Detection of novel coronavirus from chest X-rays using deep convolutional neural networks

Shashwat Sanket1 · M. Vergin Raja Sarobin1 · L. Jani Anbarasi1 · Jayraj Thakor1 · Urmila Singh1 · Sathiya Narayanan2

Abstract
With over 172 Million people infected with the novel coronavirus (COVID-19) globally and with the numbers increasing exponentially, the dire need of a fast diagnostic system keeps on surging. With shortage of kits, and deadly underlying disease due to its vastly mutating and contagious properties, the tired physicians need a fast diagnostic method to cater the requirements of the soaring number of infected patients. Laboratory testing has turned out to be an arduous, cost-ineffective and requiring a well-equipped laboratory for analysis. This paper proposes a convolutional neural network (CNN) based model for analysis/detection of COVID-19, dubbed as CovCNN, which uses the patient’s chest X-ray images for the diagnosis of COVID-19 with an aim to assist the medical practitioners to expedite the diagnostic process amongst high workload conditions. In the proposed CovCNN model, a novel deep-CNN based architecture has been incorporated with multiple folds of CNN. These models utilize depth wise convolution with varying dilation rates for efficiently extracting diversified features from chest X-rays. 657 chest X-rays of which 219 were X-ray images of patients infected from COVID-19 and the remaining were the images of non-COVID-19 (i.e. normal or COVID-19 negative) patients. Further, performance evaluation on the dataset using different pre-trained models has been analyzed based on the loss and accuracy curve. The experimental results show that the highest classification accuracy (98.4%) is achieved using the proposed CovCNN model.

 Keywords Convolutional neural network · Deep-CNN · COVID-19 detection · X-rays

1 Introduction
Coronavirus disease 2019 (COVID-19) has been at an outburst since December 2019, as one of the deadliest pandemics in over the decades and has since then been escalating. It has disrupted the economy and caused turbulences as it could never been imagined. With
its highly mutating positive centered single-stranded RNA and no cure in hand, India has reached among world’s second highest rated countries on the count of people diseased. In addition to the ranking, India has over 10,950,201 cases with a death toll of 156,014. Countries like the USA, Spain, India, Italy, China, UK, Iran etc. have been suffering from several forms of COVID-19 and which is also widely present in humans, cats, dogs, pigs, poultry and rodents in different forms. The discernible symptoms of COVID-19 are sore throat, loss of smell and taste, fatigue, fever, running nose and cough. It targets weakening the immune system and has proven to be fatal with increased chances of inception in the age bar of 45–60 years. Acute respiratory symptoms like difficulty in breathing, weakness, chest pain can be an indication. Being a contagious disease, it has been spreading all over the globe rapidly. It can proliferate through physical touch, breath contact, contact with the hand or contact with the mucus. The world has witnessed a pandemic of such severity in the bygone centuries as well. Severe acute respiratory syndrome (SARS) and Middle East respiratory syndrome (MERS) belong to the same category. The worldwide count of the infected people is appalling, with 109,594,835 people being infected and 2,424,060 resulting in death as of February 19, 2021. The fatality count has been escalating exponentially all over the world.

COVID-19 earmarks people with chronic health problems and for the elderly. The common symptoms of COVID-19 aren’t yet so futile that may further result in pneumonia, multi-organ failure and death. One way to test them is the laboratory testing which turns out to be highly time consuming and costly as well requires a well-developed laboratory for analysis. To the rescue comes the chest X-ray diagnosis which speeds up the diagnosis process. It is widely used for other typical and viral pneumonia diseases such as influenza, SARS, and MERS. One of the other better ways of medication is detection of disease at an early stage and instant quarantine due to lack of proper medication. The Chinese government maneuvered real-time polymerase chain reaction (RT-PCR) for the diagnosis of the disease. However, it did not provide accurate results because of which patients were unable to be diagnosed and treated on time. Moreover, as a result of the inaccuracy, with high false negative reports and time consumption, the infection keeps on infiltrating into a healthy person’s body, being a highly communicable disease. A suitable alternative has been found for this cumbersome process. Infected people show bilateral changes in chest X-ray images. This replacement process for detection can provide a large amount of pathological knowledge for the infection as well. This further leads to fabricating a Deep Learning-based research technique to analyze the chest X-ray without radiologist’s intervention. This paper addresses the following research problems:

- The custom CNN and the pre-trained models always follow the conventional optimization for any problem to fit. Any upgradation for the performance improvement has to be derived from the enhanced base learning by capturing required patterns.
- Most of the existing research work explores the pre-trained models and regular CNN architecture rather than customized architecture for COVID-19 detection.

The main objective of the paper is experimentation of a fast diagnostic method, i.e. chest X-ray classification for COVID-19 infected patients. Deep convolutional neural networks were employed and tested for a deep-learning method for determining if the patient is affected or not. By considering sensitivity and specificity, a multi objective fitness function is designed to classify COVID-19 infected patients. This paper proposes a convolutional neural network (CNN) based model for analysis/detection of COVID-19, dubbed as CovCNN in the remainder of this paper. The proposed CovCNN model is trained by considering the chest X-ray images of COVID-19 patients. The major contributions of the paper are:
• The proposed custom created models been developed with three deep learning architectures such as CovCNN_1, CovCNN_2, CovCNN_3 to set the baseline for the final proposed custom model i.e., CovCNN_4.
• The proposed model integrated various convolution models with effective activation layers to support better learning for extracting elite patterns to be classified.
• The data is also tested on different pre-trained transfer learning models and hyperparameter tuning has been carried out to achieve optimal accuracy.
• The results for both custom and pre-trained models are shown in the form of ROC curve, training and validation loss respectively

The remainder of the paper is structured as follows: Sect. 2 discusses the existing work in the field of COVID-19; Sect. 3 explains the proposed models, architectures developed; Sect. 4 details the experimental analysis and results of the proposed and pre-trained models and Sect. 5 concludes the work with the future scope.

2 Related work

Various machine learning models are specifically investigated by researchers for detecting the disease amongst COVID-19 X-ray dataset. Machine learning methods come in handy in critical tasks. Henceforth, deep neural technique by computer vision method can be used to detect COVID-19. Without the need for manual extraction of features, deep learning, a quite successful Artificial Intelligence (AI) research field, allows the development of end-to-end models to achieve expected outcomes with the input data. Additionally, it is also used for identification of several other problems like arrhythmia, recognition of skin cancer, breast cancer detection, diagnosis of brain disease, pneumonia identification from chest X-ray images and segmentation of the fungus images. Pneumonia in infants is diagnosed by Sousa et al. [27] through computer-aided systems from radiographic images using support vector machine (SVM), K-Nearest Neighbor, and Naïve Bayes, where SVM outperformed the others. Turk et al. [28] proposed a novel method to detect COVID-19 with residual exemplar local binary pattern (ResExLBP), using the dataset that included 87 X-ray images with COVID-19 disease that includes 26 female, 41 male and 20 are not determined. Deep learning based model is proposed by [3] using 1020 CT slices from 108 patients with AlexNet, VGG-16, VGG-19, SqueezeNet, GoogleNet, MobileNet-V2, ResNet-18, ResNet-50, ResNet-101, and Xception where ResNet-101 and Xception outperformed. RT-PCR sensitivity and chest CT, for COVID-19 detection was researched by Fang et al. [7]. The travel history and signs of two patients gives better analysis of chest sensitivity of CT and has superior detection than RT-PCR. With a whopping 97.4% accuracy, Mahmud et al. [17] proposed CovXNets for detecting the COVID-19 and other pneumonia using chest X-ray images. The Guangzhou Medical Center, China analyzed two datasets including 5856 images, parallel to the other dataset which included 305 X-rays of COVID-19 collected from Sylhet Medical College, Bangladesh. A deep neural network technique was further proposed by Panwar et al. [22] to analyze COVID-19 using nCOVnet that includes 24 layers where the first layer is the input layer and the other layers are the combination of convolution + rectified linear unit (ReLU) and max pooling layers. This scheme achieved an accuracy of 97.97% confidence. Similarly 121 chest CT images were analyzed by Berheim et al. [5] to identify the relationship between the symptoms among the patients. To distinguish between communities acquired pneumonia and other
non-pneumonia lung diseases, Li et al. [15] extracted the features from CT scans using the deep learning model COVNet. With an astounding accuracy of 98.2% sensitivity and 92.2% specificity, Gozes et al. [10] successfully performed AI based tool to test COVID-19. Shan et al. [24] developed a VB-net deep learning model to segment the infection sites in the CT scans images. For discriminating pneumonia and influenza, Lui et al. [16] processed the data using a CNN based prediction model and achieved 86.7%. Wang et al. [31] achieved 89.5% prediction level using a modified inception transfer learning model. Narin et al. [19] performed an automatic deep CNN-based prediction using chest X-ray images with ResNet50 achieving 98% accuracy. Sethy et al. [23] extracted features from the chest X-ray images using deep learning techniques and classified using SVM and achieved 95.38% accuracy. Using chest X-ray images to accurately diagnose binary and multi class classification, Ozturk et al. [21] achieved an accuracy of 98.8% for binary classification and 87.02% for multi-class using DarkNet model for COVID-19 detection. Hemdan et al. [11] made use of COVIDX-Net deep learning model to diagnose COVID-19 where Singh et al. [25] used chest CT images and fine-tuned the CNN parameters to achieve an optimized CNN to classify COVID-19 using a multi objective differential evolution. Wang et al. [30] proposed COVID-Net achieving 98.75% accuracy for multi-class classification to classify, normal, non-COVID pneumonia, and COVID-19 classes. Apostolopoulos et al. [1] achieved an accuracy of 98.75% using a deep learning model using 224 COVID-19 images. Zheng co-author [26] proposed a three-dimensional deep CNN model to detect COVID-19 from CT scan imagery and reported 90.8% accuracy. Song et al. [26] achieved an accuracy of 86% using modified Inception models for CT images.

COVIDX-Net [11], a model which involves 7 separate Deep CNN model architectures, including VGG19 and modified version of Google MobileNet, evaluated normalised intensities of the X-ray to diagnose COVID-19. CNN-based pre-trained models were used by Bi et al. [6] to diagnose contaminated patients with Coronavirus pneumonia with the help of chest CT Images. Wang et al. [32] proposed an novel transfer learning model (L2TFL) derived with an optimal layers. This model resulted with novel selection algorithm using a fusion approach. This model analysed 284 COVID-19 and 281 pneumonia images and 293 tuberculosis and 306 healthy images which achieved an accuracy of 95.61%, 96.25%, 98.30%, and 97.86% for four classes. Transfer Learning Models were experimented for classifying X-ray by Apostolopoulos [2] using off-shelf features for 3 class classifier that includes COVID-19, pneumonia and pulmonary diseases. Khan et al. analysed Xception as the head layer followed by CNN for classifying COVID-19 from chest X-ray [14]. Wang et al. [32] proposed an FGCNet a deep feature fusion model from graph convolution network (GCN) and CNN using 320 COVID-19 and 320 healthy images. The features are extracted using GCN and DFF technology is used for combining these features. This scheme achieved 15% performance improvement compared to others. Minaee et al. [18] classified COVID-19 using CNN along with models like ResNet18, ResNet50, SqueezeNet, and DenseNet-121 using 5000 chest X-ray images and achieved 98% of accuracy. Elene Firmeza Ohata et al. [20] analysed COVID-19 using CNN trained over imagenet using 194 COVID-19 chest X-rays and extracted features. These features are classified using k-Nearest Neighbor, Bayes, Random Forest, multilayer perceptron (MLP), and support vector machine (SVM).

In recent years, a lot of work has been done in the field of classifying the COVID-19 using X-ray images. It’s been challenging due to its inherent texture variations and similarity towards other diseases like pneumonia. Several other studies have been encountered to be developing for classification of COVID-19 based on computer vision algorithms.
2.1 Convolutional neural networks

Convolutional networks are influenced by biological processes \[8, 9, 12, 13\], where the pattern of communication between neurons resembles the response of a neuron in the visual cortex to a specific stimulus. Individual neurons respond to stimuli for its field of receptive zone. For spanning the entire field of vision, the receptive fields of the different neurons partly overlap. Multilayer perceptron means networks that are completely connected. Henceforth, each neuron in one layer is linked to all neurons in the next layer. CNN \[33\] is used extensively for the classification of images where the hierarchical structure and the extracted features make the CNN a dynamic model for image classification. A convolution neural network includes an input and output layer, as well as several hidden layers. The hidden layers include a series of convolution layers that convolve with an operation. The convolution layer includes convolution kernels (height and width), number of input and output channels and the convolution filters whose depth is like the feature map. The convolution operation is given in Eq. (1)

\[ F_{\text{cov}}(x, y) = (D \ast F)(x, y) = \sum_i \sum_j D(x + i, y + j)F(i, j) \]

where \(D\) refers the input matrix representing the input image, \(F\) is the 2-D filter of size \(x\) and \(y\) and \(F_{\text{cov}}\) represents the output feature map. \(D \ast F\) represent the convolution operation.

By using regularized weights over fewer parameters, the vanishing gradient and exploding gradient problems seen in traditional neural networks during backpropagation are avoided \[4, 29\]. Figure 1 includes the sample convolution operation.

The output of the convolution layer is fed to an activation function that introduces the non-linearity. Activation function works just like the human brain. The human brain decides how to react based on the perception, the activation function does the similar thing by deciding whether the neuron should be activated or not based on the inputs received and making the calculation using specific function. The non-linear activation function used in the deep learning networks is ReLU. It generates an output 0 for the value less than 0 or raw output otherwise. The mathematical representation of ReLU is given in Eq. (2).

\[ R_{\text{cov}}(x) = \max(0, x) \]

Subsampling or downsampling, also known as pooling, is a simple process to reduce the size or dimensionality of the feature map, while retaining the most important features. Max, average, and sum are the 3 types of pooling that can be applied. Pooling has the effect of reducing the dimensionality (width and height) of the previous layer by half. Moreover, it can thus remove 75% of the activations seen in the previous layer. Fully connected (FC)

![Example for convolution operation](image-url)
means that all nodes in one layer are connected to the outputs of the next layer. When each class is assigned a probability, the FC Layer outputs the class probabilities. In fully connected layer each neuron from previous layer is connected to every neuron in the next layer and every value contributes in predicting how well a value fits a particular class. The output of fully connected layer is then forwarded to an activation function which outputs the class scores. A Softmax function that measures the distribution of probability of the ‘n’ output classes is given as Eq. (3).

\[
S^{\text{Cov}} = \frac{G^m}{\sum_{j=1}^{G^p} G^j}
\]

where \( Y \) refers the input feature vector and \( S \) denotes the output. Generally, the sum of the outputs is equal to 1. The loss function used is the cross entropy defined in Eq. (4).

\[
L(DV) = - \sum_i D_i \log (V_i)
\]

\[3\]

2.2 Transfer learning models

Transfer learning in its abstract form is a method of applying knowledge acquired from one task to solve the other similar problems. Here, the knowledge gained from pre-trained networks are applied for detecting the COVID-19 infected patients. In this research work, 7 pre-trained models are used to segregate COVID-19 patients from that of normal patients: ResNet-101, VGG-16, VGG-19, Inception-V3, ResNet-50V2, InceptionResNet-V2 and Xception. These deep transfer learning pre-trained models are used as a feature extractor and on top of this extractor training the classifier is performed. ResNet-101 and ResNet-50V2 have their special residual blocks where ResNet-101 includes 33 residual blocks and is 101 layers deep. Whereas ResNet-50V2 includes 16 residual blocks and is 50 layers deep. ResNet-50V2 activates the weight layer by applying batch normalization and ReLU activation prior to the convolution operation. VGG-16 and VGG-19 architecture proposed by Simonyan and Zisserman includes five convolution blocks and three fully connected layers. The difference between them is that VGG-16 includes 13 convolution layers whereas VGG-19 has 16 convolution layers. Inception-V3 is the third version of Google’s Inception CNN which is made up of different inception blocks where each block includes different sized convolution and pooling layers. Each inception block performs multi-level feature extraction. InceptionResNet-50V2 conflates the idea of Inception and ResNet to produce hybrid architecture with better performance. The InceptionResNet-50V2 is much deeper than Inception-V3 and it consists of 164 layers. Xception model proposed by Francois Chollet derived it from Inception. In contrast to Inception, Xception consists of depth wise separable convolutions in place of inception modules. It starts with convolution layers followed by depthwise separable convolutions with a total of 71 layers.

3 Proposed architecture

In this research work, a custom CNN model is presented along with seven well-known pre-trained models to diagnose the chest X-ray image of COVID-19. The proposed work (i.e. contributions of this paper) includes four variant folds of CovCNN model of which the last version (CovCNN-4) achieves the best results.
3.1 CovCNN_1 model

CovCNN_1 model includes two 2-D convolution layers and average pooling layers followed by activation layer with ReLU as the activation function. The initial convolutional layer uses 32, 5×5 pixel filters processing the image resulting in a 32 feature map. The 3×3 kernel size is used in the average pooling layer resulting in a dimensionality reduction. Similarly in-order to extract more fine-scale features of the image, a second convolution layer block with 64, 5×5 pixel filter, followed by a 3×3 average pooling layer kernel is resulting in 23×23×64 features. Flatten layer is stacked which acts as a utility layer for converting the output into a vector. Now in order to prevent the network from overfitting, 30% drop is performed to the less contributing neurons which in turn helps to lower the generalization error. Flattening is used to convert the n-dimensional data output resulted from the convolution layers, to 1-Dimensional vector, which is then fed to the fully connected network (FC). But instead of directly feeding the features to FC, dropout is performed to prevent the model from the over-fitting. Finally, the flattened output is fed to a feed-forward neural network with two dense layers, one with 256 neurons and ReLU activation function and the other with two neurons with sigmoid as an activation function.

ReLU maintains the sparsity and reduces the likelihood of vanishing gradient that is more likely to happen in the dense model. Last layer includes sigmoid activation function to overcome dying ReLU problem that prohibits further learning. These fully connected layers act as classification layers and help in learning a nonlinear function from the non-linear combinations of the high-level features as represented by the output of the convolutional layers. Binary cross-entropy is used as a loss function and RMSprop is used as an optimizer during the training process. This choice of optimizer results in fast convergence as it uses exponentially decaying average of squared gradients and discards history from the extreme past. This model results in classification accuracy of 93.94%, specificity of 0.9091, and sensitivity of 1. As it is a shallow network, it performs decently and thus acts as a baseline model. Figure 2 shows the proposed flow diagram of CovCNN_1 model and its details. Appendix 1 shows the summary of the CovCNN_1 model.

3.2 CovCNN_2 model

CovCNN_1 model is modified resulting in a CovCNN_2 except the last layer, loss function, and optimizer. Softmax activation function is used since the last layer includes 2 neurons which require the distribution of the probability throughout each output node. The performance of CovCNN_2 model achieved an improved accuracy of 95.45% with 1.00 as sensitivity and 0.93 as specificity. The focusing point is to achieve a high sensitivity, and since this model results 1, means the model has the ability to correctly identify those actually having the disease (true positive rate). Figure 3 shows the flow diagram of CovCNN_2 model and Appendix 2 details the model summary.

3.3 CovCNN_3

CovCNN_1 was added with a new convolution block with 64, 3×3 filters and a max-pooling layer with a kernel size of 2×2. The number of neurons in the first-dense layer is decreased to 128. The newly added convolution layer performs an optimized extraction of fine-grained features. The chosen average pooling instead of max-pooling smoothens out the image resulting in the sharp features like lesion which depict the
presence of the COVID-19 virus in the chest X-ray might be identified. Max-pooling gives an upper hand as it selects the brighter pixels from the image which is very useful when the background of the image is dark in the case of X-ray images. Though there is a decrease in sensitivity and specificity value, the accuracy of this model is 96.97% which is greater than the accuracy of previous model CovCNN_2 model. In order to achieve high accuracy as well as high sensitivity we further fine-tuned the parameters and the layers to achieve so. Figure 4 shows the flow diagram of CovCNN_3 model and its model summary is shown in Appendix 3.

3.4 CovCNN_4

The fine-tuned CovCNN_4 deep network that includes 15 layers where the convolutional layers are stacked sequentially with increasing filter sizes. The filter size of all the convolutional layers is of $5 \times 5$ with ReLU as an activation function and max-pooling...
kernel size as $2 \times 2$. Three convolution blocks are introduced in this CovCNN_4 model that comprises of two 2-D convolutional layers the max-pooling layer. Each convolution block is followed by a dropout layer with a 30% rate to prevent the model from overfitting. The fine-tuned parameter helps a good gradient flow among the convolution blocks. As in the case of CovCNN_1 model, CovCNN_4 uses binary cross-entropy as loss function and RMSProp as optimizer. The performance of CovCNN_4 model outperforms every other model achieving 98.48% accuracy, 1.0 sensitivity, and 0.9773 specificity. Figure 5 shows the flow diagram of CovCNN_4 model and its model summary is given in Appendix 4.

4 Results and discussions

This section presents a comprehensive view of the dataset, experimentation, model training, and validation. A performance comparison of the proposed approach with the existing works has also been presented in this section.
4.1 Data acquisition and pre-processing

The dataset containing COVID-19 radiography images are acquired from Kaggle (https://www.kaggle.com/tawsifurrahman/covid19-radiography-database) which is the real data collected recently through various labs of the infected patients. The database includes chest X-ray images for COVID-19 positive cases along with normal and viral pneumonia images. The dataset includes 219 COVID-19 positive images, 1341 normal images and 1345 viral pneumonia images, with the total size of approximately 1.15 GB. In this work, all the images of non-COVID-19 samples (normal + pneumonia) are not included because the number of COVID-19 samples is significantly less, which creates the scenario of class-imbalance. So, 219 images are randomly selected from the normal set and 219 images from the viral pneumonia set are included, thus contributing 438 images for non-COVID-19 class. Figure 6a shows a sample COVID-19 image where arrow in the image highlights the infected part. Figure 6b shows the non-COVID-19 normal images.

4.2 Data augmentation

Datasets of COVID-19 chest X-rays are continuously sourced from public donations that are not adequate for complex training using CNN. This research adopts an augmentation technique without any loss to increase as well as maintain the sample count. The parameters of augmentation are, rotation range of 15 degree, shear range of 0.2, height shift range of 0.2, zoom range of 0.3, width shift range of 0.2 and fill mode as nearest. The images in the dataset are of different sizes and therefore all the images are converted into a size of 224 × 224 × 3 pixels. The color-gradient axis represents RGB re-ordering. Figure 7a and b shows sample images generated using augmentation process.

4.3 Hyperparameter tuning

Hyperparameter tuning was performed using Grid search on CovCNN models. The tuning process was performed using the five hyperparameters: Number of epochs, learning rate,
dropout rate in the dropout layer, batch size and gradient update optimization algorithms. The range of the hyperparameters where: dropout factor is between 10 and 35% in steps of 5%; number of epochs as 20, 25, 30, 35, 40 and 45. Batch size search space was 16, 32 and 64; gradient optimizers as ADAM or SGD or RMSprop. It was found that dropout probabilities of 25% and 30% resonated well in most of the versions of CovCNN models. The best epoch was chosen to be 40 with the batch size of 32 which yielded better performance. RMSProp was the best gradient optimizer in most of the versions, but in some versions of CovCNN, ADAM yielded better results. In order to avoid overfitting and increase the impact of generalization, a dropout of 30% has been done for regularization.

4.4 Model training and validation

The proposed CovCNN model is trained with the augmented chest X-ray images to analyze the custom models and pre-trained models and is evaluated. The distribution of the samples into training and validation dataset is in 4:1 ratio, i.e. 525 images for training and 132 images are used for testing purpose. Target classes with 525 images include 219 samples of COVID-19 and 306 samples of non-COVID-19 (normal + viral pneumonia) class. The input size of the training and testing images are $224 \times 224 \times 3$, batch size of 32 is used during the experiment. The models were trained for 40 to 50 epochs with early-stopping configuration, i.e. the training stops once the model stops improving performance on the validation set. The model was developed with the system configuration as follows 10th Generation intel core, i7 core, 16 GB Ram, 512 SSD with 4 GB NVIDIA Graphics Card.

The model is evaluated on various classification metrics that include binary accuracy (frequency with which predicted value matches true value, an idempotent operation that simply divides total by count), confusion matrix, recall, F1-score, sensitivity, and specificity. In order to improve the proposed model ability to generalize well, data set are artificially expanded by generating the different version of images in the dataset by leveraging the augmentation technique. Table 1 includes the learning process recorded by the CovCNN model with respect to the number of epochs, model loss, model accuracy curve respectively.

Fig. 7 a Different variations of a COVID-19 sample after augmentation process. b different variations of a normal sample after augmentation process
Table 1  Learning process recorded by the CovCNN model with respect to the number of epochs, model loss, model accuracy curve respectively.

| Model       | Loss % | Accuracy % |
|-------------|--------|------------|
| CovCNN_1    | ![Loss Curve](image1) ![Accuracy Curve](image2) | Accuracy: 93.43 % |
| CovCNN_2    | ![Loss Curve](image3) ![Accuracy Curve](image4) | Accuracy: 95.45%   |
| CovCNN_3    | ![Loss Curve](image5) ![Accuracy Curve](image6) | Accuracy: 96.97%   |
| CovCNN_4    | ![Loss Curve](image7) ![Accuracy Curve](image8) | Accuracy: 98.48%   |
From the learning curves detailed in Table 1, the dynamics of the model from these curves can be analyzed for under-fit, over-fit, and good fit. A model is said to be underfitted if its training loss curve in the learning curves remains flat regardless of how training or training loss continues to decrease until the end of training. A model is said to be overfitted if the plot of training loss continues to decrease with experience and the validation loss curve decreases till a certain point and then starts increasing. Finally, a model is said to be a good fit, when training loss and validation loss decreases to a point of stability, and the gap between both the curves is small.

In the case of CovCNN_1 model during the first 30 epochs in the learning curves the model depicts good fit behavior, but after 30 epochs the model started overfitting as the validation loss started increasing. CovCNN_2 model shows some improvement depicting the good-fit behavior, though the validation accuracy shows noisy movement around the training accuracy, this is due to the low-complexity of the model. CovCNN_3 trained for 32 epochs (early stopping configuration), the learning curve depicts good-fit behavior and outputs an acceptable accuracy of 96.97% with very good sensitivity and specificity value. CovCNN_4 a deep-neural network model, the learning plots depicts good-fit behavior within 25 epochs achieving an accuracy of 98.48%.

4.5 Comparison with pre-trained models

The performance of the pre-trained model with respect to the training and validation accuracy curves is shown in Table 2, and the interpretations are detailed in this section. ResNet101 generates a good fit until 10 epochs as the loss curve keeps descending. After that, there is an increase in loss function causing disturbance in the accuracy curve from which we can infer that the model overfits the data resulting in an accuracy of 90.91%. Xception model fits good up to epoch 13 resulting an acceptable accuracy of about 96.21%. However, after 13 epochs there is a gap between training and validation gaps that indicates overfitting in the model.

InceptionV3, ResNet50V2 and InceptionResNetV2, VGG16 achieved an accuracy of 96.97% resulting in a slight overfitting that can be observed from the gap between training and validation loss leading to a penalty of generalization error. VGG19 shows a good improvement over all other pre-trained models resulting in an accuracy of 98.48%. The overall validation accuracy of the proposed CovCNN model and the pre-trained model is shown in the Table 3. The Receiver operating characteristic (ROC) curve and the confusion matrix is outlined post-training on the respective validation sets. Table 4 records the ROC observations obtained for the custom CovCNN model along with the confusion matrix. Similarly the ROC curve is also obtained for the transfer learning models along with the confusion matrix and is presented in Table 5.

4.6 Visual interpretation of the trained model features

Figure 8 shows the visualization of features extracted by the inner layers of the proposed model CovCNN during the training process. This visualization helps a lot in understanding how a model is interpreting the image internally. Here image (Cov2d_45) represents visualization of the output after applying 32, $3 \times 3$ filters. Followed by image (max_pooling_2d_38), represents the output after applying max-pooling to each 32 filters. Here the edges are strengthened as compared to the Cov2d_45 image. Similarly, further convolution blocks, which helps in extracting more fine-grained features from the image. From image
(Cov2d_47), various versions (128) have been created, each of them representing some feature and contributing to the fully connected convolution layers for the final classifications.

4.7 Effectiveness of the proposed architecture

The effectiveness of the proposed CovCNN model is analyzed using various performance metrics like accuracy, sensitivity, specificity and F1-score. The resultant performance metric values are detailed in this section.

4.7.1 Binary accuracy

Binary accuracy is the frequency with which the predicted value matches with the actual value divided by total prediction made. The expression for binary accuracy is given in Eq. (5).

| Table 2 | Learning process recorded for the pre-trained models with respect to the number of epochs, model loss, model accuracy curve respectively |
|---------|-----------------------------------------------------------------------------------|
| Model   | Loss % | Accuracy % |
| ResNet101 | ![Graph](image) | ![Graph](image) | Accuracy: 90.91% |
| Xception | ![Graph](image) | ![Graph](image) | Accuracy: 96.21% |
| InceptionV3 | ![Graph](image) | ![Graph](image) | Accuracy: 96.97% |
Table 2 (continued)

| Model          | Accuracy  |
|----------------|-----------|
| InceptionResNetV2 | 96.97%    |
| VGG16          | 96.97%    |
| ResNet50V2     | 96.97%    |
| VGG19          | 98.48%    |

![Graphs showing training and validation loss and accuracy for different models](image)

\[
\text{Accuracy} = \frac{TN + TP}{TN + FN + TP + FP}
\]  

(5)
4.7.2 Sensitivity and specificity

The proportion of correctly predicted positive data is called sensitivity whereas the proportion of correctly predicted negative data is called specificity which is given in Eq. (6) and (7). In the proposed work, the number of patients who are correctly identified as COVID-19 positive is referred as sensitivity and the number of patients who are correctly identified as non-COVID-19 is termed as specificity.
Table 4  Observations recorded by the CovCNN model in the evolution of predicting Confusion matrix and ROC curve respectively

| Model Name | Confusion Matrix | ROC Curve |
|------------|------------------|-----------|
| CovCNN_1   | ![Confusion Matrix](image1) | ![ROC Curve](image2) |
| CovCNN_2   | ![Confusion Matrix](image3) | ![ROC Curve](image4) |
| CovCNN_3   | ![Confusion Matrix](image5) | ![ROC Curve](image6) |
| CovCNN_4   | ![Confusion Matrix](image7) | ![ROC Curve](image8) |
Table 5 Observations recorded by the transfer learning pre-trained models in the evolution of predicting Confusion matrix and ROC curve respectively

| Model Name | Confusion Matrix | ROC Curve |
|------------|------------------|-----------|
| VGG16      | ![Confusion Matrix](image1) | ![ROC Curve](image2) |
| ResNet101  | ![Confusion Matrix](image3) | ![ROC Curve](image4) |
| Xception   | ![Confusion Matrix](image5) | ![ROC Curve](image6) |

\[
Sensitivity = \frac{TP}{TP + FN} \quad \quad \quad (6)
\]
\[
Specificity = \frac{TN}{TN + FP} \quad \quad \quad (7)
\]
| Architecture       | Confusion Matrix | ROC Curve |
|-------------------|------------------|-----------|
| VGG19             | ![VGG19 Confusion Matrix](image) | ![VGG19 ROC Curve](image) |
| ResNet50V2        | ![ResNet50V2 Confusion Matrix](image) | ![ResNet50V2 ROC Curve](image) |
| InceptionV3       | ![InceptionV3 Confusion Matrix](image) | ![InceptionV3 ROC Curve](image) |
| InceptionResNetV2 | ![InceptionResNetV2 Confusion Matrix](image) | ![InceptionResNetV2 ROC Curve](image) |

*Table 5 (continued)*
4.7.3 F1-score

F1-score is the value obtained from a combination of precision and recall. Here, precision is the proportion of true positives from all the examples that are classified as positive.
by the neural network. Whereas recall is the same as sensitivity defined previously from Eq. (8), (9) and (10).

\[
F = 2 \times \frac{\text{Precision} \times \text{recall}}{\text{Precision} + \text{recall}}
\]

(8)

**Table 6** Quantitative performance validation results of CovCNN architectures on the COVID-19 X-ray dataset

| Network   | F1-score | Specificity | Sensitivity | Accuracy | Confusion matrix | True class | Predicted class |
|-----------|----------|-------------|-------------|----------|------------------|------------|-----------------|
|           |          | C           | 0.92        | 0.9091   | 1.0000          | 0.9394     | 44   8           |
| CovCNN_1  |          | NC          | 0.95        | 0.9318   | 1.0000          | 0.9545     | 44   6           |
| CovCNN_2  | 0.96     | 0.9659      | 0.9773      | 0.9697   | 43   3           |
|           |          | NC          | 0.98        | 0.9773   | 1.0000          | 0.9848     | 44   2           |
| CovCNN_4  |          | [Best]      | 0.98        | 0.9773   | 1.0000          | 0.9848     | 44   2           |

**Table 7** Quantitative performance validation results of pre-trained architectures on the COVID-19 X-ray dataset

| Network | F1-score | Specificity | Sensitivity | Accuracy | Confusion matrix | True class | Predicted class |
|---------|----------|-------------|-------------|----------|------------------|------------|-----------------|
|         |          | C           | 0.84        | 1.0000   | 0.7273           | 0.9091     | 32   0           |
| ResNet101|        | NC          | 0.94        | 0.9886   | 0.9773           | 0.9848     | 12   88          |
| VGG19   | 0.98     | 0.9659      | 0.9773      | 0.9697   | 43   3           |
| VGG16   | 0.96     | 0.9659      | 0.9773      | 0.9697   | 43   3           |
| Inception V3| | 0.95      | 0.9886      | 0.9318   | 0.9697          | 41   1           |
| ResNet50V2 | 0.96   | 0.9659      | 0.9773      | 0.9697   | 43   3           |
| InceptionResNetV2 | 0.96  | 0.9659      | 0.9773      | 0.9697   | 43   3           |
| Xception| 0.98     | 0.9659      | 0.9545      | 0.9621   | 42   3           |
|         |          | NC          | 0.99        | 0.9659   | 0.9545           | 0.9621     | 2    85          |
Tables 4 and 5 represent the compiled performance metric of all the models along with the confusion matrix for each model. Tables 6 and 7 represent the results of custom and pretrained models respectively. CovCNN_1 achieved an accuracy, sensitivity and specificity of 93.9%, 0.909 and 1 respectively. This initial CovCNN model has been improved as CovCNN_2 achieving an accuracy, sensitivity and specificity of 95%, 0.932 and 1 respectively. Further this model was improved as CovCNN_3 achieving an accuracy, sensitivity and specificity of 97%, 0.97 and 0.98 respectively. The best model for binary classification in the custom model category is CovCNN_4 achieving an accuracy of 98.48%, sensitivity as 1 and specificity as 0.97. In the pre-trained category, the best model is VGG19 with 98.48%, sensitivity as 97.73% and specificity as 98.86%. The accuracy achieved with the proposed algorithm is compared with the existing and is given in Table 8.

| Ref No | COVID-19 images | Type of images | Class | Link | Accuracy | Algorithm |
|--------|----------------|----------------|-------|------|----------|-----------|
| [17]   | 2780           | Chest X-ray images | coronavirus, pneumonia, and normal X-ray imagery | Guangzhou Medical Center, China | 97.4 | CovXNets |
| [22]   | 337            | Chest X-ray images | COVID-19 and Healthy X-ray images | Fukushima [9] | 97.97 | nCOVnet |
| Proposed | 876             | Chest X-ray images | COVID-19 and Healthy X-ray images | | 98.48 | CovCNN |

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{9}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{10}
\]

Tables 4 and 5 represent the compiled performance metric of all the models along with the confusion matrix for each model. Tables 6 and 7 represent the results of custom and pretrained models respectively. CovCNN_1 achieved an accuracy, sensitivity and specificity of 93.9%, 0.909 and 1 respectively. This initial CovCNN model has been improved as CovCNN_2 achieving an accuracy, sensitivity and specificity of 95%, 0.932 and 1 respectively. Further this model was improved as CovCNN_3 achieving an accuracy, sensitivity and specificity of 97%, 0.97 and 0.98 respectively. The best model for binary classification in the custom model category is CovCNN_4 achieving an accuracy of 98.48%, sensitivity as 1 and specificity as 0.97. In the pre-trained category, the best model is VGG19 with 98.48%, sensitivity as 97.73% and specificity as 98.86%. The accuracy achieved with the proposed algorithm is compared with the existing and is given in Table 8.

5 Conclusion and future scope

The significant problem in the current scenario is the precise classification and identification of the COVID-19 dataset. One of the crucial aspects is the discovery of important features that distinguish between COVID-19 and non-COVID-19 images. A deep learning approach using chest X-ray images with promising potential is presented to distinguish infection of COVID-19 from normal lung X-ray images. The objective of this research is to automatically classify the chest X-ray images using efficient deep learning intelligent model to provide faster classification. The proposed methodology aims to develop a unique CovCNN model that provides better generalization performance for classifying the COVID-19 lung X-ray images. Four deep learning custom created models
are developed of which three of them (CovCNN_1, CovCNN_2, CovCNN_3) set the baseline for the final proposed custom model i.e., CovCNN_4. Along with this, the data is also tested on different pre-trained transfer learning models and hyperparameter tuning has been implemented in order to achieve optimal accuracy. The proposed CovCNN model achieved an accuracy of 98.5% that outperforms the other versions along with the pre-trained models. This shows efficiency of the proposed CovCNN model that outperforms the existing pre-trained transfer learning models like ResNet-101, VGG-16, VGG-19, Inception-V3, ResNet-50V2, InceptionResNet-V2 and Xception. It can be extremely helpful for medical practitioners and radiologists to aid them in fastened and accurate diagnosis and following up cases. Following are the recommendations for further research.

(i) CovCNN_4 model can be further improved by fusion of CNN and pre-trained model features to enhance the accuracy.
(ii) Multi-class classification can be designed to classify various types of lung diseases.
(iii) A generalized system can be developed to detect the different mutants of COVID-19.
(iv) Hyperparameter tuning and changes in CovCNN_4 model might help in detection of different mutants of COVID-19 from x-ray images and/or CT scan images.

Appendix 1

Summary of CovCNN_1 model

| Layer (type)                     | Output Shape | Param # |
|----------------------------------|--------------|---------|
| conv2d_8 (Conv2D)                | (None, 220, 220, 32) | 2432    |
| average_pooling2d (AveragePooling2D) | (None, 73, 73, 32) | 0       |
| conv2d_9 (Conv2D)                | (None, 69, 69, 64) | 51264   |
| average_pooling2d_1 (AveragePooling2D) | (None, 23, 23, 64) | 0       |
| flatten_4 (Flatten)              | (None, 33856)   | 0       |
| dropout_4 (Dropout)              | (None, 33856)   | 0       |
| dense_8 (Dense)                  | (None, 256)     | 8667392 |
| dense_9 (Dense)                  | (None, 2)       | 514     |

Total params: 8,721,602
Trainable params: 8,721,602
Non-trainable params: 0
Appendix 2

Summary of CovCNN_2 model

| Layer (type)            | Output Shape    | Param # |
|-------------------------|-----------------|---------|
| conv2d_10 (Conv2D)      | (None, 222, 222, 32) | 896     |
| average_pooling2d_2 (Average) | (None, 111, 111, 32) | 0       |
| conv2d_11 (Conv2D)      | (None, 109, 109, 64) | 18496   |
| average_pooling2d_3 (Average) | (None, 54, 54, 64) | 0       |
| flatten_5 (Flatten)     | (None, 186624)  | 0       |
| dropout_5 (Dropout)     | (None, 186624)  | 0       |
| dense_10 (Dense)        | (None, 256)     | 47776000|
| dense_11 (Dense)        | (None, 2)       | 514     |

Total params: 47,795,906
Trainable params: 47,795,906
Non-trainable params: 0

Appendix 3

Summary of CovCNN_3 model

| Layer (type)            | Output Shape    | Param # |
|-------------------------|-----------------|---------|
| conv2d_12 (Conv2D)      | (None, 220, 220, 32) | 2432    |
| max_pooling2d_8 (MaxPooling2) | (None, 110, 110, 32) | 0       |
| conv2d_13 (Conv2D)      | (None, 106, 106, 64) | 51264   |
| average_pooling2d_4 (Average) | (None, 53, 53, 64) | 0       |
| conv2d_14 (Conv2D)      | (None, 51, 51, 64) | 36928   |
| max_pooling2d_9 (MaxPooling2) | (None, 25, 25, 64) | 0       |
| flatten_6 (Flatten)     | (None, 40000)   | 0       |
| dropout_6 (Dropout)     | (None, 40000)   | 0       |
| dense_12 (Dense)        | (None, 128)     | 5120128 |
| dense_13 (Dense)        | (None, 2)       | 258     |

Total params: 5,211,010
Trainable params: 5,211,010
Non-trainable params: 0
Appendix 4

Summary of CovCNN_4 model

| Layer (type)    | Output Shape       | Param # |
|-----------------|--------------------|---------|
| conv2d_15 (Conv2D) | (None, 220, 220, 32) | 2432    |
| conv2d_16 (Conv2D) | (None, 216, 216, 32) | 25632   |
| max_pooling2d_10 (MaxPooling) | (None, 108, 108, 32) | 0       |
| dropout_7 (Dropout) | (None, 108, 108, 32) | 0       |
| conv2d_17 (Conv2D) | (None, 104, 104, 64) | 51264   |
| conv2d_18 (Conv2D) | (None, 100, 100, 64) | 102464  |
| max_pooling2d_11 (MaxPooling) | (None, 50, 50, 64) | 0       |
| dropout_8 (Dropout) | (None, 50, 50, 64) | 0       |
| conv2d_19 (Conv2D) | (None, 46, 46, 128) | 204928  |
| conv2d_20 (Conv2D) | (None, 42, 42, 128) | 409728  |
| max_pooling2d_12 (MaxPooling) | (None, 21, 21, 128) | 0       |
| dropout_9 (Dropout) | (None, 21, 21, 128) | 0       |
| flatten_7 (Flatten) | (None, 66448) | 0       |
| dense_14 (Dense) | (None, 256) | 14450944 |
| dense_15 (Dense) | (None, 2) | 514     |

Total params: 15,247,906
Trainable params: 15,247,906
Non-trainable params: 0

References

1. Apostolopoulos ID, Mpesiana TA (2020) Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. Phys Eng Sci Med 43(2):635–640
2. Apostolopoulos ID, Aznaouridis SI, Tzani MA (2020) Extracting possibly representative COVID-19 biomarkers from X-ray images with deep learning approach and image data related to pulmonary diseases. J Med Biol Eng 40:462–469
3. Ardakani AA, Kanafi AR, Acharya UR, Khadem N, Mohammadi A (2020) Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: results of 10 convolutional neural networks. Comput Biol Med 121:103795
4. Balas VE, Kumar R, Srivastava R (eds) (2020) Recent trends and advances in artificial intelligence and internet of things, Springer International Publishing, Cham
5. Bernheim A, Mei X, Huang M, Yang Y, Fayad ZA, Zhang N, Chung M (2020) Chest CT findings in coronavirus disease-19 (COVID-19): relationship to duration of infection. Radiology 295:200463
6. Bi Q, Wu Y, Mei S, Ye C, Zou X, Zhang Z, Feng T (2020) Epidemiology and transmission of COVID-19 in Shenzhen China: analysis of 391 cases and 1,286 of their close contacts. MedRxiv 361:1767
7. Fang Y, Zhang H, Xu Y, Xie J, Pang P, Ji W (2020) CT manifestations of two cases of 2019 novel coronavirus (2019-nCoV) pneumonia. Radiology 295(1):208–209
8. Fukushima K (1980) A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol Cybern 36:193–202
9. Fukushima K (2007) Neocognitron. Scholarpedia 2(1):1717
10. Gozes O, Frid-Adar M, Greenspan H, Browning PD, Zhang H, Ji W, Siegel E (2020) Rapid AI development cycle for the coronavirus (covid-19) pandemic: initial results for automated detection & patient monitoring using deep learning and image analysis. Preprint at http://arXiv.org/abs/2003.05037
11. Hemdan EED, Shouman MA, Karar ME (2020) Covidx-net: a framework of deep learning classifiers to diagnose covid-19 in x-ray images. Preprint at http://arXiv.org/abs/2003.11055
12. Hubel DH, Wiesel TN (1970) The period of susceptibility to the physiological effects of unilateral eye closure in kittens. J Physiol 206(2):419–436
13. Kaburlasos VG, Petridis V (2000) Fuzzy lattice neurocomputing (FLN) models. Neural Netw 13(10):1145–1170
14. Khan AI, Shah JL, Bhat MM (2020) CoroNet: a deep neural network for detection and diagnosis of COVID-19 from chest X-ray images. Comput Methods Progr Biomed 196:105581
15. Li L, Qin L, Xu Z, Yin Y, Wang X, Kong B, Xia J (2020) Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT. Radiology 296:E65–E75
16. Liu KC, Xu P, Lv WF, Qiu XH, Yao JL, Gu JF, Wei W (2020) CT manifestations of coronavirus disease-2019: a retrospective analysis of 73 cases by disease severity. Eur J Radiol 126:108941
17. Mahmud T, Rahman MA, Fattah SA (2020) CovXNet: a multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. Comput Biol Med 122:103869
18. Minaee S, Kafieh R, Sonka M, Yazdani S, Soufi GJ (2020) Deep-covid: predicting covid-19 from chest X-ray images using deep transfer learning. Med Image Anal 65:101794
19. Narin A, Kaya C, Pamuk Z (2020) Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. Preprint at http://arXiv.org/abs/2003.10849
20. Ohata EF, Bezerra GM, das Chagas JVS, Neto AVL, Albuquerque AB, de Albuquerque VHC, Reboucas Filho PP (2020) Automatic detection of COVID-19 infection using chest X-ray images through transfer learning. IEEE/CAA J Autom Sin 8(1):239–248
21. Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR (2020) Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput Biol Med 121:103792
22. Panwar H, Gupta PK, Siddiqui MK, Morales-Menendez R, Singh V (2020) Application of deep learning for fast detection of COVID-19 in X-Rays using nCOVNet. Chaos Solitons Fractals 138:109944
23. Sethy PK, Behera SK, Rathia PK, Biswas P (2020) Detection of coronavirus disease (COVID-19) based on deep features and support vector machine. Int J Math Eng Manag Sci 5(4):643–651. https://doi.org/10.33889/IJMEMS.2020.5.4.052
24. Shan F, Gao Y, Wang J, Shi W, Shi N, Han M, Shi Y (2020) Lung infection quantification of COVID-19 in CT images with deep learning. Preprint at http://arXiv.org/abs/2003.04655
25. Singh D, Kumar V, Kaur M (2020) Classification of COVID-19 patients from chest CT images using multi-objective differential evolution–based convolutional neural networks. Eur J Clin Microbiol Infect Dis 39(7):1379–1389
26. Song Y, Zheng S, Li L, Zhang X, Zhang X, Huang Z, Yang Y (2020) Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. MedRxiv. https://doi.org/10.1101/TCBB.2021.3065361
27. Sousa RT, Marques O, Soares FAA, Sene II Jr, de Oliveira LL, Spoto ES (2013) Comparative performance analysis of machine learning classifiers in detection of childhood pneumonia using chest radiographs. Procedia Comput Sci 18:2579–2582
28. Tuncer T, Dogan S, Ozyurt F (2020) An automated residual exemplar local binary pattern and iterative ReliefF based COVID-19 detection method using chest X-ray image. Chemom Intell Lab Syst 203:104054
29. Venkatesan R, Li B (2017) Convolutional neural networks in visual computing: a concise guide. CRC Press, Boca Raton
30. Wang L, Lin ZQ, Wong A (2020) Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. Sci Rep 10(1):1–12
31. Wang S, Kang B, Ma J, Zeng X, Xiao M, Guo J, Xu B (2020) A deep learning algorithm using CT images to screen for corona virus disease (COVID-19). medRxiv. Preprint at https://www.medrxiv.org/content/https://doi.org/10.1101/2020.02.14, v5. Accessed 29 Oct 2020
32. Wang SH, Nayak DR, Guttery DS, Zhang X, Zhang YD (2021) COVID-19 classification by CCShNet with deep fusion using transfer learning and discriminant correlation analysis. Inf Fusion 68:131–148
33. Zbontar J, LeCun Y (2016) Stereo matching by training a convolutional neural network to compare image patches. J Mach Learn Res 17(1):2287–2318

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