID-19 Negative. The proposed model achieved high accuracy on the test sample with minimum false positive prediction. It can assist doctors and technicians with early detection of COVID-19 infection. The patient's health can further be monitored remotely with the help of connected devices with the Internet, which may be termed as the Internet of Medical Things (IoMT). The proposed IoMT-based solution for the automatic detection of COVID-19 can be a significant step toward fighting the pandemic.

1. Introduction

The Internet of Medical Things (IoMT) is the collection of medical devices and applications that connect to healthcare IT systems through online computer networks [1]. An IoMT-based solution for the automatic detection of COVID-19 is urgently needed given the infectious nature of the disease. If a system is developed that can detect the infected person remotely, then it will be helpful to minimize the spread rate of the coronavirus. The COVID-19 pandemic continues to have a devastating effect on the health and well-being of the global population and has put the world under lockdown in 2020 [2]. The population of China was infected with the Severe Acute Respiratory Syndrome (SARS) virus in February 2003. In 2019, a novel coronavirus contaminated the Chinese city of Wuhan, killing hundreds of people and infecting thousands more in just a few days. Following a December 2019 outbreak in China, the World Health Organization (WHO) classified SARS-CoV-2 as a new form of coronavirus in early 2020. As of 10th May 2021, 156,077,747 cases have been reported worldwide, including 3,256,034 deaths. The second wave has ravaged India with over 3,256,034 cases and 246,116 deaths. COVID-19 can cause acute respiratory distress, multiple organ failure, and death in many cases [3]. Medical experts have suggested social distancing and mass vaccination to curb the spread of the virus because of its high transmissibility.

Fig. 1 shows the top ten infected countries in the world till September 2021. The USA is the most infected country whereas India is present at the second position in the list. Brazil, UK and Russia stand at the 3rd, 4th and 5th position. Fig. 2, highlight the monthly COVID-19 cases since February 2020 to September 2021. During the first wave of COVID-19, September 2020 received the highest case as 2,622,328 whereas, in the second wave, it was increased to 9,016,561 in May 2021. Fig. 3 shows the month of death in India, during the peak of the first wave, i.e., in September 2020, 33,424 deaths are reported, whereas during the second wave,

[☆] This paper is for special section VSI-covid. Reviews were processed by Associate Editor Dr. Jia Hu and recommended for publication.

* Corresponding author.

E-mail addresses: pradeep.roy@iitsurat.ac.in (P.K. Roy), abhinavkumar@soa.ac.in (A. Kumar).

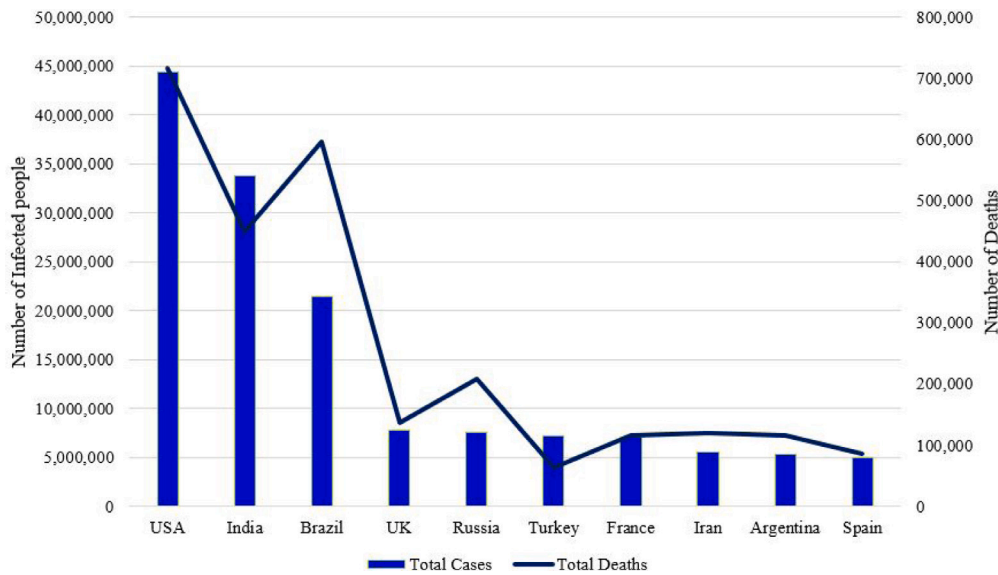


Fig. 1. Top ten infected country worldwide till September 2021.

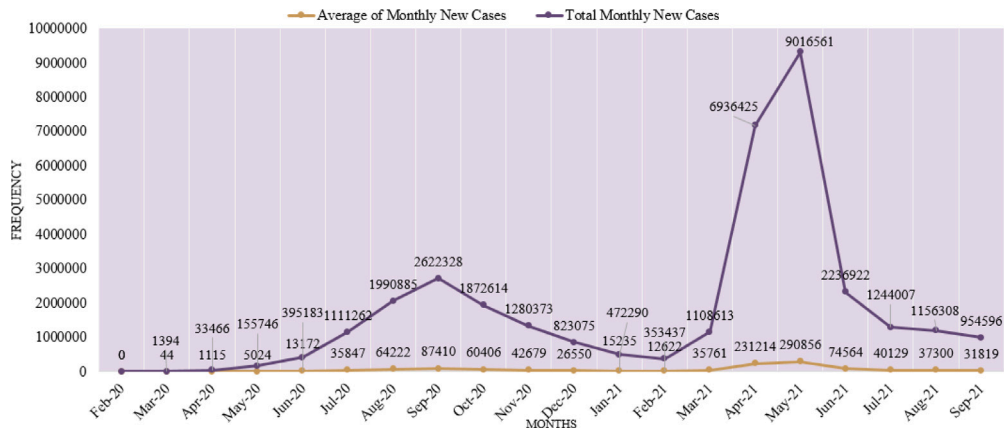


Fig. 2. Monthly COVID-19 cases in India till September 2021.

131,084 death are reported in a single month of May 2021. These statistics indicate the impact of COVID-19 on human beings. Every months thousand of lives are lost due to this virus. Even hospitals are full and unable to occupy the infected persons for treatment. During the second wave of the COVID-19, the shortage of Oxygen in various places are also reported.^{1,2}

The Reverse Transcription Polymerase Chain Reaction (RT-PCR) test is a typical real-time COVID-19 test used to assess the existence of antibodies against the virus. Furthermore, molecular testing of respiratory samples is recommended for diagnosing and laboratory confirmation of COVID-19 infection. However, it takes a long time and is prone to producing false-negative results. In the meantime, several developing countries cannot perform large-scale COVID-19 tests due to the high expense where the immediate diagnosis is based on the appearance of symptoms.

Early detection is critical for controlling and preventing COVID-19. If COVID-19 disease is diagnosed early, the disease's prevalence and the number of people who get infected further will be reduced. However, at the same time, disease detection has been slowed by a shortage of detective resources, and shortcomings in medical equipment and experts [2]. In consequence, the number of patients and casualties rises. One of the most common symptoms of COVID-19 is difficulty in breathing, which can be diagnosed using chest X-ray (CXR) imaging. Opacity related findings in the CXRs have been associated with COVID-19 [3]. This important feature may be useful in developing a Deep Learning model for screening large amounts of radiographic images for

¹ <https://www.dw.com/en/india-covid-oxygen-shortage/a-57425951>, [accessed online: 08-10-2021].

² <https://www.bbc.com/news/world-asia-india-57911638>, [accessed online: 08-10-2021].

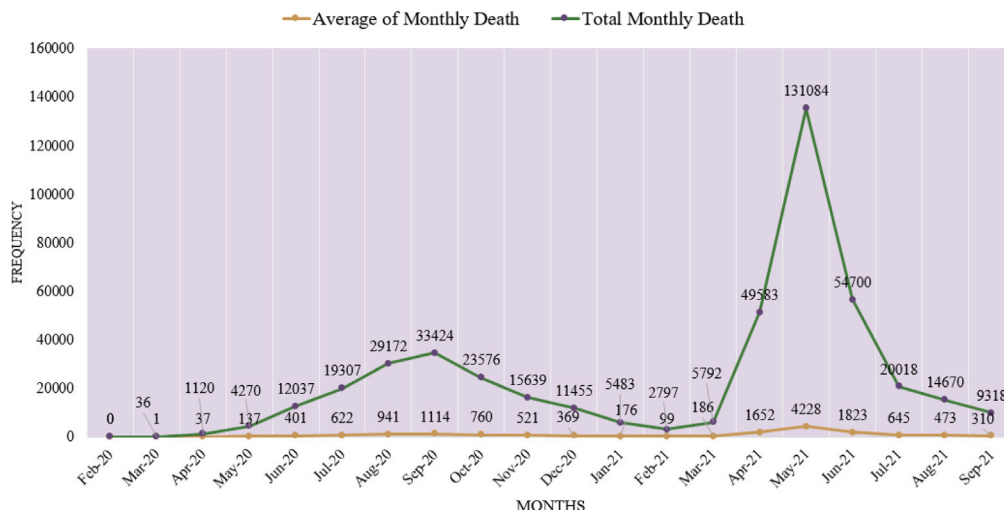


Fig. 3. Monthly COVID-19 death in India till September 2021.

COVID-19 suspect cases. The IoMT plays an important role, where the patient health information will be monitored remotely and reduce the physical load of the medical hospitals.

Deep Neural Networks, especially the Convolutional Neural Network (CNN), have achieved state-of-the-art accuracy in a variety of domains over the last decade, ranging from computer vision to text classification [4]. Deep Learning (DL) have been used for the diagnosis of various diseases ranging from Tumor analysis [5], Cancer classification, Diabetic Retinopathy etc. This study looks into the problem of implementing a Transfer Learning models (a DL based approach) for COVID-19 predictions. Transfer learning architecture has been shown to not only produce state-of-the-art results but also to outperform in many computer vision tasks [6]. The objective of this study is to assess the efficacy of state-of-the-art pre-trained CNN architectures. On Keras platform, currently, 26 transfer learning models' architecture is listed with their size, accuracy, depth of the network, time concerning CPU and GPU.³ Seven particular transfer learning architectures namely (i) ResNet50, (ii) DenseNet201, (iii) InceptionV3, (iv) VGG-16, (v) VGG-19, (vi) Xception, and (vii) MobilenetV2 have been explored for the same based on their performances in recent research on COVID-19.

Many frameworks are reported recently with the individual transfer learning models. However, they suffer from false-positive prediction issues. The false-positive prediction means even though a person is not infected from the COVID-19, the model predicts COVID Positive, which might not be acceptable as it causes frustration. To minimize the false positive predictions and fill the research gap, this research aims to utilize a large number of COVID-19 Positive and COVID-19 Negative samples to build the model. This research used an ensemble learning framework that combined the outcomes of seven transfer learning models. The ensemble model leads to high prediction accuracy, and the false-positive prediction was zero. The proposed ensemble system combines a diverse group of learners to improve the model's consistency and predictive capacity. The ensemble approach resolves the bias-variance trade-off and makes the model more robust to a new dataset. The model can be used at the backend of IoT and smart devices to improve existing testing methodologies. If the proposed ensemble model is used in association with smart devices, it will function as follows: The desktop manager can display the digital X-ray image captured by the machine. The proposed model will be loaded and executed on the desktop manager's onboard processing chip. In the IoMT environment, the model should predict the image as either COVID-19 Positive or COVID-19 Negative and notify that as its final diagnosis. The major contributions are as follows:

- An ensemble of transfer learning algorithms is suggested for reliable COVID-19 patient prediction with a low false-positive rate.
- The performance of the proposed ensemble model is compared with existing state-of-the-art transfer learning models.
- The proposed ensemble framework achieved high prediction accuracy; hence it may be implemented in smart devices for early COVID-19 patient prediction.

The rest of the article is organized as follows: Section 2 discusses the existing research. Section 3 discusses the working of the proposed ensemble framework and highlights the transfer learning models. The experimental outcomes of the proposed methodology are explained in Section 4. Finally, Section 5 concludes the work.

2. Literature review

This section discusses an extensive literature survey about the different types of deep learning models that have been proposed to detect COVID-19. Rahimzadeh and Attar [7] used a concatenated framework of transfer learning models to detect COVID-19 from

³ <https://keras.io/api/applications/>.

CXR images. They used a dataset that contained images belonging to three classes: (i) COVID-19 –180 images, (ii) Pneumonia - 6054 images and (iii) Normal - 8851 images which were collected from GitHub⁴ and Kaggle⁵ respectively in March–April 2020.

Apostolopoulos and Mpesiana [8] used transfer learning models on CXR scans to detect COVID-19. The data set was divided into two sections to evaluate the model: (i) bacterial pneumonia and (ii) viral pneumonia. The images were rescaled for optimal feature extraction. A total of 1428 X-ray images were examined, with 224 images from COVID19 disease, 700 images of common bacterial pneumonia, and 504 images of normal conditions. Models are analyzed using two types of classification accuracies: (i) three-class (covid, pneumonia, and normal) and (ii) two-class (covid and normal). VGG-19 and MobileNetV2 are found to outperform other models in classification, with an accuracy of 98.75% for the two-class classifier, and 97.40% for the three-class classifier.

Loey et al. [9] introduced a novel system that uses deep transfer learning models combined with data augmentation and Conditional Generative Adversarial Network (CGAN) on 742 CT images (divided into COVID-19, Non-COVID-19). They gathered pre-prints from bioRxiv⁶ and medRxiv⁷ that recorded COVID-19 CT patient cases from January 19 to March 25, 2020. The dataset was divided into three categories (i) train, (ii) validation, and (iii) test. ResNet50 was found to be the best classifier for detecting COVID-19 in the CT dataset by achieving the accuracy, sensitivity and specificity values as 81.38%, 88.85% and 81.90%, respectively. Tareh et al. [10] did a comparative analysis of transfer learning models on CXR images to determine the efficacy of pre-trained CNN in the automated diagnosis of COVID-19. VGG-16 was superior to other models, with the highest accuracy and F1-score values of 98.72%, and 97.59%, respectively. Fan et al. [11] also did an analysis of transfer learning models on CXR images for binary classification of COVID-19. Two datasets were used to validate the transfer learning models' performance on X-ray images. The first dataset⁸ contained 74 "normal" and 74 "pneumonia" images for training taken from GitHub. 20 "normal" and 20 "pneumonia" images were used to test the integrity of models. For the second dataset, the same number of images were used, with "normal" scans⁹ and infected "pneumonia" scans.¹⁰ The models were built on MATLAB using a 10-fold procedure and trained on two separate datasets for 10 epochs with 140 iterations. According to the performance assessment metrics, the MobileNetv2 and Xception models could accurately diagnose COVID-19 from CXR images with 95% and 96% accuracy, respectively.

Horry et al. [12] used various models for COVID-19 classification on three imaging modes (i) X-ray, (ii) Ultrasound and (iii) CT Scan. The dataset was collected from open source repositories and analyzed on 11 May 2020. It was found that VGG-19 architecture outperformed other models on ultrasound images and achieved an F1-score value of 0.99. Civit-Masot et al. [13] used deep learning to classify the radiographic images into COVID-19, Healthy, and Pneumonia. Their model achieved an AUC value greater than 90%.

Ko et al. [14] developed a 2D deep learning framework called Fast-Track COVID-19 classification network (FCNet). They classified CT Scanned images into COVID-19 pneumonia, other pneumonia, and non-pneumonia. A dataset consisting of 3993 CT scanned images was collected from various hospitals from January 19 and March 25, 2020. It was observed that ResNet50 outperformed other state-of-the-art transfer learning models and achieved the sensitivity, precision, and accuracy of 97.39%, 99.64%, and 98.67%, respectively.

Azemin et al. [15] used a transfer learning model for COVID-19 prediction on the data collected from source.¹¹ Their binary classification model achieved an accuracy of 71.90%. Narayan et al. [16] also used the transfer learning approach for COVID-19 prediction. The dataset was collected from 6 different publicly available sources with 2504 COVID-19 images and 6807 Non-COVID-19 images downloaded on 16 May 2020. ResNet50 outperforms other Transfer Learning models such as Inception-v3, DenseNet201, and Xception. Using 10-fold cross-validation, it was observed that ResNet50 had an accuracy (%) of 99.34 ± 0.35 .

Wang et al. [17] introduced COVID-Net, which is a deep CNN that detects COVID-19 from CXR images. COVIDx dataset was created by combining 13975 CXR images from 13870 patients from various open access data sources. The images were divided into three labels (i) COVID-19, (ii) Normal, and (iii) Pneumonia. The following hyperparameters were used for training: learning rate= $2e-4$, number of epochs=22, batch size=64, factor=0.7, patience=5. COVID-Net achieves good accuracy by achieving 93.3% test accuracy with a sensitivity of 91% to detect COVID-19.

Jain et al. [18] used pre-trained transfer learning models to predict the class probabilities and classify the radiographic images as belonging to COVID-19, Normal or Pneumonia. There were 6432 total CXR images in the database,¹² which were divided into a training set (5467) and a validation set (965). XceptionNet had the best performance among all the discussed models with an overall accuracy of 97%. Pham [19] did a study to look into the potential of parameter adjustments in the transfer learning of three popular pre-trained CNNs: AlexNet, GoogLeNet, and SqueezeNet, which are known to have the shortest prediction and training iteration times among pre-trained CNNs.

Islam et al. [20] proposed a combined CNN-LSTM model that detects COVID-19 from CXR images. CNN acted as a feature extractor in the model while LSTM for classification. The COVID-19 images were classified with good sensitivity, specificity, and F1-score of 99.3%, 99.2%, and 98.9%, respectively. Many research works have been proposed since the COVID-19 began. However, due to less number of training samples or high false positive prediction, the research continues to boost it. Some limitations of the existing research are highlighted in Table 1. This research aims to address the existing issues and fill the research gap.

⁴ <https://github.com/ieee8023/covid-chestxray-dataset>.

⁵ <https://www.kaggle.com/c/rsna-pneumonia-detection-challenge>.

⁶ <https://www.biorxiv.org>.

⁷ <https://www.medrxiv.org>.

⁸ <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>.

⁹ <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>.

¹⁰ <https://github.com/ieee8023/covid-chestxray-dataset>.

¹¹ <https://github.com/ieee8023/covid-chestxray-dataset> and shorturl.at/uxBPR

¹² <https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia>.

Table 1
Limitations of the existing research.

| Source | Limitations |
|---------------------------------|---|
| Rahimzadeh and Attar [7] | The high performance may be a result of over-fitting because of the less dataset available for training and testing. |
| Apostolopoulos and Mpesiana [8] | They used a small dataset for model, the medical community may determine the likelihood of integrating X-rays into disease diagnosis after analyzing and checking the findings with experts. |
| Taresh et al. [10] | Due to the class imbalance of data, the problem remained to identify which of the classifiers would perform better in confirming COVID-19 cases. Hence an optimal set of hyper parameters along with an increased dataset is required for better generalizability of the network. |
| Fan et al. [11] | Due to low availability of the dataset, the model should not be used for diagnosis and generalizability without consulting with the concerned medical authorities. |
| Ko et al. [14] | The research dataset was derived from the same sources as the training dataset, potentially raising generalizability and over fitting concerns. |
| Azemin et al. [15] | They built the model with a small number of publicly accessible COVID-19 CXR images. |
| Narayan et al. [16] | Their work was undervalues the significance of data augmentation techniques, such as Generative Adversarial Networks (GANs), which generate more training images synthetically even when the COVID-19 dataset is insufficient to train a Deep Learning model from scratch. |
| Wang et al. [17] | One of the most significant bottlenecks is the need for expert radiologists to interpret radiography images. As a result, radiologists are in desperate need of computer-assisted diagnostic systems. |
| Jain et al. [18] | The high accuracy obtained could be cause for concern because it could be due to over fitting. Thus, the model needs to be validated on a large scale public dataset and consulted with the medical fraternity. |
| Pham [19] | The study did not recognize COVID-19 sub-classification into mild, moderate, or extreme disease due to restricted data labeling. Another problem was that each patient only received a single CXR sequence. Because of this data constraint, it is impossible to tell whether patients developed radiographic findings as their illness progressed. |
| Islam et al. [20] | Due to the limited size of the dataset, the network's generalizability must be enhanced. Only the posterior-anterior X-ray view were functional; lateral views and anterior-posterior views were not. The model does not classify COVID-19 disease types (mild, severe), which could be improved with a larger dataset. |

3. Proposed methodology

This research aims to build a system that predicts COVID-19 infected people in the early stage with minimum false positive predictions. The developed system can be embedded with smart devices to remotely monitor the infected people's health using the IoMT facility. The IoMT enabled devices can pass the real-time monitoring information to the medical experts. The steps involved in developing the predictive system is shown in Fig. 4. Total seven transfer learning models (i) ResNet50, (ii) DenseNet201, (iii) InceptionV3, (iv) VGG-16, (v) VGG-19, (vi) Xception, and (vii) MobineNetV2 are experimented separately and found all models having false-positive predictions. Hence, to minimize this problem, an ensemble learning-based framework was developed, utilizing the predictions of individual transfer learning models and providing the final prediction. The outcomes of the ensemble learning framework were superior to those of the individual models with negligible false positive prediction.

The dataset used in this research was collected from <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>, which has images from various sources. Considering the computational resource restraints, a part of the dataset was used for the experiment consisting of 2000 images. The model classified the image into two classes (i) COVID-19 Positive and (ii) COVID-19 Negative, respectively. Sample images of COVID-19 Positive and COVID-19 Negative are shown in Figs. 5 and 6.

The images corresponding to COVID-19 Positive class and COVID-19 Negative class had been converted to .npy files along with their appropriate labels. The .npy file format is NumPy's basic binary file format for storing a single NumPy array on a disk. This way, the shape and the data type information necessary to create the array on a system with different architecture remains intact. This process leads to faster processing of data. All of the images from the source were 299 × 299 pixels in nature, which had been converted to appropriate sizes to suit the respective transfer learning architecture. The code structure appends the data, the images according to their respective classes, the labels and stores it in a standard binary format, .npy file. Threading was used for the parallel execution and utilized the multiprocessing capacity to its optimal usage. A total of 2000 chest X-ray images were used for the experiment. The training, validation, and testing sets have been split into 60%, 20%, 20% ratios, respectively, for optimal model performance. The detailed break-down of the number of images in the data set has been shown in Table 2.

3.1. Background: Transfer learning

This section mainly highlights the configuration and working of the transfer learning models and ensemble frameworks. In recent years, along with technological developments, we have become progressively better at training deep learning neural networks. However, the generalizability issue is still observed when the model is used on different datasets, which has not been seen before.

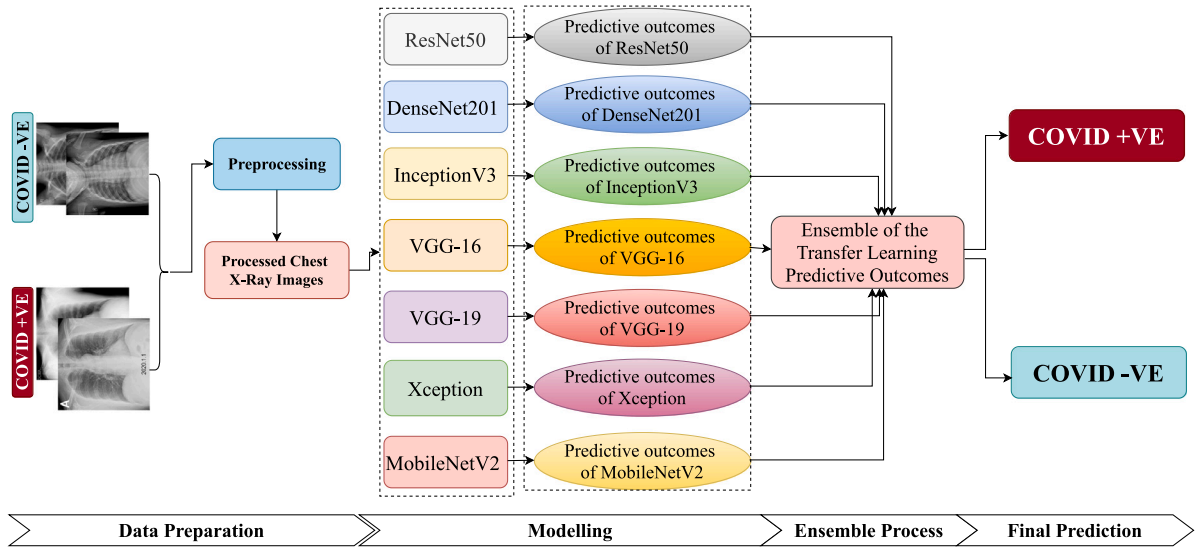


Fig. 4. Proposed Ensemble Framework for COVID-19 Disease Prediction with Chest X-ray.

Table 2

The dataset description.

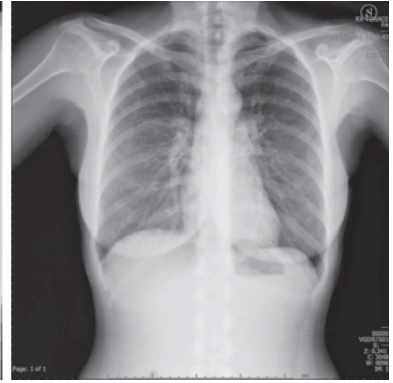
| Class | Train | Test | Validation | Total |
|-------------------|-------|------|------------|-------|
| COVID-19 Positive | 600 | 200 | 200 | 1000 |
| COVID-19 Negative | 600 | 200 | 200 | 1000 |
| Total | 1200 | 400 | 400 | 2000 |



(a) COVID-19 Positive



(b) COVID-19 Positive



(c) COVID-19 Positive

Fig. 5. Chest X-rays of COVID-19 Positive Patient.

Transfer Learning is the art of using the knowledge derived from solving one problem and using it for a different but similar problem. After experimenting with several transfer learning models it was found that the (i) ResNet50, (ii) DenseNet201, (iii) InceptionV3, (iv) VGG-16, (v) VGG-19, (vi) Xception, and (vii) MobineNetV2 have performed comparably better for predicting the COVID-19 instances.

3.1.1.1. ResNet50

ResNet50 is a model having 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer [21]. The general intuition is that with an increasing number of convolutional layers, the performance of the model will improve. However, it has been observed that the model's actual performance degrades in doing so because of over-fitting and vanishing gradient problems. ResNet address this problem by introducing skip connections before the ReLu activation function is applied. Skip Connections address the problem of vanishing gradient by allowing the alternate shortcut path for the gradient to flow through. They also allow the model to learn an identity function that guarantees that the higher layer performs at least and the lower layer, if not better. The input size of the

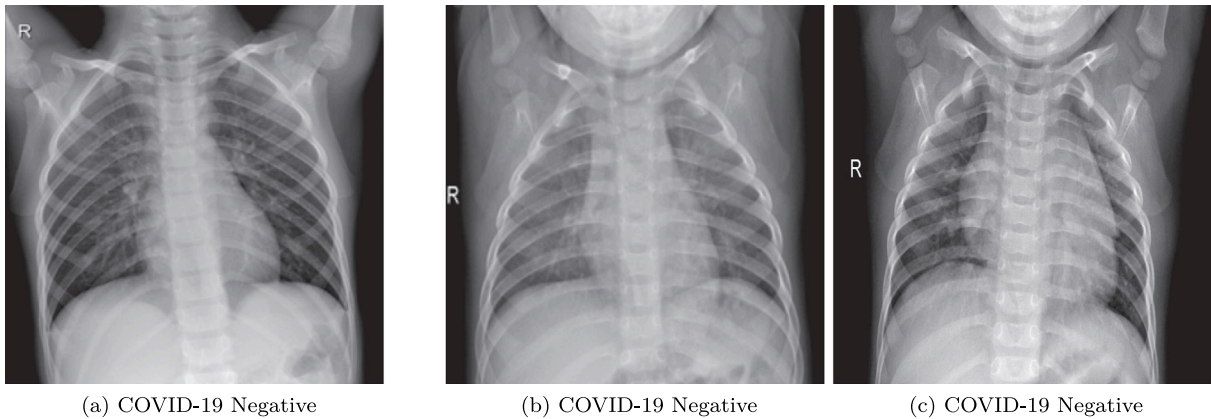


Fig. 6. Chest X-rays of COVID-19 Negative Patient.

images accepted by the ResNet50 model is 224×224 . The ResNet50 model is divided into five stages, each with its own convolution and identity block. Each convolution block has three layers, and each identity block also has three layers. Finally, the network has an Average Pooling layer followed by a fully connected (FC) layer having 1000 neurons. ResNet50V2 applies Batch Normalization and ReLU activation to the input before convolution operation. The second non-linearity in ResNet50V2 is primarily focused on making it an identity mapping. The contribution of the addition operation between the identity mapping and the residual mapping should be transferred directly to the next block for processing.

3.1.2. DenseNet201

To solve the vanishing gradient problem that occurs with the increase in depth of the CNN, DenseNet (Dense Convolutional Network) was introduced [22]. The network is connected in such a way that each layer gets input from the preceding layers and passes the feature maps of its own to the subsequent layers. For a network containing K layers, there would be $(K * (K + 1))/2$ connections. The input size of the images accepted by the DenseNet201 model is 224×224 . The model is deep trained in such a way that along with the feature maps, information that needs to be preserved is also sent. The DenseNet architecture distinguishes between information that is introduced to the network and data that is preserved. They add only a small set of feature maps to the network and keep the remaining unaltered. The classifier makes the decision depending upon the feature maps in the network.

3.1.3. InceptionV3

InceptionNet is important because, before it, the models would stack layers in the network to get better performance [23]. However, very deep networks are prone to overfitting, especially when we have small data to train on. Because of the orientation of images, there can be a huge variation in locating the features. This makes choosing a kernel of the right size difficult. A large kernel can be used to locate global features, whereas small kernels can be used to locate local features. One solution is to move from FC architecture to sparsely connected architectures where the information from different kernels ($1 \times 1, 3 \times 3, 5 \times 5$) is concatenated. The input size of the images accepted by the InceptionV3 model is 299×299 . To detect individual parts of an image, there should be kernel size of varied dimensions. This is where the inception layer comes into play. It allows the internal layers to select which filter size is appropriate for learning the necessary information. In InceptionV2 two 3×3 convolutions are used in place of 5×5 convolutions. because of the reduction of parameters, 3×3 convolutions are computationally better than 5×5 convolutions. They also involve factoring $n \times n$ convolutions into $1 \times n$ or $n \times 1$ convolutions. InceptionV3 is similar to InceptionV2 but involves the use of RMSProp optimizer, 7×7 factorized convolutions and Batch Normalization in auxiliary classifiers to improve the convergence of the network.

3.1.4. VGG-16 and VGG-19

VGG-16 and VGG-19 models were trained and tested with an ImageNet dataset consisting of 1000 classes and more than 14 million images. VGG-16 and VGG-19 are the improved version of AlexNet. The model's input is 224×224 sized three channeled (R, G, B) images, passing to the stacked convolutional layer for processing. The convolution operation takes place with a 3×3 kernel to capture the smallest features from the image. Three FC layers are present at the end of stacked convolution layers. The first and second layer of the FC network consists of 4096 neurons, whereas the third layer has 1000 neurons. The final layer of the network is the softmax layer which predicts the probabilities for all classes.

3.1.5. Xception

Xception model is another CNN-based transfer learning model which involves depth-wise separable convolutions and is widely used for various image processing tasks. The input to the model is a size of 299×299 images [24]. Keras library provided the flexibility to initialize the model's weight. The weight can either be randomly assigned or can be loaded with pre-trained weights of ImageNet. The model supports global max pooling or average pooling to optimize the feature dimensions. The model can be used for binary classification of multiclass classification purposes by changing the number of neurons at the output layer of the network.

3.1.6. MobileNetV2

MobileNetV2 is another lightweight transfer learning model that achieved better performances on various benchmark datasets than the other models. The MobileNetV2 architecture consists of a lesser number of parameters as compared to the existing models and outperforms the existing model. The input size of the images accepted by the MobileNetV2 model is 32×32 . A detailed study about the network configuration and working of the model can be studied from the resource [25].

3.2. Loss function

The purpose of the loss function in deep learning models is to measure the overall performance. The loss value must be minimum for any model to achieve better performance. The objective behind using the loss function is to evaluate the actual predictions of the deep learning models and, if needed, do the weight updation of preceding layers to achieve better performance. Loss functions are mainly categorized in two categories: Probabilistic Loss and Regression Loss.¹³ This research uses a loss function that belongs to the Probabilistic Loss category, namely crossentropy. The crossentropy loss function evaluates the losses of the output layer by considering the probability values of the different classes present at the output layer. The deep learning model classifies the input images into two classes, i.e., COVID-19 Positive and COVID-19 Negative only. Hence, a binary crossentropy function is used. However, for the multiclass classification problem, Keras supported another loss function called Categorical crossentropy. The crossentropy-based loss function is suitable for the inputs with larger boundaries than other loss functions. In parallel, it helps the model to find the saturation point quickly.

Mathematically, it can be defined as follows: Suppose for the inputs $(x_1, x_2, x_3, \dots, x_n)$, the output is $(y_1, y_2, y_1, \dots, y_1)$ respectively. Means, x_i having the actual output as y_i and their output predicted by the model is \hat{y} , where \hat{y} for i th input is calculated using Eq. (1):

$$\hat{y}_i = f(w_i^T * x_i + b_i) \quad (1)$$

Here, \hat{y}_i is the predicted output for i th input, w_i^T is the corresponding weights and b_i is the bias values, f is a non-linear activation function called Rectified Linear Unit(ReLU) which helps to convert the negative value to zero and positive values remain unchanged. The ReLU activation mathematically defined in Eq. (2).

$$ReLU(f) = \max(0, x) \quad (2)$$

Finally, the loss function ϕ is for the binary class (i.e., C=2) is defined in Eq. (3).

$$\phi(\hat{y}^i, y^i) = - \sum_{i=1}^C y^i \log \hat{y}^i \quad (3)$$

The cost function for k training samples is calculated using the Eq. (4).

$$\lambda(w, b) = -\frac{1}{k} \sum_{i=1}^k \phi(\hat{y}^i, y^i) \quad (4)$$

The cost function helps to updated the associated weights and biases of the network.

3.3. Ensemble learning

The ensemble approach involves combining the predictive power of various learners to improve the overall performance and robustness of the model. The error in the predictive capacity of a model can be decomposed into three errors: Bias error of the model, Variance error of the model and variance of the Irreducible error. That is, the model error can be described as: Model Error = Error due to Bias + Error due to Variance + Irreducible Error. The term Bias error is used to describe how much the expected values vary from the actual value on average. A high bias results from underperformance where the model misses important trends during the training phase. Variance indicates the predictive capacity of a model on the same observation. A model tends to overfit with high variance and would perform worse with a validation dataset. The optimal scenario is to reach a minimum of bias error and a minimum of variance error. Various ensemble approaches have been proposed to address this bias variance trade-off. The reducible error (Bias Error and Variance Error) is the element that can be improved. We reduce the quantity when the model learns on a training dataset. We try to get this quantity as close to zero to reduce the overall error in the model's predictive capacity. The irreducible error is the error that cannot be removed. The error is generated because of noise in the observations or outliers in the dataset. The popular Ensemble techniques are (i) Max-voting, (ii) Probability Averaging, and (iii) Weighted Probability average. To provide a final class label to the chest x-ray image, the current study used a probability averaging approach. The probability averaging ensemble approach can be defined mathematically as follows:

Mathematically, the ensemble process can be defined as follows: suppose a total of m transfer learning models are used $(M_1, M_2, M_3, \dots, M_m)$ having n classes. The probability value of each classes for i th input (x_i) with model M_1 can be expressed as:

$$O_{M1}^{(i)} = (p_{1M1}^i, p_{2M1}^i, p_{3M1}^i, \dots, p_{nM1}^i) \quad (5)$$

¹³ <https://keras.io/api/losses/>.

Table 3
Hyper-parameter settings for different transfer learning models.

| Models | Epochs | Activation functions | Batch Size | Loss | Optimizer | Learning rate |
|---|--------|----------------------|------------|---------------------|-----------|---------------|
| VGG-16 (M1) | 100 | Softmax | 16 | Binary crossentropy | Adam | 0.001 |
| ResNet50 (M2) | 100 | Softmax | 16 | Binary crossentropy | Adam | 0.001 |
| VGG-19 (M3) | 100 | Softmax | 16 | Binary crossentropy | Adam | 0.001 |
| Xception (M4) | 100 | Softmax | 16 | Binary crossentropy | Adam | 0.001 |
| InceptionV3 (M5) | 100 | Softmax | 16 | Binary crossentropy | Adam | 0.001 |
| MobileNetV2 (M6) | 100 | Softmax | 16 | Binary crossentropy | Adam | 0.001 |
| DenseNet201 (M7) | 100 | Softmax | 16 | Binary crossentropy | Adam | 0.001 |
| Ensemble (M1 + M2 + M3 + M4 + M5 + M6 + M7) | 100 | Softmax | 16 | Binary crossentropy | Adam | 0.001 |

The probability value of each classes for the same i th input (x_i) with model M_2 can be expressed as:

$$O_{M_2}^{(i)} = (p_{1M_2}^i, p_{2M_2}^i, p_{3M_2}^i, \dots, p_{nM_2}^i) \quad (6)$$

This way, the outcomes of the i th input (x_i) was evaluated using all m models ($O_{M_1}^{(i)}, O_{M_2}^{(i)}, \dots, O_{M_m}^{(i)}$). The final prediction score obtained using probability averaging can be expressed using the Eq. (7)

$$O_{Ensemble} = \left[\frac{\sum_{k=1}^m p_{1Mk}^i}{m}, \frac{\sum_{k=1}^m p_{2Mk}^i}{m}, \frac{\sum_{k=1}^m p_{3Mk}^i}{m}, \dots, \frac{\sum_{k=1}^m p_{nMk}^i}{m} \right] \\ = [o_1, o_2, o_3, \dots, o_m] \quad (7)$$

The final prediction of the input will be decided by finding the maximum values from the predicted values, i.e., $\max(o_1, o_2, o_3, \dots, o_m)$. The predicted class will be compared with the actual label of the input to find the error and accuracy value.

3.4. Model training

As the performance of deep learning-based models is very sensitive to the chosen hyper-parameters, extensive experiments were performed to choose the best-suited hyper-parameters. We varied the learning rate, batch size, epochs to find the best-suited hyper-parameters. The best-suited hyper-parameters of the proposed models are listed in Table 3. To prevent RAM overhead and for optimal utilization of resources, a callback function was explicitly defined with a patience value of 10 to keep a check on the validation loss and cut off the training when the validation loss did not improve over 10 epochs.

4. Results

This section describes the experimental simulations and results. It consists of a description and testing results of the classification model used to classify chest X-ray images. It also discusses the performance of the model on unforeseen data, that is, on validation dataset. The Precision, recall, F1-score, accuracy, and AUC-ROC are the main metrics used to evaluate the performance. These are described as follows (Eq. (8) to (11)):

$$Accuracy = \frac{TP + TN}{TN + FP + TP + FN} \quad (8)$$

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$F1 - score = 2 \times \frac{Precision * Recall}{Precision + Recall} \quad (11)$$

Where a True Positive (TP) indicates a positive sample is correctly classified (correctly predicted COVID-19 Positive cases), a True Negative (TN) occurs when a negative sample is correctly classified (correct classification of COVID-19 Negative). A False Positive (FP) occurs when a negative sample is mistakenly classified as positive (COVID-19 Negative is classified as COVID-19 Positive). A False Negative (FN) occurs when a positive sample is mistakenly classified as negative (COVID-19 Positive is classified as COVID-19 Negative). The AUC-ROC metric is also used to evaluate the performance of classifiers at various threshold settings. The ROC represents the probability curve, and the AUC represents the degree of separability measure. The higher the AUC, the more accurate the model is in forecasting [26].

The experiments were started with the selected transfer learning models on the preprocessed dataset. First, the VGG-16 model was trained and tested without adding any dropout layer in it. The outcomes of the VGG-16 model is shown in Table 4. The model achieved 0.94 precision, recall and F1-score for COVID Positive cases prediction whereas, for COVID Negative, the precision, recall and F1-score value were 0.93. The AUC value for the same model was 0.937657, which indicates the VGG-16 misclassifying both the COVID Positive and Negative instances.

Table 4
Result of the transfer learning models without using Dropout.

| Models | Class | Precision | Recall | F1-score | AUC |
|-------------|-------------------|-----------|--------|----------|-----------|
| VGG-16 | COVID_19 Positive | 0.94 | 0.94 | 0.94 | 0.937657 |
| | COVID_19 Negative | 0.93 | 0.92 | 0.93 | |
| | Weighted Avg. | 0.93 | 0.93 | 0.93 | |
| ResNet50 | COVID_19 Positive | 0.95 | 0.96 | 0.95 | 0.957245 |
| | COVID_19 Negative | 0.94 | 0.95 | 0.95 | |
| | Weighted Avg. | 0.95 | 0.95 | 0.95 | |
| VGG-19 | COVID_19 Positive | 0.95 | 0.92 | 0.93 | 0.9423658 |
| | COVID_19 Negative | 0.96 | 0.92 | 0.93 | |
| | Weighted Avg. | 0.95 | 0.92 | 0.93 | |
| Xception | COVID_19 Positive | 0.92 | 0.91 | 0.92 | 0.920356 |
| | COVID_19 Negative | 0.93 | 0.92 | 0.92 | |
| | Weighted Avg. | 0.92 | 0.92 | 0.92 | |
| InceptionV3 | COVID_19 Positive | 0.96 | 0.94 | 0.95 | 0.951269 |
| | COVID_19 Negative | 0.94 | 0.94 | 0.94 | |
| | Weighted Avg. | 0.95 | 0.94 | 0.95 | |
| MobileNetV2 | COVID_19 Positive | 0.92 | 0.96 | 0.94 | 0.9524859 |
| | COVID_19 Negative | 0.93 | 0.95 | 0.94 | |
| | Weighted Avg. | 0.93 | 0.95 | 0.94 | |
| DenseNet201 | COVID_19 Positive | 0.89 | 0.91 | 0.9 | 0.9102536 |
| | COVID_19 Negative | 0.91 | 0.91 | 0.91 | |
| | Weighted Avg. | 0.90 | 0.91 | 0.91 | |

Table 5
Result of the transfer learning models with Dropout and a FC connected layer at the end.

| Models | Class | Precision | Recall | F1-score | AUC |
|-------------|-------------------|-----------|--------|----------|-----------|
| VGG-16 | COVID_19 Positive | 0.94 | 0.94 | 0.94 | 0.9355865 |
| | COVID_19 Negative | 0.93 | 0.92 | 0.93 | |
| | Weighted Avg. | 0.93 | 0.93 | 0.94 | |
| ResNet50 | COVID_19 Positive | 0.95 | 0.96 | 0.95 | 0.962535 |
| | COVID_19 Negative | 0.94 | 0.96 | 0.96 | |
| | Weighted Avg. | 0.95 | 0.96 | 0.96 | |
| VGG-19 | COVID_19 Positive | 0.91 | 0.92 | 0.92 | 0.932565 |
| | COVID_19 Negative | 0.95 | 0.94 | 0.94 | |
| | Weighted Avg. | 0.93 | 0.93 | 0.93 | |
| Xception | COVID_19 Positive | 0.93 | 0.92 | 0.92 | 0.935689 |
| | COVID_19 Negative | 0.93 | 0.94 | 0.93 | |
| | Weighted Avg. | 0.93 | 0.93 | 0.93 | |
| InceptionV3 | COVID_19 Positive | 0.97 | 0.95 | 0.96 | 0.953656 |
| | COVID_19 Negative | 0.94 | 0.95 | 0.95 | |
| | Weighted Avg. | 0.96 | 0.95 | 0.95 | |
| MobileNetV2 | COVID_19 Positive | 0.94 | 0.95 | 0.95 | 0.9545786 |
| | COVID_19 Negative | 0.96 | 0.94 | 0.95 | |
| | Weighted Avg. | 0.95 | 0.95 | 0.95 | |
| DenseNet201 | COVID_19 Positive | 0.91 | 0.92 | 0.91 | 0.9215463 |
| | COVID_19 Negative | 0.93 | 0.92 | 0.92 | |
| | Weighted Avg. | 0.92 | 0.92 | 0.92 | |

Further, other six transfer learning models, ResNet50, VGG-19, Xception, InceptionV3, MobileNetV2, and DenseNet201, were experimented with to find the best suitable model. However, the misclassification of COVID Positive and Negative instances are present in each model's prediction. One of the reasons behind the image misclassification is the model over-fitting. The model overfitting issue was handled by introducing the Regularization parameter called Dropout. Hence, we re-experimented all selected transfer learning models with the Dropout layer. The outcomes of the model with Dropout is shown in Table 5. The best performing model without and with Dropout layer is ResNet50, where the weighted average precision, recall and F1-score for COVID Positive and Negative class are 0.95, 0.95, 0.95, 0.95, 0.96, and 0.96 respectively. These experiments were conducted with a FC layer at the end of the network consisting of 64 neurons. Further, we extended the number of FC layers from one to two by fixing the neurons as 128 and 64 before the output layer.

The outcomes of the modified network consisting of 128 and 64 neurons are shown in Table 6. The best performing model with this setting is InceptionV3, where the weighted precision, recall and F1-score was 0.96, 0.96, and 0.96 respectively. The ResNet50 model achieved the weighted precision, recall and F1-score was 0.95, 0.95, and 0.95 respectively. The performance of the ResNet50 model is closer to the best performing InceptionV3 model.

Table 6

Result of the transfer learning models with Dropout and two FC layers with 128 and 64 neurons.

| Models | Class | Precision | Recall | F1-score | AUC |
|-------------|-------------------|-----------|--------|----------|----------|
| VGG-16 | COVID_19 Positive | 0.92 | 0.94 | 0.93 | 0.946548 |
| | COVID_19 Negative | 0.95 | 0.94 | 0.94 | |
| | Weighted Avg. | 0.94 | 0.94 | 0.94 | |
| ResNet50 | COVID_19 Positive | 0.96 | 0.95 | 0.96 | 0.958796 |
| | COVID_19 Negative | 0.95 | 0.95 | 0.95 | |
| | Weighted Avg. | 0.95 | 0.95 | 0.95 | |
| VGG-19 | COVID_19 Positive | 0.93 | 0.92 | 0.93 | 0.945263 |
| | COVID_19 Negative | 0.95 | 0.96 | 0.96 | |
| | Weighted Avg. | 0.94 | 0.94 | 0.94 | |
| Xception | COVID_19 Positive | 0.96 | 0.96 | 0.96 | 0.951456 |
| | COVID_19 Negative | 0.95 | 0.95 | 0.95 | |
| | Weighted Avg. | 0.95 | 0.95 | 0.95 | |
| InceptionV3 | COVID_19 Positive | 0.97 | 0.97 | 0.97 | 0.965869 |
| | COVID_19 Negative | 0.96 | 0.95 | 0.96 | |
| | Weighted Avg. | 0.96 | 0.96 | 0.96 | |
| MobileNetV2 | COVID_19 Positive | 0.93 | 0.95 | 0.94 | 0.942536 |
| | COVID_19 Negative | 0.95 | 0.95 | 0.95 | |
| | Weighted Avg. | 0.94 | 0.95 | 0.94 | |
| DenseNet201 | COVID_19 Positive | 0.93 | 0.93 | 0.93 | 0.932154 |
| | COVID_19 Negative | 0.93 | 0.94 | 0.93 | |
| | Weighted Avg. | 0.93 | 0.93 | 0.93 | |

Table 7

Result of the transfer learning models without using Dropout and three FC layers with 256, 128, and 64 neurons respectively.

| Models | Class | Precision | Recall | F1-score | AUC |
|-------------|-------------------|-----------|--------|----------|----------|
| VGG-16 | COVID_19 Positive | 0.96 | 0.96 | 0.96 | 0.962535 |
| | COVID_19 Negative | 0.95 | 0.95 | 0.95 | |
| | Weighted Avg. | 0.96 | 0.96 | 0.96 | |
| ResNet50 | COVID_19 Positive | 0.97 | 0.97 | 0.97 | 0.965869 |
| | COVID_19 Negative | 0.96 | 0.96 | 0.96 | |
| | Weighted Avg. | 0.96 | 0.96 | 0.96 | |
| VGG-19 | COVID_19 Positive | 0.95 | 0.95 | 0.95 | 0.953628 |
| | COVID_19 Negative | 0.96 | 0.95 | 0.96 | |
| | Weighted Avg. | 0.95 | 0.95 | 0.95 | |
| Xception | COVID_19 Positive | 0.97 | 0.98 | 0.97 | 0.975836 |
| | COVID_19 Negative | 0.97 | 0.96 | 0.97 | |
| | Weighted Avg. | 0.97 | 0.96 | 0.97 | |
| InceptionV3 | COVID_19 Positive | 0.98 | 0.97 | 0.98 | 0.975893 |
| | COVID_19 Negative | 0.96 | 0.97 | 0.96 | |
| | Weighted Avg. | 0.97 | 0.97 | 0.97 | |
| MobileNetV2 | COVID_19 Positive | 0.96 | 0.94 | 0.95 | 0.952365 |
| | COVID_19 Negative | 0.96 | 0.96 | 0.96 | |
| | Weighted Avg. | 0.96 | 0.95 | 0.95 | |
| DenseNet201 | COVID_19 Positive | 0.95 | 0.96 | 0.96 | 0.962154 |
| | COVID_19 Negative | 0.94 | 0.96 | 0.95 | |
| | Weighted Avg. | 0.95 | 0.96 | 0.96 | |

Next, the number of FC layers increased from two to three with the number of neurons 256, 128, and 64, respectively. The other settings remain unchanged. With this setting, the InceptionV3 model achieved the weighted precision, recall and F1-score was 0.97, 0.97, and 0.97 respectively, which is a better performance than other experimental models as shown in [Table 7](#). Next, experiments were carried out with four FC layers having 512, 256, 128, and 64 neurons. The experimental outcomes of the four FC layered models are presented in [Table 8](#). This model achieved the best-weighted precision, recall and F1-score was 0.99, 0.99, and 0.99, respectively. Even though the model achieved the best performance, it also has false-positive predictions. To reduce the false prediction, we perform an ensemble process in which the predictions of all individual transfer learning models are considered. After training all the transfer learning models, the probability averaging ensemble method is used to get the final class value for each data sample. The outcomes of the ensemble learning model are shown in [Table 8](#). The proposed ensemble learning model achieved the weighted precision, recall and F1-score of 1.00, 1.00, 1.00, respectively, which indicates that the misclassification rate is almost zero. The confusion matrix and ROC curve for the proposed ensemble model can be seen in [Figs. 7](#) and [8](#). The Accuracy vs Epochs and Loss vs Epochs plots can be seen in [Fig. 9\(a\)](#) and [Fig. 9\(b\)](#), respectively.

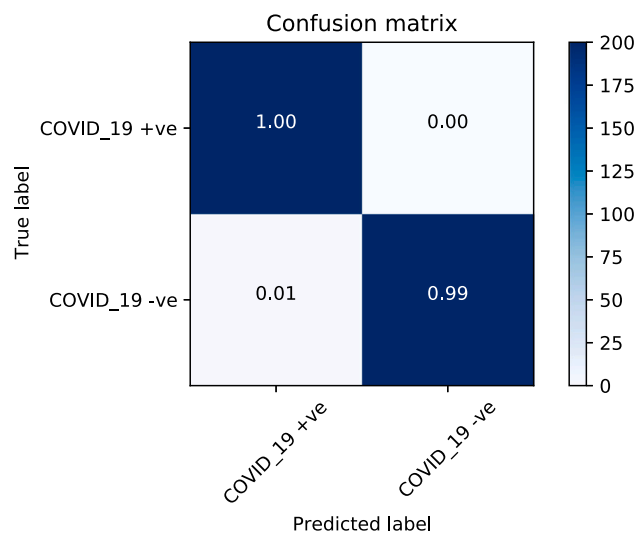


Fig. 7. Confusion matrix of the proposed ensemble model for COVID-19 disease prediction with chest X-ray.

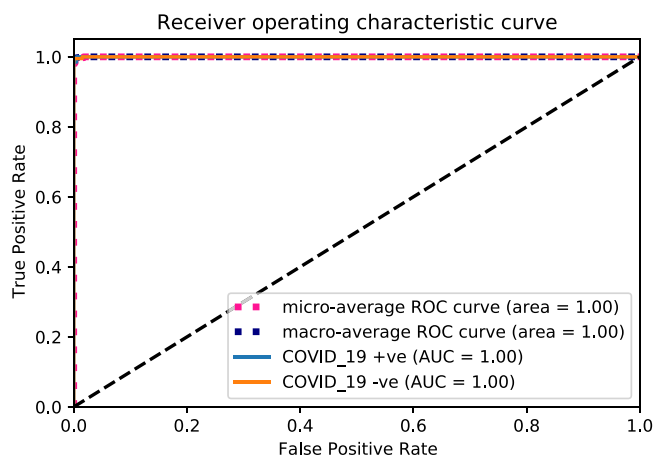
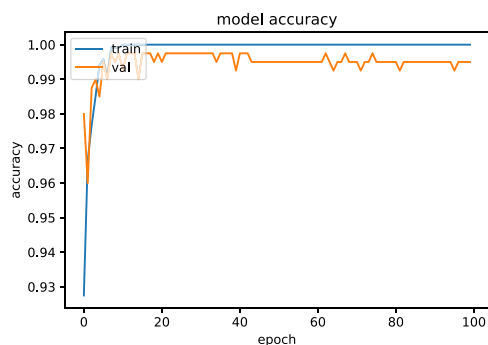
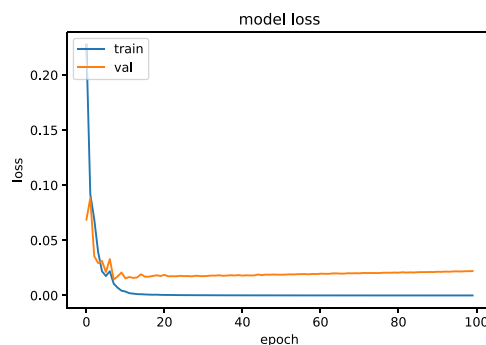


Fig. 8. ROC curve of the proposed ensemble model for COVID-19 disease prediction with chest X-ray.



(a) Accuracy Vs Epochs plot for the proposed ensemble-based model



(b) Loss Vs Epochs plot for the proposed ensemble-based model

Fig. 9. Accuracy vs Epochs and Loss vs Epochs plots for the proposed ensemble-based model.

Table 8

Result of the transfer learning models with Dropout and four FC layers with 512, 256, 128, and 64 neurons respectively and taking their ensemble.

| Models | Class | Precision | Recall | F1-score | TP | TN | FP | FN | AUC |
|---------------------------------|-------------------|-----------|--------|----------|-----|-----|----|----|----------|
| VGG16 (M1) | COVID_19 Positive | 0.99 | 0.98 | 0.99 | 196 | 197 | 4 | 3 | 0.996875 |
| | COVID_19 Negative | 0.99 | 0.99 | 0.99 | | | | | |
| | Weighted Avg. | 0.99 | 0.99 | 0.99 | | | | | |
| ResNet50 (M2) | COVID_19 Positive | 0.99 | 0.99 | 0.99 | 197 | 197 | 3 | 3 | 0.991875 |
| | COVID_19 Negative | 0.98 | 0.99 | 0.99 | | | | | |
| | Weighted Avg. | 0.98 | 0.99 | 0.99 | | | | | |
| VGG19 (M3) | COVID_19 Positive | 0.99 | 0.99 | 0.99 | 197 | 198 | 3 | 2 | 0.990925 |
| | COVID_19 Negative | 0.99 | 0.99 | 0.99 | | | | | |
| | Weighted Avg. | 0.99 | 0.99 | 0.99 | | | | | |
| Xception (M4) | COVID_19 Positive | 0.99 | 0.99 | 0.99 | 198 | 198 | 2 | 2 | 0.992412 |
| | COVID_19 Negative | 0.99 | 0.99 | 0.99 | | | | | |
| | Weighted Avg. | 0.99 | 0.99 | 0.99 | | | | | |
| InceptionV3 (M5) | COVID_19 Positive | 0.98 | 0.99 | 0.99 | 197 | 194 | 3 | 6 | 0.984156 |
| | COVID_19 Negative | 0.96 | 0.98 | 0.97 | | | | | |
| | Weighted Avg. | 0.97 | 0.98 | 0.98 | | | | | |
| MobileNetV2 (M6) | COVID_19 Positive | 0.99 | 0.99 | 0.99 | 197 | 192 | 3 | 8 | 0.988851 |
| | COVID_19 Negative | 0.99 | 0.97 | 0.98 | | | | | |
| | Weighted Avg. | 0.99 | 0.98 | 0.99 | | | | | |
| DenseNet201 (M7) | COVID_19 Positive | 0.99 | 0.99 | 0.99 | 198 | 192 | 2 | 8 | 0.989125 |
| | COVID_19 Negative | 0.98 | 0.97 | 0.98 | | | | | |
| | Weighted Avg. | 0.98 | 0.98 | 0.98 | | | | | |
| Ensemble (M1+M2+M3+M4+M5+M6+M7) | COVID_19 Positive | 1.00 | 1.00 | 1.00 | 200 | 198 | 0 | 2 | 0.999875 |
| | COVID_19 Negative | 1.00 | 0.99 | 1.00 | | | | | |
| | Weighted Avg. | 1.00 | 1.00 | 1.00 | | | | | |

Table 9

Average execution time taken by transfer learning models.

| Models | Execution time in microseconds |
|---------------------------------|--------------------------------|
| VGG-16 (M1) | 65.50 |
| ResNet50 (M2) | 56.23 |
| VGG-19 (M3) | 80.61 |
| Xception (M4) | 122.31 |
| InceptionV3 (M5) | 53.74 |
| MobileNetV2 (M6) | 46.98 |
| DenseNet201 (M7) | 90.33 |
| Ensemble (M1+M2+M3+M4+M5+M6+M7) | 145.53 |

Table 10

Performance comparison of the proposed ensemble model with the existing models.

| Source | Data Type | Classes | Precision | Recall | F1-Score | Accuracy |
|---------------------------|-----------|---------|-----------|--------|----------|----------|
| Apostolopoulos et al. [8] | CXR | Multi | 94.46 | 98.66 | – | 96.78 |
| Loey et al. [9] | CT | Binary | 81.90 | 80.85 | – | 81.38 |
| Taresh et al. [10] | CXR | Multi | 98.69 | 98.78 | 97.59 | 98.72 |
| Fan et al. [11] | CXR | Binary | 97.50 | 96.50 | 97.00 | 97.00 |
| Horry et al. [12] | CXR | Binary | 86.00 | 86.00 | 86.00 | – |
| Civit-Masot et al. [13] | CXR | Multi | 84.00 | 100 | 92.00 | – |
| Ko et al. [14] | CT | Multi | 100 | 99.58 | – | 99.87 |
| Azemin et al. [15] | CXR | Binary | 71.80 | 77.30 | – | 71.90 |
| Narayanan et al. [16] | CXR | Binary | 99.00 | 91.00 | 90.00 | 99.34 |
| Proposed | CXR | Binary | 100 | 100 | 100 | – |

Table 10 shown in experimental outcomes of the existing research on COVID-19. Most of the research were performed using chest X-ray (CXR) data [8,10–13,15,16], whereas few researchers used CT scan of the infected people to build the predictive model [9,14]. Among the listed research of Table 10, the best specificity (precision) was obtained by Ko et al. [14], i.e., 100% which was similar to our proposed model, the sensitivity (recall) value was 99.58, and the accuracy value was 99.87%. The model suggested in [15] achieved poor predictions where the precision, recall, and accuracy value was 71.80%, 77.30%, and 71.0%, respectively. The model suggested by Loey et al. [9] have a precision value of 81.90%, indicating that most of the infected people are not properly detected by the model. On the other hand, the proposed model achieved remarkable performance where the precision, recall and F1-score was 100%, 100%, and 100%, respectively, which outperformed the existing research.

All the experimentation were performed on Google Colaboratory¹⁴ having following hardware configurations: GPU: 1xTesla K80, compute 3.7, having 2496 CUDA cores, and 12 GB GDDR5 VRAM. The time to execute a testing sample on all the implemented transfer learning models can be seen in Table 9. We found that the MobileNetV2 took lesser time to execute on an average among all implemented models, whereas the proposed ensemble model took more execution time slightly but performed significantly well.

5. Conclusion

This study suggested a framework that detected the COVID-19 infected people with high accuracy. The same can be implemented into any Internet-enabled device with a monitoring facility to connect with medical experts. The COVID-19 disease is infectious. Hence remote monitoring with IoMT based technology is the best solution to control it. As the cases have been increasing at an exponential rate, there is a need to identify every positive case during this emergency. In this study a Deep Ensemble Learning Framework is proposed that uses the predictive accuracies of individual transfer learning models namely (i) ResNet50, (ii) DenseNet201, (iii) InceptionV3, (iv) VGG-16, (v) VGG-19, (vi) Xception, and (vii) MobineNetV2 to predict the COVID-19 infected people. The ensemble learning framework uses the individual strength of the transfer learners to detect COVID-19 from the chest X-ray images. Despite its heavy computational requirements and its complex structure, this framework is practical enough as it provides optimal results on validation data set as well. However the proposed system has limitations. First of all the data set is quite small to test generalization of the system. This can be resolved if more images are used to train the model. Second, as of now the model works on the posterior–anterior (PA) view of X-rays. Hence it cannot differentiate anterior–posterior (AP), lateral views etc. Third, COVID-19 with varied degrees cannot be identified as of now. If we can classify COVID-19 further into mild and severe cases, this would reduce the load on existing healthcare infrastructure. Also, there is a need for efficient radiologists to identify and confirm the results of the proposed model.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to <https://doi.org/10.1016/j.compeleceng.2022.108018>.

References

- [1] Ugandran I, Mahendran A, Anandakumar S, Hamada M, Alnumay WS, Ghosh U, et al. A novel cryptosystem using dna sequencing and contextual array splicing system for medical internet of things. *Comput Electr Eng* 2021;96:107429.
- [2] Chahar S, Roy PK. Covid-19: A comprehensive review of learning models. *Arch Comput Methods Eng* 2021;1–26.
- [3] Vieira PA, Magalhães DM, Carvalho-Filho AO, Veras RM, Rabêlo RA, Silva RR. Classification of covid-19 in x-ray images with genetic fine-tuning. *Comput Electr Eng* 2021;107467.
- [4] Roy PK. Multilayer convolutional neural network to filter low quality content from quora. *Neural Process Lett* 2020;52(1):805–21.
- [5] Das TK, Roy PK, Uddin M, Srinivasan K, Chang CY, Syed-Abdul S. Early tumor diagnosis in brain mr images via deep convolutional neural network model. *Comput Mater Contin* 2021;68(2):2413–29.
- [6] Li X, Grandvalet Y, Davoine F, Cheng J, Cui Y, Zhang H, et al. Transfer learning in computer vision tasks: Remember where you come from. *Image Vis Comput* 2020;93:103853.
- [7] Mohammad R, Abolfazl A. A new modified deep convolutional neural network for detecting covid-19 from x-ray images, arXiv preprint arXiv:2004.08052.
- [8] Apostolopoulos ID, Mpesiana TA. Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Phys Eng Sci Med* 2020;43(2):635–40.
- [9] Loey M, Manogaran G, Khalifa NEM. A deep transfer learning model with classical data augmentation and cgan to detect covid-19 from chest ct radiography digital images. *Neural Comput Appl* 2020;1–13.
- [10] Tareh M, Zhu N, Ali TAA. Transfer learning to detect covid-19 automatically from x-ray images. In: *Using convolutional neural networks*. 2020, medRxiv.
- [11] Fan Z, Jamil M, Sadiq MT, Huang X, Yu X. Exploiting multiple optimizers with transfer learning techniques for the identification of covid-19 patients. *J Healthcare Eng* 2020;2020.
- [12] Horry MJ, Chakraborty S, Paul M, Ulhaq A, Pradhan B, Saha M, et al. Covid-19 detection through transfer learning using multimodal imaging data. *IEEE Access* 2020;8:149808–24.
- [13] Civit-Masot J, Luna-Perejón F, Domínguez Morales M, Civit A. Deep learning system for covid-19 diagnosis aid using x-ray pulmonary images. *Appl Sci* 2020;10(13):4640.
- [14] Ko H, Chung H, Kang WS, Kim KW, Shin Y, Kang SJ, et al. Covid-19 pneumonia diagnosis using a simple 2d deep learning framework with a single chest ct image: model development and validation. *J Med Internet Res* 2020;22(6):e19569.
- [15] Che Azemin MZ, Hassan R, Mohd Tamrin MI, Md Ali MA. Covid-19 deep learning prediction model using publicly available radiologist-adjudicated chest x-ray images as training data: preliminary findings. *Int J Biomed Imaging* 2020;2020.
- [16] Narayanan BN, Hardie RC, Krishnaraja V, Karam C, Davuluru VSP. Transfer-to-transfer learning approach for computer aided detection of covid-19 in chest radiographs. *AI* 2020;1(4):539–57.
- [17] Wang L, Lin ZQ, Wong A. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Sci Rep* 2020;10(1):1–12.
- [18] Jain R, Gupta M, Taneja S, Hemanth DJ. Deep learning based detection and analysis of covid-19 on chest x-ray images. *Appl Intell* 2021;51(3):1690–700.
- [19] Pham TD. Classification of covid-19 chest x-rays with deep learning: new models or fine tuning? *Health Inform Sci Syst* 2021;9(1):1–11.
- [20] Islam MZ, Islam MM, Asraf A. A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images. *Inform Med Unlock* 2020;20:100412.
- [21] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 770–8.

¹⁴ <https://colab.research.google.com/>.

- [22] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 4700–8.
- [23] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 2818–26.
- [24] Chollet F. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 1251–8.
- [25] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 4510–20.
- [26] Kowsari K, Jafari Meimandi K, Heidarysafa M, Mendu S, Barnes L, Brown D. Text classification algorithms: A survey. *Information* 2019;10(4):150.

Pradeep Kumar Roy received his Ph.D. degree in Computer Science and Engineering from the National Institute of Technology Patna in 2018. He received a Certificate of Excellence for securing a top rank in the M. Tech course. He is currently an Assistant Professor at IIIT Surat, India. His area of specialization straddles across machine learning, deep learning, and computer vision.

Abhinav Kumar is currently working as an Assistant Professor, Department of CSE, Siksha 'O' Anusandhan, Deemed to be University, Bhubaneswar, Odisha, India. He has obtained a Ph.D. degree in Computer Science & Engineering from the Department of CSE of the NIT Patna, India. His research interest includes machine learning, deep learning, crisis informatics, natural language processing, and social networks.