PROJECT REPORT

ON

"COVID-19 DETECTION USING DEEP LEARNING"

Submitted in partial fulfilment of the requirements for the partial completion of

MINI PROJECT [19EC3PWMP3]

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

SUBMITTED BY:

Bharath Gajula **1BM18EC051**

Under the Guidance of

BHAVANA H T

(Assistant Professor, ECE, BMSCE)

August 21 - January 22

Department of Electronics and Communication Engineering

B.M.S COLLEGE OF ENGINEERING

(Autonomous College Affiliated to VTU, Belgaum)
Bull Temple Road, Basavanagudi, Bangalore-560019

DECLARATION

We undersigned students of pre-final semester B.E in Electronics and Communication Engineering, BMS College of Engineering, Bangalore, hereby declare that the dissertation entitled “COVID-19 DETECTION USING DEEP LEARNING”, embodies the report of our project work carried out independently by us under the guidance of Bhavana H T, Assistant Professor, E&C Department, BMSCE, Bangalore in partial fulfilment for the award of Bachelor of Engineering in Electronics and Communication from Visvesvaraya Technological University, Belgaum during the academic year 2020-2021.

We also declare that to the best of our knowledge and belief, this project has not been submitted for the award of any other degree on earlier occasion by any student.

Place: Bangalore
Date:

B.M.S COLLEGE OF ENGINEERING

(Autonomous College under VTU)

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the project entitled "**COVID-19 DETECTION USING DEEP LEARNING**" is a bonafide work carried out by **Bhushan Avalakki**(USN:1BM18EC027), **Chinmay Biradar**(USN:1BM18EC032), **Bharath Gajula**(USN:1BM18EC051) and **Harsha Narayan**(USN:1BM18EC057) in fulfilment for the completion of MINI-PROJECT (19EC7PWMP3) during the academic year October 2021-January 2022.

Bhavana H T Assistant Professor, ECE, BMSCE	**Dr. Siddappaji** Professor and HOD, ECE, BMSCE	**Dr. S Muralidhara** Principal, BMSCE

External Examination: **Signature with date:**

1.

2.

ABSTRACT:

COVID-19 global pandemic affects health care and lifestyle worldwide, and its early detection is critical to control cases' spreading and mortality. The actual leader diagnosis test is the Reverse transcription Polymerase chain reaction (RT-PCR), result times and cost of these tests are high, so other fast and accessible diagnostic tools are needed. Inspired by recent research that correlates the presence of COVID-19 to findings in Chest X-ray images, this papers' approach uses existing deep learning models (VGG19 and U-Net) to process these images and classify them as positive or negative for COVID-19. The proposed system involves a pre-processing stage with lung segmentation, removing the surroundings which does not offer relevant information for the task and may produce biased results, after this initial stage comes the classification model trained under the transfer learning scheme.

CONTENTS:

LIST OF FIGURES:

LIST OF TABLES:

1.INTRODUCTION:

The COVID-19 is a deadly disease caused by the newly recognized coronavirus. In December 2019, coronavirus (SARS-COV-2 showed in fig1) infected the human body for the first time, and it can spread principally among humans through the droplets formed by the infected persons when they speak, cough or sneeze As the droplets are too heavy to travel far, they cannot spread person-to-person without coming in close contact Although the exact time is not yet known, a new study has estimated that the COVID-19 can be viable in the air for up to 3 hours, on copper for 4 hours and up to 72 hours on plastic and stainless steel. However, the exact answers to these questions are still not agreed upon by the general health research community and currently under investigation. COVID-19 attacks the lung and damages the tissues of an infected person. At the early stage, some people may not find any symptoms where most of the people had fever and cough as the core symptoms. Other secondary symptoms could be body aches, sore throat, and a headache could be all possible.

At present, COVID-19 disease is increasing daily due to the lack of quick detection methods. All over the world, a huge number of people died of this disease in 2020. The respiratory tract and lungs are the media where the virus can spread easily. As a result, inflammation occurs, and air sacs can be filled with fluid and discharge. The process is responsible for creating an obstacle in oxygen intake. Quick and accurate detection of the virus is a major challenge for doctors and health professionals around the world in order to reduce the death rate caused by this virus.

The proposed work here provides an intelligent machine learning architecture in order to detect COVID-19 disease using chest X-ray images.

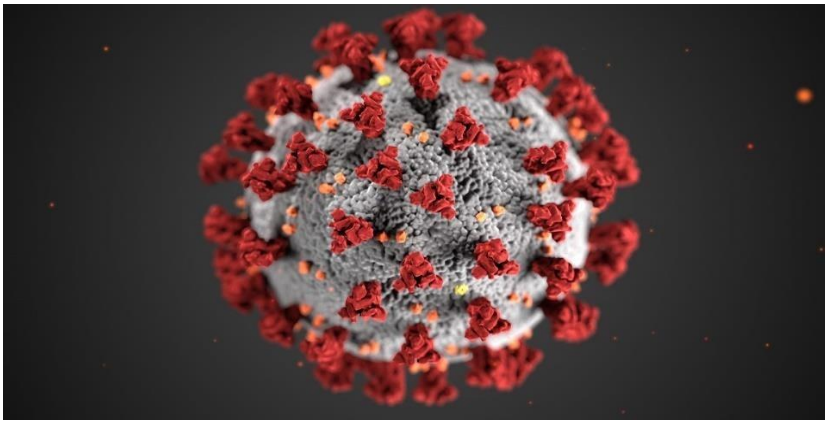


Fig 1: Coronavirus

2.LITERATURE SURVEY:

In recent months, researchers have investigated and analysed chest X-ray images using deep learning algorithms to detect COVID-19. First, the images are pre-processed using the CNN technique for extracting better features, which are fed in deep learning algorithms for image classification.

Ahamed et al. [1] proposed a deep neural network based system where CNN provided high accuracy (94.03%). The authors trained the system with normal, pneumonia and COVID-19 patient's chest X-ray images. The limitation of the work was that a dataset with only 285 images was used for developing the system, and this small number of data was not perfect for training a deep learning-based system for the COVID-19 prediction.

Chowdhury et al. [2] worked with chest X-ray images to develop a novel framework named PDCOVIDNet based on parallel-dilated CNN. In the proposed method, the authors used a dilated convolution in the parallel stack that could capture and stretch necessary features for obtaining a detection accuracy of 96.58%.

Sahlol et al. [3] proposed an improved hybrid classification approach using CNNs and marine predators' algorithm for classifying COVID-19 images, which were obtained from international cardiothoracic radiologists. Inception architecture of CNNs was employed to extract features, and a swarm-based marine predators' algorithm was used to select the most relevant features from the images. However, the research work did not consider any fusion approach to improve the classification and feature extraction of the COVID-19 images.

Azemin et al. [4] used a deep learning method based on the ResNet-101 CNN model. In their proposed method, thousands of images were used in the pre-trained phase to recognize meaningful objects and retrained to detect abnormality in the chest X-ray images. The accuracy of this method was only 71.9%.

Sekeroglu et al. [5] developed a model using deep learning and machine learning classifiers where a total of 38 experiments was conducted by CNN for the detection of the COVID-19 using the chest X-ray images with high accuracy. Among them, 10 experiments were performed using 5 different machine-learning algorithms, and 14 experiments were carried out by the state-of-the-art pre-trained network for transfer learning. The system demonstrated 98.50% accuracy, 99.18% specificity and 93.84% sensitivity. They concluded that the system developed by CNN was capable of achieving COVID-19 detection.

El-Rashidy et al. [6] introduced a framework consisted of three layers: patient layer, cloud layer and hospital layer. A set of data was collected from the patient layer using some wearable sensors and a mobile app. A neural network-based deep learning model was used to detect COVID-19 using the patient X-ray images. The proposed model achieved 97.9% accuracy and 98.85% specificity.

Minaee et al. [7] reported a deep learning-based framework to detect COVID-19 from chest Xray images using four tuning models like ResNet18, ResNet50, SqueezeNet and DensNet-121. The proposed method took advantage of data augmentation to create a transformed version of the COVID-19 images, which increased the number of samples and finally achieved 98% sensitivity and 90% specificity.

3.PROBLEM DEFINITION:

In many poorer countries have only recently begun to vaccinate in significant numbers, so waves or surges of infections are still being experienced. No other medication with high efficacy against COVID-19 has been developed. These factors mean that spread of COVID is hard to monitor, detect and overcome in less developed countries, particularly with the emergence of newer more infectious strains. RT-PCR-based tests respiratory swabs from nasopharyngeal or oropharyngeal. Although these may fail to identify COVID-19 cases in the early phases when viral load is low in the sampled tissues. However, a more significant issue is that RT-PCR is expensive, requires highly developed facilities and technical expertise that in many countries there is limited accessibility outside of large towns.

The Laboratory confirmation of SARS-CoV-2 is performed with a virus-specific RT– PCR, but the test can take up to 2 days to complete. Although these methods are to a large extent accurate, they have many drawbacks. Due to the patient number increase, there is a lack of test kits, the experienced operators' availability. There is need for a new and better method to test for the virus which yields reliable and timely results.

Researchers are working on overcoming the limitations of RT– PCR testing to enhance diagnosing and detection of the COVID-19. According to the recommendations by WHO chest imaging examination is an effective method for the detection of clinical symptoms of people who have been affected and recovered from the virus.

4. PROPOSED SOLUTION:

We propose to use multiple benchmark CNN models have been created in our proposed work which have been trained individually to make independent predictions to predict covid positive or negative using x- rays where CNN is that each layer takes the feature maps of all preceding layers as inputs. This helps to strengthen feature propagation and encourages feature reuse for making predictions. In real life, we always prefer to come up with a medical diagnosis based on multiple medical expert views. The combined opinion of the medical experts helps in reaching a more reliable conclusion. Following the same philosophy, multiple benchmark CNN models have been adopted in our proposed work. They have been trained individually to make independent predictions. Then the models are combined, using a new method of weighted average ensemble technique, to predict a class value. This newly proposed assembling method is expected to make the prediction more robust

1. Therefore we employed a pre-processing technique such as data Augmentation on the image dataset to enable the dataset to be accurately and efficiently analysed by our deep learning model.
2. We developing an extended VGG-16-based deep learning model where fine-tuning was performed to facilitate rapid detection and diagnosis of COVID-19 cases with high accuracy.
3. Classifying COVID-19 patient images from normal and typical pneumonia cases.
4. We conducted a comparative performance analysis of our proposed methodology with other state-of-the-art approaches and showed that our model can identify COVID-19 cases with high accuracy using chest CT-scan

Fig 2: Training the model

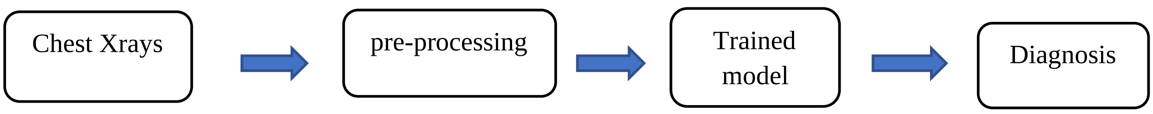


Fig 3: Deploying trained model for practised use

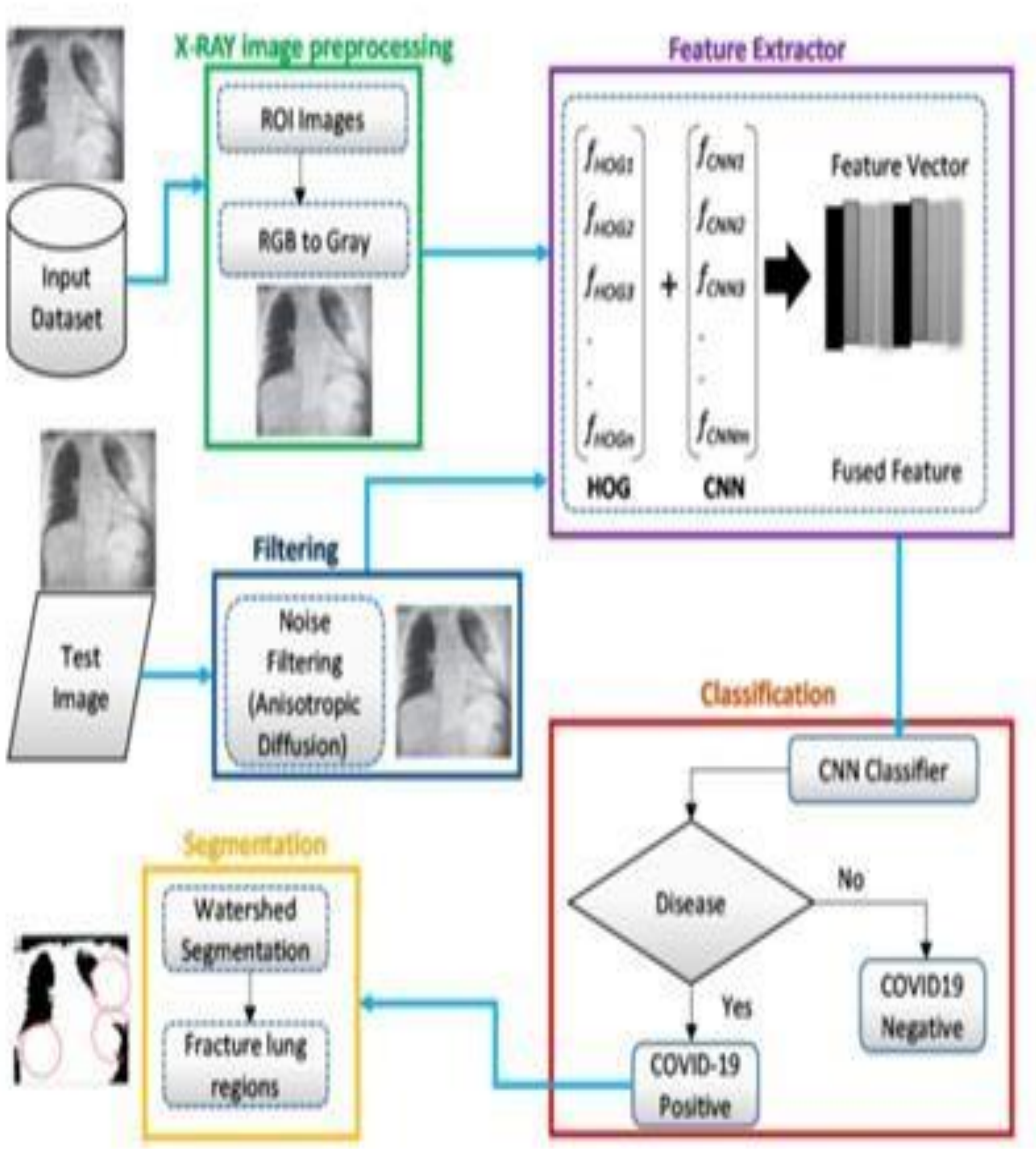


Fig 4: Overview of the proposed intelligent system architecture for identifying COVID-19 from chest X-ray images.

5. MATERIALS AND METHODS:

5.1 DATASET:

In the experiments of this study, a primary dataset containing 4131 X-ray images has been used as a base dataset. Of 4131 images, 1327 X-ray images belonged to confirmed COVID-19 patients and other 2804 images belonged to normal or people with other diseases like pneumonia. The dataset used is available on GitHub and Kaggle. The basic dataset consists of three classes of COVID-19 with 1327 samples and others with 4131 samples. Some of the pictures of the X-rays are shown in the figure 5. Thus, the dataset was imbalanced and needed pre-processing to achieve promising results. As a first attempt, CNN was trained on the given original dataset and around 66% accuracy was achieved, which was not worthy of the current application domain. The main dataset sources used in this study are enlisted as follows:

(i)For dataset balancing, a collection of chest X-ray images were collected from Kaggle.

(ii)Independent validation Primary chest X-ray image dataset of COVID-19 containing a collection of COVID-19 and normal X-ray images for the real-world testing of the proposed CNN was collected from different sources such as using github.

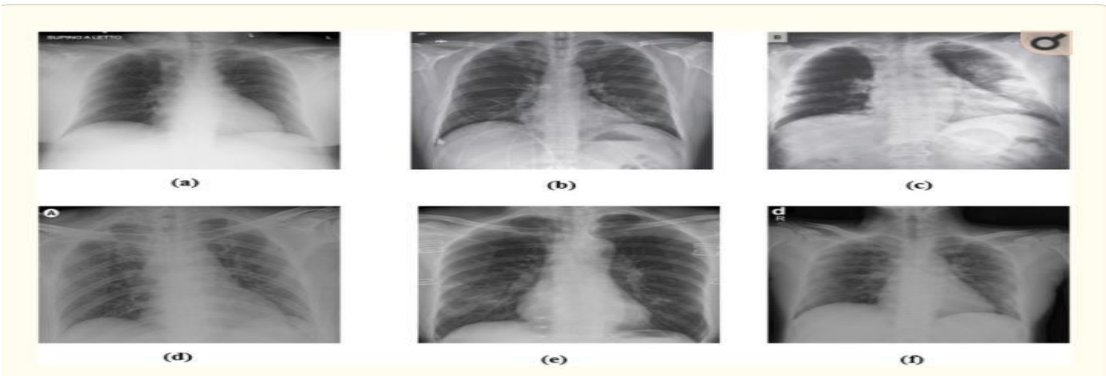


Fig 5.1: A few COVID-19 cases and findings by dataset: (a) Cardio-vasal shadow within limits, (b) Increasing left basilar opacity is visible, arousing concern about pneumonia, (c)Progressive infiltrate and consolidation, (d) Small consolidation in right upper lobe and ground-glass opacities in both lower lobes (e) Infection demonstrates right airspace opacities, (f) Progression of prominent bilateral perihilar infiltration and ill-defined patchy opacities at bilateral lungs.

5.1.1 BALANCING DATASET CLASSES:

To balance the given dataset, in order to improve the performance of the proposed CNN models in the detection of COVID-19 cases, 4131 normal chest X-ray images have been used. These concatenated extra X-ray images were downloaded from Kaggle and GitHub. After balancing the dataset when the models have been trained again on the resulting dataset, the accuracy of the given CNN models was improved. Still, the performance given by the models in terms of accuracy and other measures was not justified as an effective system for COVID-19 detection.

5.2 DATA AUGMENTATION:

Data augmentation is a technique that can significantly increase the data instances of a dataset to train a model. In the case of image datasets, the technique uses the basic image processing operations, such as flipping, rotating, cropping, or padding for augmentation. We can use data augmentation to significantly boost the diversity of the data samples for the training models. Image augmentation approaches may help to reduce network generalization errors, improve training amenities, and address data overfitting concerns. The dataset is then extended by these transformed images resulted from the existing image set, which increases the size of dataset to train the neural networks . To solve the problem of the availability of a small size dataset that was affecting the performance of the proposed CNN, the data augmentation method has been used in this study. This technique increased the size of the dataset; in addition, it provides more learning features to the learning model. Two image processing operations, flipping, zooming and rotation, have been used in this study for data augmentation.

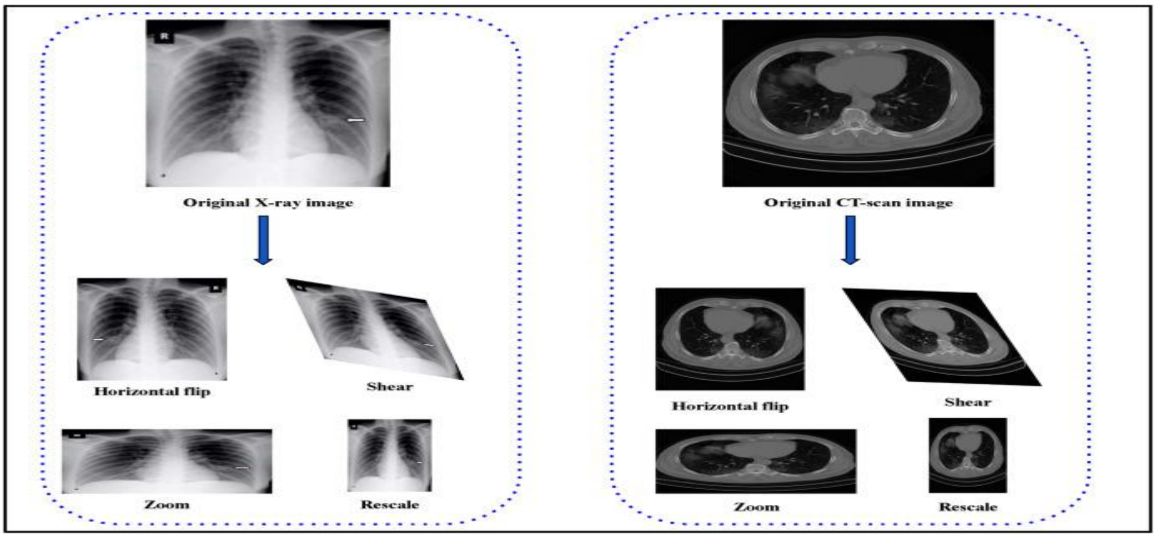


Fig 5.2: Data Augmentation of X-ray Images

5.3 CONVOLUTIONAL NEURAL NETWORKS (CNN):

As one of the most important and prominent models in deep learning methods, The convolutional neural network (CNN/ConvNet shown in figure 6) is a class of deep learning neural networks, most commonly applied to analyze visual imagery inspired by visual system of human brain. The idea behind the CNNs is to make computers capable of viewing the world as humans view it. This way CNNs can be used in the fields of image recognition and analysis, image classification, and natural language processing.

In mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

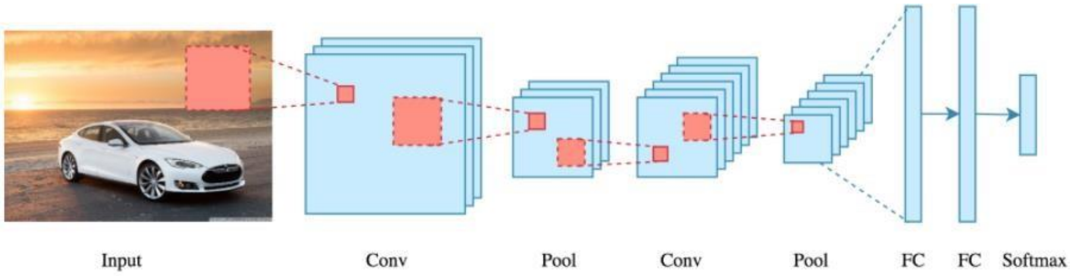


Fig 6: CNN Working

Convolutional neural networks are composed of multiple layers of artificial neurons and convolutional layers , max pooling, and nonlinear activation layers. The convolutional layer, considered as a main layer of a CNN, performs the operation called “convolution” that gives CNN its name. The mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. When you input an image in a ConvNet, each layer generates several activation functions that are passed on to the next layer. The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces. All the outputs of the convolutional layers are convolved as a feature map. In this study, the Rectified Linear Unit (ReLU) has been used in the activation function with a convolutional layer which is helpful to increase the nonlinearity in input image, as the images are fundamentally nonlinear in nature. Thus, CNN with ReLU in the current scenario is easier and faster. Since the ReLU is zero for all negative inputs which made it possible to train deep CNNs more effectively and achieve better performance.

5.3.1 TRANSFER LEARNING CNN ARCHITECTURE MODELS:

In the ImageNet LSVRC-2012 competition, a revolutionary CNN architecture named AlexNet demonstrated that deep CNNs are able to achieve excellent performance on highly challenging datasets with purely supervised learning. CNN developed a wide range of network settings and training skills, such as ReLU, dropout, pooling and local response normalization, In recent years, more advanced networks based on AlexNet were created, such as VGG, GoogLeNet, ResNet, DenseNet, MobileNet, SqueezeNet, etc. In 2013.VGG proposed a template for using loops and subroutines to design new networks. Taking advantage of repeated convolutional blocks proposed in VGG and the structure of NiN, GoogLeNet combines convolution kernels with different size, uses Inception blocks and employs 1×1 convolutions to reduce channel dimensionality. In 2015, a novel type of CNN architecture named ResNet was proposed, which profoundly influenced network structure design thereafter. ResNet realized identity mapping through inserting shortcut connections between layers, which well-defines the function complexity for adding new residual blocks and obviously improves CNNs' classification performance. DenseNet extended the architecture of ResNet by using concatenation as the cross-layer connections, and making each layer densely connected to the last layer in these connections.

The above CNN architectures mainly focus on improving model accuracy. Another stream in this field was developed to improve the training efficiency of CNNs, which focused more on reducing the computational budget of CNNs with reasonable compromise of accuracy.

5.3.2 LAYERS OF CNN NETWORKS:

The pooling layer or subsampling layer is also an important building block of CNN. On each feature map extracted through the convolution layer, the pooling layer operates independently as shown in the figure 7 To minimize overfitting and the number of extracted features, it decreases the spatial size of the feature map and returns the important features. Pooling can be the max, average, and sum in the CNN model. In this study, max pooling has been used because others may not identify the sharp features easily as compared to max pooling. In addition, the batch normalization layer has been used in this study as it involved the training of a very deep neural network. So the technique adjusts the scaling and activation to normalize the input layer and speed up the learning procedure between hidden units.

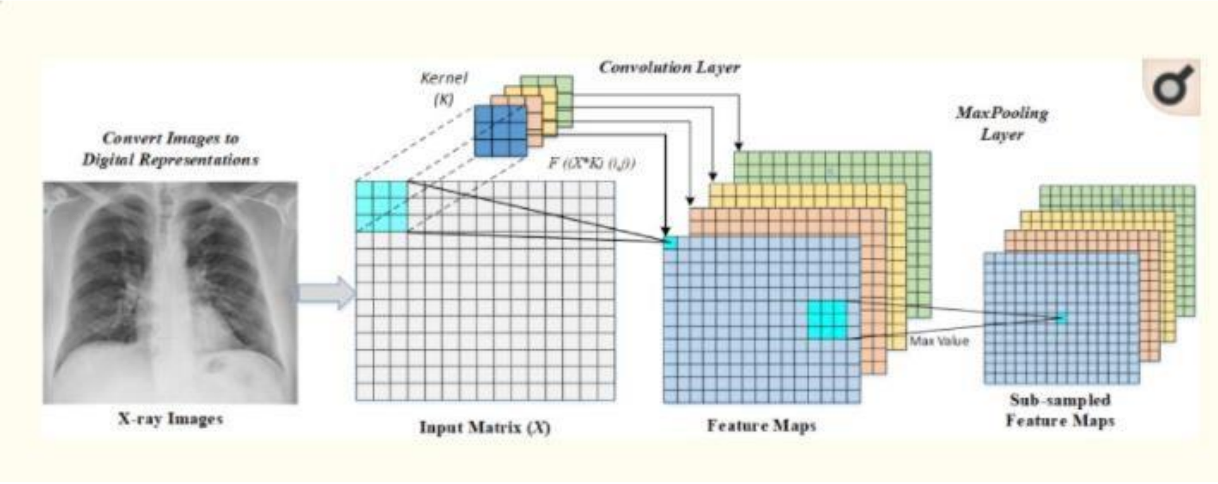


Fig 7: A schematic representation of convolution and Max-pooling layer operations.

The dropout layer with a 20% dropout rate has also been used, which drops the neurons during the training chosen at random to reduce the overfitting problem. Towards the last stage of the CNN used in the study, there is a flatting layer to convert the output of convolutional layers into a single-dimensional feature vector. In other words, the flattening layer arranges all the pixel data output produced by convolutional layers in one vector. After flattening, the vector data is given as an input to the next layers of the CNN called fully connected layers or dense layers. In a fully connected layer, each neuron of the previous layer is directly connected to each of the neurons in its next layer. The main functionality of dense layers is to take flattened output results from the convolution and pooling layers and as input and classify the image to a specific class label. Each value of the flattened feature set represents the probability of a feature belonging to a specific class. Thus, on the basis of these probabilities, the fully connected network with dense layers finally drives the classification decision.

5.4 PERFORMANCE METRICS:

To evaluate the performance of the proposed approach, the metrics adopted are classification accuracy, sensitivity and F1-score, measured as follows

Classification accuracy =TP+TN / TP+TN+FP+FN

Sensitivity =TP/TP+FN

$F1$ Score = 2×sensitivity×precision / sensitivity+precision where TP stands for True Positive, FP for False Positive, FN for False Negative and TN for True Negative. In a confusion matrix, the COVID-19 +ve cases that are correctly classified by the model are termed as True Positive and incorrectly classified as COVID –ve are termed as False Positive. Similarly, COVID –ve subjects classified correctly are termed as True Negative and incorrectly classified as COVID +ve are termed as False Negative.

5.5 VGG16:

A convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. VGG16 was trained for weeks and was using NVIDIA Titan Black GPU's, VGG-16 is shown in the figure 8.

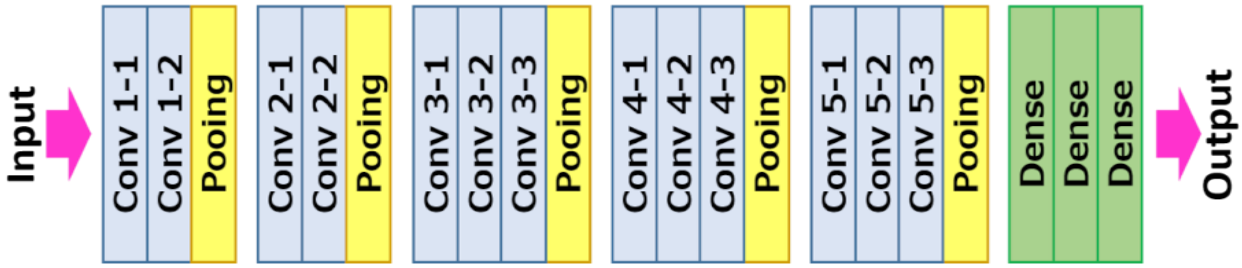


Fig 8: VGG-16

THE ARCHITECTURE:

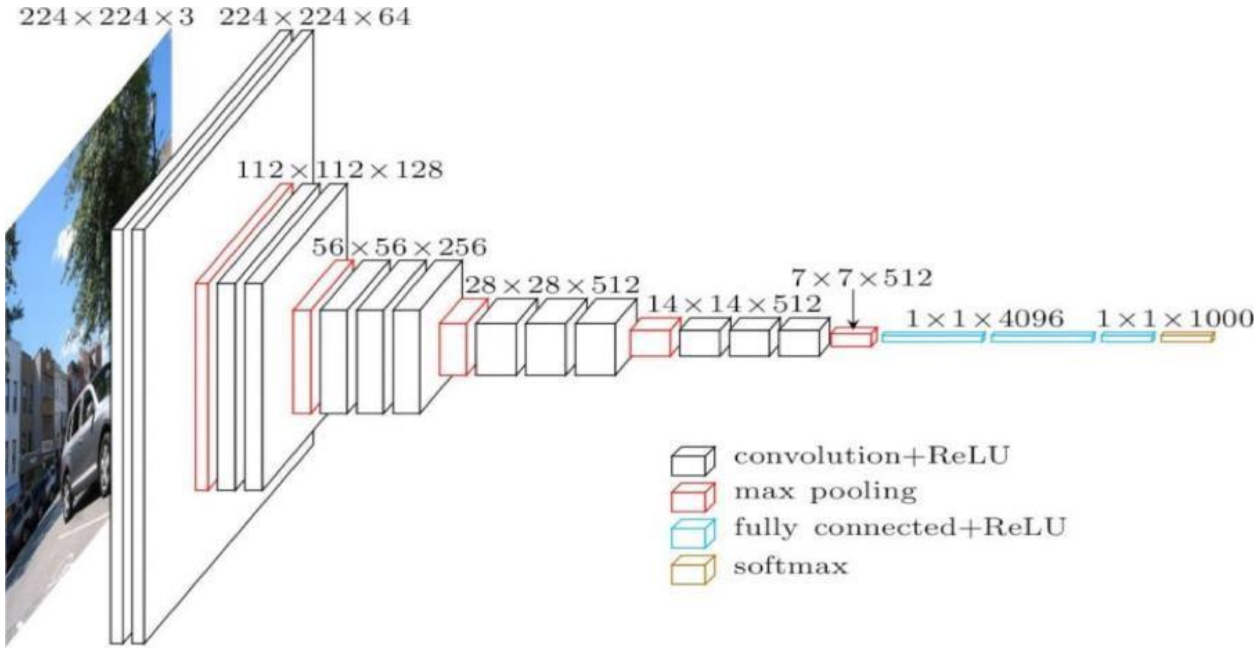


Fig 9: VGG-16 Architecture

As shown in the figure 9 VGG-16 architecture the input to conv1 layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, it also utilizes 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e., the padding is 1-pixel for 3×3 conv layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2×2-pixel window, with stride 2. Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks. All hidden layers are equipped with the rectification (ReLU) nonlinearity. It is also noted that none of the networks (except for one) contain Local Response Normalisation (LRN), such normalization does not improve the performance on the ILSVRC dataset, but leads to increased memory consumption and computation time.

6. PROPOSED CNN ARCHITECTURE MODELS:

We made five models for checking different CNN architectures accuracy with different convolutional layers for classification. We Created three models with different convolution layers and dropout layers which kept improving over time. The second model with five layers of CNN models with increasing neurons from 16, 32, 64, 128 back to 64 in model 3 with 2 layers of Convolutional layer then I finished off with a flatten and the dense layers with the classes. These base models we not learning as a proper machine learning algorithm due to the fact that our training set had a lower accuracy than our validation. This means that our model was underfitted so we did not pursue these in spite of the great scores.

Model 4 was Transfer Learning Models Unsatisfied by my results from my base models, I tried to build on top of what I had already made. I began with VGG16 transfer learning model pretrained for image datasets. On top of the VGG16 Model I added another dense layer for the number of classes and flattened the images. We continued by using various dropouts functions on VGG16 and adding more complexity but we still saw no improvement in our scores. But it seemed that the best results for these models was to keep it simple and barely add anything to it and just add the Dense layer with the class names.

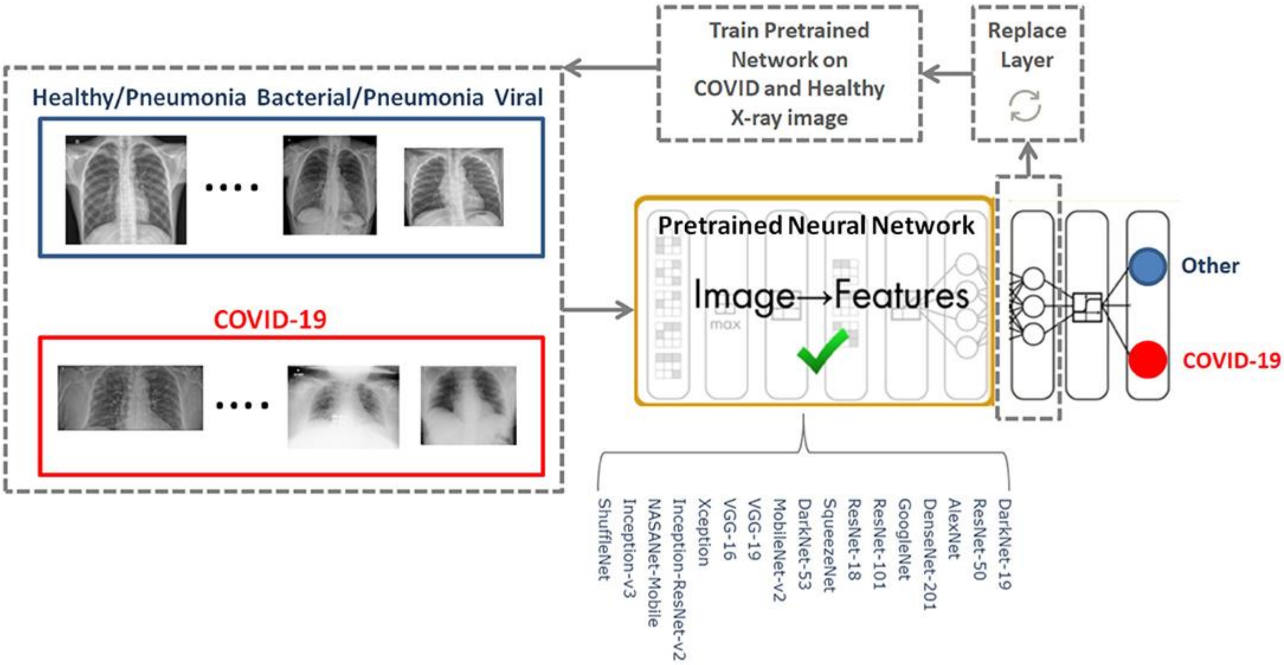
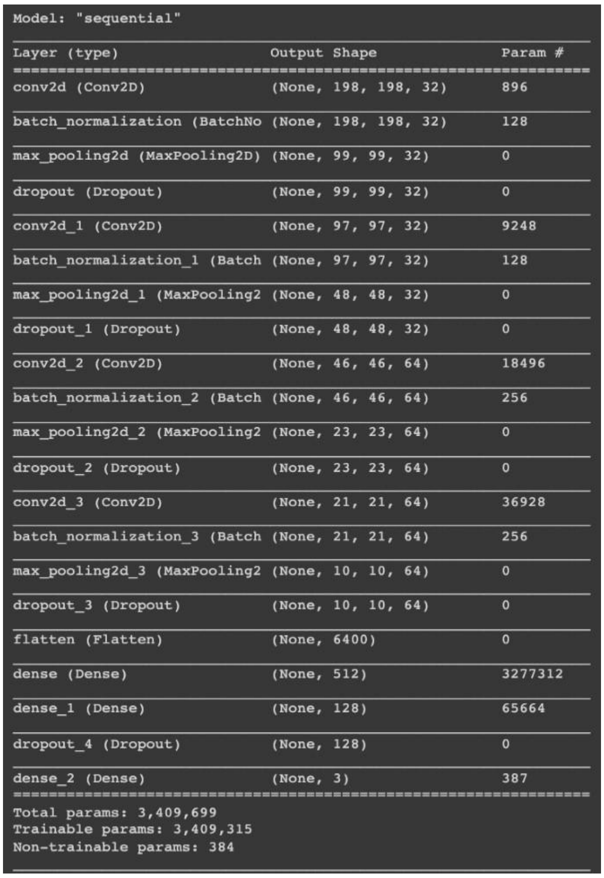


Fig 9.2:CNN Transfer learning Networks

Model 1:

This base model 1 was made with 4 with 32 ,32 ,64 and 64 filters with max pool. The model has Train Accuracy: 0.597, Train Loss: 2.957, Test Accuracy:0.663166403770446 as shown in the figure 10.1, Test Loss:1.6551761627197266 as shown in the figure 10.2



Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 198, 198, 32)	896
batch_normalization (BatchNo	(None, 198, 198, 32)	128
max_pooling2d (MaxPooling2D)	(None, 99, 99, 32)	0
dropout (Dropout)	(None, 99, 99, 32)	0
conv2d_1 (Conv2D)	(None, 97, 97, 32)	9248
batch_normalization_1 (Batch	(None, 97, 97, 32)	128
max_pooling2d_1 (MaxPooling2	(None, 48, 48, 32)	0
dropout_1 (Dropout)	(None, 48, 48, 32)	0
conv2d_2 (Conv2D)	(None, 46, 46, 64)	18496
batch_normalization_2 (Batch	(None, 46, 46, 64)	256
max_pooling2d_2 (MaxPooling2	(None, 23, 23, 64)	0
dropout_2 (Dropout)	(None, 23, 23, 64)	0
conv2d_3 (Conv2D)	(None, 21, 21, 64)	36928
batch_normalization_3 (Batch	(None, 21, 21, 64)	256
max_pooling2d_3 (MaxPooling2	(None, 10, 10, 64)	0
dropout_3 (Dropout)	(None, 10, 10, 64)	0
flatten (Flatten)	(None, 6400)	0
dense (Dense)	(None, 512)	3277312
dense_1 (Dense)	(None, 128)	65664
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 3)	387

Total params: 3,409,699
Trainable params: 3,409,315
Non-trainable params: 384

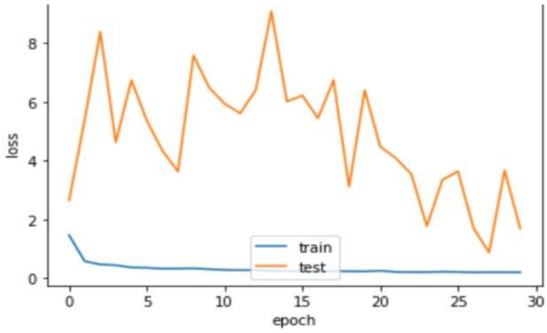


Fig 10.1: Model 1 loss vs epoch

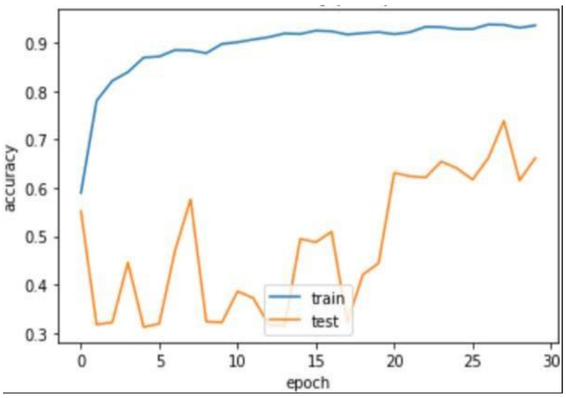


Fig 10: Model 1 Classifier Summary

Fig 10.2: Model 1 accuracy vs epoch

Model 2:

The second model with five layers of CNN models with increasing neurons from 16, 32, 64, 128. The model has Train Accuracy of 0.59, Test Accuracy of 0.619 as shown in the figure 11.1, Test Loss: 2.59 and Train Loss of 3.515 as shown in the figure 11.2.

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_14 (Conv2D)           (None, 198, 198, 32)      896
_________________________________________________________________
batch_normalization_14 (Batc (None, 198, 198, 32)      128
_________________________________________________________________
max_pooling2d_14 (MaxPooling (None, 99, 99, 32)        0
_________________________________________________________________
conv2d_15 (Conv2D)           (None, 97, 97, 64)        18496
_________________________________________________________________
dropout_13 (Dropout)         (None, 97, 97, 64)        0
_________________________________________________________________
batch_normalization_15 (Batc (None, 97, 97, 64)        256
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 48, 48, 64)        0
_________________________________________________________________
conv2d_16 (Conv2D)           (None, 46, 46, 64)        36928
_________________________________________________________________
batch_normalization_16 (Batc (None, 46, 46, 64)        256
_________________________________________________________________
max_pooling2d_16 (MaxPooling (None, 23, 23, 64)        0
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 21, 21, 128)       73856
_________________________________________________________________
dropout_14 (Dropout)         (None, 21, 21, 128)       0
_________________________________________________________________
batch_normalization_17 (Batc (None, 21, 21, 128)       512
_________________________________________________________________
max_pooling2d_17 (MaxPooling (None, 10, 10, 128)       0
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 8, 8, 128)         147584
_________________________________________________________________
dropout_15 (Dropout)         (None, 8, 8, 128)         0
_________________________________________________________________
batch_normalization_18 (Batc (None, 8, 8, 128)         512
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 4, 4, 128)         0
_________________________________________________________________
flatten_3 (Flatten)          (None, 2048)              0
_________________________________________________________________
dense_7 (Dense)              (None, 128)               262272
_________________________________________________________________
dropout_16 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_8 (Dense)              (None, 3)                 387
=================================================================
Total params: 542,083
Trainable params: 541,251
Non-trainable params: 832
```

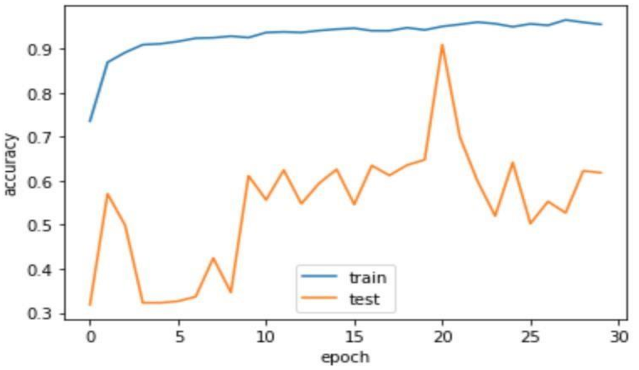


Fig 11.1: Model 2 loss vs epoch

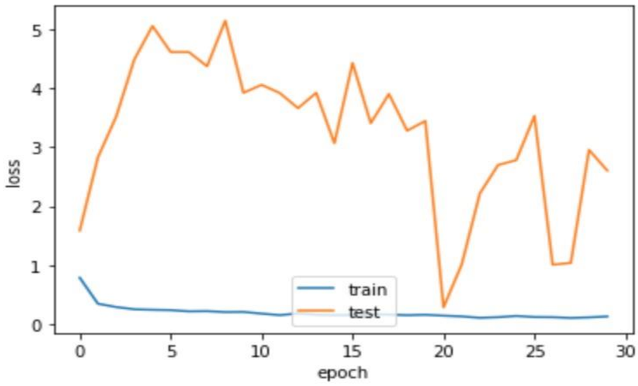


Fig 11: Model 2 Classifier Summary

Fig 11.2: Model 2 accuracy vs epoch

Model 3:

This base model 3 was made with 2 convolution filters with 32 and 32 filters with max pool and 2 dense layers. The model has Train Accuracy of 0.75,Test Accuracy of 0.688 , Test Loss of 0.9478 and Train Loss of 0.894 as shown in the figures 12.1 and 12.2.

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_19 (Conv2D)           (None, 198, 198, 32)      896
_________________________________________________________________
batch_normalization_19 (Batc (None, 198, 198, 32)      128
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 99, 99, 32)        0
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 97, 97, 32)        9248
_________________________________________________________________
dropout_17 (Dropout)         (None, 97, 97, 32)        0
_________________________________________________________________
batch_normalization_20 (Batc (None, 97, 97, 32)        128
_________________________________________________________________
max_pooling2d_20 (MaxPooling (None, 48, 48, 32)        0
_________________________________________________________________
flatten_4 (Flatten)          (None, 73728)             0
_________________________________________________________________
dense_9 (Dense)              (None, 128)               9437312
_________________________________________________________________
dropout_18 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_10 (Dense)             (None, 3)                 387
=================================================================
Total params: 9,448,099
Trainable params: 9,447,971
Non-trainable params: 128
_________________________________________________________________
```

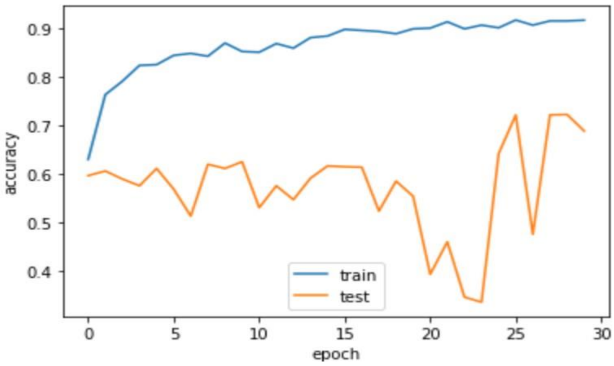


Fig 12.1: Model 3 accuracy vs epoch

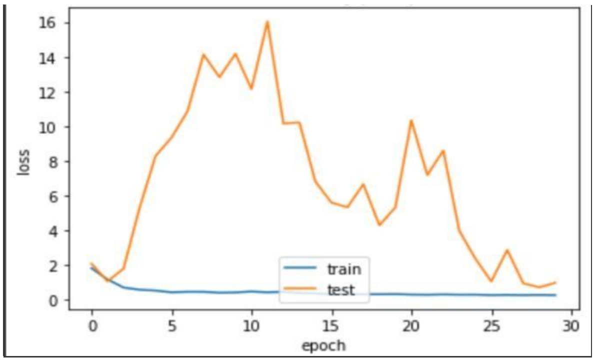
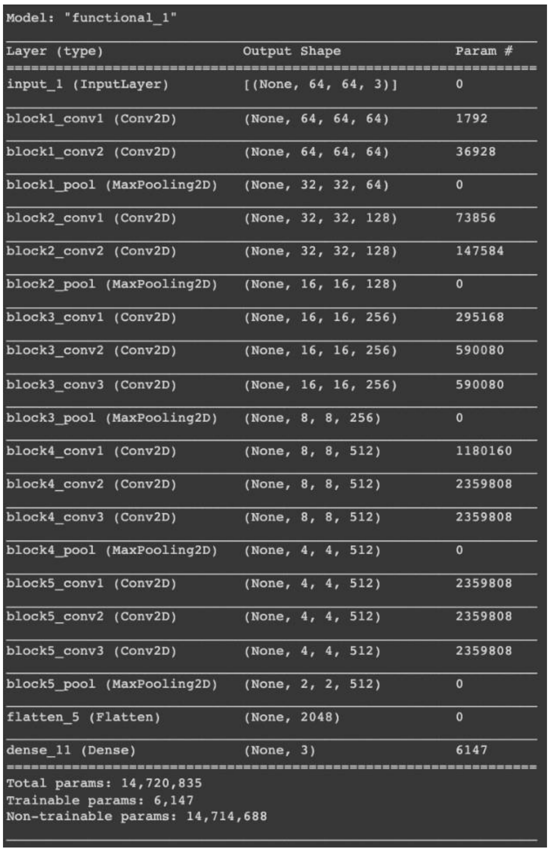


Fig 12: Model 3 Classifier Summary

Fig 12.2: Model 3 loss vs epoch

Model 4:

VGG16 transfer learning model pre trained for image datasets. On top of the VGG16 Model We added another dense layer for the number of classes and flattened the images. We continued by using various dropouts functions. The model has Train Accuracy of 0.9246, Test Accuracy of 0.859,Test Loss of 0.35 and Train Loss: 0.20 as shown in the figures 13.1 and 13.2.



Model: "functional_1"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 64, 64, 3)]	0
block1_conv1 (Conv2D)	(None, 64, 64, 64)	1792
block1_conv2 (Conv2D)	(None, 64, 64, 64)	36928
block1_pool (MaxPooling2D)	(None, 32, 32, 64)	0
block2_conv1 (Conv2D)	(None, 32, 32, 128)	73856
block2_conv2 (Conv2D)	(None, 32, 32, 128)	147584
block2_pool (MaxPooling2D)	(None, 16, 16, 128)	0
block3_conv1 (Conv2D)	(None, 16, 16, 256)	295168
block3_conv2 (Conv2D)	(None, 16, 16, 256)	590080
block3_conv3 (Conv2D)	(None, 16, 16, 256)	590080
block3_pool (MaxPooling2D)	(None, 8, 8, 256)	0
block4_conv1 (Conv2D)	(None, 8, 8, 512)	1180160
block4_conv2 (Conv2D)	(None, 8, 8, 512)	2359808
block4_conv3 (Conv2D)	(None, 8, 8, 512)	2359808
block4_pool (MaxPooling2D)	(None, 4, 4, 512)	0
block5_conv1 (Conv2D)	(None, 4, 4, 512)	2359808
block5_conv2 (Conv2D)	(None, 4, 4, 512)	2359808
block5_conv3 (Conv2D)	(None, 4, 4, 512)	2359808
block5_pool (MaxPooling2D)	(None, 2, 2, 512)	0
flatten_5 (Flatten)	(None, 2048)	0
dense_11 (Dense)	(None, 3)	6147

Total params: 14,720,835
Trainable params: 6,147
Non-trainable params: 14,714,688

Fig 13: Model 4 Classifier Summary

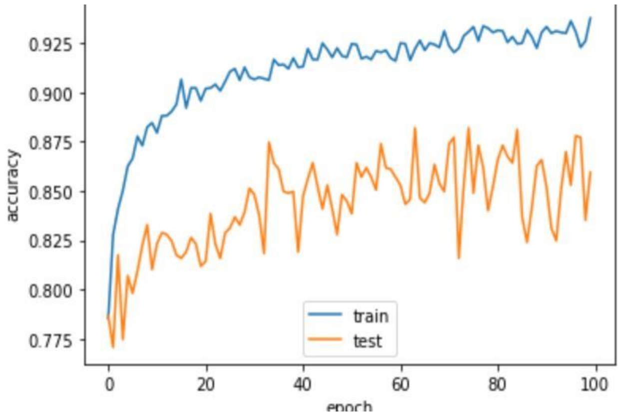


Fig 13.1: Model 4 accuracy vs epoch

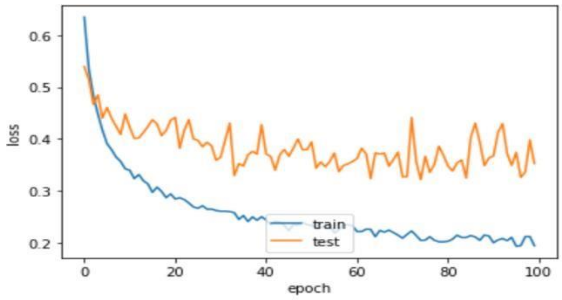
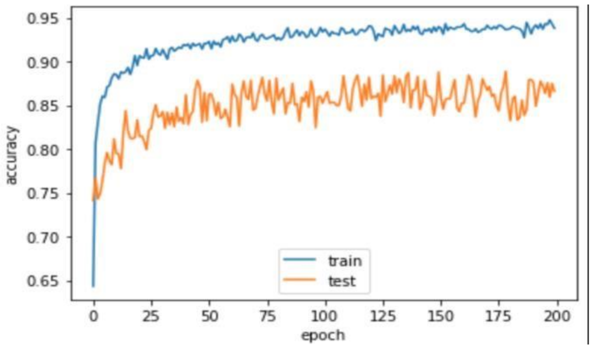


Fig 13.2: Model 4 loss vs epoch

Model 5:

VGG16 transfer learning model pre trained for image datasets. On top of the VGG16 Model We added another dense layer for the number of classes and flattened the images. We continued by using various dropouts functions. With higher epochs for training the data. After increasing the epochs, model has Train and Test Accuracy of 0.944 and 0.86752 ,Test and Train Loss of 0.36 and 0.17363 as shown in the figures 14 and 15.



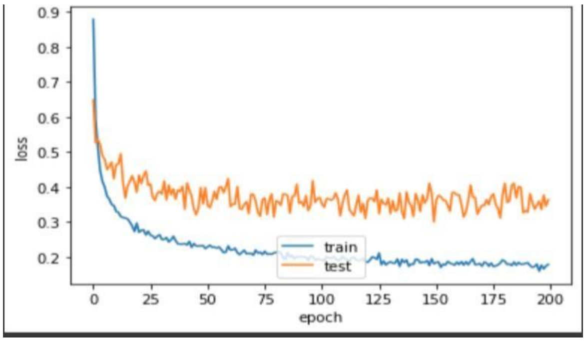


Fig 14: Model 5 accuracy vs epoch **Fig 15: Model 5 loss vs epoch**

7. FINAL MODEL PERFORMANCE MATRIX:

```
VGG16 Confusion Matrix- Validation
                       pred: COVID19  pred: NORMAL  pred: Viral_Pneumonia
true: COVID19                    368             6                       24
true: NORMAL                       0           388                       16
true: Viral_Pneumonia              0           112                      336


Classification Report
                 precision    recall  f1-score   support

        COVID19       1.00      0.92      0.96       398
         NORMAL       0.77      0.96      0.85       404
Viral_Pneumonia       0.89      0.75      0.82       448

       accuracy                           0.87      1250
      macro avg       0.89      0.88      0.88      1250
   weighted avg       0.89      0.87      0.87      1250
```

Fig 16: Final Model Performance Matrix

8. ACCURACY OF DIFFERENT MODELS:

CNN Model	Train Accuracy	Test Accuracy
Model 1 - 4 layer CNN	0.597	0.663
Model 2- 5 layer CNN	0.59	0.619
Model 3- 2 layer CNN with 2 dense layer	0.759	0.688
Model 4 - VGG16	0.924	0.867
Model 5 - VGG16 (200 Epochs)	0.944	0.867

Table 1. Accuracy Of Different Models

9. PROJECT EXPECTED TIMELINE:

Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Planning						
Selection of project						
Design of solution						
Synopsis review						
Development of Project						
Testing and Debugging						
Final Submission						

Table 2. Project Expected Timeline

10. CONCLUSION:

In this study, we have proposed a deep learning-based model to detect and classify COVID-19 cases from X-ray images. Our model is fully automated with an end-to-end structure without the need for manual feature extraction. This system can be used in remote places in countries affected by COVID-19 to overcome a shortage of radiologists. Also, such models can be used to diagnose other chest-related diseases including tuberculosis and pneumonia. A limitation of the study is the use of a limited number of COVID-19 X-ray images. We intend to make our model more robust and accurate by using more such images from our local hospitals.

- CNN with extra convolutional layers (e.g., six layers have been used in the CNN proposed in this study) performs best in COVID-19 diagnosis.
- CNN models require a sufficient number of images for efficient and more accurate image classification.
- Data augmentation techniques are very effective to improve the CNN model performance remarkably by generating more data from an existing limited-size dataset.
- Data augmentation is also effective in image classification as it gives the ability of invariance to CNNs.
- CNN-based diagnosis using X-ray imaging can be very effective for medical sector to handle the mass testing situations in pandemics like COVID-19.
- We intend to make CNN-based diagnosis using X-ray imaging more robust and accurate for medical sector to handle the mass testing situations in pandemics like COVID-19.

11.FUTURE SCOPE

CNN transfer models can be used in AI Health Diagnosis using X-ray and CT Images for early detection and preventive measures. The CNN models can used in app to make the diagnosis patient X-rays in Remote places using App analysed by the CNN. The Future X -ray and CT Scans images can be used to diagnose multiple diseases at earlier stages and at cheaper rate.

12.REFERENCES:

1.Ahammed, K.; Satu, M.S.; Abedin, M.Z.; Rahaman, M.A.; Islam, S.M.S. Early Detection of Coronavirus Cases Using Chest X-ray Images Employing Machine Learning and Deep Learning Approaches. medRxiv 2020. medRxiv 2020.06.07.20124594

2.Chowdhury, N.K.; Rahman, M.M.; Kabir, M.A. PDCOVIDNet: A parallel-dilated convolutional neural network architecture for detecting COVID-19 from chest X-ray images. Health Inf. Sci. Syst. 2020, 8, 1–14.

3. Sahlol, A.T.; Yousri, D.; Ewees, A.A.; Al-Qaness, M.A.; Damasevicius, R.; Abd Elaziz, M. COVID-19 image classification using deep features and fractional-order marine predators algorithm. Sci. Rep. 2020, 10, 1–15.

4. Che Azemin, M.Z.; Hassan, R.; Mohd Tamrin, M.I.; Md Ali, M.A. COVID-19 Deep Learning Prediction Model Using Publicly Available Radiologist-Adjudicated Chest X-Ray Images as Training Data: Preliminary Findings. Int. J. Biomed. Imaging 2020, 2020.

5. Sekeroglu, B.; Ozsahin, I. Detection of COVID-19 from Chest X-Ray Images Using Convolutional Neural Networks. SLAS Technol. Transl. Life Sci. Innov. 2020, 25, 553–565.

6. El-Rashidy, N.; El-Sappagh, S.; Islam, S.M.R.; El-Bakry, H.M.; Abdelrazek, S. End-To-End Deep Learning Framework for Coronavirus (COVID-19) Detection and Monitoring. Electronics 2020, 9, 1439.

7. Minaee, S.; Kafieh, R.; Sonka, M.; Yazdani, S.; Soufi, G. Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning. Med. Image Anal. 2020, 65, 101794.

8. Benbrahim, H.; Hachimi, H.; Amine, A. Deep transfer learning with apache spark to detect COVID-19 in chest X-ray images. Rom. J. Inform. Sci. Technol. 2020, 23, S117–S129.

9. Cohen, J.P.; Hashir, M.; Brooks, R.; Bertrand, H. On the limits of cross-domain generalization in automated X-ray prediction. In Medical Imaging with Deep Learning.

10. Chen, M.; Ludwig, S.A. Color image segmentation using fuzzy C-regression model. Adv. Fuzzy Syst. 2017, 2017.

11.Krizhevsky A., Sutskever I., Hinton G.E. *Advances in Neural Information Processing Systems.* 2012. Imagenet classification with deep convolutional neural networks; pp. 1097–1105.

12. GUIDE RECOMMENDATIONS:

13.REVIWER RECOMMENDATIONS:

13. REVIEWER RECOMMENDATION: