# Artificial Intelligence(CSE3013)

**Team Members**

Shashwat Sanket 17BCE1007

Jayraj Thakor 17BCE1017

Urmilla Singh 18BCE1259

# Application of deep learning in detection of Novel Coronavirus from Chest X-Rays using Cov19Net and well-known pre-trained Convolutional Neural Networks.

**Abstract**

In a world with over 25,298,875 infected people and with the numbers rising 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 arduous, not cost-effective and requiring a well- equipped laboratory for analysis. This paper uses the patient's chest X-ray images for the diagnosis of novel coronavirus with an aim to assist the medical practitioners in-order to fasten the diagnostic process amongst high workload conditions. In our work, we have made a deep convolutional neural network (CNN) based architecture, named as Cov19Net. Our work consists of 5 versions of Cov19Net in which each successive versions are evolved from previous versions. These models utilise depthwise convolution with varying dilation rates for efficiently extracting diversified features from chest X-rays. 657 chest x-rays with laboratory proven CoVID-19 were included of which 219 were X-ray images of patients infected from COVID-19 and the remaining were the images of non-COVID patients. In the later part of the paper we have evaluated the dataset on different pre-trained models and have made inference from the loss and accuracy curve for all the models. Our proposed model Cov19Net, which is trained and tested on a validation dataset shows the experimental result of 98.48% accuracy. From the pre-trained models, MobileNet was the best performing model with an accuracy of 99.24%.

**INTRODUCTION**

Coronavirus disease 2019 or (COVID-19) has been at an outburst since March 2020, as one of the deadliest pandemic in over the decades and has since then been escalating. It has disrupted the economy and caused turbulences as it could never be 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 52 lakh cases with a death toll of 1 lakh. 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, runny 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) belonging to the same category. The world wide count of the infected people is appalling, with 25,298,875people being infected and 847,602 resulting in death as of September 1, 2020. 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 modes of transmission of the virus is through coughing, sneezing, respiratory droplets from one human to the others. These common symptoms which ain’t yet so futile may further result in pneumonia, multi-organ failure and death. One way to test them is the Laboratory testings 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 atypical and viral pneumonia diseases such as influenza, severe acute respiratory syndrome (SARS), and Middle East respiratory syndrome (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. Further investigation is required for improving the performance of the radiologist. The Chinese government manoeuvred Real-time polymerase chain reaction (RT-PCR) for the diagnosis of the disease. However, it did not provide accurate results as a consequence 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 nCov infection as well. This further leads to fabricating a Deep Learning-based Research Technique to analyse the chest X-ray without radiologist's intervention.

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 (CNN) 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.. The proposed model is trained by considering the chest X-ray images of COVID-19 patients. The further summary of the paper is as follows: the literature review section discusses the existing work in the field of COVID-19; the “Proposed model” section discusses the different model architectures developed; the “Experimental analysis and Result” section analyses the performance of our proposed and pre-trained models along with the discussion about the training and testing process and the paper is finally concluded with a conclusion and future scope in the “Conclusion section”.

**Literature Survey**

Our study determines various machine learning models specifically trained for detecting the disease amongst Covid-19 X-Ray dataset. Machine learning methods come in handy in critical tasks. Henceforth, we made use of deep neural technique by computer vision method 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 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 , segmentation of the fungus images. Further, pneumonia diagnosis in infants classified by Sousa et al [1] through computer-aided systems from radiographic images using SVM, K-Nearest Neighbor, and Naïve Bayes, resulted in SVM outperforming the others. Turker et al [2] 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 which included 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.[4]. 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 whooping 97.4% accuracy, Tanvir et al [5] proposed CovXNets for detecting the Covid-19 and other pneumonia using chest X-ray images. The Guangzhou Medical Center, China analysed 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 [6] to analyse COVID-19 dates using nCOVnet includes 24 layers where the first layer is the input layer and the other 18 layers are the combination of Convolution+ReLU and Max Pooling layers. They achieved an accuracy of 97.97% confidence. 121 patients' chest CT scans were analysed by Berheim et al. [7] to identify the relationship between the symptoms. To distinguish between community acquired pneumonia and other non-pneumonia lung diseases, Li et al. [8] 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. [9] successfully performed AI based tool to test COVID-19

Shan et al. [10] developed a VB-net deep learning model to segment the infection sites in the CT scans images. For discriminating pneumonia and influenza- A viral pneumonia, Xu et al [11] studied the work using a CNN based prediction model based on deep learning techniques and achieved 86.7%. Wang et al[12] achieved 89.5% prediction level using a modified inception transfer learning model. Narin et al. [13] performed an automatic deep convolutional neural network based prediction using chest X-ray images with ResNet50 achieving 98% accuracy. Sethy et al. [14] 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, Tulin et al. [19] achieved an accuracy of 98.8 for binary classification and 87.02 for multi class using DarkNet model using an automatic COVID-19 detection. and Singh et al. Hemdan et al. [15] made use of COVIDX-Net deep learning model to diagnose COVID-19. Singh et al. [20] using chest CT images where fine tuning of CNN parameters performed an optimized CNN to classify COVID-19 and multi objective differential evolution.

Wang and Wong [16] proposed COVID-Net achieving 98.75% accuracy for two class classification for classifying, normal, non-COVID pneumonia, and COVID-19 classes. Ioannis et al [17] achieved an accuracy of 98.75 using a deep learning model using 224 COVID-19 images.Zheng et al. [48] proposed a three-dimensional deep CNN model to detect COVID-19 from CT scan imagery and reported a 90.8% accuracy. Ying et al. [18] achieved an accuracy of 86% using modified Inception models for CT images.

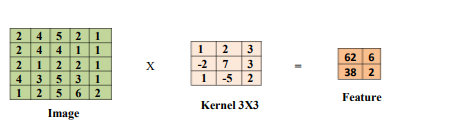
In recent years, a lot of work has been done in the field of classifying the Covid-19 data using X-ray images. It's been challenging due to its inherent texture variations and similarity towards other diseases like pneumonia. Several other studies have been encountered to be developing for classification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

**Methodologies**

**CNN**

Convolutional networks are influenced by biological processes[22][23][24][25], 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 perceptrons means networks that are completely connected. Henceforth, each neuron in one layer is linked to all neurons in the next layer.

Convolution Neural Networks (CNN)[21] 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. Various activation functions like Relu, tanh, sigmoid, softmax etc are used. subsequently followed by additional convolutions like pooling layers, normalization layers, fully connected layers. The Convolution layer is designed as structures of three dimensions: width, height and depth. The Convolution layer includes Convolution kernels (height and width), number of input and output channels and the Convolution filters depth similar to the feature map. By using regularized weights over fewer parameters, the vanishing gradient and exploding gradient problems seen in traditional neural networks during backpropagation are avoided[26][27].

****

**Fig 0.1 Example for convolution operation**

**Pooling Layers**

Subsampling or downsampling, also known as pooling, is a simple process where we reduce the size or dimensionality of the Feature Map to reduce the number of parameters needed to train, whilst retaining the most important features.Max, Average, and Sum are the 3 types of Pooling we can apply. In our models we have mainly used average and max pooling. Typically Pooling is done using 2x2 windows with a Stride of 2 and with no padding applied for smaller input images, for larger images we use larger pool windows of 3x3. According to the above information, pooling has the effect of reducing the dimensionality (width and height) of the previous layer by half. Moreover, it can thus remove ¾ or 75% of the activations seen in the previous layer. Pooling makes our model more invariant to small and minor transformations or distortions in our input image since we’re now averaging or taking the max output from a small area of our image.

**Fully connected Layer**

Fully Connected 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. All probabilities must sum to 1, e.g: (0,2, 0.5, 0.3).The Softmax Function acts as the outputs of the FC layer (last layer) and is in the form of probabilities.The activation function is used to produce these probabilities. Let's say the output of the last FC layer was [2, 1, 1]. According to the softmax ‘squashes’, these real value numbers are then turned into probabilities that sum to one: the output would therefore be: [0.7,0.2, 0.1].

**Performance evaluation of networks**

**TP -** True Positive

**TN -** True Negative

**FN -** False Negative

**FP -** False Positive

**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 as (1).

*(1)*

**Sensitivity and Specificity**

The proportion of correctly predicted positive data is called sensitivity whereas the proportion of correctly predicted negative data is called specificity. Here in our case, the number of patients who are correctly identified as Covid19 positive is called sensitivity and the number of patients who are correctly identified as non-Covid19 is called specificity.

(2)

(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 (3).

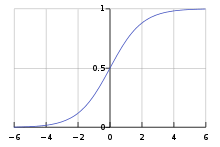
(4)

***Activation Functions***

In neural networks each of the nodes consists of an activation function. 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 functions used in our deep learning networks are: Sigmoid and ReLU.

***Sigmoid***

𝛔(z) = (5)

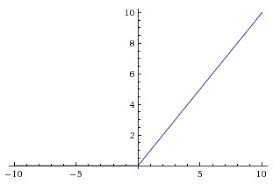


**Fig 0.2 Graph for sigmoid function**

The sigmoid function is a non-linear continuous differentiable function. It takes the input values and outputs the value between 0 and 1.

***ReLU***

ReLU stands for Rectified Linear Unit function. It is one of the most commonly used activation functions in training neural networks.



**Fig 0.3 Graph for ReLu Activation function**

**F(x) = max(0,X) (6)**

The clamping value at 0 accounts for it’s non linear behaviour. ReLU simply changes all the negative values to 0 while leaving the positives values unchanged.

**Dropout**

Dropout is the dropping nodes (both hidden and visible) in a neural network with the aim of reducing overfitting. While training, certain parts of the neural network are ignored during some forward and backward propagations.To help reduce interdependent learning amongst the neurons, dropout is utilised to regularize the neural networks. Thus the neural network learns more robust or meaningful features. In Dropout we set a parameter ‘P’ that sets the probability of which nodes are kept or (1-p) for those that are dropped. Dropout almost doubles the time to converge in training.

**Batch Normalization**

Batch Norm was proposed by Ioffe and Szgedy in 2015 in their paper titled “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. Normally, we apply a generic form of Normalization scaling for our image data from values between 0-255 to values between 0 to 1. We do this to reduce the influence of larger data points. Batch Normalization however is used to normalize the activations of an input tensor before passing it into the next layer in the network. Batch Normalization reduces the number of Epochs it takes our network to converge. It aids in regularization (reducing overfitting) and allows us to improve stability of our training, thus allowing us to use large learning rates.

**Optimizers**

Optimizers are the algorithms used to minimize the losses.These algorithms help by manipulating the learning rate to allow faster convergence and better validation accuracy. The two main optimizers used in our networks are RMSprop and Adam. we have to set our hyperparameters to control our learning rate schedule. This often ends up being a process of trial and error leading to time wastage. To solve this problem, adaptive learning rate methods (Optimizer algorithms) have been developed.

***RMSprop optimizer***

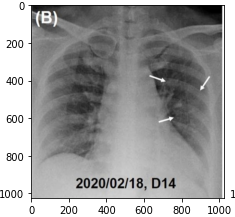
RMSprop stands for Root mean square propagation with plain momentum. This algorithm tries to dampen the vertical oscillations, thereby allowing to take larger steps in horizontal direction which results in faster convergence by increasing the learning rates. It maintains the learning rates by keeping track of average of recent gradient values of weights.

***Adam optimizer***

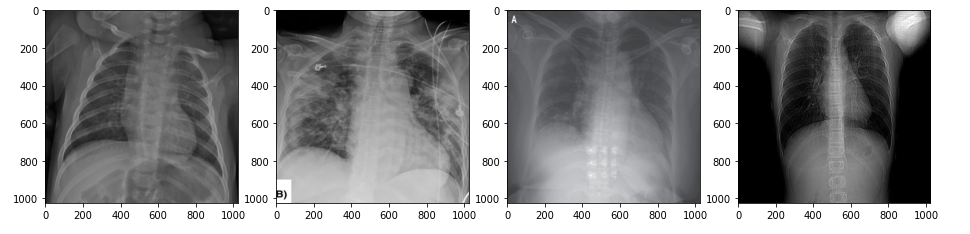
Adam stands for Adaptive moment optimization. It is an extension of the Stochastic Gradient descent method. Adam utilizes the benefits and heuristics of RMSprop and Momentum. Instead of keeping track of average of first moment as compared to RMSprop, Adam also uses the average of second moments of the gradients making it computationally efficient

**Dataset Acquisition and Pre-Processing**

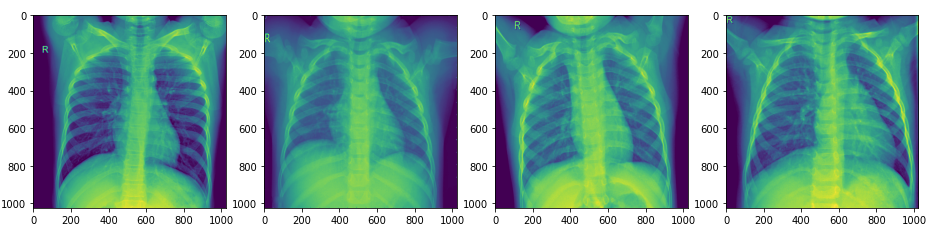
For this research the dataset is acquired from kaggle covid19-radiography-database. The database consists of chest X-ray images for COVID-19 positive cases along with Normal and Viral Pneumonia images. There are 219 COVID-19 positive images, 1341 normal images and 1345 viral pneumonia images, with the total size of approximately 1.15GB. In this study, we are not using all the images of NON-COVID samples (Normal + pneumonia) because the number of COVID samples is significantly less, thus creating the scenario of class-imbalance. So we have randomly selected 219 images from the Normal set and 219 images from the Viral Pneumonia set, thus contributing 438 images for NON-COVID class (Binary classification - [Covid, Non-Covid]). Fig 1.1 to 1.4 are some sample images from the database.



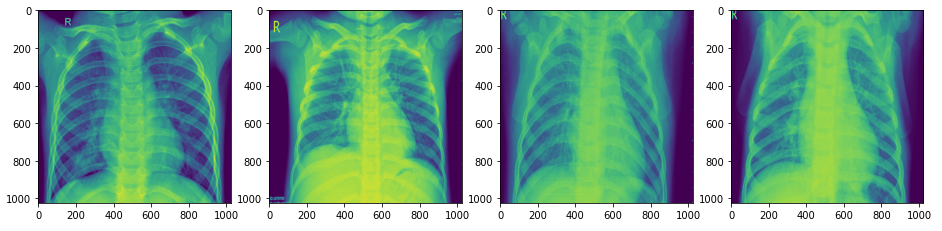
**Fig 1.1 Covid Sample  
Arrow in the image highlighting infected part**

****

**Fig 1.2 More Covid Samples**

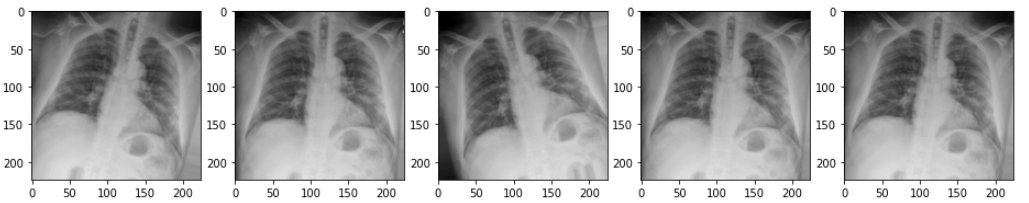
****

**Fig 1.3 Normal Case [NON-COVID Class]**

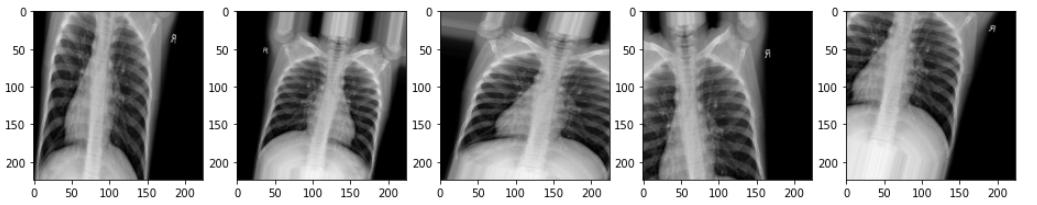
****

**Fig 1.4 Viral Pneumonia [NON-COVID Class]**

The collected images in the dataset are of different sizes and therefore we converted all the images into the same size of 224x224x3 pixels. The color-gradient axis represents RGB re-ordering. Since the size of the dataset is small, so for our model to generalize well, we performed image augmentation. The parameters of augmentation were, Rotation range of 15, 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. Fig 1.5 and 1.6 shows images generated through augmentation.

****

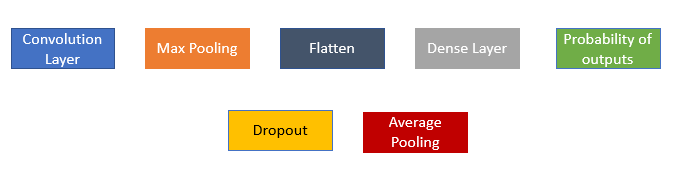
**Fig 1.5 Different variations of a COVID sample after augmentation process**

****

**Fig 1.6 Different variations of a Normal sample after augmentation process**

**Model Architecture and Development**

In this study, we present a custom CNN model called Cov19Net and nine well-known pre-trained models that can diagnose the chest x-ray image whether it is infected with the novel coronavirus or not. We have presented five incrementally improved architectures of which the last version gives the best results.

****

**Fig <Initial> Block diagram Abbreviations**

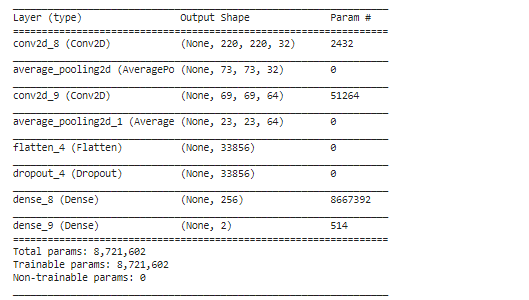
**Cov19net\_model\_1**

**Model Block Diagram:**

****

**Fig 2.0 Block diagram of Cov19net\_mode\_1**

**Model Summary**



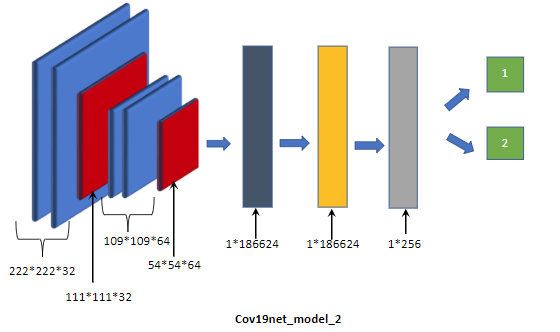
**Table 1.0 Model Summary of Cov19net\_mode\_1**

**Model Description**

The first initial sequential model "Cov19net\_mode\_1'' in Fig 2.0, takes an image of the input dimension of 224x224x3. Followed by the input image are two 2D convolution layers and average pooling layers after each convolution layer with rectified linear unit (ReLU) as an activation unit. The first convolutional layer uses 32 5x5 pixel filters that are applied to each part of the image which in return gives 32 channels of activation values as a feature map. The 3x3 kernel size is used in the average pooling layer and is responsible for dimensionality reduction and returns the average pixel value for a particular kernel. Similarly in-order to extract more fine-scale features of the image, a second convolution layer block with 64 5x5 pixel filter, followed by a 3x3 average pooling layer kernel is added which finally returns the output of shape 23x23x64. Then the 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, we choose to drop 30% less contributing neurons which in turn helps to lower the generalization error. For this, we use a dropout layer. Finally, the flattened output is fed to a feed-forward neural network with two dense layers, one with 256 neurons and ReLU as activation function and the other with 2 neurons with sigmoid as an activation function. 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. For training, binary cross-entropy is used as a loss function and RMSprop is used as an optimizer. This model gives the classification accuracy of 93.18%, specificity of 1.0, and sensitivity of 0.89. As it is a shallow network, it performs decently and thus acts as a baseline model.

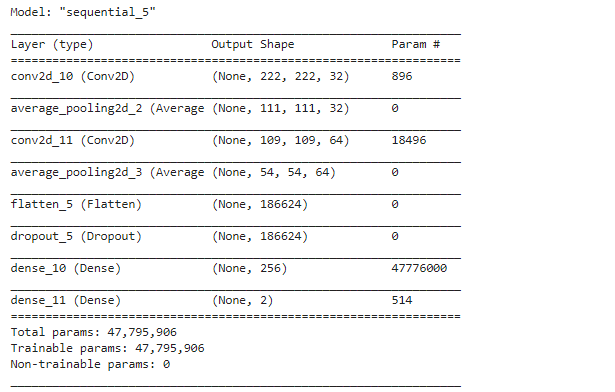
**Cov19net\_model\_2**

**Block Diagram**

****

**Fig 2.1 Block Diagram for Cov19net\_model\_2**

**Model Summary**

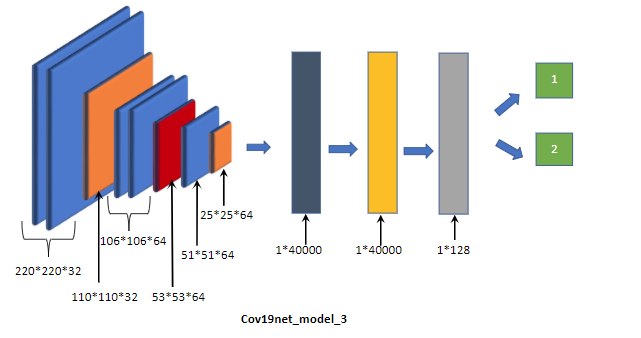
****

**Table 1.1 Model Summary for Cov19net\_model\_2**

**Model Description**

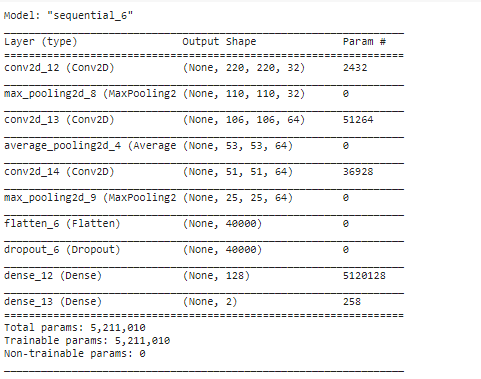
The second model "Cov19net\_model\_2" in Fig 2.1 is very similar to that of "Cov19net\_model\_1". Except for the last layer, loss function, and optimizer. Here we have used the softmax activation function because the last layer has 2 neurons instead of 1 and due to this we need to distribute the probability throughout each output node. Since we are using softmax activation, the loss function will be cross-entropy and Adam as an optimizer. The performance of this model increased with an accuracy of 94.70% with 1.00 as sensitivity and 0.92 as specificity. The focusing point here is that we want the sensitivity of the model to be high, as here in this case is 1, which means the model has the ability to correctly identify those actually having the disease (true positive rate).

**Cov19net\_model\_3**

****

**Fig 2.2 Block diagram for Cov19net\_model\_3**

**Model Summary**

****

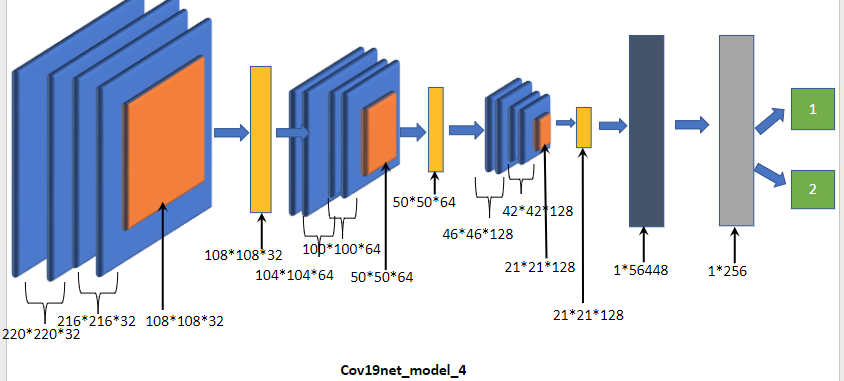
**Table 1.2 Model summary for Cov19net\_model\_3**

**Model Description**

The third model Cov19net\_model\_3 in Fig 2.2 derived from the Cov19net\_model\_1 introduces a new convolution block with 64 3x3 filters and a max-pooling layer with a kernel size of 2x2 stacked before the first 2D convolution layer of Cov19net\_model\_1. The number of neurons in the first-dense layer is decreased to 128. The reason for introducing a new convolution layer is to make the model extract more fine-grained features. And the reason for choosing average pooling over max-pooling is that the average pooling smooths out the image and hence the sharp features like lesion which depict the presence of the COVID19 virus in the chest X-RAY may not 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 like here in the case of X-ray images. Though there is a decrease in sensitivity and specificity value the Accuracy of this model was 96.21% which is greater than the accuracy of previous model Cov19net\_model\_2. From here our final aim becomes more clear as we need to get high accuracy as well as high sensitivity.

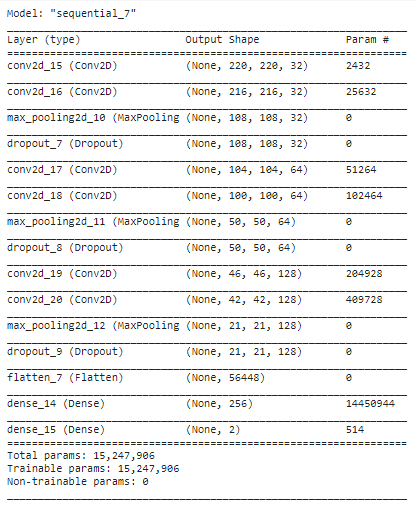
**Cov19net\_model\_4**

**Block Diagram**

****

**Fig 2.3 Block Diagram for Cov19net\_model\_4**

**Model Summary**

****

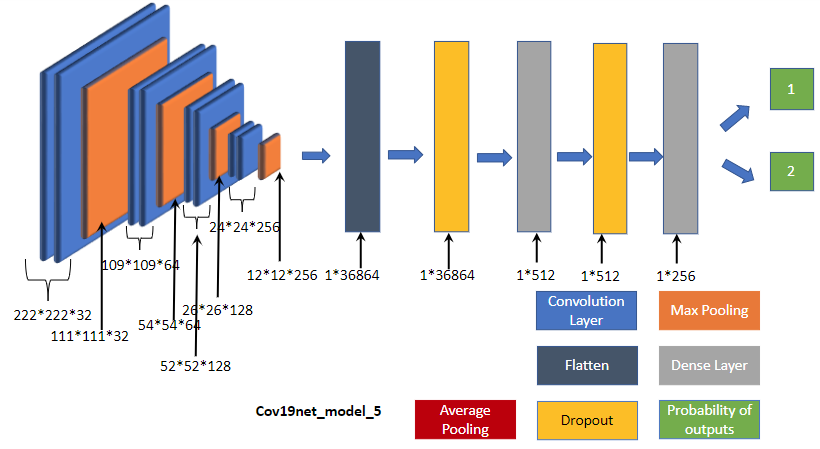
**Table 1.3 Model summary for Cov19net\_model\_4**

**Model Description**

The next model Cov19net\_model\_4 in Fig 2.3 is a deep network of 15 layers as compared to previous models. Here, convolutional layers are stacked sequentially with increasing filter sizes. The filter size of all the convolutional layers is of 5x5 with ReLU as an activation function. and max-pooling kernel size as 2x2. Here every two 2D convolutional layers plus the max-pooling layer is termed as one convolutional block. Therefore it has three convolutional blocks. Each block is followed by a dropout layer with a 30% rate which prevents the model from overfitting. This model uses binary cross-entropy as a loss function and RMSProp as an optimizer. The performance of this outperforms every previous Cov19Net model described above. The model obtained 97.73% accuracy, 0.95 sensitivity, and 0.98 specificity.

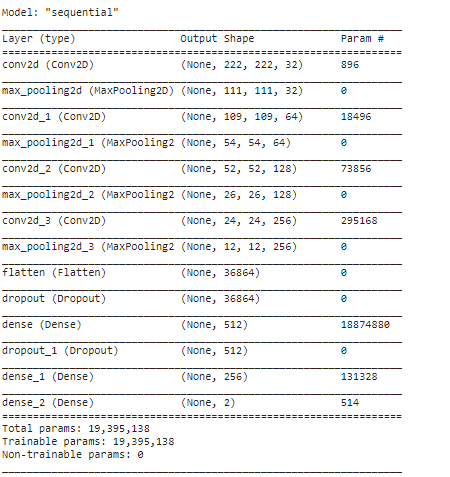
**Cov19net\_model\_5**

**Block Diagram**

****

**Fig 2.4 Block Diagram for Cov19net\_model\_5**

**Model Summary**

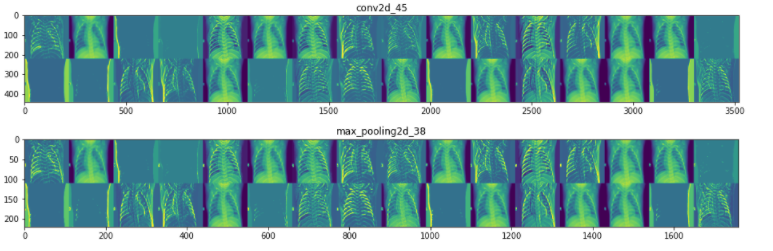
****

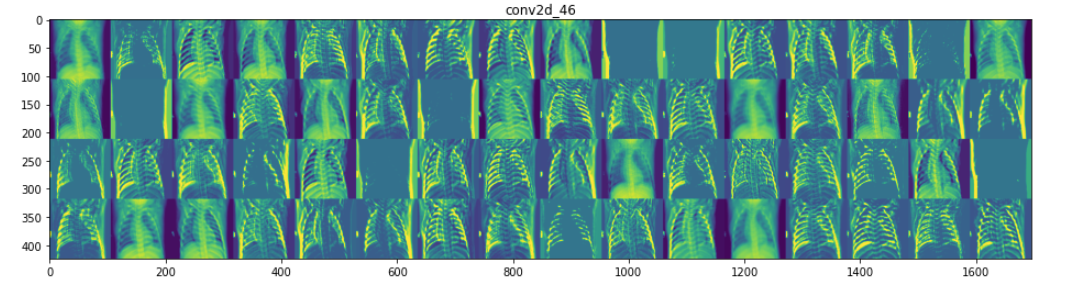
**Table 1.4 Model Summary for Cov19net\_model\_5**

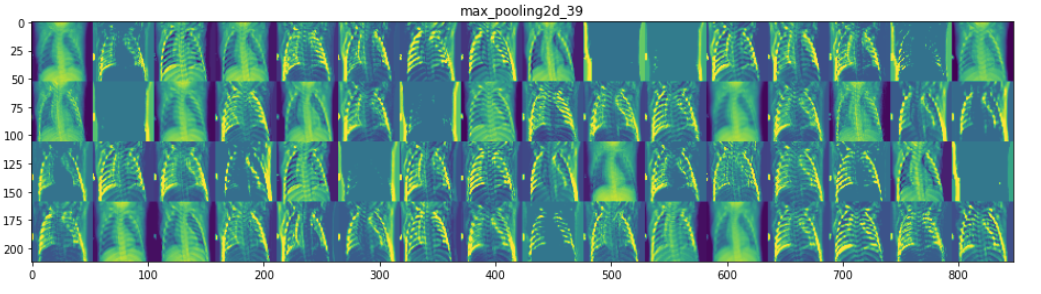
**Model Description**

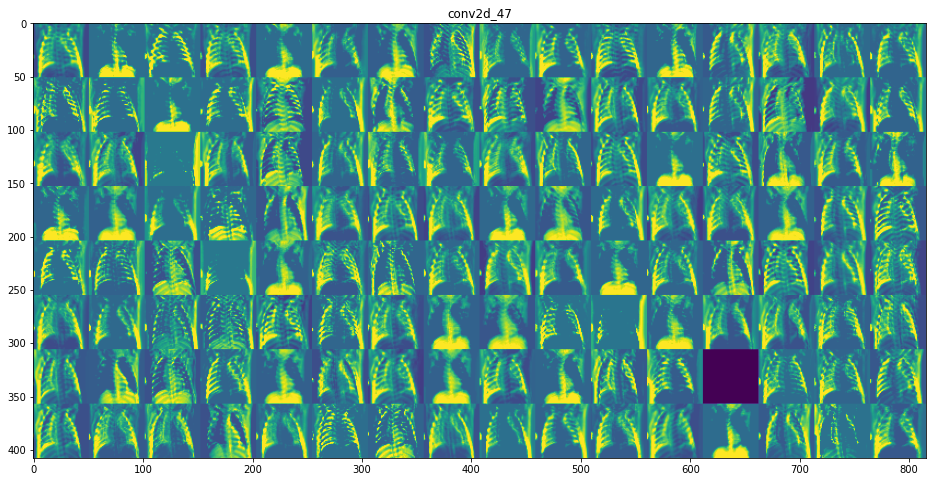
The final model Cov19net\_model\_5 in fig 2.4 gives us the best performance. This model also comes under the deep neural network category with 14 layers. Here the convolutional block consists of one 2D convolution layer and one max-pooling layer. It also follows the same increasing filter sizes convention as that of cov19net\_model\_4 except for the filter size of 3x3 in the convolutional layer and 2x2 as kernel size in the max-pooling layer. In this architecture, there are 3 dense layers with the number of neurons as 512, 256, and 2. ReLU activation function is used as an activation function in the entire network except for the final layer which uses sigmoid activation function. The model compiles with binary cross-entropy as loss function and RMSProp as an optimizer. The current architecture outperforms all previous versions with 98.48% accuracy, 1 as sensitivity, and 0.97 as specificity.

**Cov19net\_model\_5 : Inner layer visualization**









**Fig 2.5 Layer wise visualization**

Fig 2.5 shows the visualization of features extracted by the inner-layers of our best model Cov19net\_model\_5 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 3x3 filters. Followed by image (max\_pooling\_2d\_38), represents the output after applying max-pooling to each 32 filters. Here we can see 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) we can see different versions (here 128) are created, each of them representing some feature and contributing to the FCC layers for the final classifications.

**Pre-Trained Model Description**

**Transfer Learning Study**

Transfer learning in its abstract form is a method of applying knowledge acquired from one task to solve the other similar problems. Here, in our case we use the knowledge gained from pre-trained networks and apply the same to our classification problem to detect Covid-19 infected patients. In this study, we use 9 pre-trained models to segregate Covid-19 patients from that of Normal patients: 1- ResNet-101, 2- VGG-16, 3- VGG-19, 4- Inception-V3, 5- ResNet-50V2, 6- InceptionResNet-V2, 7- MobileNet-V2, 8- Xception, 9- MobileNet. We have used these deep transfer learning pre-trained models as a feature extractor and on top of this extractor we train the classifier. All the layers from the input layer and first convolution layer to the last pooling layer are used as feature extractor. The remaining part of the model is the classifier. Both ResNet-101 and ResNet-50V2 have their special residual blocks derived from ResNet. ResNet-101 consists of 33 residual blocks and is 101 layers deep. Whereas ResNet-50V2 consists of 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 was proposed by Simonyan and Zisserman. Both the architecture consists of five convolution blocks and three fully connected layers. The difference between them is that VGG-16 contains 13 convolution layers whereas VGG-19 has 16 convolution layers. Inception-V3 is the third version of Google’s Inception CNN. Inception-V3 is made up of different inception blocks where each block consists of different sized convolution and pooling layers. Each inception block does the work of multi-level feature extraction. All these similar types of Inception blocks combine to form a layered architecture called Inception-V3. InceptionResNet-50V2 conflates the idea of Inception and ResNet to produce a hybrid architecture with better performance. The InceptionResNet-50V2 is much deeper than Inception-V3 and it consists of 164 layers. MobileNet and MobileNet-V2 both are Light Weight depth wise separable convolution models. This depth wise separable convolution block in MobileNet consists of two parts where the first part is made of 3\*3 depth wise convolution and second part is 1\*1 pointwise convolution. Both the convolutions are followed by Batch Normalization and ReLU activation. The MobileNet consists of 13 such depth wise convolution blocks. MobileNet-V2 consists of similar convolution blocks but with three convolution layers namely Expansion Layer, Depthwise layer, Projection layer. Xception was created by Francois Chollet and is derived 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. It consists of a total 71 layers.

**Experimental Analysis and Results**

In this section, we will describe the analysis of custom models and pre-trained models. For each model, training and validation accuracy was recorded for each Epoch. The split between training and test dataset is in 4:1 ratio, giving 525 images in training and 132 images in the test set. Considering the target classes there were 219 samples having COVID-19 and rest 306 samples belonging to NON-COVID (Normal + Viral Pneumonia) class. The image shape was fixed to 224x224x3 for all the models. The batch size of 32 was set throughout the experimental procedure. The models were trained for 40-50 Epochs with early-stopping configuration, i.e. the training stops once the model stops improving performance on the validation set. For calculating and analysing the performance of the model, binary accuracy (frequency with which y\_pred matches y\_true, an idempotent operation that simply divides total by count), f1-score, confusion matrix, sensitivity, and specificity were computed. Since the dataset size is less, so in order to improve our model ability to fit well, in-other words generalize well, we need to artificially expand its size by generating the different version of images in the dataset by leveraging the augmentation technique. For this we used ImageDataGenerator provided by keras library, Data acquisition and preprocessing section elucidate in details about different parameters used to configure the data generator. This data-generator is then feeded into the model for the training and validation process.

**Cov19net models (Custom Models)**

Table 3.1 contains Training vs Validation loss and Training vs Validation accuracy curves for different version of Cov19net model,

|  |  |  |
| --- | --- | --- |
| **Model** | **Training vs Validation loss** | **Training and Validation Accuracy** |
| **Cov19net\_model\_1**  **ACC: 93.18 %** | **(a)** | **(b)** |
| **Cov19net\_model\_2**  **ACC: 94.70 %** | **(C)** | **(d)** |
| **Cov19net\_model\_3**  **ACC: 96.21%** | **(e)** | **(f)** |
| **Cov19net\_model\_4**  **ACC: 96.97%** | **(g)** | **(h)** |
| **Cov19net\_model\_5**  **ACC: 98.48%**  **[ BEST ]** | **(i)** | **(j)** |

**Table 3.1 Learning Curves for Cov19net models**

From the learning curves in table 3.1, we can diagnose the dynamics of the model from these curves as under-fit, over-fit, and good-fit. A model is said to be under-fitted 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 over-fitted 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 Cov19net\_model\_1, during the first 30 epochs in the learning curves (a) and (b) the model depicts good fit behaviour, but after that model started overfitting as the validation loss started increasing. The second model Cov19net\_model\_2 shows some improvement from the plot (c) and (d). The plot (c) depicts the good-fit behaviour. Though the validation accuracy shows noisy movement around the training accuracy, this is due to the low-complexity of the model. The third model that is Cov19net\_model\_3 trained for 32 epochs (early-stopping configuration), the learning curve (e) and (f) depicts good-fit behavior and outputs an acceptable accuracy of 96.21% with very good sensitivity and specificity value. The fourth model Cov19net\_model\_4 which comes under deep-neural network category, the learning plots (g) and (h) depicts good-fit till epoch 20, after which it started overfitting as well as the generalisation error also started increasing which can be seen from the gap between the training loss and validation loss thus requiring better regularization parameters or the current model is too-complex for the data. The final model which is Cov19net\_model\_5, the learning curves (i) and (j) for our best-model depicts good-fit behaviour till epoch 25 after which, it also starts overfitting. But the overall performance of the model outperforms every other version of Cov19net.

**Pre-trained models**

Table 3.2 contains Training vs Validation loss and Training vs Validation accuracy curves for different pre-trained model,

|  |  |  |
| --- | --- | --- |
| **Model** | **Training vs Validation loss** | **Training vs Validation Accuracy** |
| **ResNet101**  **ACC: 83.33%** |  |  |
| **VGG19**  **ACC: 91.67%** |  |  |
| **VGG16**  **ACC: 93.94%** |  |  |
| **InceptionV3**  **ACC: 96.97%** |  |  |
| **ResNet50V2**  **ACC: 97.73%** |  |  |
| **Inception-**  **ResNetV2**  **ACC: 97.73%** |  |  |
| **MobileNetV2**  **ACC: 97.73%** |  |  |
| **Xception**  **ACC: 98.48%** |  |  |
| **MobileNet**  **ACC: 99.24%**  **[BEST]** |  |  |

From the loss and accuracy curves of different pre-trained models shown in table 3.2, numerous interpretations can be made. The first model ResNet101 shows a good fit till 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. The second model VGG19 shows a good improvement over the previous model by increasing the accuracy with a margin of about 8%. However, we can observe that there is a gap between training and validation loss which suggests that we can move towards a more complex model. From the loss and accuracy curve of VGG16 which are somewhat coincident over the entire course, we can infer that it gives a very good performance. The fourth model InceptionV3 shows a good fit till 6 epochs. However, after that there is a slight overfitting which can be observed from the gap between training and validation loss indicating an increase in generalization error. ResNet50V2 and InceptionResNetV2 show the overfitting and suggest that the models are too complex i.e. there are too many parameters for memorizing a limited amount of information, thus resulting in poor generalization. The next model MobileNetV2 shows a good fit till 8 epochs for loss curve, after which it also starts overfitting the data. But still this model gives a good accuracy of about 97.7%. Our second best model is the Xception model. Here after 15 epochs the model fits good and gives an acceptable accuracy of about 98.5%. The last and the best performing model is MobileNet. This model gives the best result with a comprehensive accuracy of 99.24%. From the both curves, the loss and accuracy curve, it can be inferred that the validation and training curves almost perfectly coincides i.e good fit, indicating that it is the model which gives best results.

**Combined Graphs**

|  |  |  |
| --- | --- | --- |
| **Graph-Type** | **Custom Models** | **Pre-trained Models** |
| **Accuracy- Sensitivity-**  **Specificity-**  **F1** |  |  |
| **Training Accuracy** |  |  |
| **Validation Accuracy** |  |  |

**Table 3.3 Represents the combined graphical analysis of custom Cov19net models as well as pre-trained models.**

**Results**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Network** | **f1 - score** | | | **Specificity** | **Sensitivity** | **Accuracy** | **Confusion Matrix** | | **Predicted Class** |
| **True Class** | |
| **C** | **NC** |
| **Cov19net\_model\_1** | **C** | | **0.91** | **0.8977** | **1.0000** | **0.9318** | **44** | **9** | **C** |
| **NC** | | **0.95** | **0** | **79** | **NC** |
| **Cov19net\_model\_2** | **C** | **0.93** | | **0.9205** | **1.0000** | **0.9470** | **44** | **7** | **C** |
| **NC** | **0.96** | | **0** | **81** | **NC** |
| **Cov19net\_model\_3** | **C** | **0.94** | | **0.9886** | **0.9091** | **0.9621** | **40** | **1** | **C** |
| **NC** | **0.97** | | **4** | **87** | **NC** |
| **Cov19net\_model\_4** | **C** | **0.97** | | **0.9886** | **0.9545** | **0.9773** | **42** | **1** | **C** |
| **NC** | **0.98** | | **2** | **87** | **NC** |
| **Cov19net\_model\_5**  **(Best model)** | **C** | **0.98** | | **0.9773** | **1.0000** | **0.9848** | **44** | **2** | **C** |
| **NC** | **0.99** | | **0** | **86** | **NC** |

**Table 3.4 Compiled Performance metric for Cov19net model**

**Results of Pre-Trained Models**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Network** | **f1 - score** | | **Specificity** | **Sensitivity** | **Accuracy** | **Confusion Matrix** | |  |
| **True Class** | |
| **C** | **NC** |
| **ResNet101** | **C** | **0.71** | **0.9432** | **0.6136** | **0.8333** | **27** | **5** | **C** |
| **NC** | **0.88** | **17** | **83** | **NC** |
| **VGG19** | **C** | **0.86** | **0.9886** | **0.7727** | **0.9167** | **34** | **1** | **C** |
| **NC** | **0.94** | **10** | **87** | **NC** |
| **VGG16** | **C** | **0.90** | **0.9886** | **0.8409** | **0.9394** | **37** | **1** | **C** |
| **NC** | **0.96** | **7** | **87** | **NC** |
| **Inception V3** | **C** | **0.95** | **0.9886** | **0.9318** | **0.9697** | **41** | **1** | **C** |
| **NC** | **0.98** | **3** | **87** | **NC** |
| **ResNet50V2** | **C** | **0.97** | **0.9886** | **0.9545** | **0.9773** | **42** | **1** | **C** |
| **NC** | **0.98** | **2** | **87** | **NC** |
| **InceptionResNetV2** | **C** | **0.96** | **1.0000** | **0.9318** | **0.9773** | **41** | **0** | **C** |
| **NC** | **0.98** | **3** | **88** | **NC** |
| **MobileNetV2** | **C** | **0.96** | **1.0000** | **0.9318** | **0.9773** | **41** | **0** | **C** |
| **NC** | **0.98** | **3** | **88** | **NC** |
| **Xception** | **C** | **0.98** | **0.9773** | **1.0000** | **0.9848** | **44** | **2** | **C** |
| **NC** | **0.99** | **0** | **86** | **NC** |
| **MobileNet**  **(Best Model)** | **C** | **0.99** | **1.0000** | **0.9773** | **0.9924** | **43** | **0** | **C** |
| **NC** | **0.99** | **1** | **88** | **NC** |

**Table 3.5 Compiled Performance metric for Pre-trained models**

Tables 3.4 and 3.5 represent the compiled performance metric of all the models along with the confusion matrix for each model. Table 3.4 is for custom models and Table 3.5 is for pre-trained models. The best model for binary classification in the custom model category is Cov19Net\_model\_5 with the accuracy of 98.48%, sensitivity as 100% and specificity as 97.73%. In the pre-trained category the best model is MobileNet with 99.24%, sensitivity as 100% and specificity as 97.73%.

**Conclusion and Future Scope**

In conclusion, we have designed a custom model namely Cov19net for the detection of novel coronavirus from X-ray image dataset and have made comparisons with various pre-trained models. Initially, the X-ray image dataset is pre-processed and then augmented. Five versions of Cov19net have been created with each version improving over the previous ones. Our best Cov19net model gives an accuracy of about 98.5%, whereas MobileNet in pre-trained models gives an accuracy of 99.24%. Cov19net indicates promising results and an inference that with an improvised model and increase in dataset, better results can be seen. Overall, the proposed model advances the existing binary classification models for X-ray images in detection of novel coronavirus. It can be extremely helpful for medical practitioners and radiologists to aid them in fastened and accurate diagnosis and following up cases. The work can be extended for developing a generalised system which can detect different mutants of covid. Hyperparameter tuning and changes in model may help in detection of different mutants of covid from x-ray images. Apart from this, the state of the art result achieved in this paper can be used for detection of covid from CT scan images.

**References**

1. R. T. Sousa, O. Marques, F. A. A. M. N. Soares, I. I. G. Sene, L. L. G. De Oliveira, and E. S. Spoto, “Comparative performance analysis of machine learning classifiers in detection of childhood pneumonia using chest radiographs,” Procedia Comput. Sci., vol. 18, pp. 2579–2582, 2013, doi: 10.1016/j.procs.2013.05.444
2. Turker Tuncer, Sengul Dogan, Fatih Özyurt, An automated Residual Exemplar Local Binary Pattern and iterative Relief based COVIDCOVID-19 detection method using chest X-ray image, Chemometrics and Intelligent Laboratory Systems, 203 (2020) 104054
3. Ali Abbasian Ardakani, Alireza Rajabzadeh Kanafi, U. Rajendra Acharya, Nazanin Khadem, Afshin Mohammadi, “ Application of deep learning technique to manage COVIDCOVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks”, Computers in Biology and Medicine 121 (2020) 103795
4. 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 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
5. Mahmud, Tanvir, Md Awsafur Rahman, and Shaikh Anowarul Fattah. "CovXNet: A multi-dilation convolutional neural network for automatic COVIDCOVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization." Computers in Biology and Medicine (2020): 103869.
6. Panwar, Harsh, et al. "Application of Deep Learning for Fast Detection of COVIDCOVID-19 in X-Rays using nCOVnet." Chaos, Solitons & Fractals (2020): 109944.
7. Bernheim A, Mei X, Huang M et al (2020) Chest CT findings in coronavirus disease-19 (COVID-19): relationship to duration of infection https://doi.org/10.1148/radiol.2020200463
8. Li L et al (2020) Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT. Radiology. https://doi.org/10.1148/radiol.2020200905
9. Gozes O et al (2020) Rapid AI development cycle for the coronavirus (COVID-19) pandemic: initial results for automated Detection & patient monitoring using deep learning CT image analysis. arXiv preprint arXiv:2003.05037
10. Shan F, Gao Y,Wang J, ShiW, Shi N, HanM, Xue Z, Shi Y (2020) Lung infection quantification of COVID-19 in CT images with deep learning. arXiv preprint arXiv:2003.04655, 1–19, 2020
11. Liu K-C, Xu P, Lv W-F, Qiu X-H, Yao J-L, Jin-Feng G (2020) CT manifestations of coronavirus disease-2019: a retrospective analysis of 73 cases by disease severity. Eur J Radiol 108941. https://doi.org/ 10.1016/j.ejrad.2020.108941
12. Wang S, Kang B, Ma J, Zeng X, Xiao M, Guo J,Cai M, Yang J, Li Y, Meng X, Xu B (2020) A deep learning algorithm using CT images to screen for coronavirus disease (COVID-19). medRxiv preprint. https://doi.org/10.1101/2020.02.14.20023028, 1–26
13. Narin A, Kaya C, Pamuk Z (2020) Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. arXiv preprint arXiv:2003.10849
14. Sethy PK, Behera SK Detection of coronavirus disease (COVID- 19) based on deep features. Preprints 2020, 2020030300. https:// doi.org/10.20944/preprints202003.0300.
15. E.E.D. Hemdan, M.A. Shouman, M.E. Karar, COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images, 2020 arXiv preprint arXiv:2003.11055.
16. L. Wang, A. Wong, COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest Radiography Images, 2020 arXiv preprint arXiv:2003.09871.
17. Ioannis D. Apostolopoulos1, Tzani Bessiana, COVID-19: Automatic Detection from X-Ray Images Utilizing Transfer Learning with Convolutional Neural Networks, arXiv:2003.11617
18. Y. Song, S. Zheng, L. Li, X. Zhang, X. Zhang, Z. Huang, Y. Chong, Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images, medRxiv (2020).
19. Ozturk, Tulin, et al. "Automated detection of COVID-19 cases using deep neural networks with X-ray images." Computers in Biology and Medicine (2020): 103792.
20. Singh, Dilbag, Vijay Kumar, and Manjit Kaur. "Classification of COVID-19 patients from chest CT images using multi-objective differential evolution–based convolutional neural networks."European Journal of Clinical Microbiology & Infectious Diseases (2020): 1-11.
21. 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
22. Fukushima,K.(2007). "Neocognitron". Scholarpedia. 2 (1): 1717. Bibcode: 2007 SchpJ...2.1717F.
23. Hubel, D. H.; Wiesel, T. N. (1968-03-01). "Receptive fields and functional architecture of monkey striate cortex".The Journal of Physiology.195(1): 215– 243. doi:10.1113/jphysiol.1968.sp008455. ISSN 0022-3751. PMC 1557912. PMID 4966457.
24. Fukushima, Kunihiko (1980). "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position" (PDF). Biological Cybernetics. 36 (4): 193– 202. doi:10.1007/BF00344251. PMID 7370364. S2CID 206775608. Retrieved 16 November 2013.
25. Matusugu, Masakazu; Katsuhiko Mori; Yusuke Mitari; Yuji Kaneda (2003). "Subject independent facial expression recognition with robust face detection using a convolutional neural network" (PDF). Neural Networks. 16 (5): 555– 559. doi:10.1016/S0893-6080(03)00115-1. PMID 12850007. Retrieved 17 November 2013.
26. Venkatesan, Ragav; Li, Baoxin (2017-10-23).Convolutional Neural Networks in Visual Computing: A Concise Guide. CRC Press. ISBN 978-1-351-65032-8.
27. Balas, Valentina E.; Kumar, Raghvendra; Srivastava, Rajshree (2019-11-19). Recent Trends and Advances in Artificial Intelligence and Internet of Things. Springer Nature. ISBN 978-3-030-32644-9.