



University of Asia Pacific

Department of Electrical and Electronic Engineering

Performance analysis of different Deep Learning architecture using COVID X-ray images

A thesis submitted in partial fulfilment of the requirements for the degree of
Bachelor of Science in Electrical and Electronic Engineering

By

Full Name

Registration No

Tanjina Tasnim Asha

18108094

MD. Rifat Jahan

18108118

Saiful Islam

18108119

SUPERVISED BY

A.H.M Zadidul Karim

(Associate Professor)

Department of Electrical and Electronic Engineering

University of Asia Pacific

74/A, Green Road, Farmgate, Dhaka-1205, Bangladesh

June 2022.

DECLARATION

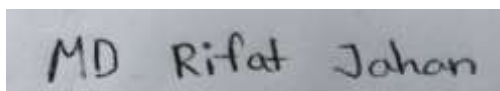
We do hereby declare that the research work presented in this undergraduate thesis has not been carried out by us and has not been previously submitted to any other University / Institution / Organization for an academic qualification / certificate / diploma or Degree.



.....

Tanjina Tasnim Asha

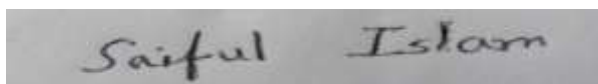
Registration No: 18108094



.....

MD. Rifat Jahan

Registration No: 18108118



.....

Saiful Islam

Registration No: 18108119

ACCEPTANCE

The undersigned certifies that he has read, recommended for acceptance, a thesis entitled” **Performance analysis of different deep learning architecture using COVID X-ray images**” in the partial fulfilment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering.

.....

A.H.M Zadidul Karim

Associate Professor

Department of Electrical and Electronic Engineering

University of Asia Pacific

ACKNOWLEDGEMENTS

All acclaims go to the all-powerful Allah for the fruitful completion of this proposal and satisfaction of our dream of reality.

Words cannot express our gratitude to our honorable supervisor A. H. M Zadidul Karim, Associate Professor, Department of Electrical and Electronic Engineering, the University of Asia Pacific for his invaluable patience and feedback. We could not have undertaken this journey without his generously provided knowledge and expertise.

We would be delinquent in not saying our family, particularly our guardians. Their conviction in us has kept our spirits and inspiration tall amid this preparation.

Tanjina Tasnim Asha
Reg. No.:18108094

MD. Rifat Jahan
Reg. No.:18108118

Saiful Islam
Reg. No.:18108119

ABSTRACT

The novel coronavirus infection 2019 (COVID-19) is an infectious disease that has caused thousands of passings and contaminated millions of worlds. In this way, different innovations that permit for the quick location of COVID-19 contaminations with tall exactness can offer healthcare experts much-needed assistance. This consideration is pointed at assessing the adequacy of the state-of-the-art pre-trained Convolutional Neural Systems (CNNs) on the programmed conclusion of COVID-19 from chest X-rays (CXRs). The dataset utilized within the tests comprises 240 CXR pictures from people with COVID-19 and 120 CXR pictures from healthy individuals. In this paper, the adequacy of artificial intelligence (AI) within the fast and exact recognizable proof of COVID-19 from CXR pictures has been investigated based on different pre-trained deep learning calculations and fine-tuned to boost location precision to recognize the most excellent calculations. The results showed that deep learning with X-ray imaging is valuable in collecting basic organic markers related to COVID-19 contaminations. VGG19 got the most elevated exactness of 98.28%. Be that as it may, VGG19 outflanked all other models in COVID-19 discovery with an exactness, F1 score, accuracy, specificity, and sensitivity of 98%, 97%, 96%, 100% separately. The exceptional execution of these pre-trained models can make strides in the speed and exactness of COVID-19 determination. Be that as it may, a bigger dataset of COVID-19 X-ray pictures is required for a more exact and solid distinguishing proof of COVID-19 contaminations when utilizing profound exchange learning. This would be greatly advantageous in this widespread when the malady burden and the requirements for preventive measures are in a struggle with the now accessible assets.

Table of Contents

DECLARATION	ii
ACCEPTANCE	iii
ACKNOWLEDGEMENTS	iv
ABSTRACT.....	v
Table of Contents.....	vi
List of Figures	viii
List of Tables	xiv
ABBREVIATION.....	xv
CHAPTER 1	1
INTRODUCTION	1
1.1 Short Overview	2
1.2 Motivation	4
1.3 Objectives.....	5
1.4 Organization of the Dissertation	5
CHAPTER 2	7
COVID-19 and Different architecture	7
2.1 History.....	7
2.2 Coronavirus outbreaks in Bangladesh.....	8
2.3 Reason Behind choosing Deep Learning to detect COVID-19.....	9
2.4 Convolutional Neural Networks.....	11
2.5 Different CNN architecture	12
2.5.1 AlexNet	12
2.5.2 VGG16.....	13
2.5.3 VGG19.....	15
2.5.4 GoogleNet	17
2.5.5 ResNet50.....	18
2.5.6 Inception v3	19
2.5.7 DenseNet.....	20
CHAPTER-3	23
Literature Review.....	23
3.1 Introduction	23
3.2 Literature review	24
CHAPTER-4.....	27

Methodology	27
4.1 Introduction	27
4.2 DATASET	27
4.3 Feature Extraction Steps.....	29
4.3.1 AlexNet	29
4.3.2 VGG19	32
4.3.3 ResNet50.....	36
4.3.4 Inception v3	40
4.3.5 GoogleNet	44
4.3.6 VGG16.....	47
CHAPTER 5	53
Result and Discussion	53
5.1 VGG19	53
5.2 VGG16 model	57
5.3 ResNet50	60
5.4 Inception3.....	64
5.5 GoogleNet	68
5.6 AlexNet	71
5.7 DenseNet	74
5.8 Accuracy, Sensitivity and Specificity in all our proposed model	77
5.8.1 Final result	78
Chapter-6.....	80
Conclusion and future work.....	80
6.1 Recommendation and Future work	80
6.2 Advantages of Deep Learning	80
6.3 Limitation of Deep Learning.....	81
Reference	83

List of Figures

Figure-1	Basic architecture of CNN	1
Figure1.1(a, b, c, d):	COVID-19 infected chest X-ray images	3
Figure-2.1:	This graph is a visual representation of how quickly the virus can spread from person to person, country to country.	7
Figure-2.2:	COVID-19 outbreaks in Bangladesh.	8
Figure-2.5.1:	Structure of AlexNet	13
Figure-2.5.2(a):	VGG-16 architecture	14
Figure-2.5.2(b):	VGG-16 architecture map	15
Figure-2.5.3:	VGG19 architecture	16
Figure-2.5.4:	GoogleNet architecture	17
Figure-2.5.5:	ResNet50 architecture	18
Figure-2.5.6(a):	Architecture of Inception v3	19
Figure-2.5.6(b):	Working principle of Inception v3	20
Figure-2.5.7(a):	DenseNet structure	21
Figure-2.5.7(b):	A schematic representation of DenseNet	21
Figure-4.1:	Samples from the dataset used in this study. (a) X-ray of a patient with COVID-19 from, Dataset A; (b) X-ray of a healthy patient from Dataset A;(c)X-ray of a patient with COVID-19 from Dataset B; (d) X-ray of a	28

healthy patient from Dataset B. We choose the images randomly from our dataset.

Figure-4.3.1 (a):	Flowchart for our model	30
Figure-4.3.1(b):	Number of epochs, loss and accuracy of AlexNet model	31
Figure-4.3.2(a):	Flowchart of our proposed VGG19 model	34
Figure-4.3.2(b):	Number of epochs, accuracy, losses, validation accuracy and validation losses for our VGG19 model	35
Figure-4.3.3(a):	Flowchart of our proposed ResNet50 model	37
Figure-4.3.3(b):	How we did our residual mapping	38
Figure-4.3.3(c):	Number of epochs, accuracy, losses, validation accuracy and validation losses for our ResNet50 model	39
Figure-4.3.4(a):	Efficiently reduced grid size	41
Figure-4.3.4(b):	Number of epochs, accuracy, losses, validation accuracy and validation losses for our Inception v3 model	41
Figure-4.3.4(c):	Flowchart of our proposed Inception v3 model	43
Figure4.3.5(a):	Layer by Layer architectural details of GoogleNet	44
Figure-4.3.5(b):	Flowchart of our proposed GoogleNet model	45
Figure-4.3.5(c):	Number of epochs, accuracy, losses, validation accuracy and validation losses for our GoogleNet model	46

Figure-4.3.6(a):	Flowchart of our proposed VGG16 model	48
Figure-4.3.6(b):	Number of epochs, accuracy, losses, validation accuracy and validation losses for our ResNet50 model	49
Figure-4.3.7(a):	DenseNet Architecture	50
Figure-4.3.7(b)	Flowchart of DenseNet model	51
Figure-4.3.7(c):	Number of epochs, accuracy, losses, validation accuracy and validation losses for our DenseNet model	52
Figure-5.1 (a):	Here we can see the difference between train loss and validation loss for our given dataset.	53
Figure-5.1(b):	Here we can see the difference between train accuracy and validation accuracy for our given dataset.	54
Figure-5.1(c):	Here we can see the ROC curve for VGG19 model where x axis denotes false positive rate and y axis denote true positive rate.	54
Figure-5.1(d):	Confusion Matrix with Normalized Values	55
Figure-5.1(e):	Confusion Matrix without Normalization	56
Figure-5.1(f):	Class activation report for VGG19	56
Figure-5.2 (a):	Here we can see the difference between train loss and validation loss for our given dataset.	57
Figure-5.2 (b):	Here we can see the difference between train accuracy and validation accuracy for our given dataset.	57

Figure-5.2(c):	Here we can see the ROC curve for VGG16 model where x axis denotes false positive rate and y axis denote true positive rate.	58
Figure-5.2(d):	Confusion Matrix with Normalized Values	59
Figure-5.2(e):	Class activation report for VGG16	59
Figure-5.3 (a):	Here we can see the difference between train loss and validation loss for our given dataset.	60
Figure-5.3 (b):	Here we can see the difference between train accuracy and validation accuracy for our given dataset.	60
Figure-5.3(c):	Here we can see the ROC curve for ResNet50 model where x axis denotes false positive rate and y axis denote true positive rate.	61
Figure-5.3(d):	Confusion Matrix without Normalization	62
Figure-5.3(e):	Confusion Matrix with Normalized Values	62
Figure-5.3(f)	Class activation report for ResNet50	63
Figure-5.4 (a):	Here we can see the difference between train loss and validation loss for our given dataset.	64
Figure-5.4 (b):	Here we can see the difference between train accuracy and validation accuracy for our given dataset.	64
Figure-5.4(c):	Here we can see the ROC curve for Inception v3 model where x axis denotes false positive rate and y axis denote true positive rate.	65

Figure-5.4(d):	Confusion Matrix without Normalization	66
Figure-5.4(e):	Confusion Matrix with Normalized Values	66
Figure-5.4(f):	Class activation report for Inception3	67
Figure-5.5 (a):	Here we can see the difference between train accuracy and validation accuracy for our given dataset.	68
Figure-5.5 (b):	Here we can see the difference between train loss and validation loss for our given dataset.	68
Figure-5.5(c):	Here we can see the ROC curve for Inception v3 model where x axis denotes false positive rate and y axis denote true positive rate.	69
Figure-5.5(d):	Confusion Matrix without Normalization	70
Figure-5.5(e):	Classification report for DenseNet	70
Figure-5.6 (a):	Here we can see the difference between train accuracy and validation accuracy for our given dataset.	71
Figure-5.6 (b):	Here we can see the difference between train loss and validation loss for our given dataset.	71
Figure-5.6(c):	Here we can see the ROC curve for Inception v3 model where x axis denotes false positive rate and y axis denote true positive rate.	72
Figure-5.6(d):	Confusion Matrix without Normalization	73
Figure-5.6(e):	Classification report for AlexNet	73
Figure-5.7 (a):	Here we can see the difference between train loss and validation loss for Our given dataset.	74

Figure-5.7 (b):	Here we can see the difference between train accuracy and validation accuracy for our given dataset.	74
Figure-5.7(c):	Here we can see the ROC curve for DenseNet model where x axis denotes false positive rate and y axis denote true positive rate.	75
Figure-5.7(d):	Confusion Matrix without Normalization	76
Figure-5.7(e):	Visual representation of classification report of DenseNet	76
Figure-5.8:	Visual representation of accuracy, sensitivity and specificity of different architecture	77
Figure-5.9(a):	Accuracy curve of different architecture.	78
Figure-5.9(b):	Validation accuracy curve of different architecture.	78
Figure-5.9(c):	Loss curve of different architecture.	79
Figure-5.9(d):	Validation Loss curve of different architecture	79

List of Tables

Figure4.1:	Our chosen dataset	26
Table-5.1:	Classification report for our VGG19 model	55
Table-5.2:	Classification report for our VGG19 model	59
Table-5.3:	Classification report for our ResNet50 model	63
Table-5.4:	Classification report for our Inception v3 model	67
Table-5.5:	Classification report for our GoogleNet model	71
Table-5.6:	Classification report for our AlexNet model	75
Table-5.7:	Classification report for our DenseNet model	79

ABBREVIATION

- i. **CNN** = Convolutional Neural Network
- ii. **DL** = Deep Learning
- iii. **ML** = Machine Learning
- iv. **COVID** = Corona Virus Disease
- v. **SARS** = Severe acute respiratory syndrome
- vi. **ANN** = Artificial Neural Network
- vii. **VGG16** = Visual Geometry Group 16
- viii. **VGG19** = Visual Geometry Group 19
- ix. **ResNet** = Residual neural network
- x. **ROC** = Region over curve
- xi. **RGB** = Red Green Blue
- xii. **CT** = Computed Tomography
- xiii. **ReLU** = rectified linear activation function
- xiv. **RT-PCR** = Reverse transcription-polymerase chain reaction

CHAPTER 1

INTRODUCTION

A CNN architecture is formed by a stack of distinct layers that transform the input volume into an output volume (e.g. holding the class scores) through a differentiable function. A few distinct types of layers are commonly used [1]. It is used in image recognition and processing which is designed to process pixel data.

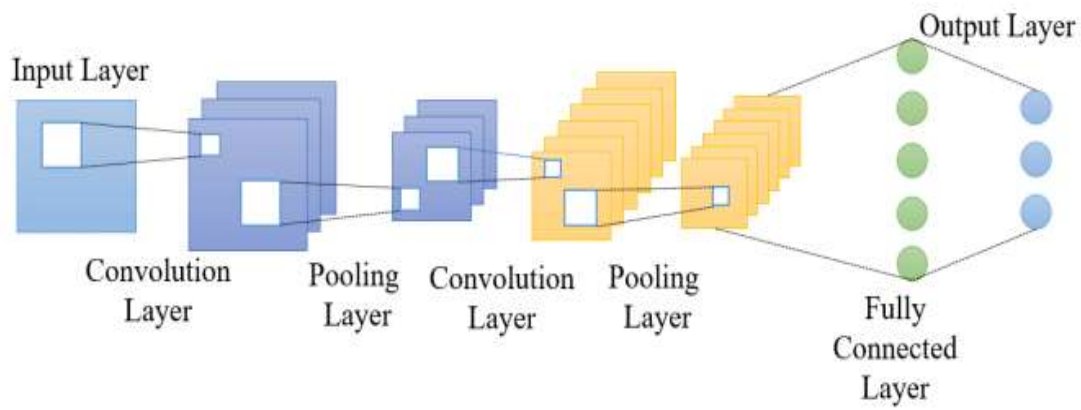


Figure-1: Basic architecture of CNN

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of artificial neural network (ANN), most commonly applied to analyze visual imagery [2]. Its contribution to the field of computer vision and image analysis is undeniable. CNNs are a class of Deep Neural Systems that can recognize and classify specific highlights from pictures and are broadly utilized for analyzing visual pictures. Their applications run from picture and video recognition, picture classification, restorative picture examination, computer vision and characteristic dialect processing. The term ‘Convolution’ in CNN denotes the mathematical function of convolution which is a special kind of linear operation wherein two functions are multiplied to produce a third function which expresses how the shape of one function is modified by the other. In simple terms, two images which can be represented as matrices are multiplied to give an output that is used to extract features from the image [3].

1.1 Short Overview

Testing of all individuals for SARS-CoV-2, counting those who have no side effects, those who appear side effects of infection such as inconvenience breathing, fever, sore throat or misfortune of the sense of scent and taste, and who may have been uncovered to the infection will offer assistance anticipate the spread of COVID-19 by recognizing individuals who are in require of care in a convenient design. A positive test early within the course of the ailment empowers people to confine themselves – decreasing the chances that they will taint others and permitting them to look for treatment prior, likely decreasing malady seriousness and the hazard of long-term inability, or passing. It is spreading through close contact or the droplets when the infected coughs, sneezes or talks [4]. For a vaccine to develop for COVID-19, diagnosis through real-time reverse transcription-polymerase chain reaction (RT-PCR) is confirmed [5]. RT-PCR needs time management. Because of this, it may emerge that the contaminated individual may not be recognized and may not get the appropriate treatment and may spread the infection to a sound populace and cannot be worthy in this widespread circumstance. X-rays and CT pictures are sensitive to the screening of COVID-19 among patients[6]. At first, a lung X-ray is performed for suspected or affirmed patients through particular circuits. It demonstrates to be a separating component wherein the patient's advance conclusion depends on the clinical circumstance and the chest X-ray film that comes about. Many recent works depending on the elaborated image features with clinical and diagnostic results may help in the early detection of COVID-19 [7–9]. Technological advancement using machine and deep learning applications for automatic prediction of COVID-19 traces in people has taken over people's minds [10]. A work on X-ray image classification is introduced by Tulin et al. [11]. They used the DarkNet model for classifying the detected YOLO-based object in the input images. Another work on X-ray images called Decompose, Transfer, and Compose (DeTraC), is introduced by Asmaa et al. in [12]. DeTraC used the class decomposition mechanism to deal with class boundaries leading to an accuracy of 0.931 and sensitivity of 1. A work by Taban et al. on chest X-ray images used the 12-off-the-shelf CNN architectures in transfer learning [13]. GANs have also been used in the process of COVID-19 disease analysis by Mohamed et al. [14]. It used deep transfer learning based on GANs for 3-class classification among COVID-19, Normal

and Pneumonia X-ray images. They utilized the dataset created by Joseph et al. [15] Yifan et al. introduced the use of CNN in the classification process of COVID-19 [16].

Some COVID-19 positive X-ray images are given below:



(a)



(b)



(c)



(d)

Figure-1.1(a,b,c,d): COVID-19 infected chest X-ray images

1.2 Motivation

The inspiration behind this study was to set up a distant better; a much better, a higher, a stronger and an improved test for coronavirus disease. The RT-PCR test is utilized to analyze the coronavirus habitually and returned a negative result for an infected person. Moreover, this test remains restrictively costly for most citizens, and not everybody may manage it due to monetary hardship. A proficient imaging approach is developed for the assessment of lung conditions, which has been done by analyzing the chest X-ray or chest CT of an infected individual. Deep Learning is the well-suited subspace of Artificial Intelligence [AI] innovation, which offers supportive examination to consider more chest X-ray pictures that can essentially have an impact on coronavirus screening. The objective of this research is to cluster the X-ray pictures displayed within the dataset into COVID-19 positive and COVID19 negative by making utilize of the manufactured neural systems. The preparing dataset was fine-tuned with seven already prepared convolutional neural networks. The appraisal of the models on a test appears that AlexNet, VGG16, VGG19, GoogleNet, Inception v3, ResNet50,DenseNet as the best performing models.

1.3 Objectives

The purpose of this thesis is to **Performance analysis of different Deep learning architecture using COVID X-ray images**. The objectives are

1. To know about CNN architectures.
2. To know about different models of CNN architectures.
3. To know about image processing in CNN.
4. To learn about deep learning.
5. To learn how to use python in Google Colab.
6. To learn about feature engineering
7. To learn how to use a multi-model machine learning approach.
8. Identifying the features and training and then testing the framework with the features and different datasets.
9. To analyze the loss rate for different machine learning algorithms with a different set of Features.
10. Finally find out the highest accuracy rate, the F1 factor.

1.4 Organization of the Dissertation

The thesis consists of 6 chapters. A short explanation is introduced here:

Chapter-1 Introduction

Chapter 1 is an introductory chapter. It contains a short overview of the related work, our motivation for work and objectives.

Chapter-2 COVID19

Chapter 2 contains a discussion of COVID19 where its history, its outbreak in the world, the outbreak in Bangladesh, its present situation, and its effects in our daily life all is described.

Chapter-3 Literature review

Chapter 3 is a discussion about previous and recent work on deep learning and machine learning on COVID19.

Chapter-4 Methodology

Chapter 5 contains a block diagram and framework of our proposed algorithms, a description of every block, how we use the multi-model machine learning approach, which machine learning algorithms we used etc.

Chapter-5 Results and Discussion

Chapter 5 shows the results and calculations of the models shown in chapter 5. We also provide here a discussion about the accuracy and loss rate of our methodology.

Chapter-6 Conclusion and future work

Chapter 6 contains the conclusion and direction of future work.

CHAPTER 2

COVID-19 and Different architecture

2.1 History

The infection first showed up on a little scale in November 2019 with the primary huge cluster showing up in Wuhan, China in December 2019. It was, first thought that SARS-CoV-2 made the bounced to people at one of Wuhan, China’s open-air “wet markets. “Though the original source of viral transmission to humans remains unclear that it may have begun as a natural weapon in a lab in China, but there is no proof. Most people contaminated with the infection will experience mild to direct respiratory sickness and recover without requiring special treatment. Be that as it may, a few will end up seriously sick and require therapeutic consideration. More seasoned people and those with basic medical conditions like cardiovascular infection, diabetes, constant respiratory infection, or cancer are more likely to create serious sickness. Anybody can get wiped out with COVID-19 and get to be seriously sick or die at any age.

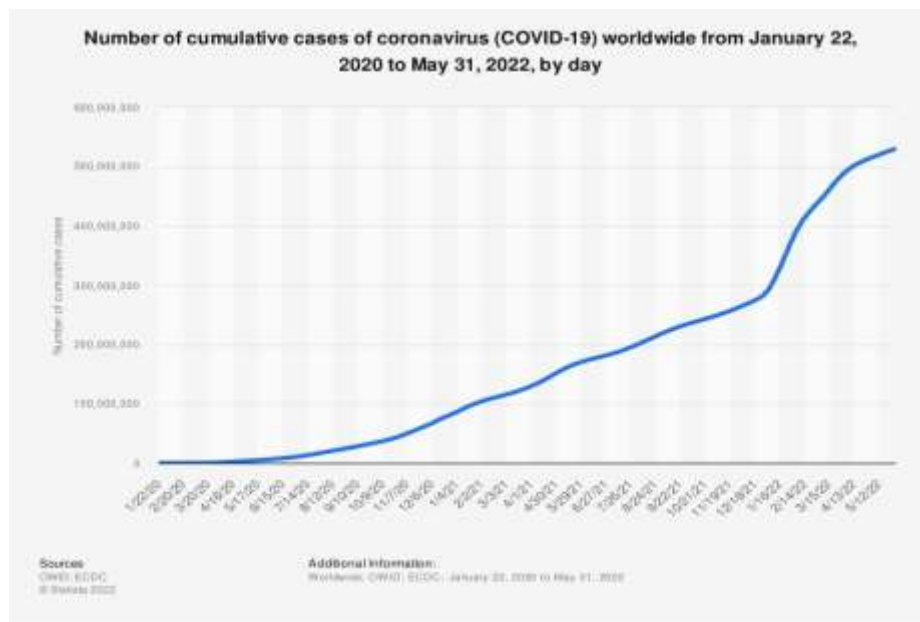


Figure 2.1: This graph is a visual representation of how quickly the virus can spread from person to person, country to country.

The disease was officially named coronavirus disease 2019 (COVID-19) by 11 February 2020[18], and was declared a pandemic on 11 March 2020[19]. The outbreak of novel coronavirus-2 (nCoV-2) pronounced a widespread and worldwide open well-being crisis by the World Wellbeing Organization (WHO), and the whole world is working to address it. It may be a quickly advancing and developing circumstance.

2.2 Coronavirus outbreaks in Bangladesh

The virus was confirmed to have spread to Bangladesh in March 2020. The first three known cases were reported on 8 March 2020 by the country's epidemiology institute, IEDCR. Since then, the pandemic has spread day by day over the whole nation and the number of affected people has been increasing. Bangladesh is the second most affected country in South Asia, after India.[17]As COVID-19 is mainly transmitted from person to person by inhaling the contaminated air which already bears the droplets /aerosols and small airborne particles containing the virus, the government of Bangladesh declared a lockdown which includes travel bans, further offline office activities and most critically, social separation on 23 March 2020 in order to save the lives of its citizens. But in the first week of April Bangladesh experienced the COVID-19 outbreaks.

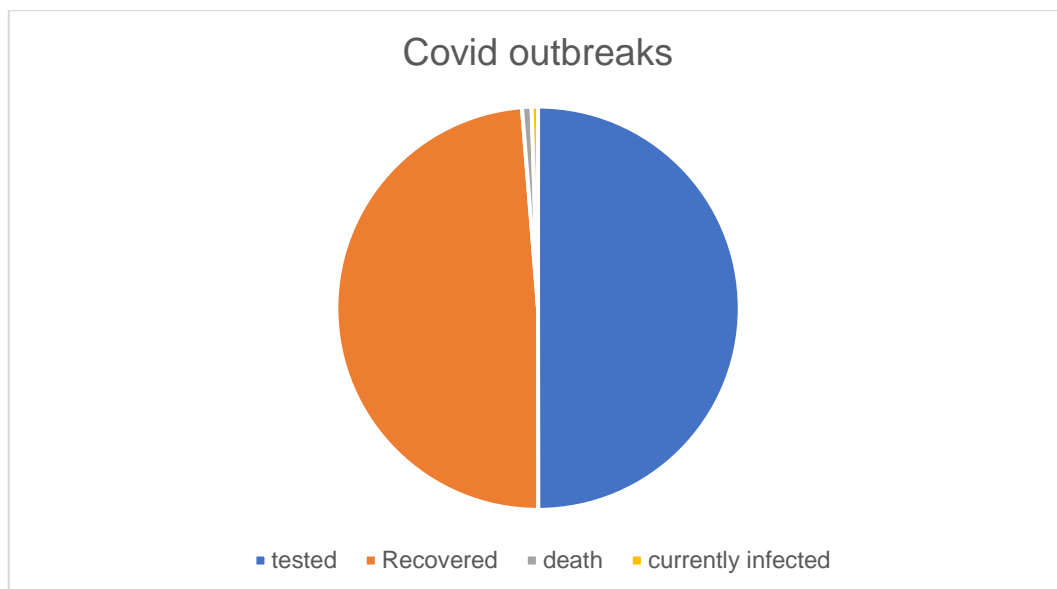


Figure 2.2: COVID-19 outbreaks in Bangladesh.

Human civilization has been attacked over and over by infectious disease outbreaks, such as plagues and pandemics; as a result, mankind has had to adjust, alter courses of life, create

adapting techniques and for a few illnesses found cures—eradicating those outbreaks. These pandemics are not as it were dangerous to physical well-being, but challenge the mental well-being of all people in any case of statistical characteristics or whether they were affected.

For a densely populated and financially poor country, social distancing is troublesome and it isn't possible for the people of Bangladesh to remain home for a long time without earning their livelihood. Before curing we have to detect COVID-19 which is very costly for the people of Bangladesh.

2.3 Reason Behind choosing Deep Learning to detect COVID-19

COVID-19 spread at a breakneck speed, so we ought to be able to identify the virus as long as possible. all the possible covid discovery methods are time-consuming. This infection is exceptionally contagious and seems to spread among masses before the test result might be found. In some cases, the test comes about is off-base because it appears the inverse result. On the off chance that a non-covid infected person gets the result as positive, it doesn't matter since it won't hurt anybody. But on the off chance that an infected person gets the resulting negative at that point, they will begin meandering around openly which would influence numerous people. But on the off chance that we attempt to detect covid-19 through CNN architecture by image processing not as it did, we get a precise result but moreover we know almost the false-negative and false-positive comes about. CNN architecture works based on comparing both COVID-19 infected and normal X-ray images.

PCR with turnaround translation (RT–PCR) is the test of choice for diagnosing COVID-19, imaging can complement its utilization to attain more noteworthy demonstrative certainty or indeed be a surrogate in some nations where RT–PCR isn't promptly accessible. In a few cases, chest X-ray anomalies are obvious in patients who at first had a negative RT–PCR test and a few studies have appeared that chest computed tomography (CT) includes a higher affectability for COVID-19 than RT–PCR, and may be considered an essential device for

diagnosis. In reaction to the widespread, analysts have surged to create models utilizing manufactured insights (AI), in specific machine learning, to back clinicians. Machine learning methods offer an extraordinary promise for quick and exact detection and guess of coronavirus infection 2019 (COVID-19) from standard-of-care chest radiographs (CXR) and from CT scans of the chest. Applying machine learning strategies to COVID-19 radiological imaging for progressing the exactness of conclusion, compared with the gold-standard RT-PCR, while giving profitable knowledge for the forecast of understanding results. These models have the potential to misuse the huge sum of multimodal information collected from patients and seem, on the off chance that fruitful, change discovery, determination and triage of patients with suspected COVID-19. Of most noteworthy potential utility could be a show that cannot as it were to recognize patients with COVID-19 from patients without COVID-19 but to observe elective sorts of pneumonia such as those of bacterial or other viral aetiologies. With no standardization, AI algorithms for COVID-19 have been created with an awfully wide run of applications, information collection methods and execution appraisal measurements. Maybe, as a result, none are right now prepared to be sent clinically. Like any test, RT-PCR gives false negatives come about that can be corrected by clinicians by going up against clinical, organic and imaging information. The combination of RT-PCR and chest-CT seem to move forward conclusion execution, but this would require impressive assets for its quick utilization in all patients with suspected COVID-19. The potential commitment of machine learning in this circumstance has not been completely assessed.

Moreover, deep learning has appeared to play a critical part in recognizing viral and bacterial pneumonia and diagnosing the foremost common thoracic infections. Besides, the challenge is to create a calculation competent for recognizing an understanding of COVID-19. By the by, this errand remains challenging as COVID-19 can share comparable radiographic highlights with other sorts of pneumonia. In [20], the authors mentioned the poor performance of MobileNet in distinguishing cases of COVID-19 from other pneumonia cases when the training dataset included only bacterial pneumonia cases. We hence endeavor to recognize COVID-19 from viral pneumonia (not bacterial pneumonia) by pointing to quickly distinguish clusters of COVID-19 caused by a novel infection. Besides, the COVID-19 versus non-COVID classification may be an extreme lopsidedness issue with respect to the number of

COVID-19 versus non-COVID-19 tests due to the trouble of getting a satisfactory number of positive COVID-19 tests.

Testing of individuals who have been in contact with others who have a reported disease is additionally critical. A negative test doesn't mean anyone within the clear; we'll end up irresistible afterwards. In this manner, indeed in the event that anyone tests negative, he/she would like to proceed to secure himself/herself and others by washing his/her hands regularly, physically distancing, and wearing a confront mask. A positive test makes it clear merely ought to confine himself, which others with whom they've got been in contact since the time of their presentation ought to too get tested. Since it is recognized that about half of all SARS-CoV-2 diseases are transmitted by individuals who are not appearing any indications, distinguishing tainted people who are presymptomatic, as well as those who are asymptomatic, will play a major part in ceasing the widespread.

2.4 Convolutional Neural Networks

An input layer, hidden layers, and an output layer make up a convolutional neural network. Any middle layers in a feed-forward neural network are referred to as hidden since the activation function and final convolution mask their inputs and outputs. The hidden layers of a convolutional neural network include convolutional layers. This usually comprises a layer that does a dot product of the convolution kernel with the input matrix of the layer. The activation function of this product is usually ReLU, and it is usually the Frobenius inner product. The convolution procedure generates a feature map as the convolution kernel slides along the input matrix for the layer, which then contributes to the input of the following layer. Then there are further levels, such as pooling layers, fully connected layers, and normalization layers.

The CNN is regarded as suitable for transfer learning in the event that it is found to be able to extricate the most important picture highlights. Then, in transfer learning, CNN is utilized to analyze a new dataset of a different nature and extricate its features according to the information procured from the, beginning with preparation. One

common procedure to misuse the capabilities of the pre-trained CNN is called feature extraction by means of transfer learning. This approach implies that the CNN will hold its architecture and weights between its layers; hence, the CNN is used only as a feature extractor. The features are afterwards utilized in a second network/classifier that will handle its classification. The exchange learning approach is generally utilized to work around computational costs of preparing a organize from scratch or to keep the include extractor prepared amid the primary task. This performance is linked to various extraction parameters that are typically not allowed as they cause overfitting of the network [31]. That said, feature extraction performed with transfer learning allows a large number of features to be extracted by generalizing the problem and avoiding excessive adjustments [32]. The use of transfer learning also allows the use of the internet of things (IoT) systems to classify medical images. For example, Dourado Jr et al. [33] proposed an IoT system to detect a stroke in CT images. Rodrigues et al. [34] used the system proposed by Dourado Jr et al. [33] to classify EEG signals.

2.5 Different CNN architecture

2.5.1 AlexNet

AlexNet is the name of a convolutional neural network (CNN) architecture, designed by Alex Krizhevsky in collaboration with Ilya Sutskever and Geoffrey Hinton, who was Krizhevsky's PhD advisor.[35] AlexNet permits multi-GPU preparation by putting half of the model's neurons on one GPU and the other half on another GPU. Not as it were does this mean that a greater demonstration can be prepared, but it moreover cuts down on the preparation time. It has 8 layers with learnable parameters. The input to the Demonstrate is RGB images. It has 5 convolution layers with a combination of max-pooling layers. Then it has 3 completely connected layers. The activation function used in all layers is Relu. It utilized two Dropout layers. The actuation work utilized within the output layer is Softmax. They add up to the number of parameters in this engineering is 62.3 million.

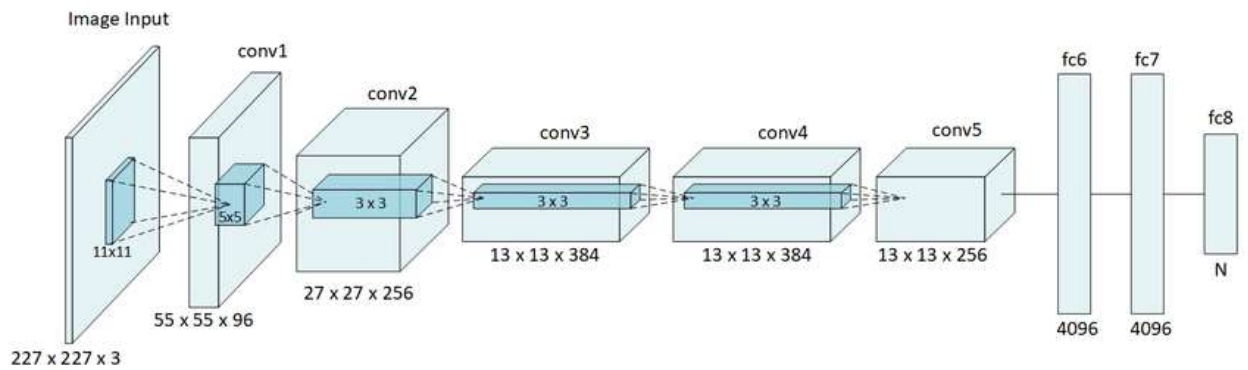


Figure 2.5.1: Structure of AlexNet .

AlexNet is considered one of the foremost persuasive papers distributed in computer vision, having impelled numerous more papers distributed utilizing CNNs and GPUs to quicken profound learning.

2.5.2 VGG16

VGG16 demonstrated to be a noteworthy turning point within the quest of mankind to form computers “see” the world. A part of exertion has been put into progressing this capacity beneath the teaching of Computer Vision (CV) for a number of decades. VGG16 is one of the noteworthy advancements that cleared the way for several developments that have taken after in this field. It may be a Convolutional Neural Organize (CNN) show proposed by Karen Simonyan and Andrew Zisserman at the College of Oxford. The thought of demonstrating was proposed in 2013, but the real show was submitted amid the ILSVRC ImageNet Challenge in 2014. The ImageNet Huge Scale Visual Acknowledgment Challenge (ILSVRC) was a yearly competition that evaluated calculations for picture classification (and question location) at an expansive scale. They did well within the challenge but couldn’t win.

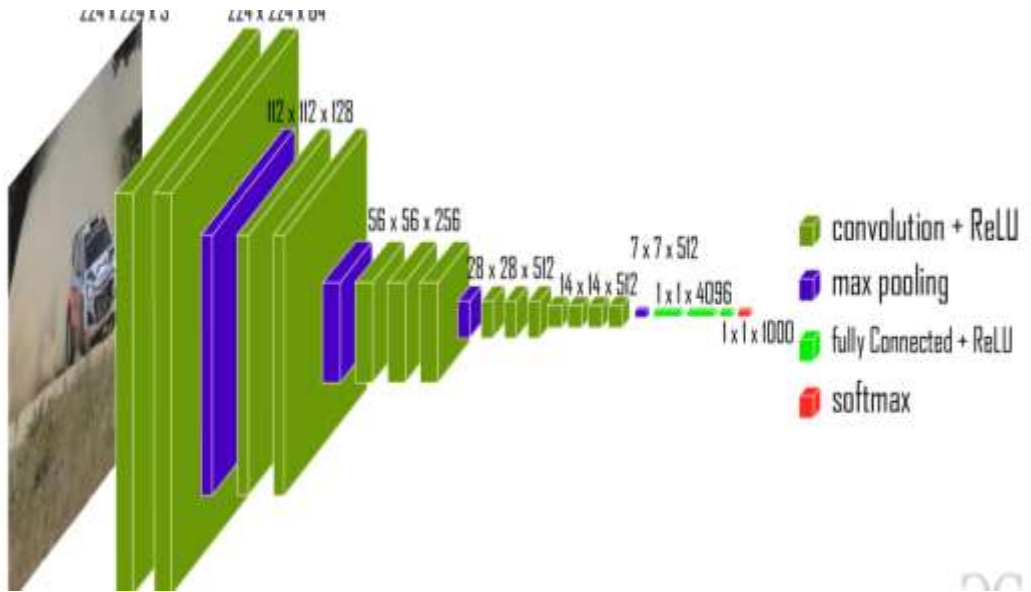


Figure 2.5.2(a) : VGG16 architecture

VGG16 was recognized to be the finest performing show on the ImageNet dataset. Image is passed through the primary stack of 2 convolution layers of the exceptionally little open measure of 3 x 3, taken after by ReLU enactments. Each of these two layers contains 64 channels. The convolution walk is settled at 1 pixel, and the cushioning is 1 pixel. This arrangement the spatial determination, and the estimate of the yield actuation outline is the same as the input picture measurements. The actuation maps are at that point passed through spatial max pooling over a 2 x 2-pixel window, with a walk of 2 pixels. This parts the estimate of the actions. In this way, the estimate of the activations at the conclusion of the primary stack is 112 x 112 x 64.

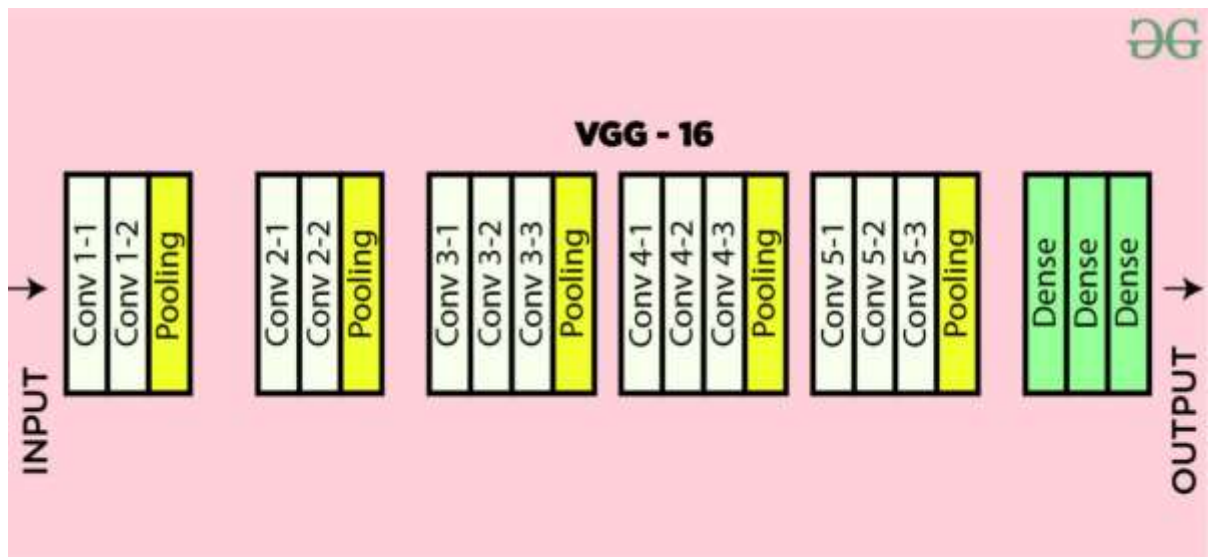


Figure-2.5.2(b): VGG-16 architecture map

The activations at that point stream through a comparative moment stack, but with 128 channels as against 64 within the to begin with one. Thus, the measure after the moment stack gets to be $56 \times 56 \times 128$. This can be taken after by the third stack with three convolutional layers and a max pool layer. The no. of channels connected here is 256, making the output size of the stack $28 \times 28 \times 256$. This can be taken after by two stacks of three convolutional layers, with each containing 512 channels. The yield at the conclusion of both these stacks will be $7 \times 7 \times 512$. The stacks of convolutional layers are taken after by three completely associated layers with a straightening layer in-between. The primary two have 4,096 neurons each, and the final completely associated layer serves as the yield layer and has 1,000 neurons compared to the 1,000 conceivable classes for the ImageNet dataset. The yield layer is taken after by the Softmax actuation layer utilized for categorical classification.

2.5.3 VGG19

VGG19 could be a variation of the VGG show which in brief comprises 19 layers (16 convolution layers, 3 Completely associated layers, 5 MaxPool layers and 1 SoftMax layer).

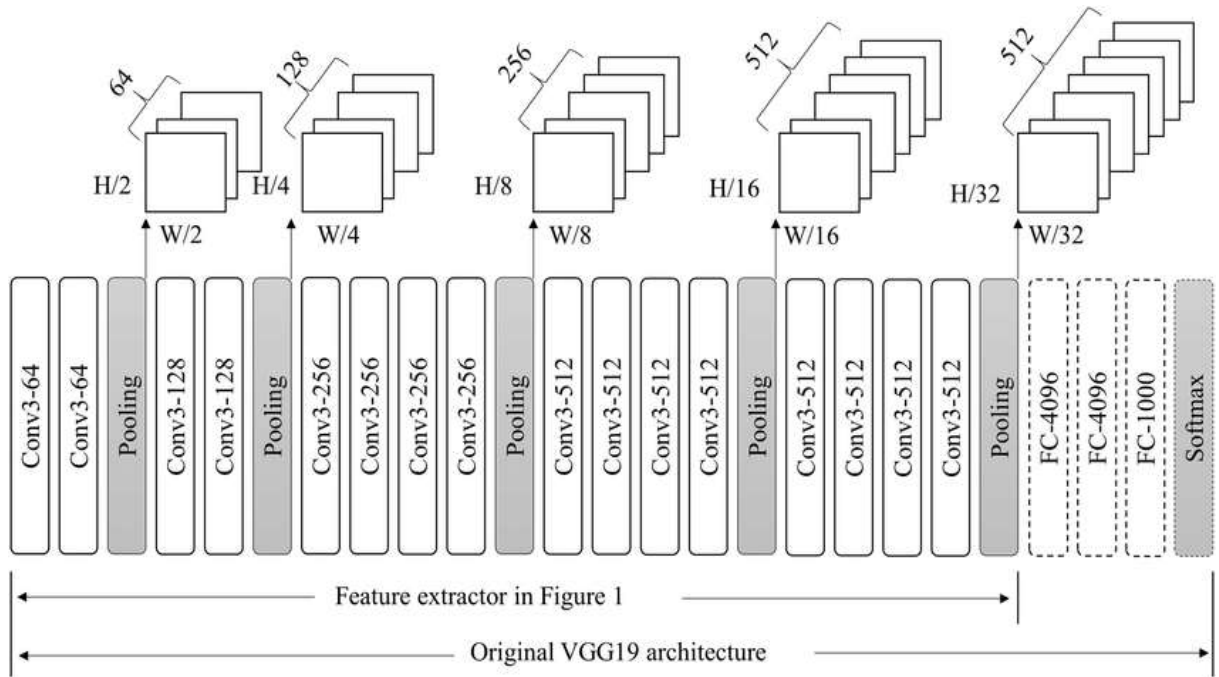


Figure 2.5.3: VGG19 architecture

Fixed estimate of (224 * 224) RGB picture was given as input to this organization which suggests that the network was of shape (224,224,3). As it was preprocessing that was done is that they subtracted the cruel RGB esteem from each pixel, computed over the complete preparing set. Used parts of (3 * 3) estimate with a walk measure of 1 pixel, this empowered them to cover the total idea of the image. spatial cushioning was utilized to protect the spatial determination of the image. max-pooling was performed over 2 * 2-pixel windows with side 2. this was taken after by Corrected straight unit(ReLU) to present non-linearity to form the show classify superior and to progress computational time as the past models utilized tanh or sigmoid capacities this demonstrated much superior to those. implemented three fully connected layers from which the first two were of measure 4096 and after that, a layer with 1000 channels for 1000-way ILSVRC classification and the ultimate layer may be a SoftMax function.

2.5.4 GoogleNet

Google Net (or Inception V1) was proposed by research at Google (with the collaboration of various universities) in 2014 in the research paper titled “Going Deeper with Convolutions”.[36] The GoogleNet architecture is exceptionally distinctive from past state-of-the-art structures such as AlexNet and ZF-Net. It employs numerous diverse sorts of strategies such as 1×1 convolution and worldwide normal pooling that enables it to make deeper architecture.

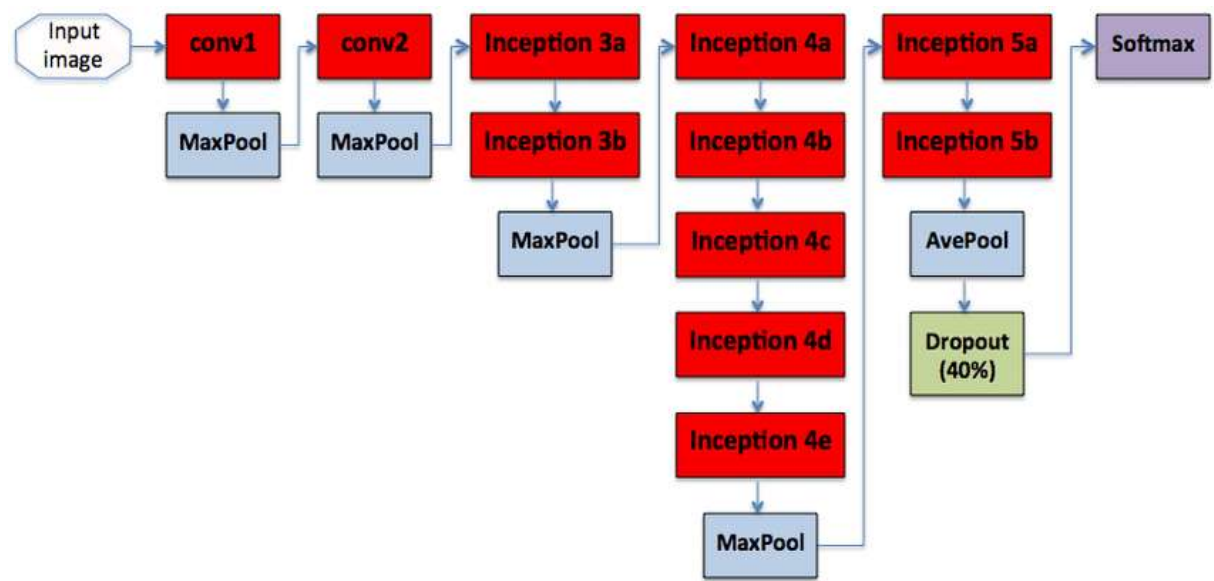


Figure 2.5.4: GoogleNet architecture

Inception architecture utilized a few middle classifier branches within the center of the design, these branches are utilized amid training as it were. These branches comprise a 5×5 normal pooling layer with a walk of 3, a 1×1 convolutions with 128 channels, two completely associated layers of 1024 yields and 1000 yields and a softmax classification layer. The created misfortune of these layers included to add up to misfortune with a weight of 0.3. These layers offer assistance in combating gradient vanishing issues and additionally give regularization.

2.5.5 ResNet50

ResNet, short for Residual Network is a specific type of neural network that was introduced in 2015 by Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun in their paper “Deep Residual Learning for Image Recognition”[37] After the primary CNN-based design (AlexNet) that win the ImageNet 2012 competition, Each consequent winning architecture employs more layers in a deep neural arrange to decrease the mistake rate.

This works for less number of layers, but when we increment the number of layers, there's a common issue in deep learning related to that called the Vanishing/Exploding angle. This causes the slope to end up or as well expansive. Hence when we increment the number of layers, the preparing and test blunder rate moreover increments.

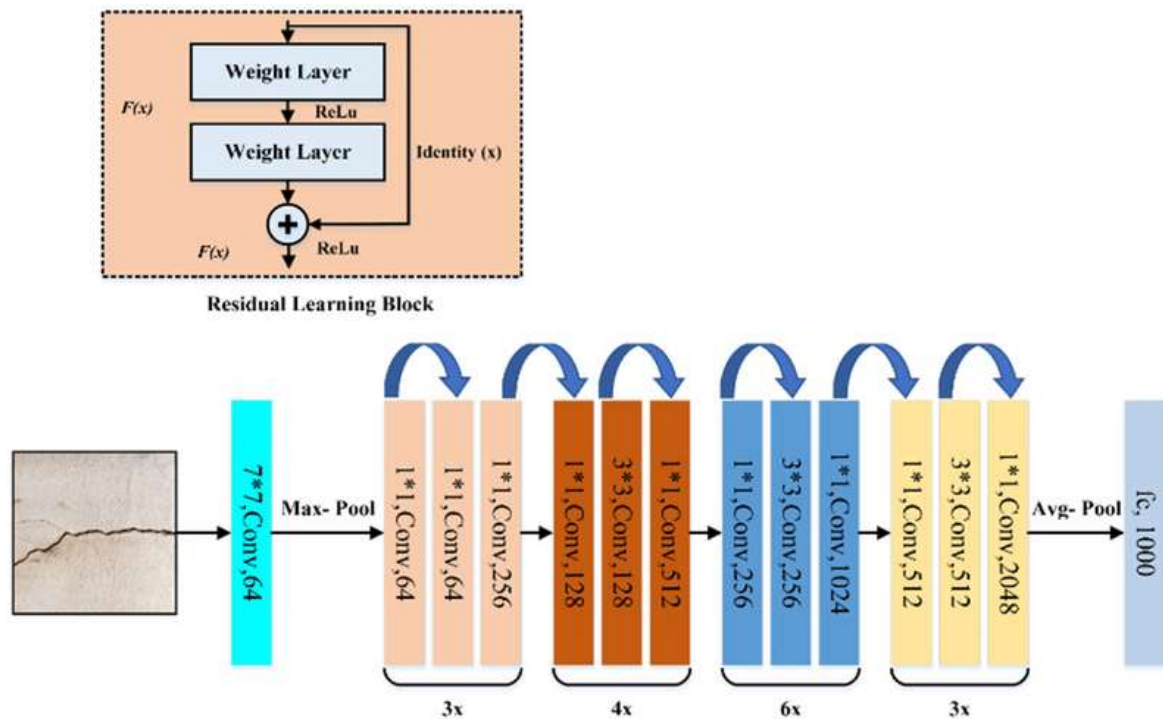


Figure 2.5.4: ResNet50 architecture

In order to unravel the issue of the vanishing/exploding slope, this engineering presented the concept called Remaining Organize. In this network, we utilize a strategy called skip connections. The skip connection skips preparing from a couple of layers and interfaces specifically to the output. The advantage of including this sort of skip

association is that in the event that any layer harmed the execution of architecture at that point it'll be skipped by regularization. So, this results in preparing a really deep neural network without the issues caused by vanishing/exploding angles.

2.5.6 Inception v3

Inception-v3 is a convolutional neural network architecture from the Inception family that makes several improvements including using Label Smoothing, Factorized 7 x 7 convolutions, and the use of an auxiliary classifier to propagate label information lower down the network (along with the use of batch normalization for layers in the side head).[26]

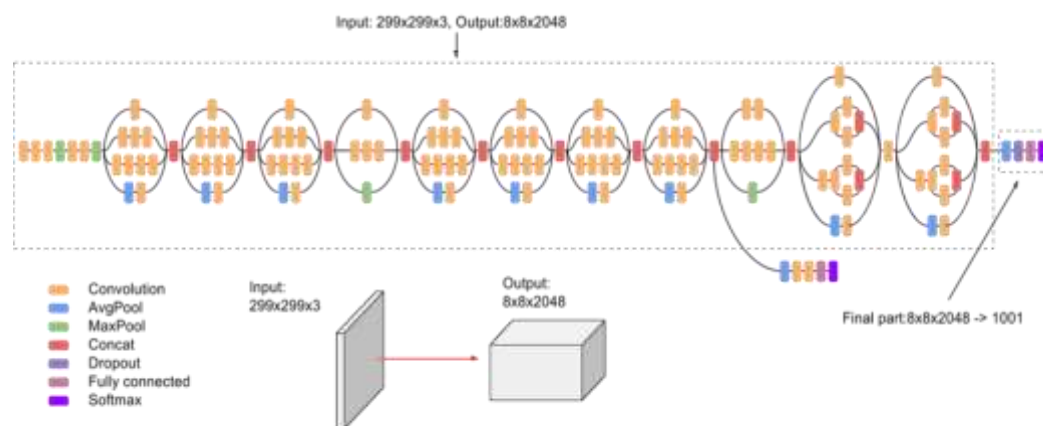


Figure-2.5.6(a): Architecture of Inception 3

Inception V3 used a number of network enhancement techniques to better match the model. Compared to the other version of Inception V1 and V2, it has a deeper network, but its speed is not compromised. It is cheaper than the other versions. Use sub-categories when ordering.

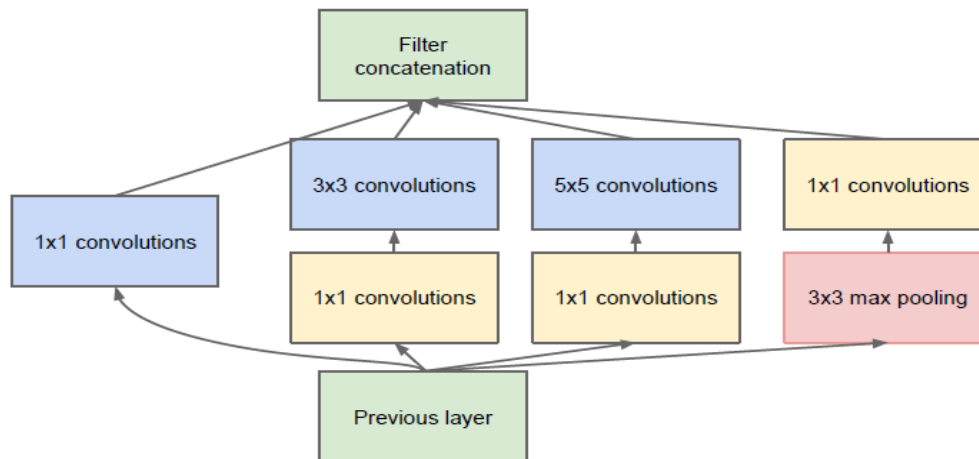


Figure-2.5.6(b): working principle

The inception v3 model was released within the year 2015, it encompasses an add-up of 42 layers and a lower blunder rate than its forerunners. The major adjustments done on the Inception V3 show are Factorization into Littler Convolutions Spatial Factorization into Asymmetric Convolutions Utility of Assistant Classifiers Efficient Framework Estimate Reduction. The inception V3 is just to the progressed and optimized adaptation of the inception V1 demonstrate. The Inception V3 show utilized a few techniques for optimizing the arrangement for superior demonstration adaptation. It has higher efficiency. It encompasses a deeper arrangement compared to the Inception V1 and V2 models, but its speed isn't compromised. It is computationally less expensive. It employments assistant Classifiers as regularizes.

2.5.7 DenseNet

A DenseNet is a type of convolutional neural network that utilizes dense connections between layers, through Dense Blocks, where we connect all layers (with matching feature-map sizes) directly with each other.[27]

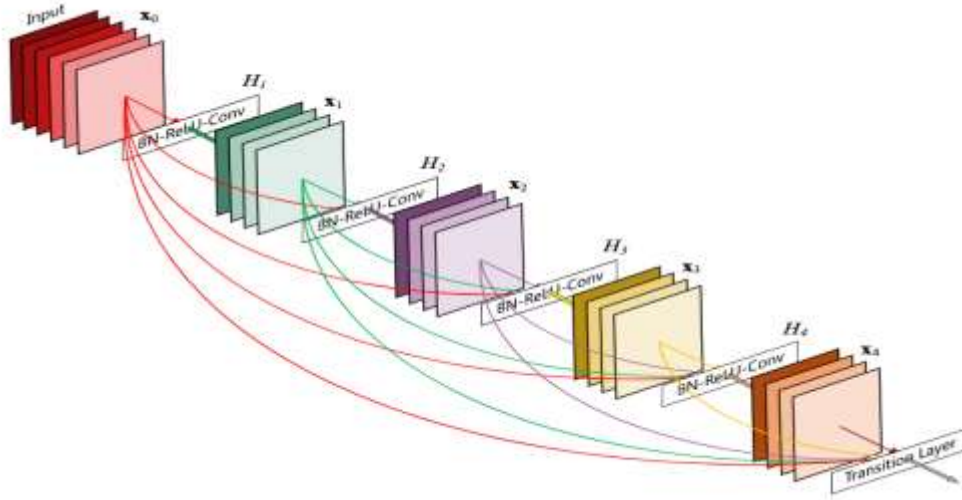


Figure-2.5.7(a) : Densely structure

DenseNet is one of the modern disclosures in neural systems for visual protest acknowledgement. DenseNet is very comparable to ResNet with a few crucial contrasts. ResNet employments an added substance strategy (+) that combines the past layer (personality) with the end of the layer, though DenseNet concatenates (.) the yield of the last layer with the long-haul layer. Get in-depth information on ResNet in this guide. DenseNet was created to make strides in the declined precision caused by the vanishing slope in high-level neural systems. In easier terms, due to the long way between the input layer and the yield layer, the data vanishes some time recently coming to its goal.

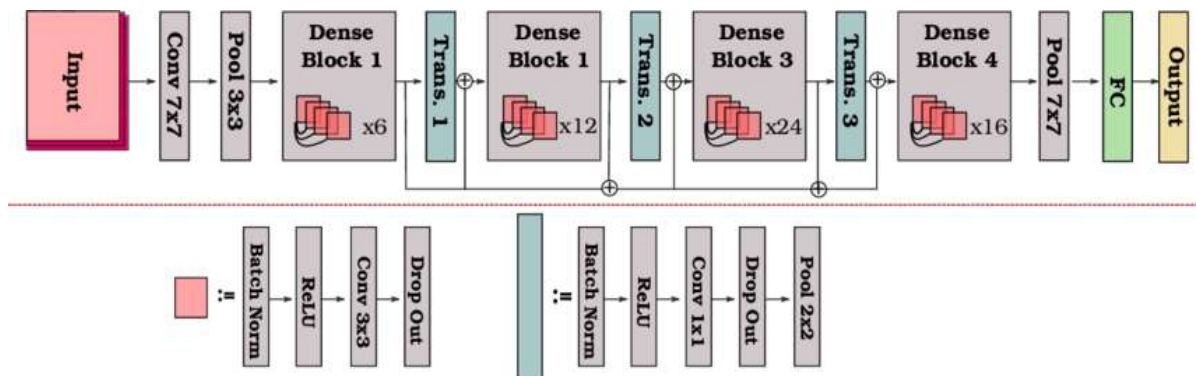


Figure-2.5.7(b): A schematic representation

DenseNet resolves issues by adjusting the standard CNN design and disentangling the network design between layers. In a DenseNet architecture, each layer is associated

straightforwardly with each other layer, consequently the title Densely Associated Convolutional Arrange. For 'L' layers, there are $L(L+1)/2$ coordinate associations.

CHAPTER-3

Literature Review

3.1 Introduction

Deep learning, commonly referred to as deep structured learning, is one of several machine learning techniques built on representation learning and artificial neural networks. Unsupervised, semi-supervised, and supervised learning are all possible.

In fields like computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection, and board game programs, deep-learning architectures like deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks, and convolutional neural networks have been used. These applications have led to results that are comparable to and in some cases even better than those of traditional approaches.

The usage of several network layers is indicated by the term "deep" in deep learning. Early research shown that a network with one hidden layer of unlimited breadth and a nonpolynomial activation function may be a universal classifier but a linear perceptron cannot. A more recent form, known as deep learning, focuses on an infinite number of layers with bounded sizes, allowing for practical application and efficient implementation while maintaining theoretical universality under benign circumstances. For the sake of efficiency, trainability, and understandability, deep learning also allows the layers to be diverse and diverge greatly from physiologically informed connectionist models, hence the "structured" aspect.

Most modern deep learning models are based on artificial neural networks, specifically convolutional neural networks (CNN)s, although they can also include propositional formulas or latent variables organized layer-wise in deep generative models such as the nodes in deep belief networks and deep Boltzmann machines.[21]

3.2 Literature review

A work on X-ray image classification is presented by Tulin et al. where They utilized the DarkNet show for classifying the identified YOLO-based question within the input pictures.[22] Another work on X-ray pictures called Decompose, Transfer, and Compose (DeTraC), is presented by Asmaa et al. in DeTraC utilized the lesson decomposition component to bargain with lesson boundaries driving to the exactness of 0.931 and affectability of 1.[23]. Taban et al. also worked on chest X-ray images where he utilized the 12-off-the-shelf CNN designs in transfer learning [24]. Mohamed et al. used GANs for covid19 analysis[25].

two deep learning architectures are proposed by Muhammet Fatih Aslan,Muhammed Fahri Unlersen, Kadir Sabanci, Akif Durdu that automatically detect positive COVID-19 cases using Chest CT X-ray images. Lung segmentation (preprocessing) in CT images, which are given as input to those proposed architectures, is performed automatically with Artificial Neural Networks (ANN). Since both architectures contain AlexNet architecture, the recommended method could be a transfer learning application. However, the second proposed architecture may be a hybrid structure because it contains a Bidirectional Long Short-Term Memories (BiLSTM) layer, which also takes into consideration the temporal properties. While the COVID-19 classification accuracy of the primary architecture is 98.14%, this value is 98.70% within the second hybrid architecture. The results prove that the proposed architecture shows outstanding success in infection detection and, therefore this study contributes to previous studies in terms of both deep architectural design and high classification success[28]

Shervin Minaee, Rahele Kafieh, Milan Sonka, Shakib Yazdani, Ghazaleh Jamalipour Soufi study the application of deep learning models to detect COVID-19 patients from their chest radiography images. We first prepare a dataset of 5000 Chest X-rays from the publicly available datasets. Images exhibiting COVID-19 disease presence were identified by board-certified radiologist. Transfer learning on a subset of 2000 radiograms was accustomed to train four popular convolutional neural networks, including ResNet18, ResNet50,SqueezeNet, and DenseNet-121, to spot COVID-19

disease within the analyzed chest X-ray images. We evaluated these models on the remaining 3000 images, and most of those networks achieved a sensitivity rate of 98% ($\pm 3\%$), while having a specificity rate of around 90%. Besides sensitivity and specificity rates, we also present the receiver operating characteristic (ROC) curve, precision-recall curve, average prediction, and confusion matrix of every model. We also used a way to get heatmaps of lung regions potentially infected by COVID-19 and show that the generated heat maps contain most of the infected areas annotated by our board certified radiologist. While the achieved performance is incredibly encouraging, further analysis is required on a bigger set of COVID-19 images, to possess a more reliable estimation of accuracy rates.[29]

Mohamed Berrimi, Skander Hamdi, Raoudha Yahia Cherif, Abdelouahab Moussaoui, Mourad Oussalah, Mafaza Chabane they propose deep learning models using Transfer learning to detect COVID-19. Both X-ray and CT scans were considered to evaluate the proposed methods.[30]

Michael J. Horry, Subrata Chakraborty, Manoranjan Paul, Anwaar Ulhaq, Biswajeet Pradhan, Manas Saha, Nagesh Shukla demonstrate how transfer learning from deep learning models are often accustomed to perform COVID-19 detection using images from three most typically used medical imaging modes X-Ray, Ultrasound, and CT scan. The aim is to produce over-stressed medical professionals a second pair of eyes through intelligent deep learning image classification models. We identify an acceptable Convolutional Neural Network (CNN) model through initial comparative study of several popular CNN models. We then optimize the chosen VGG19 model for the image modalities to point out how the models will be used for the highly scarce and challenging COVID-19 datasets. We highlight the challenges (including dataset size and quality) in utilizing current publicly available COVID-19 datasets for developing useful deep learning models and the way it adversely impacts the trainability of complex models. We also propose a picture pre-processing stage to make a trustworthy image dataset for developing and testing the deep learning models. The new approach is aimed to scale back unwanted noise from the photographs so deep learning models can specialize in detecting diseases with specific features from them. Our results indicate that Ultrasound images provide superior detection accuracy

compared to X-Ray and CT scans. The experimental results highlight that with limited data, most of the deeper networks struggle to coach well and provide less consistency over the three imaging modes we are using. The chosen VGG19 model, which is then extensively tuned with appropriate parameters, performs in considerable levels of COVID-19 detection against pneumonia or normal for all three lung image modes with the precision of up to 86% for X-Ray, 100% for Ultrasound and 84% for CT scans.[31]

Taban Majeed, Rasber Rashid, Dashti Ali, and Aras Asaad proposed a quantitative chemical analysis to gauge 12 off-the-shelf convolutional neural networks (CNNs) for the purpose of COVID-19 X-ray image analysis. Specifically, CNN transfer learning procedure was adopted because of the little number of images available for investigation. They also proposed a straightforward CNN architecture with a little number of parameters that perform well on distinguishing COVID-19 from normal X-rays. A qualitative investigation performed to examine the decisions made by CNNs employing a technique referred to as class activation maps (CAM). Using CAMs, one can map the activations contributed most to the choice of CNNs back to the first image to visualise the foremost discriminating regions on the input image. Chest X-ray images employed in this work are coming from 3 publicly available sources. Two COVID-19 X-ray image datasets and an outsized dataset of other non-COVID-19 viral infections, bacterial infections and normal X-rays utilised. We conclude that CNN decisions shouldn't be taken into consideration, despite their high classification accuracy, until clinicians can visually inspect the regions of the input image utilized by CNNs that cause its prediction.[38]

CHAPTER-4

Methodology

4.1 Introduction

In this segment, we display the proposed strategy for classifying an X-ray as being of a solid persistent or a patient affected by COVID-19. To begin with, we depict the datasets of images utilized in this consideration. At that point, we clarify the method of feature extraction, which is based on the exchange learning theory. After that, we show the classification techniques applied and the steps of their preparing handle. Finally, we define the metrics we utilize to assess the comes about and to compare it to other approaches. Table- 1 presents the information of the proposed approach; each step is clarified within the following subsections.

4.2 DATASET

In our study, we use frontal-view chest X-ray images.

Table-1

Architecture	Number of Train data	Number of Validation data	Image size(pixel)
AlexNet	6000+2000	4000+1000	240*240
VGG16	6000+2000	4000+1000	240*240
ResNet50	6000+2000	4000+1000	240*240
VGG19	6000+2000	4000+1000	240*240
GoogleNet	6000+2000	4000+1000	240*240

Inception3	6000+2000	4000+1000	240*240
DenseNet	6000+2000	4000+1000	240*240

Figure-4.1 : Our chosen dataset

Though we used 2 datasets from Kaggle but here we mention one of them. We divided the samples into two classes: X-ray pictures of patients analyzed with COVID-19 and X-ray pictures of sound patients. For a superior assessment of the proposed strategy, we built two datasets: Dataset A and dataset B. Dataset A means train dataset. Dataset B means validation dataset. Then again, we divided datasets A and B into subdivisions. Both datasets have different images for the COVID-19 class and for the healthy class. In both datasets, the classes are balanced, consisting of total 8000 images for the A dataset and total 5000 images for the B dataset.

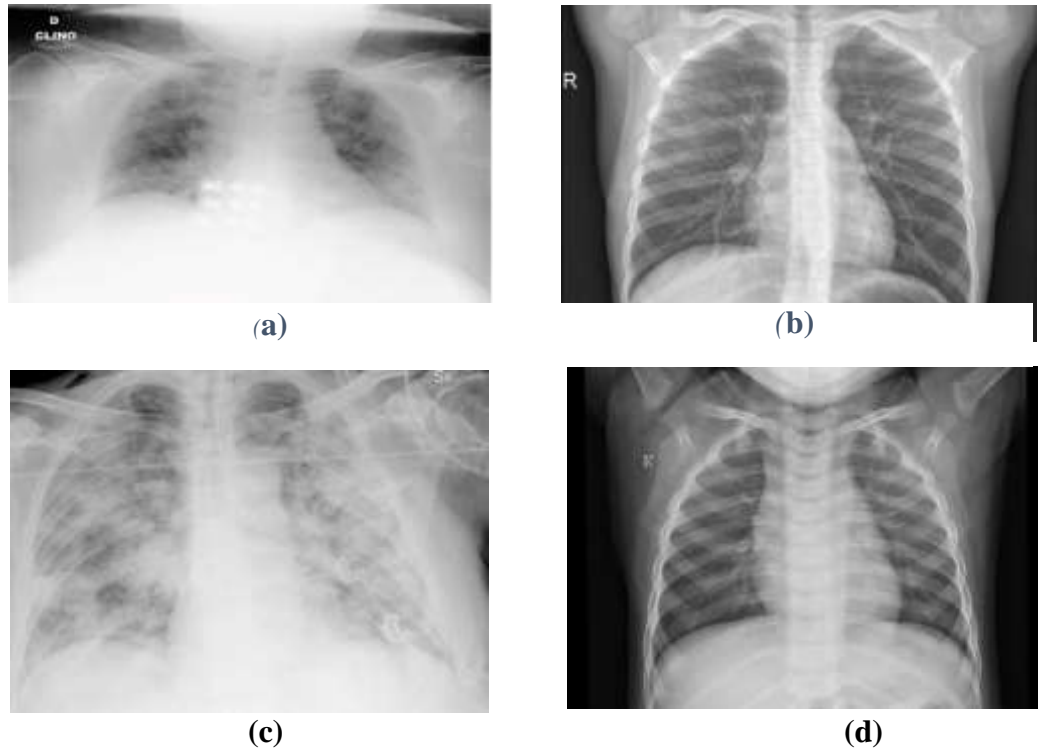


Figure-4.1: Samples from the dataset used in this study. (a) X-ray of a patient with COVID-19 from, Dataset A; (b) X-ray of a healthy patient from Dataset A;(c)X-ray of a patient with COVID-19 from Dataset B; (d) X-ray of a healthy patient from Dataset B. We choose the images randomly from our dataset.

4.3 Feature Extraction Steps

For extracting features from the X-ray images, we utilize the transfer learning concept. We selected diverse CNN designs that accomplished excellent performance on the ImageNet dataset. We evacuate any fully connected layers from these arrangements. Convolutional and pooling layers are capable of extricating features from the image, while the completely associated ones are dependable for classifying the features and, thus, the picture. In this way, evacuating these layers is vital to turn a CNN into a feature extractor. After this step, the unused output of the adjusted CNN could be a set of features extricated from an input image. Since we have worked on 5 types of architecture, we discuss the further steps of them separately.

4.3.1 AlexNet

AlexNet is a deep architecture. It has eight layers with learnable parameters. The model comprises five layers with a combination of max pooling taken after by 3 completely associated layers and they utilize ReLu activation in each of these layers but the output layer.

The input to this model is the images of size 224×224 . Then we apply the first convolution layer with 96 filters of size 11×11 with stride 4. The activation function used in this layer is ReLu. After that we have the first Max Pooling layer, of size 3×3 and stride 2. Next, we apply the second convolution operation. This time the filter size is reduced to 5×5 and we have 256 such filters. The stride is 1 and padding 2. The activation function used is again ReLu. Then again we applied a max-pooling layer of size 3×3 with stride 2.

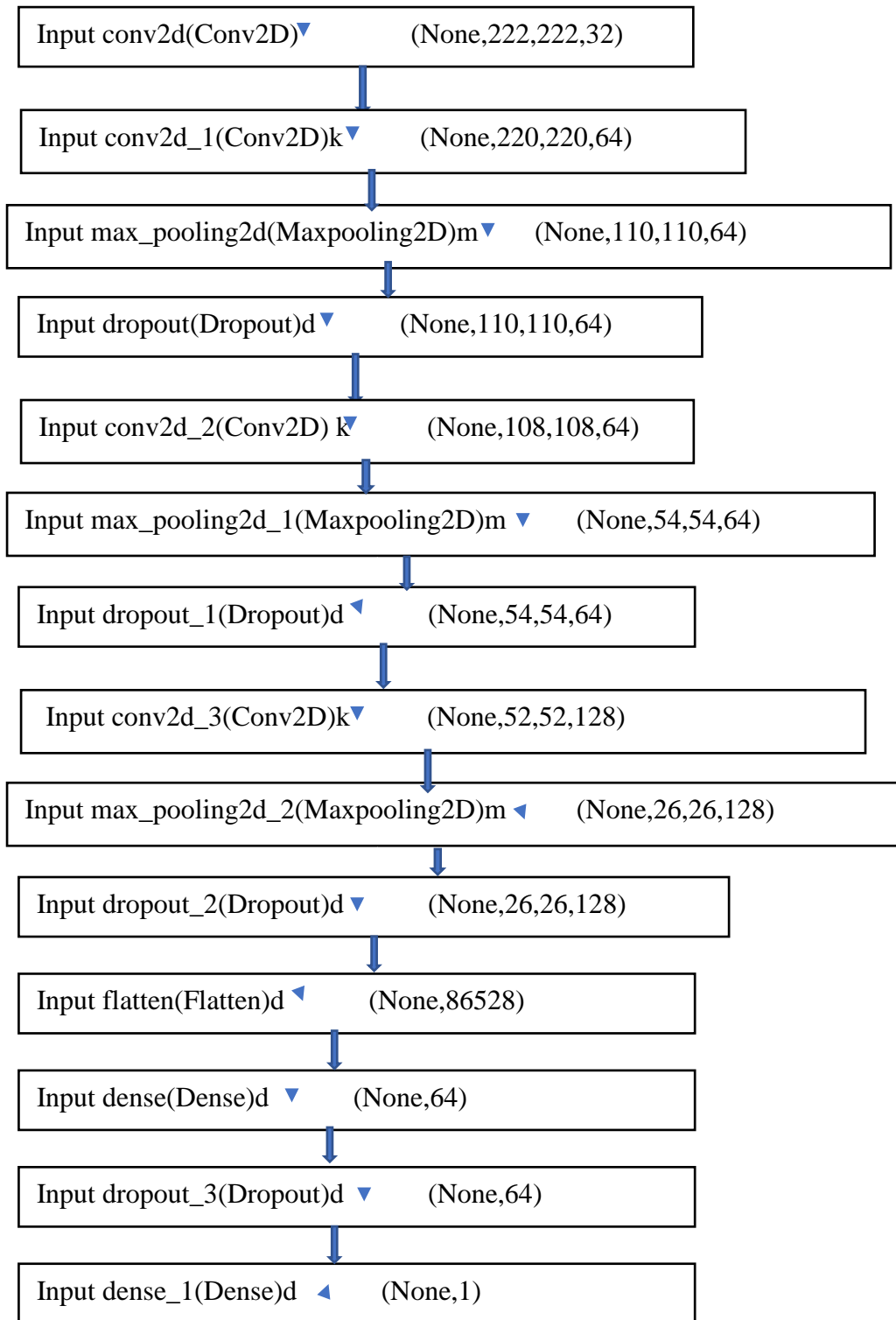


Figure-4.3.1(a): Flow chart for our model

Then we applied the third convolutional operation where the activation function used is ReLu. The number of filters is expanding as we are going deeper. Consequently, it is extricating more features as we move deeper into the architecture. Moreover, the channel measure is diminishing, which suggests the introductory filter was bigger and as we go ahead the filter size is decreasing, resulting in a diminished within the included outline shape. After this, we have to begin with the dropout layer. The dropout rate is set to be 0.5. Then we have the primary completely associated layer with a ReLu actuation work. The estimate of the yield is 4096. Another comes another dropout layer with the dropout rate settled at 0.5. This taken after by a moment completely associated layer with 4096 neurons and ReLu activation. Finally, we have the final completely associated layer or yield layer with 1000 neurons as we have 10000 classes within the information set. The activation layer we utilized at this layer is Sigmoid because Sigmoid is comparable to a 2-element Softmax, where the moment component is accepted to be zero. In this manner, sigmoid is for the most part utilized for double classification.

The number of epoches we used in our model is 8 and steps per epoch is 2. The optimizer we used in our model is ‘Adam’ as it is best among the adaptive optimizer and also more suitable for our model. We train our model by giving it two types of datasets. One is covid positive and the other one is negative. As we used two types of datasets that's why our class mode is binary.

```
Epoch 1/8
6/6 [=====] - 67s 11s/step - loss: 1.0287 - accuracy: 0.4844 - val_loss: 0.6890 - val_accuracy: 0.5000
Epoch 2/8
6/6 [=====] - 67s 11s/step - loss: 0.6513 - accuracy: 0.6146 - val_loss: 0.6288 - val_accuracy: 0.7333
Epoch 3/8
6/6 [=====] - 65s 11s/step - loss: 0.5370 - accuracy: 0.7448 - val_loss: 0.4694 - val_accuracy: 0.9333
Epoch 4/8
6/6 [=====] - 63s 11s/step - loss: 0.3912 - accuracy: 0.8438 - val_loss: 0.3156 - val_accuracy: 0.9333
Epoch 5/8
6/6 [=====] - 67s 11s/step - loss: 0.3344 - accuracy: 0.8438 - val_loss: 0.2878 - val_accuracy: 0.9333
Epoch 6/8
6/6 [=====] - 63s 10s/step - loss: 0.3020 - accuracy: 0.8802 - val_loss: 0.2057 - val_accuracy: 0.9667
Epoch 7/8
6/6 [=====] - 63s 10s/step - loss: 0.2467 - accuracy: 0.8958 - val_loss: 0.1415 - val_accuracy: 0.9833
Epoch 8/8
6/6 [=====] - 63s 10s/step - loss: 0.1931 - accuracy: 0.9010 - val_loss: 0.1093 - val_accuracy: 0.9667
```

Figure-4.3.1(b): Number of epochs, loss of AlexNet

Here, we can clearly notice that the maximum accuracy is 89% while the maximum validation accuracy is 96%. We also discern the minimum loss is 19% while the minimum validation loss is 10%.

For calculating accuracy, specificity and sensitivity we just used two simple equations where we used value from our confusion matrix. Now the equations are given below:

$$\begin{aligned}\text{Accuracy} &= (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \\ &= (25+5)/(25+5+5+10) \\ &= 30/45 \\ &= 0.66\end{aligned}$$

$$\begin{aligned}\text{Sensitivity} &= \text{TP}/(\text{TP}+\text{FN}) \\ &= 25/(25+10) \\ &= 0.71\end{aligned}$$

$$\begin{aligned}\text{Specificity} &= \text{TN}/(\text{TN}+\text{FP}) \\ &= 5/(5+5) \\ &= 5/10 \\ &= 0.5\end{aligned}$$

4.3.2 VGG19

VGG19 is a pre-trained network which we used for feature extraction. In our covid detection model we used VGG19 based fusion framework to observe the accuracy percentage and we also compared the accuracy of this model with other CNN architecture-based models' accuracy.

Because of artifacts and erroneous tagging, the KAGGLE collection contains a significant number of indefinable images. To recognize the type, picture segmentation methods are applied. We give our images as input in this architecture then it subtracted the RGB value from each pixel from the whole training set which was given by us. Then it used Kernel size to enable them to cover the whole notion of the image. Then for the spatial resolution of the image preservation we used spatial padding. After that max-pooling was performed by ReLu to introduce non-linearity to classify the model better and improve computation time as previous models using tanh or sigmoid

functions turned out to be much better than this one. Then we implemented three fully connected layers. The primary goal of picture segmentation now is to identify hard and soft exudates, hemorrhages, and microaneurysms as white patches for ease of class annotation. To improve the contrast for segmentation, the RGB split and contrast limited adaptive histogram adjustment are used first. After that, background subtraction was used to identify a region of interest, which was then segmented with the use of a mask image. During the image segmentation process, feature detection identifies visually salient regions from an image and develops the saliency map, which is a useful tool for identifying low-level luminance visual features from color images. This method was widely used for evaluating items and image regions automatically without any prior information. As a result, changes between pixels and their immediate surroundings are specified.

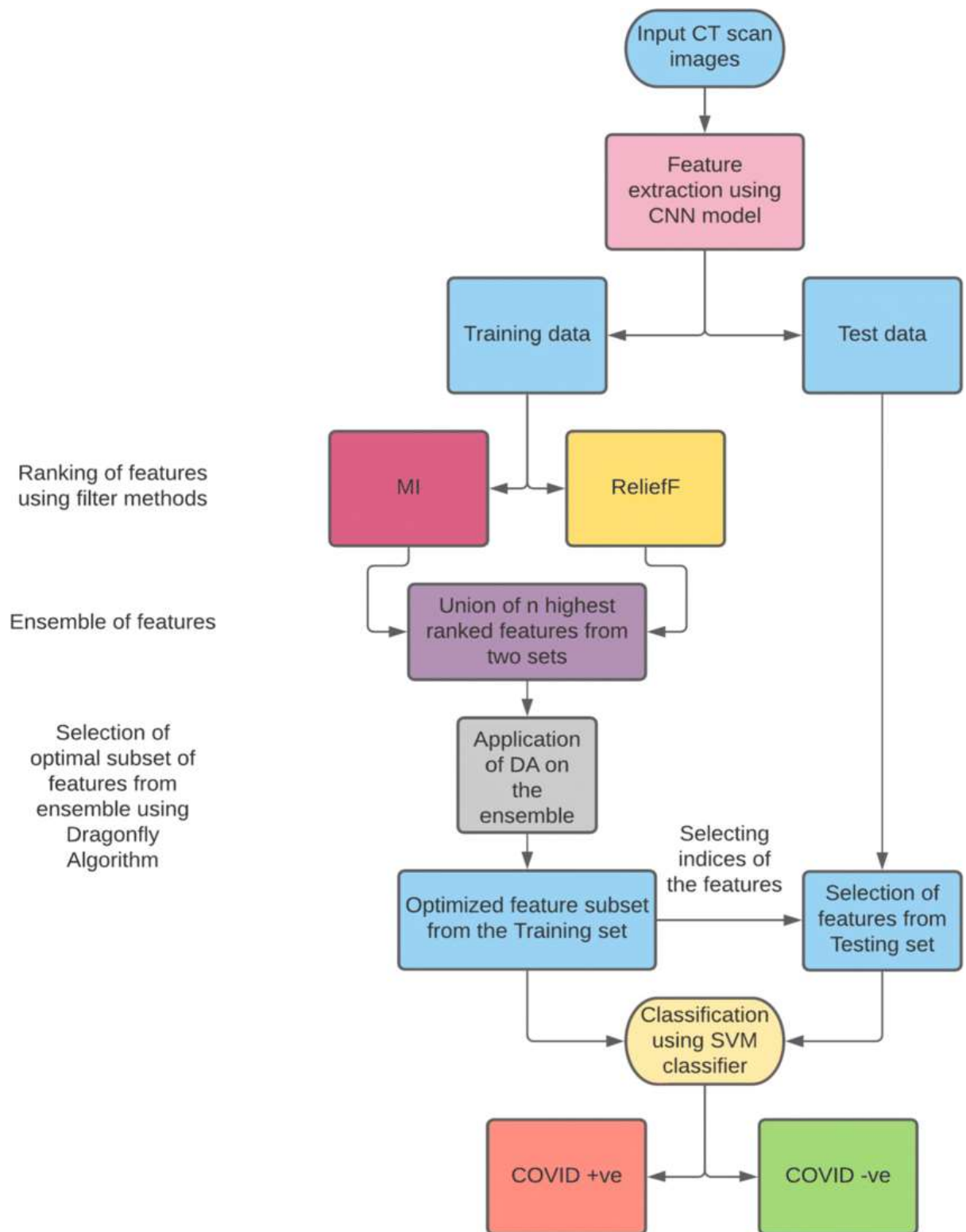


Figure-4.3.2(a): Flowchart of our proposed VGG19 model

The number of epochs we used in our model is 8 and steps per epoch is 2. The optimizer we used in our model is ‘Adam’ as it is best among the adaptive optimizer and also more suitable for our model. We trained our model by giving it two types of datasets. One is covid positive and the other one is negative. As we used two types of datasets that’s why our class mode is binary.

```
Epoch 1/8
5/5 [=====] - 151s 28s/step - loss: 1.7162 - accuracy: 0.4663 - val_loss: 0.4444 - val_accuracy: 0.7609
Epoch 2/8
5/5 [=====] - 146s 29s/step - loss: 0.6676 - accuracy: 0.7022 - val_loss: 0.1520 - val_accuracy: 0.9565
Epoch 3/8
5/5 [=====] - 145s 27s/step - loss: 0.3029 - accuracy: 0.8596 - val_loss: 0.1412 - val_accuracy: 0.9783
Epoch 4/8
5/5 [=====] - 146s 27s/step - loss: 0.1106 - accuracy: 0.9494 - val_loss: 0.0571 - val_accuracy: 0.9565
Epoch 5/8
5/5 [=====] - 145s 27s/step - loss: 0.1544 - accuracy: 0.9213 - val_loss: 0.1511 - val_accuracy: 0.9565
Epoch 6/8
5/5 [=====] - 144s 29s/step - loss: 0.1424 - accuracy: 0.9382 - val_loss: 0.1004 - val_accuracy: 0.9783
Epoch 7/8
5/5 [=====] - 146s 29s/step - loss: 0.1043 - accuracy: 0.9607 - val_loss: 0.0391 - val_accuracy: 0.9565
Epoch 8/8
5/5 [=====] - 146s 27s/step - loss: 0.1101 - accuracy: 0.9607 - val_loss: 0.0749 - val_accuracy: 0.9783
```

Figure-4.3.2(b): Number of epochs, accuracy, losses, validation accuracy and validation losses for our **VGG19** model.

Here, we can clearly notice that the maximum accuracy is 96% while the maximum validation accuracy is 97%. We also discern the minimum loss is 10% while the minimum validation loss is 3%.

For calculating accuracy, specificity and sensitivity we just used three simple equations where we used value from our confusion matrix. Now the equations are given below:

$$\begin{aligned}
 \text{Accuracy} &= (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \\
 &= (22+23)/(22+23+1+0) \\
 &= 45/46 \\
 &= 0.97
 \end{aligned}$$

$$\begin{aligned}
 \text{Sensitivity} &= \text{TP}/(\text{TP}+\text{FN}) \\
 &= 22/(22+0) \\
 &= 1
 \end{aligned}$$

$$\begin{aligned}
\text{Specificity} &= \text{TN}/(\text{TN}+\text{FP}) \\
&= 23/(23+1) \\
&= 23/24 \\
&= 0.96
\end{aligned}$$

4.3.3 ResNet50

It's harder to train deep neural networks. Resnet makes it possible to overcome the challenges of training very deep neural networks. ResNet comes in a variety of flavors, each with a different number of layers but the same basic premise. The term Resnet50 refers to a variation that can work with up to 50 neural network layers. The Resnet50 architecture is based on the above model, however there is one significant distinction. Due to concerns about the time required to train the layers, the building block was redesigned into a bottleneck design. Instead of the previous two levels, this one had three.

As a result, each of the Resnet34's 2-layer blocks was replaced with a Resnet 50's 3-layer bottleneck block. Compared to the 34-layer ResNet model, this one is far more accurate. ResNet's 50 layers reach 3.8 billion FLOPS of performance.

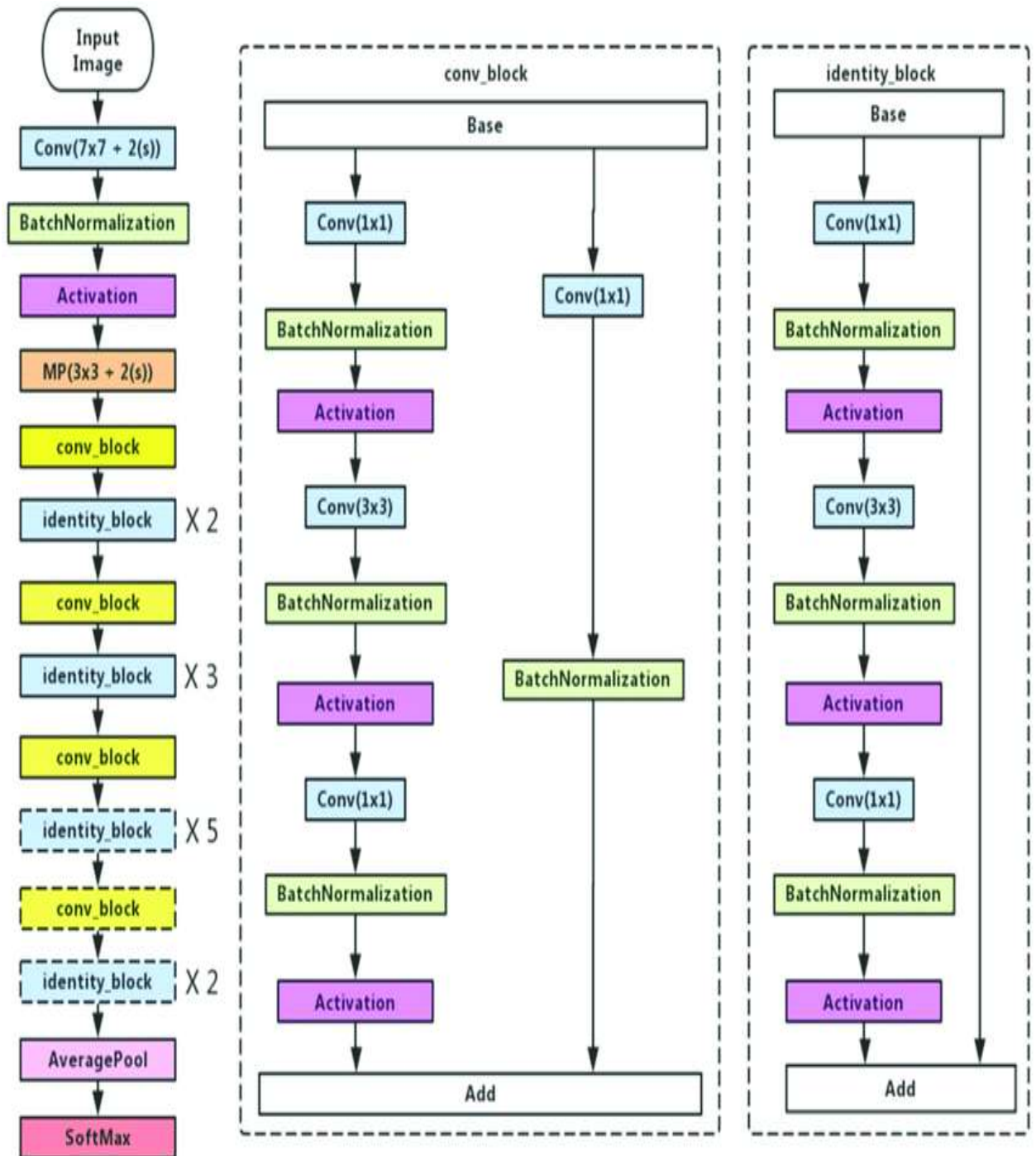


Figure-4.3.3(a) : Flowchart of our proposed ResNet50 model

To change the CNN into a residual network and generate the convolution block, we run a code to define the identity blocks. The next step is to combine both blocks to create a 50-layer Resnet model. There are two paths for gradients to transit back to the input layer when traversing a residual block during backpropagation. There are two pathways shown in the next diagram: pathway-1 is for identity mapping, while pathway-2 is for residual mapping.

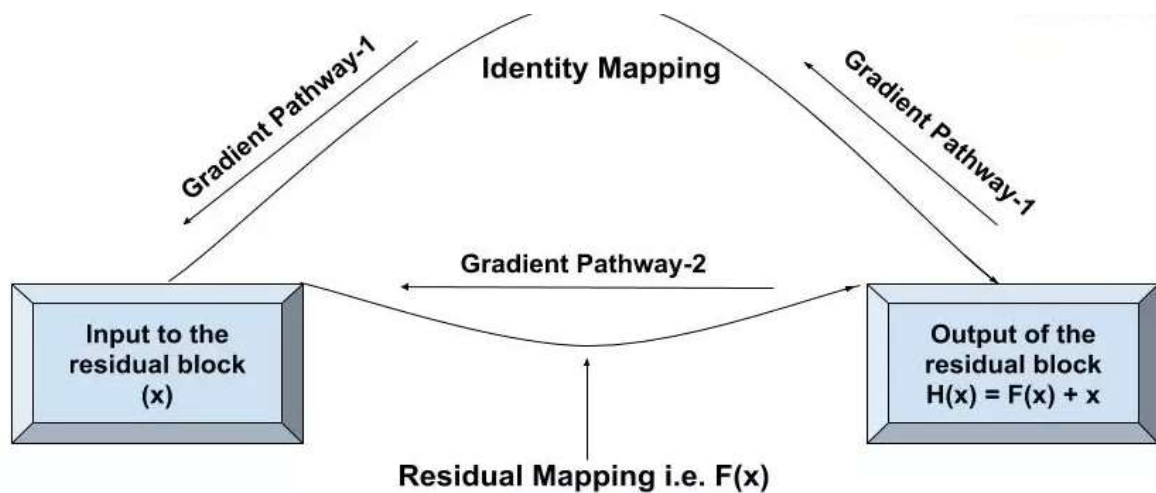


Figure-4.3.3(b): How we did our residual mapping

Finally, we prepare the model for the task at hand. We can quickly provide a detailed overview of the network architecture you created with Keras. For future use, this can be saved or printed. The number of epochs we used in our model is 8 and steps per epoch is 2. The optimizer we used in our model is ‘Adam’ as it is best among the adaptive optimizer and also more suitable for our model. We train our model by giving it two types of datasets. One is covid positive and the other one is negative. As we used two types of datasets that's why our class mode is binary. The addition of the identification connection does not result in the addition of any additional parameters. As a result, the complexity of computing for simple deep networks and deep residual networks is nearly identical. For the addition operation to work, the dimensions of x and F must be the same. One of the following methods can be used to match the dimensions:

For increased dimensions, extra Zero entries should be padded. There will be no more parameters introduced as a result of this. To match the dimensions, a projection shortcut might be employed (1 x 1 convolutions).

$$y = F(x, \{W_i\}) + W_s x$$

```
Epoch 1/8
5/5 [=====] - 30s 5s/step - loss: 5.7185 - accuracy: 0.4663 - val_loss: 6.4936
Epoch 2/8
5/5 [=====] - 26s 5s/step - loss: 3.4533 - accuracy: 0.5169 - val_loss: 1.2719
Epoch 3/8
5/5 [=====] - 27s 5s/step - loss: 1.6880 - accuracy: 0.5787 - val_loss: 1.6231
Epoch 4/8
5/5 [=====] - 29s 5s/step - loss: 1.8167 - accuracy: 0.5337 - val_loss: 1.5816
Epoch 5/8
5/5 [=====] - 26s 5s/step - loss: 1.2595 - accuracy: 0.5730 - val_loss: 0.2932
Epoch 6/8
5/5 [=====] - 25s 5s/step - loss: 0.6262 - accuracy: 0.7697 - val_loss: 0.4862
Epoch 7/8
5/5 [=====] - 25s 5s/step - loss: 0.6984 - accuracy: 0.6966 - val_loss: 0.1766
Epoch 8/8
5/5 [=====] - 25s 5s/step - loss: 0.5012 - accuracy: 0.8034 - val_loss: 0.2333
```

Figure-4.3.3(c): Number of epochs, accuracy, losses, validation accuracy and validation losses for our **ResNet50** model.

Here, we can clearly notice that the maximum accuracy is 80% while the maximum validation accuracy is 95%. We also discern the minimum loss is 50% while the minimum validation loss is 17%.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

$$= (23+18)/(23+18+0+5)$$

$$= 41/46$$

$$= 0.89$$

$$\text{Sensitivity} = \text{TP}/(\text{TP}+\text{FN})$$

$$= 23/(23+5)$$

$$= 23/28$$

$$= 0.82$$

$$\text{Specificity} = \text{TN}/(\text{TN}+\text{FP})$$

$$= 18/(18+0)$$

$$= 18/18$$

$$= 1$$

4.3.4 Inception v3

Inception-v3 is a convolutional neural network design from the Inception family that includes Label Smoothing, factorized 7 x 7 convolutions, and the inclusion of an auxiliary classifier to transport label information lower down the network, among other advancements.

The model must be trained on a large number of labeled photos before it can be used to recognize images. ImageNet is a widely used dataset.

Over ten million URLs of captioned photos are available on ImageNet. The ImageNet dataset is composed v respectively for this model. The training and evaluation datasets are intentionally kept separate. To train the model, only images from the training dataset are used, and only images from the evaluation dataset are used to assess model accuracy.

Inception v3's present implementation is on the verge of being input-bound. Images are retrieved from the file system, encoded, and pre-processed before being used. There are a variety of preprocessing stages to choose from, ranging from simple to sophisticated. The training pipeline becomes much more complex if we use the most complex preprocessing phases. Image preprocessing is an important aspect of the system that can affect the model's maximum accuracy during training. Images must be decrypted and scaled to fit the model at the very least. To minimize the grid size of feature maps, max pooling and average pooling were traditionally utilized. The activation dimension of the network filters is enhanced in the inception V3 model in order to lower the grid size efficiently.

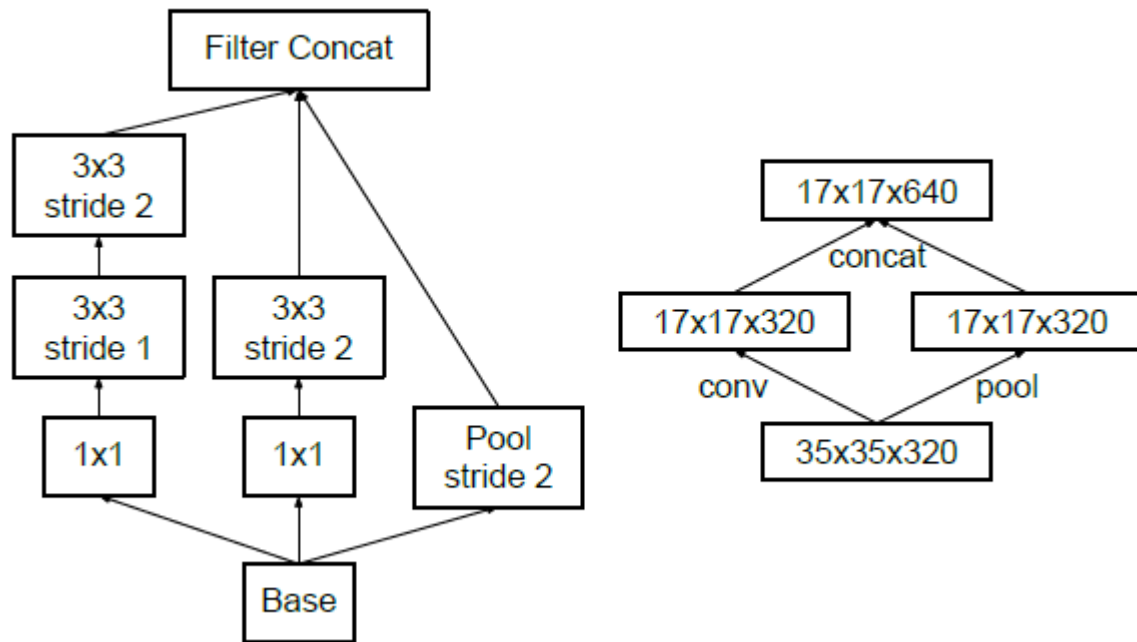


Figure-4.3.4(a): Efficiently reduced grid size

The number of epochs we used in our model is 8 and steps per epoch is 2. The optimizer we used in our model is ‘Adam’ as it is best among the adaptive optimizer and also more suitable for our model. We train our model by giving it two types of datasets. One is covid positive and the other one is negative. As we used two types of datasets that's why our class mode is binary.

```

Epoch 1/8
5/5 [=====] - 38s 6s/step - loss: 4.8718 - accuracy: 0.6517 - val_loss: 0.2768 - val_accuracy: 0.9783
Epoch 2/8
5/5 [=====] - 35s 6s/step - loss: 0.8328 - accuracy: 0.9438 - val_loss: 3.0669 - val_accuracy: 0.8043
Epoch 3/8
5/5 [=====] - 34s 6s/step - loss: 0.6597 - accuracy: 0.9382 - val_loss: 0.8614 - val_accuracy: 0.9348
Epoch 4/8
5/5 [=====] - 26s 5s/step - loss: 0.4228 - accuracy: 0.9831 - val_loss: 0.0645 - val_accuracy: 0.9783
Epoch 5/8
5/5 [=====] - 25s 5s/step - loss: 0.5807 - accuracy: 0.9607 - val_loss: 0.2939 - val_accuracy: 0.9565
Epoch 6/8
5/5 [=====] - 25s 5s/step - loss: 0.3493 - accuracy: 0.9831 - val_loss: 0.2123 - val_accuracy: 0.9565
Epoch 7/8
5/5 [=====] - 25s 5s/step - loss: 0.3267 - accuracy: 0.9888 - val_loss: 0.4845 - val_accuracy: 0.9348
Epoch 8/8
5/5 [=====] - 26s 5s/step - loss: 0.3589 - accuracy: 0.9775 - val_loss: 0.9661 - val_accuracy: 0.9348

```

Figure-4.3.4(b): Number of epochs, accuracy, losses, validation accuracy and validation losses for our **Inception3** model

Here, we can clearly notice that the maximum accuracy is 98% while the maximum validation accuracy is 97%. We also discern the minimum loss is 32% while the minimum validation loss is 6%.

$$\begin{aligned}\text{Accuracy} &= (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \\ &= (20+23)/(20+23+3+0) \\ &= 43/46 \\ &= 0.93\end{aligned}$$

$$\begin{aligned}\text{Sensitivity} &= \text{TP}/(\text{TP}+\text{FN}) \\ &= 20/(20+0) \\ &= 1\end{aligned}$$

$$\begin{aligned}\text{Specificity} &= \text{TN}/(\text{TN}+\text{FP}) \\ &= 23/(23+3) \\ &= 23/26 \\ &= 0.88\end{aligned}$$

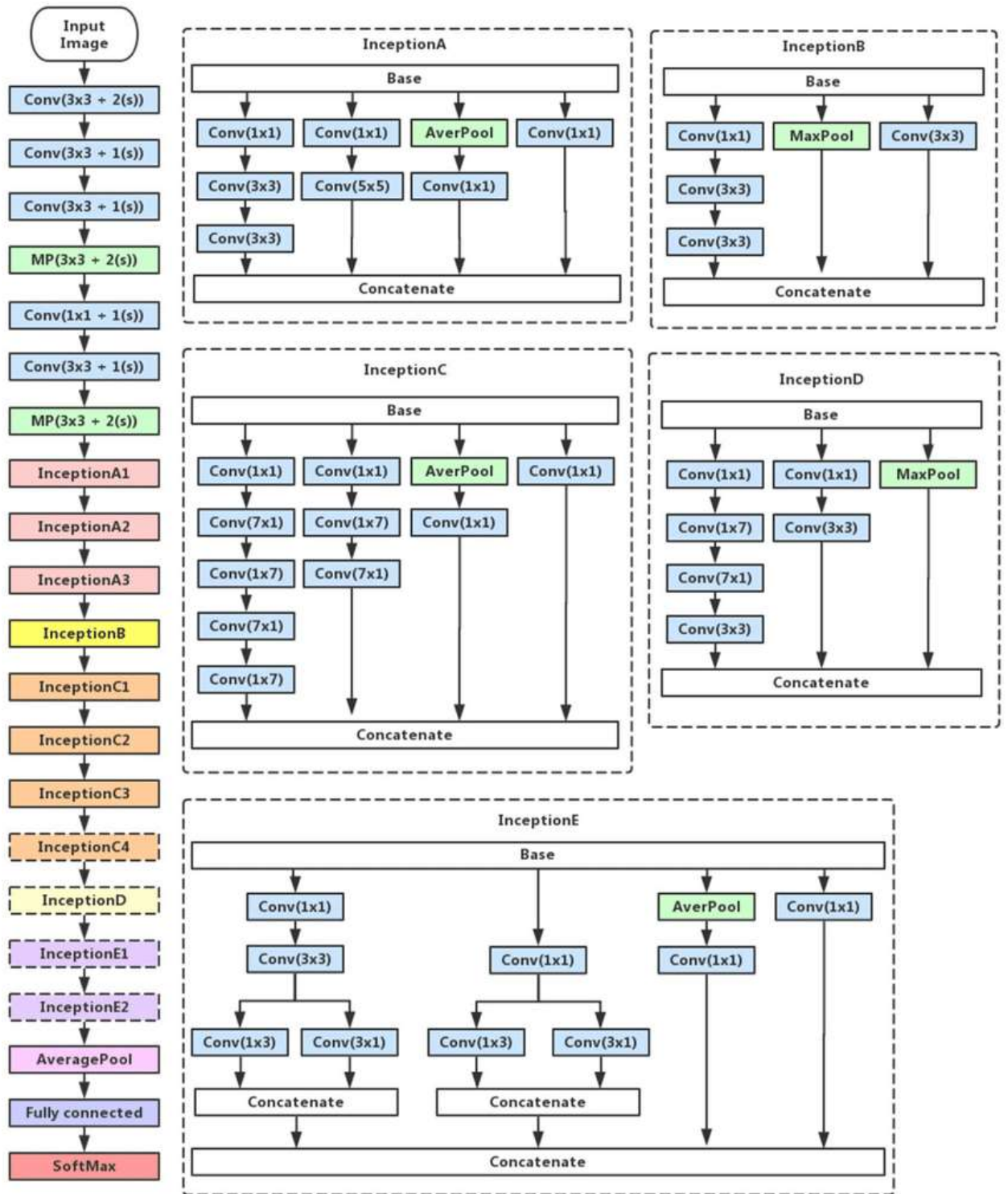


Figure-4.3.4(c): Flowchart of our proposed Inception v3 model

4.3.5 GoogleNet

It creates deeper architecture by employing a variety of techniques such as 11 convolution and global average pooling. Some of these strategies will be discussed in the architecture. At the end of the GoogleNet design, a method known as global average pooling is employed. This layer takes a feature map with a value of 77% and averages it to a value of 11%. This also reduces the number of trainable parameters to zero and boosts top-1 accuracy by 0.6 percent. Each layer in this architecture has a set convolution size.

33%, 55%, and 33% max pooling are conducted in parallel at the input of the Inception module, and the outputs of these are layered together to give the final result. The concept

The idea behind convolution filters of different sizes will handle objects at multiple scales better.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 32)	896
conv2d_1 (Conv2D)	(None, 252, 252, 64)	18496
max_pooling2d (MaxPooling2D)	(None, 126, 126, 64)	0
dropout (Dropout)	(None, 126, 126, 64)	0
conv2d_2 (Conv2D)	(None, 124, 124, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 64)	0
dropout_1 (Dropout)	(None, 62, 62, 64)	0
conv2d_3 (Conv2D)	(None, 60, 60, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 128)	0
dropout_2 (Dropout)	(None, 30, 30, 128)	0
flatten (Flatten)	(None, 115200)	0
dense (Dense)	(None, 64)	7372864
dropout_3 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65
Total params: 7,503,105		
Trainable params: 7,503,105		

Figure4.3.5(a): Layer by Layer architectural details of GoogleNet

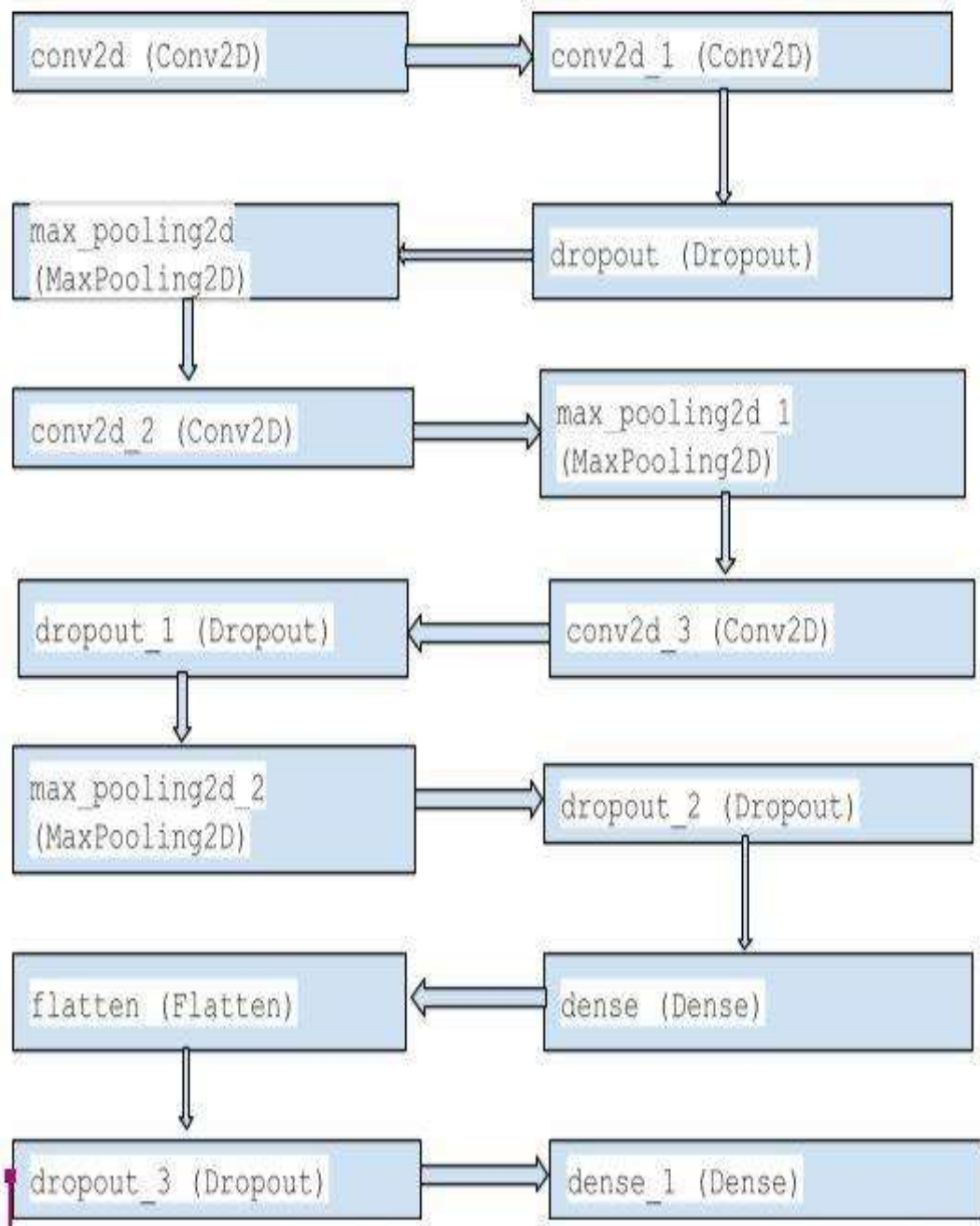


Figure-4.3.5(b): Flowchart of our proposed GoogleNet model

Dropout Regularization with dropout ratio = 0.25 when activation layer is ReLu

Dropout Regularization with dropout ratio = 0.50 when activation layer is sigmoid

The number of epochs we used in our model is 8 and steps per epoch is 2. The optimizer we used in our model is 'Adam' as it is best among the adaptive optimizer and also more suitable for our model. We train our model by giving it two types of datasets. One is covid positive and the other one is negative. As we used two types of datasets that's why our class mode is binary.

```
Epoch 1/10
8/8 [=====] - 60s 8s/step - loss: 0.6902 - accuracy: 0.5234 - val_loss: 0.6906 - val_accuracy: 0.5667
Epoch 2/10
8/8 [=====] - 59s 7s/step - loss: 0.6884 - accuracy: 0.5234 - val_loss: 0.6740 - val_accuracy: 0.8167
Epoch 3/10
8/8 [=====] - 60s 8s/step - loss: 0.6022 - accuracy: 0.7109 - val_loss: 0.5296 - val_accuracy: 0.7333
Epoch 4/10
8/8 [=====] - 59s 7s/step - loss: 0.4740 - accuracy: 0.7734 - val_loss: 0.4060 - val_accuracy: 0.9333
Epoch 5/10
8/8 [=====] - 59s 7s/step - loss: 0.4149 - accuracy: 0.8047 - val_loss: 0.2695 - val_accuracy: 0.9167
Epoch 6/10
8/8 [=====] - 59s 7s/step - loss: 0.2704 - accuracy: 0.8750 - val_loss: 0.2179 - val_accuracy: 0.9667
Epoch 7/10
8/8 [=====] - 60s 8s/step - loss: 0.2536 - accuracy: 0.9141 - val_loss: 0.2217 - val_accuracy: 0.9000
Epoch 8/10
8/8 [=====] - 59s 7s/step - loss: 0.2425 - accuracy: 0.8984 - val_loss: 0.1116 - val_accuracy: 0.9667
Epoch 9/10
8/8 [=====] - 59s 7s/step - loss: 0.2693 - accuracy: 0.8906 - val_loss: 0.3117 - val_accuracy: 0.9167
Epoch 10/10
8/8 [=====] - 58s 7s/step - loss: 0.2309 - accuracy: 0.9453 - val_loss: 0.0988 - val_accuracy: 0.9833
```

Figure-4.3.5(c): Number of epochs, accuracy, losses, validation accuracy and validation losses for our **GoogleNet** model

Here, we can clearly notice that the maximum accuracy is 94% while the maximum validation accuracy is 98%. We also discern the minimum loss is 23% while the minimum validation loss is 9%.

$$\begin{aligned}\text{Accuracy} &= (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \\ &= (20+22)/(20+22+3+1) \\ &= 42/46 \\ &= 0.91\end{aligned}$$

$$\begin{aligned}\text{Sensitivity} &= \text{TP}/(\text{TP}+\text{FN}) \\ &= 20/(20+1) \\ &= 0.95\end{aligned}$$

$$\begin{aligned}
\text{Specificity} &= \text{TN}/(\text{TN}+\text{FP}) \\
&= 22/(22+3) \\
&= 22/25 \\
&= 0.88
\end{aligned}$$

4.3.6 VGG16

A fixed size 224 by 224 image with three channels – R, G, and B – is regarded as the input to any of the network configurations. The only pre-processing done is to normalize each pixel's RGB values. Every pixel is subtracted from the mean value to achieve this. Following ReLu activations, information is sent through a first stack of two convolution layers with a very small receptive size of 3×3 . There are 64 filters in each of these two layers. The padding is 1 pixel, while the convolution stride is fixed at 1 pixel. The spatial resolution is preserved in this arrangement, and the output activation map is the same size as the input image dimensions.

After that, the activation maps are passed through. The activation maps are then run via spatial max pooling with a stride of 2 pixels over a 2×2 -pixel window. The size of the activations is reduced by half. As a result, the activations at the bottom of the first stack are $112 \times 112 \times 64$ in size.

The Keras architecture for creating the VGG16 model The Keras Applications library also includes a VGG16 model that has already been trained. The ImageNet weights are included in the pre-trained model. We may employ transfer learning methods to train on your custom images while using the pre-trained model. But first, let's go over how we can construct it from the ground up.

In Keras, we created a sequential model object:

```
Sequential = model ()
```

Two convolutional layers with 64 filters of size 3×3 are followed by a 2×2 max-pooling layer with stride 2 in this stack. In addition, the size of the input image is set to $224 \times 224 \times 3$.

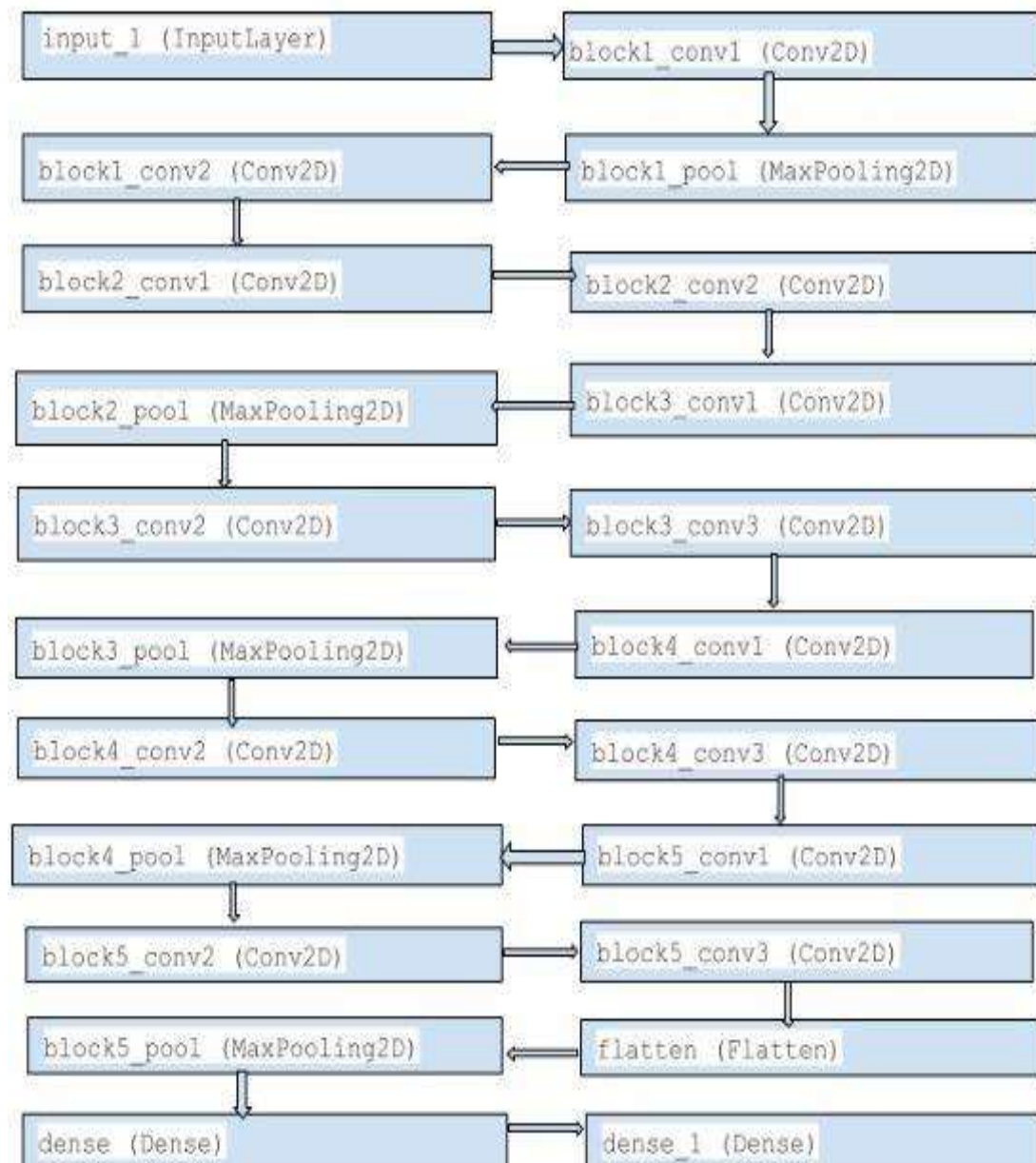


Figure-4.3.6(a): Flowchart of our proposed VGG16 model

The number of epochs we used in our model is 8 and steps per epoch is 2. The optimizer we used in our model is ‘Adam’ as it is best among the adaptive optimizer and also more suitable for our model. We train our model by giving it two types of datasets. One is covid positive and the other one is negative. As we used two types of datasets that’s why our class mode is binary.

```
Epoch 1/8
5/5 [=====] - 107s 23s/step - loss: 0.2536 - accuracy: 0.8938 - val_loss: 0.0552 - val_accuracy: 0.9833
Epoch 2/8
5/5 [=====] - 104s 22s/step - loss: 0.0439 - accuracy: 0.9875 - val_loss: 0.0508 - val_accuracy: 0.9833
Epoch 3/8
5/5 [=====] - 107s 23s/step - loss: 0.0147 - accuracy: 1.0000 - val_loss: 0.0330 - val_accuracy: 0.9667
Epoch 4/8
5/5 [=====] - 106s 22s/step - loss: 0.0068 - accuracy: 1.0000 - val_loss: 0.0357 - val_accuracy: 0.9833
Epoch 5/8
5/5 [=====] - 104s 22s/step - loss: 0.0083 - accuracy: 1.0000 - val_loss: 0.0305 - val_accuracy: 0.9833
Epoch 6/8
5/5 [=====] - 104s 22s/step - loss: 0.0041 - accuracy: 1.0000 - val_loss: 0.0343 - val_accuracy: 0.9833
Epoch 7/8
5/5 [=====] - 105s 22s/step - loss: 0.0115 - accuracy: 0.9937 - val_loss: 0.0480 - val_accuracy: 0.9833
Epoch 8/8
5/5 [=====] - 106s 22s/step - loss: 0.0012 - accuracy: 1.0000 - val_loss: 0.0746 - val_accuracy: 0.9833
```

Figure-4.3.6(b): Number of epochs, accuracy, losses, validation accuracy and validation losses for our **Vgg16** model

Here, we can clearly notice that the maximum accuracy is 100% while the maximum validation accuracy is 98%. We also discern the minimum loss is 0.1% while the minimum validation loss is 3%.

For calculating accuracy, specificity and sensitivity we just used two simple equations where we used value from our confusion matrix. Now the equations are given below:

$$\begin{aligned}\text{Accuracy} &= (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \\ &= (20+19)/(20+19+10+11) \\ &= 39/60 \\ &= 0.65\end{aligned}$$

$$\begin{aligned}\text{Sensitivity} &= \text{TP}/(\text{TP}+\text{FN}) \\ &= 20/(20+11) \\ &= 20/31 \\ &= 0.64\end{aligned}$$

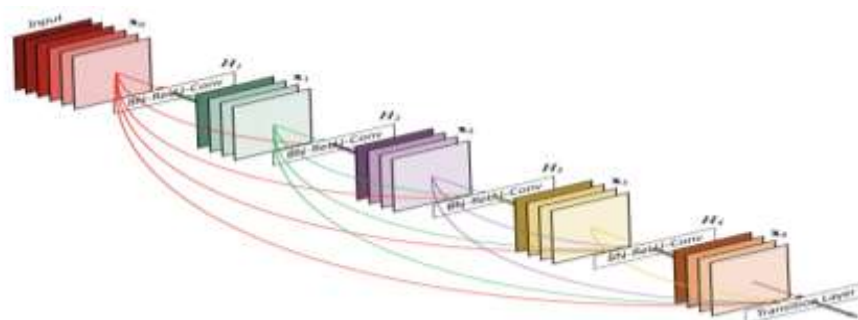
$$\begin{aligned}
\text{Specificity} &= \text{TN}/(\text{TN}+\text{FP}) \\
&= 19/(19+10) \\
&= 19/29 \\
&= 0.65
\end{aligned}$$

4.3.7 DenseNet

DenseNet was created specifically to increase accuracy in high-level neural networks caused by the vanishing gradient, which occurs when information vanishes before reaching its destination due to the great distance between input and output layers. There will be roughly L and L plus one by two connections $L(L+1)/2$ in a DenseNet. So, because a dense net has less layers than the other model, we can easily train more than 100 layers of the model using this technique. As we get farther into the network, this becomes unsustainable, and a feature map explosion occurs. To solve this problem, we establish a dense block here, with a predetermined number of layers inside each dense block. as well as the result and the output from that dense block is passed to a transition layer, which is like one-by-one convolution followed by Max pooling to lower the size of the dense block.

The maps of the features as a result, the transition layer allows for Max pooling, which reduces the size of your feature maps.

Structure of DenseNet is given below:



DenseNet Structure

$$a^{[l]} = g([a^{[0]}, a^{[1]}, a^{[2]}, \dots, a^{[l-1]}])$$

Figure-4.3.7(a): Structure of DenseNet

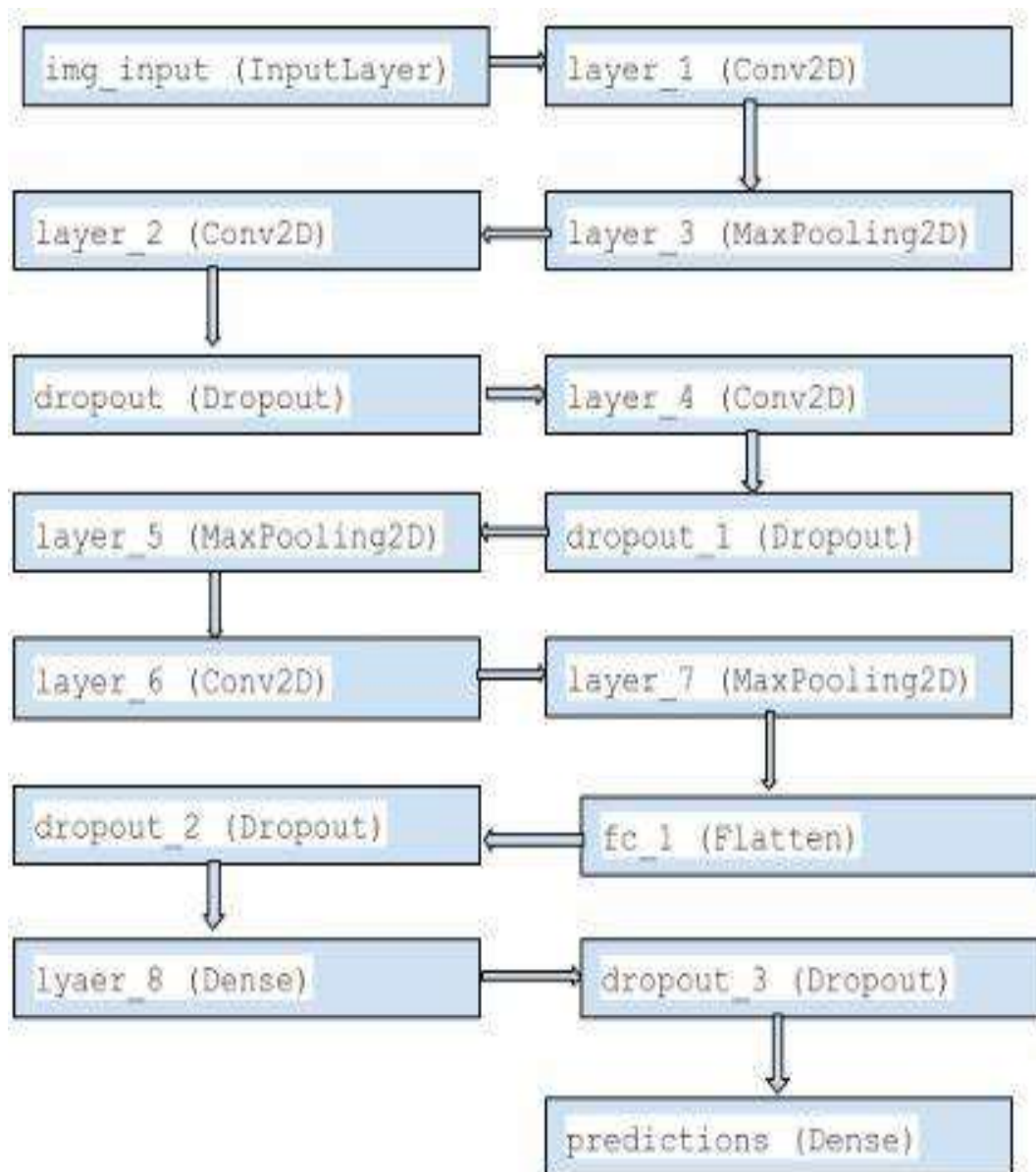


Figure-4.3.7(b): Flowchart of DenseNet model

The number of epochs we used in our model is 8 and steps per epoch is 2. The optimizer we used in our model is ‘Adam’ as it is best among the adaptive optimizer and also more suitable for our model. We train our model by giving it two types of datasets. One is covid positive and the other one is negative. As we used two types of datasets that’s why our class mode is binary.

```
Epoch 1/8
7/7 [=====] - 87s 12s/step - loss: 1.3751 - accuracy: 0.6875 - val_loss: 0.6643 - val_accuracy: 0.7812
Epoch 2/8
7/7 [=====] - 74s 11s/step - loss: 0.4242 - accuracy: 0.8482 - val_loss: 0.3355 - val_accuracy: 0.9062
Epoch 3/8
7/7 [=====] - 75s 11s/step - loss: 0.3825 - accuracy: 0.8571 - val_loss: 0.4434 - val_accuracy: 0.8438
Epoch 4/8
7/7 [=====] - 74s 11s/step - loss: 0.3281 - accuracy: 0.8616 - val_loss: 0.2746 - val_accuracy: 1.0000
Epoch 5/8
7/7 [=====] - 80s 12s/step - loss: 0.2086 - accuracy: 0.9018 - val_loss: 0.1651 - val_accuracy: 0.9688
Epoch 6/8
7/7 [=====] - 80s 11s/step - loss: 0.1321 - accuracy: 0.9509 - val_loss: 0.1465 - val_accuracy: 0.9375
Epoch 7/8
7/7 [=====] - 74s 11s/step - loss: 0.1355 - accuracy: 0.9554 - val_loss: 0.1357 - val_accuracy: 0.9688
Epoch 8/8
7/7 [=====] - 73s 10s/step - loss: 0.1012 - accuracy: 0.9598 - val_loss: 0.0565 - val_accuracy: 1.0000
```

Figure-4.3.7(c): Number of epochs, accuracy, losses, validation accuracy and validation losses for our **DenseNet** model

Here, we can clearly notice that the maximum accuracy is 95% while the maximum validation accuracy is 100%. We also discern the minimum loss is 10% while the minimum validation loss is 5%.

For calculating accuracy, specificity and sensitivity we just used two simple equations where we used value from our confusion matrix. Now the equations are given below:

$$\begin{aligned}\text{Accuracy} &= (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \\ &= (13+12) / (13+12+17+18) \\ &= 25/60 \\ &= 0.41\end{aligned}$$

$$\begin{aligned}\text{Sensitivity} &= \text{TP} / (\text{TP} + \text{FN}) \\ &= 13 / (13+30) \\ &= 0.30\end{aligned}$$

$$\begin{aligned}\text{Specificity} &= \text{TN} / (\text{TN} + \text{FP}) \\ &= 12 / (12+17) \\ &= 12/29 \\ &= 0.41\end{aligned}$$

CHAPTER 5

Result and Discussion

In this chapter we will describe several tables and plots which contain maximum accuracy, minimum loss, confusion matrix, ROC curve, Classification report of our proposed methods.

5.1 VGG19

Now we are going to show the loss curve and accuracy curve for our VGG19 model:

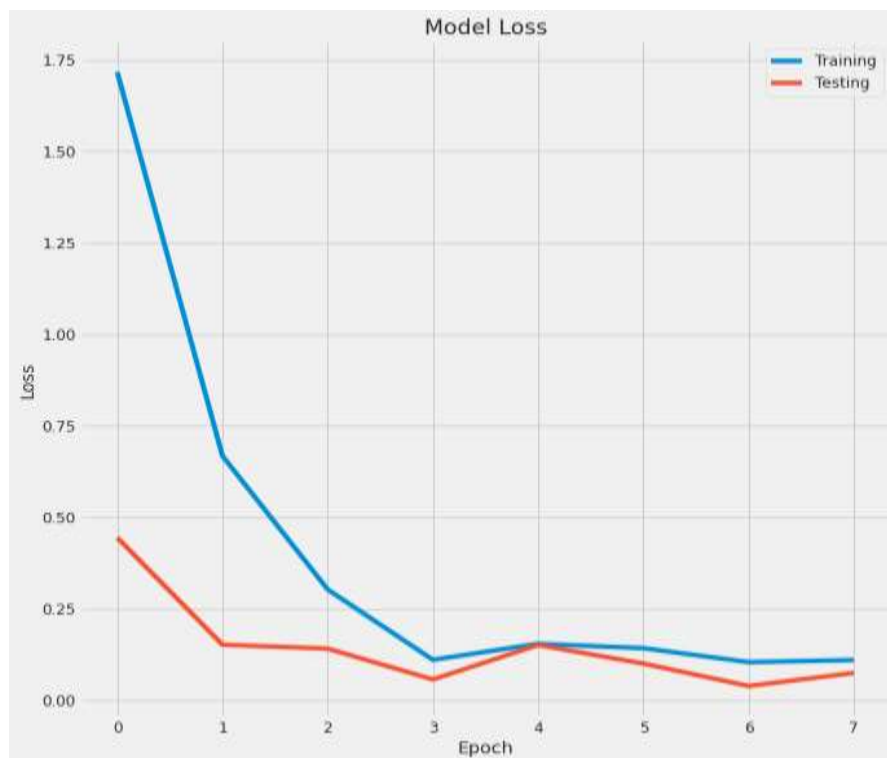


Figure-5.1 (a): Here we can see the difference between train loss and validation loss for our given dataset.

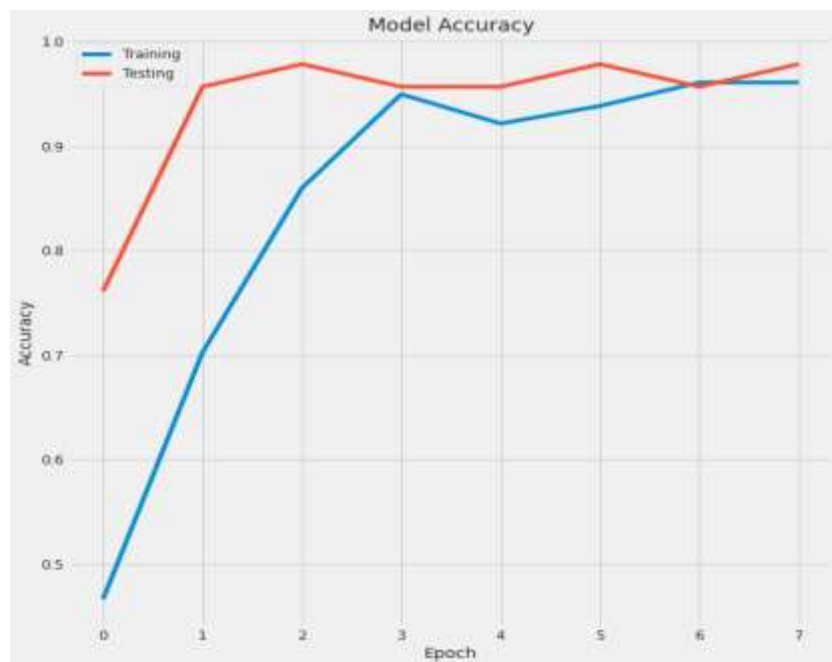


Figure-5.1(b): Here we can see the difference between train accuracy and validation accuracy for our given dataset.

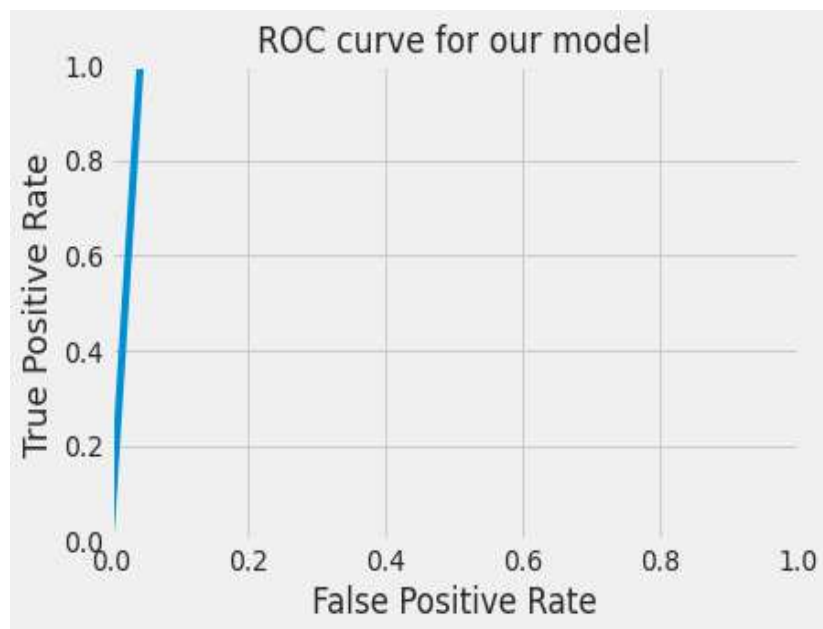


Figure-5.1(c): Here we can see the ROC curve for VGG19 model where x axis denotes false positive rate and y axis denotes true positive rate.

Now we are going to show the classification report for our VGG19 model:

	precision	recall	f1-score	No of test images
Covid positive	1.00	0.96	0.98	23
normal	0.96	1.00	0.98	23
accuracy			0.98	46
macro avg	0.98	0.98	0.98	46
weighted avg	0.98	0.98	0.98	46

Table-5.1: Classification report for our VGG19 model.

Confusion matrix for VGG19 model is given below

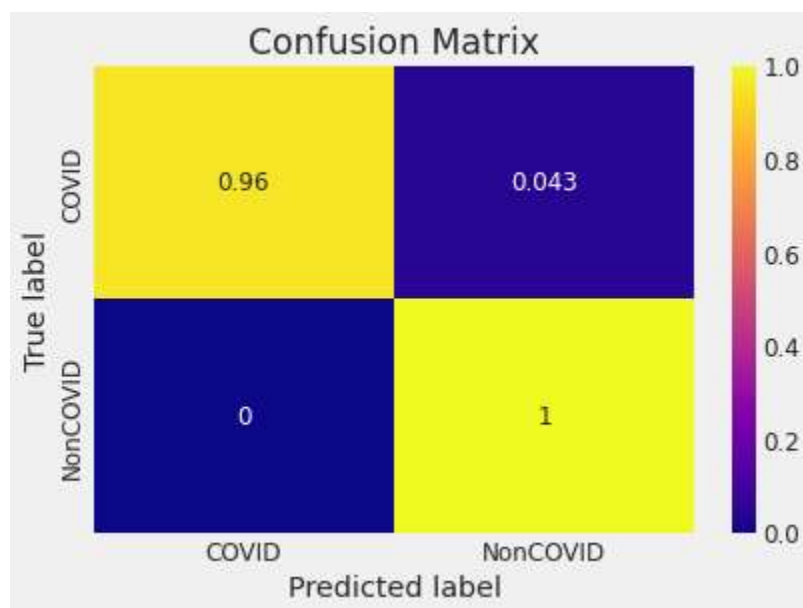


Figure-5.1(d): Confusion Matrix with Normalized Values

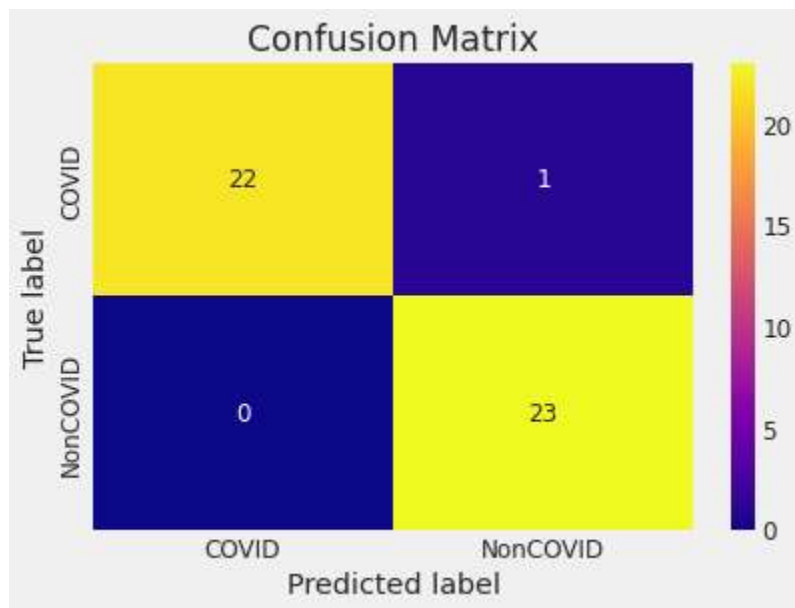


Figure-5.1(e): Confusion Matrix without Normalization

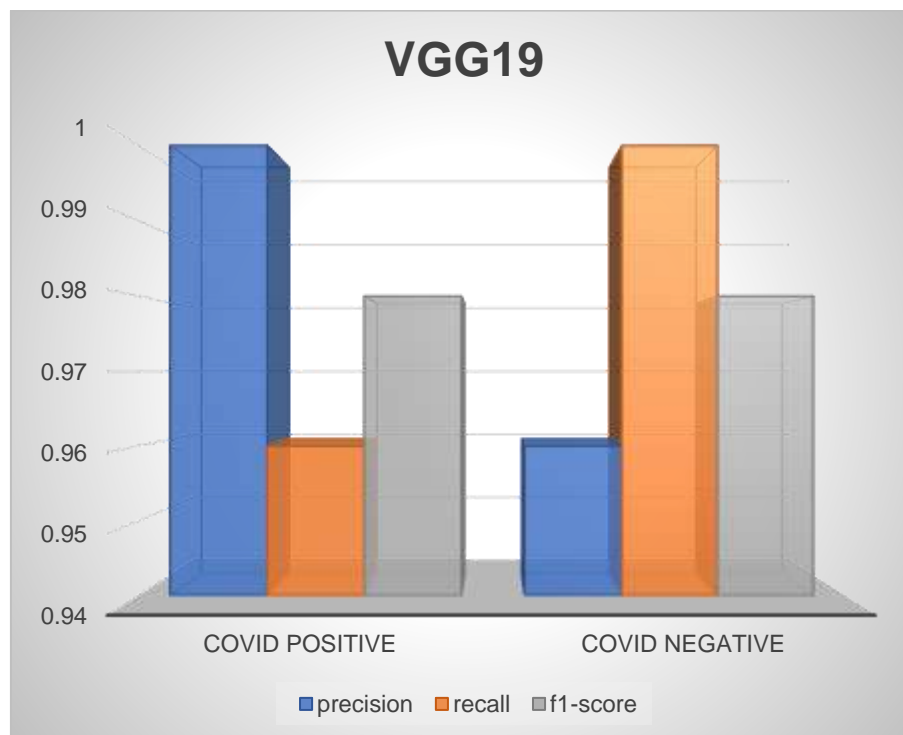


Figure-5.1(f): Class activation report for VGG19

5.2 VGG16 model

Now we are going to show the loss curve and accuracy curve for our VGG16 model:

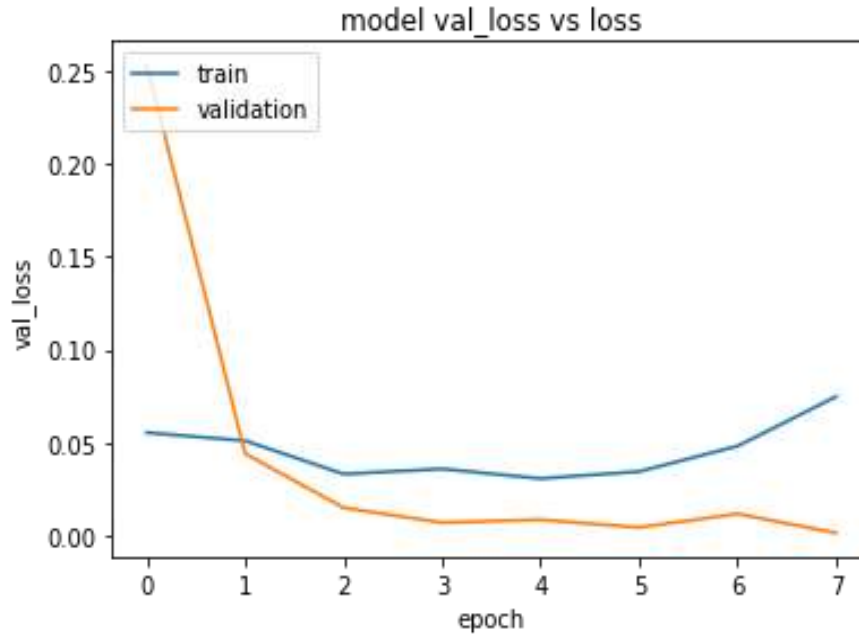


Figure-5.2 (a): Here we can see the difference between train loss and validation loss for our given dataset.

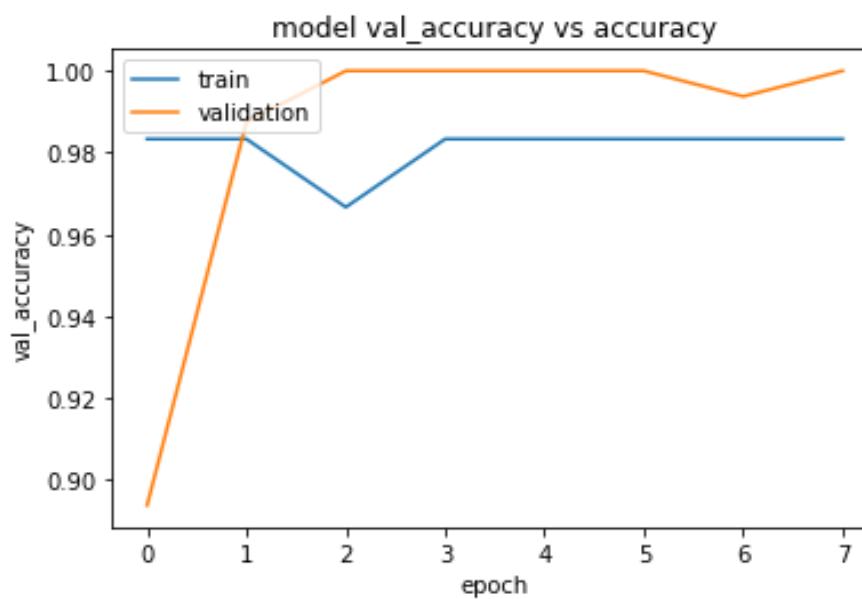


Figure-5.2 (b): Here we can see the difference between train accuracy and validation accuracy for our given dataset.

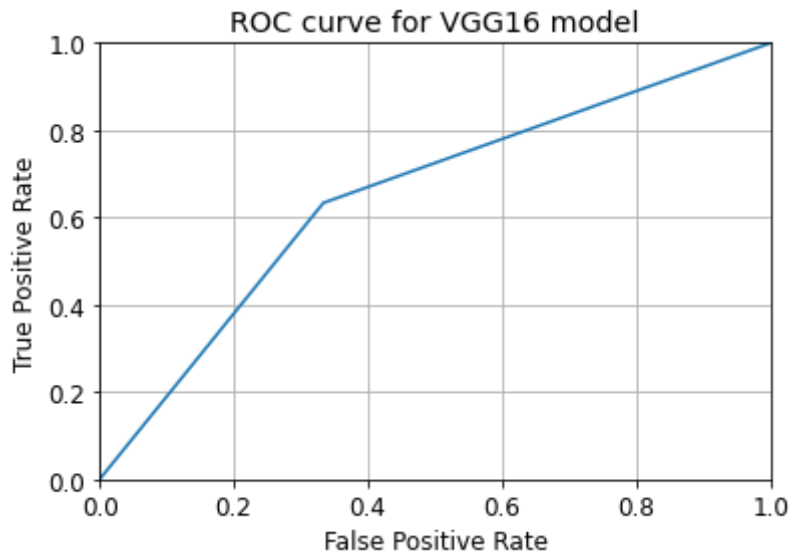


Figure-5.2(c): Here we can see the ROC curve for VGG16 model where x axis denotes false positive rate and y axis denote true positive rate.

Now we are going to show the classification report for our VGG16 model:

	Precision	Recall	f1-score	No of test images
COVID+	0.65	0.67	0.66	30
COVID-	0.66	0.63	0.64	30
Accuracy			0.65	60
Macro avg	0.65	0.65	0.65	60
Weighted avg	0.65	0.65	0.65	60

Table-5.2: Classification report for our VGG19 model.

Confusion matrix for VGG16 model is given below

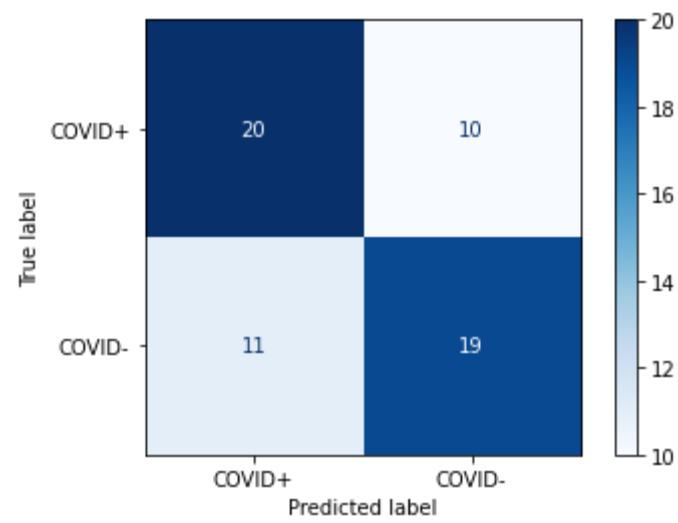


Figure-5.2(d): Confusion Matrix with Normalized Values

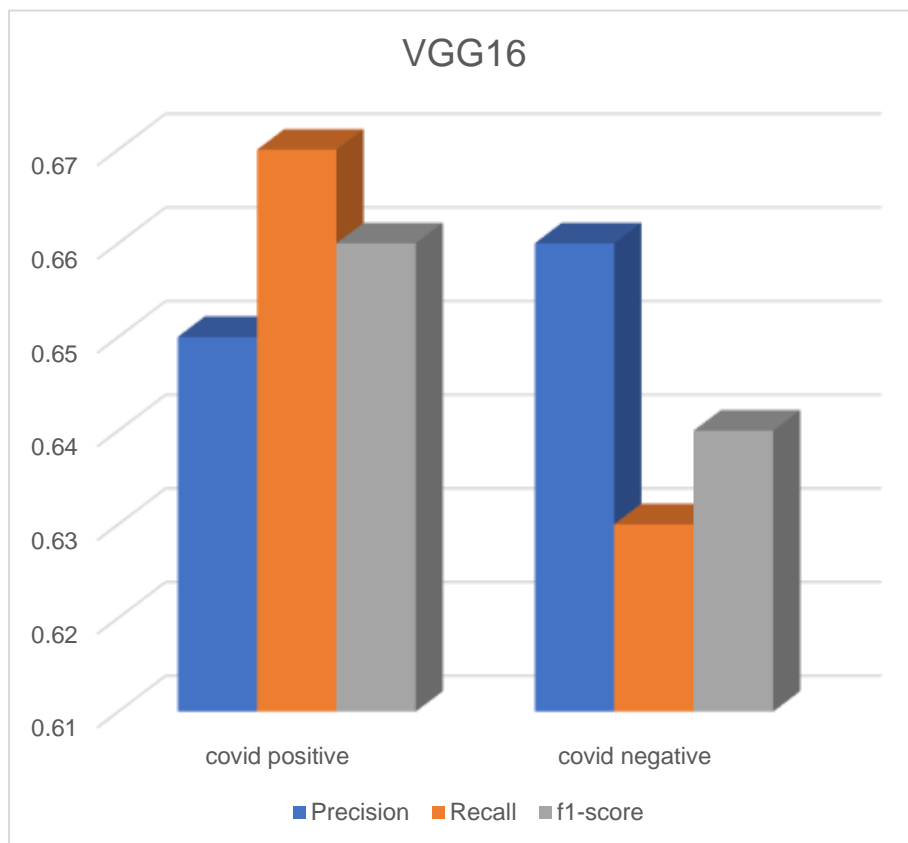


Figure-5.2(e): Class activation report for VGG16

5.3 ResNet50

Now we are going to show the loss curve and accuracy curve for our ResNet50 model:

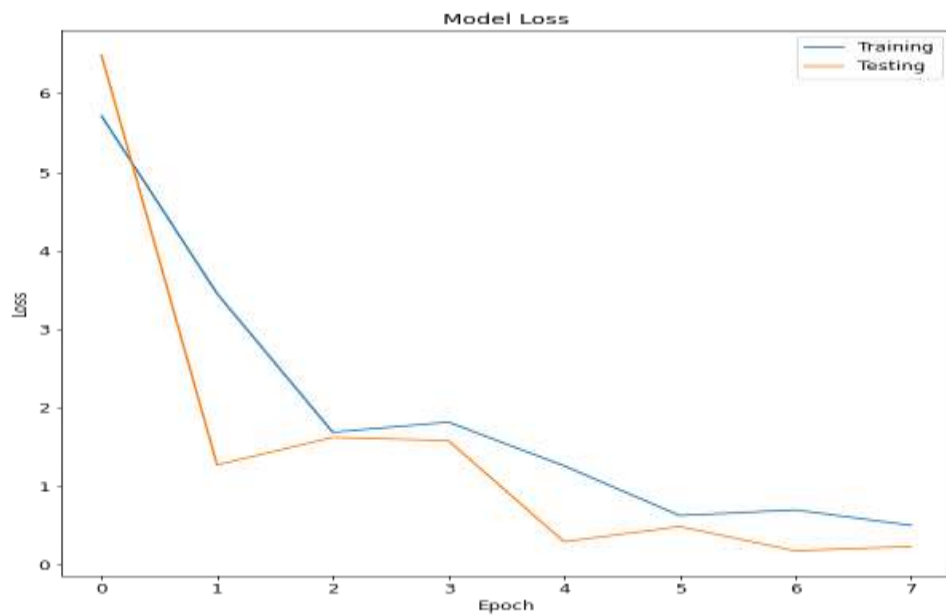


Figure-5.3 (a): Here we can see the difference between train loss and validation loss for our given dataset.

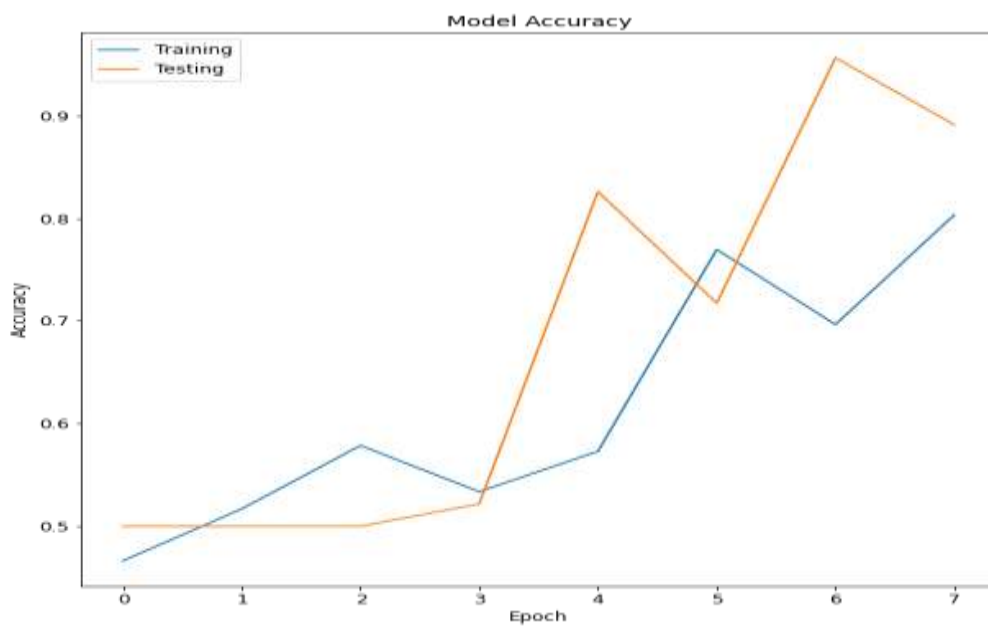


Figure-5.3 (b): Here we can see the difference between train accuracy and validation accuracy for our given dataset.

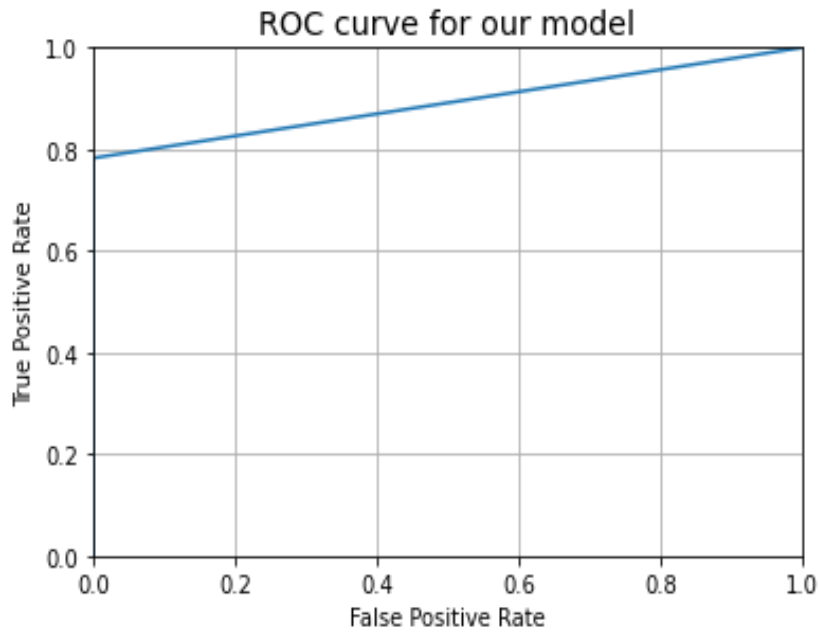


Figure-5.3(c): Here we can see the ROC curve for ResNet50 model where x axis denotes false positive rate and y axis denote true positive rate.

Now we are going to show the classification report for our ResNet50 model:

	Precision	Recall	f1-score	No of test images
COVID+	0.82	1.00	0.90	23
COVID-	1.00	0.78	0.88	23
Accuracy			0.89	46
Macro avg	0.91	0.89	0.89	46
Weighted avg	0.91	0.89	0.89	46

Table-5.3: Classification report for our ResNet50 model.

Confusion matrix for ResNet50 model is given below:

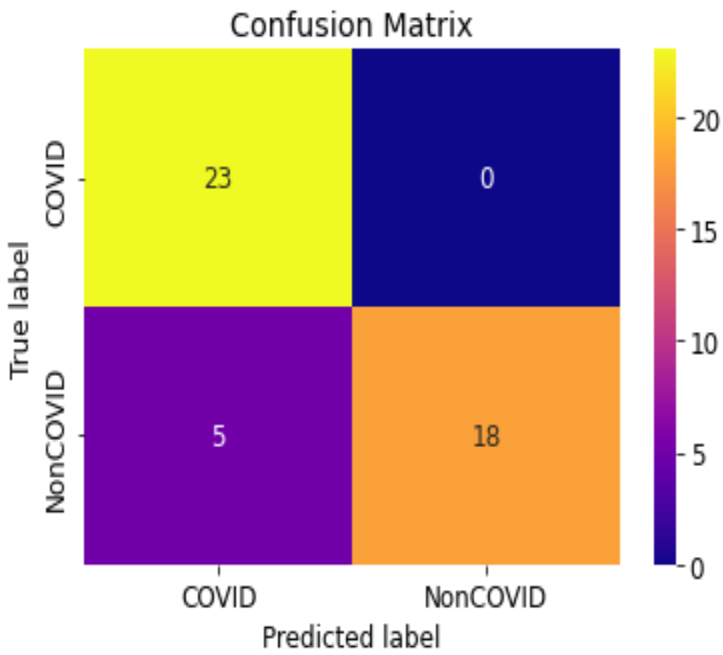


Figure-5.3(d): Confusion Matrix without Normalization

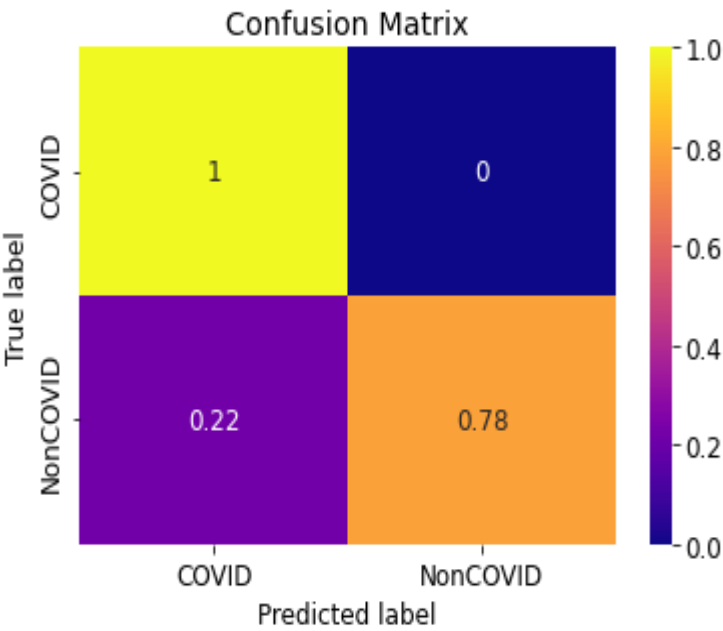


Figure-5.3(e): Confusion Matrix with Normalized Values

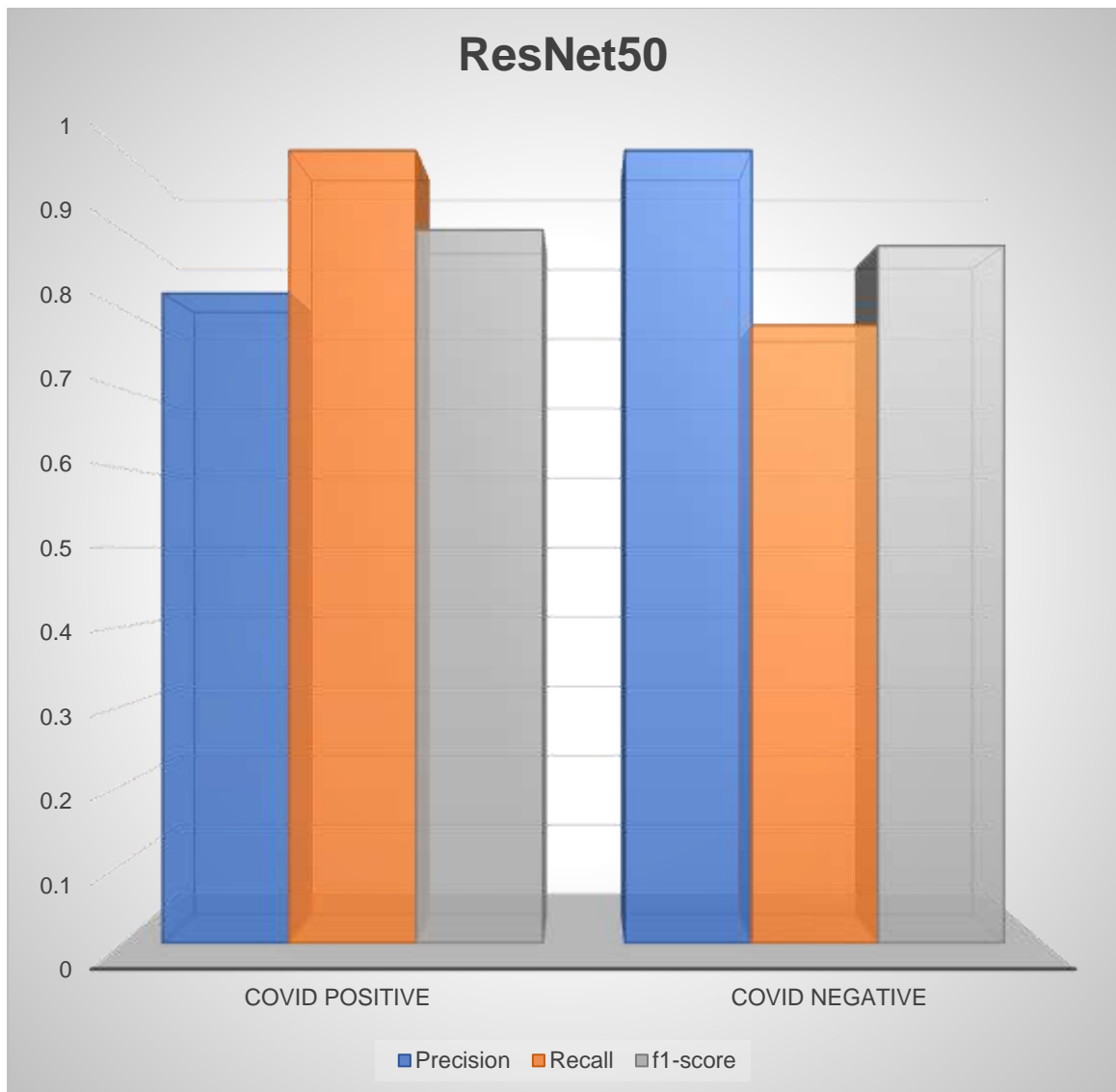


Figure-5.3(f): Class activation report for ResNet50

5.4 Inception3

Now we are going to show the loss curve and accuracy curve for our Inception V3 model:

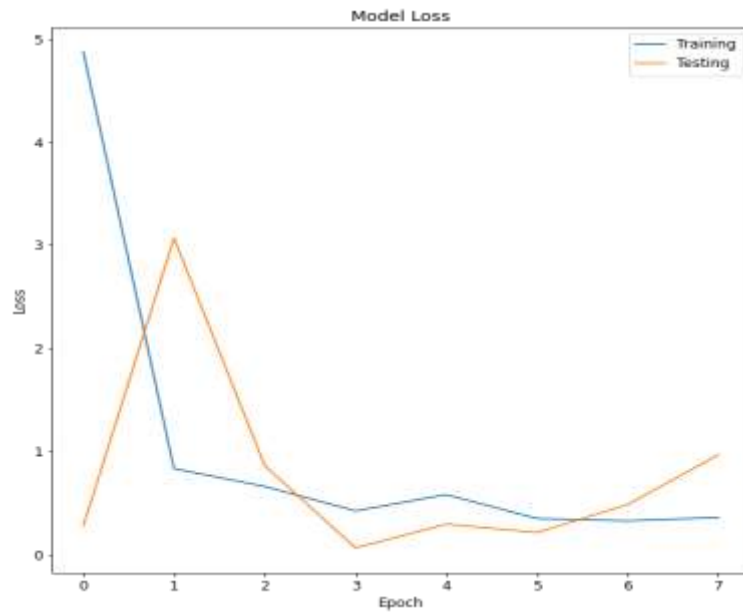


Figure-5.4 (a): Here we can see the difference between train loss and validation loss for our given dataset.

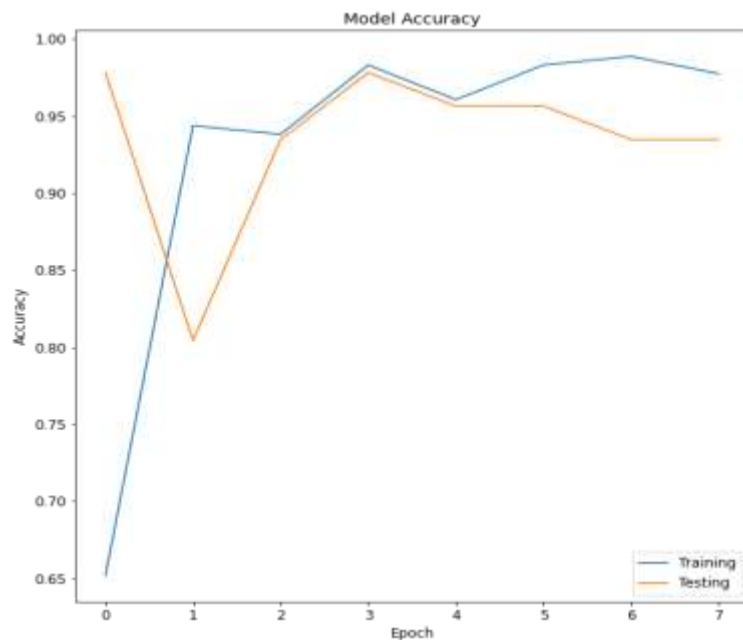


Figure-5.4 (b): Here we can see the difference between train accuracy and validation accuracy for our given dataset.

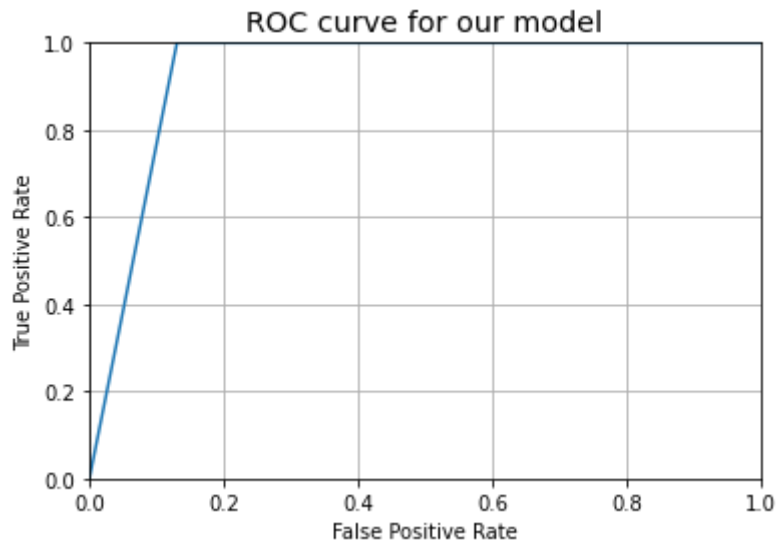


Figure-5.4(c): Here we can see the ROC curve for Inception v3 model where x axis denotes false positive rate and y axis denote true positive rate.

Now we are going to show the classification report for our Inception v3 model:

	Precision	Recall	f1-score	No of test images
COVID+	1.00	0.87	0.93	23
COVID-	0.88	1.00	0.94	23
Accuracy			0.93	46
Macro avg	0.94	0.93	0.93	46
Weighted avg	0.94	0.93	0.93	46

Table-5.4: Classification report for our Inception v3 model

Confusion matrix for Inception v3 model is given below

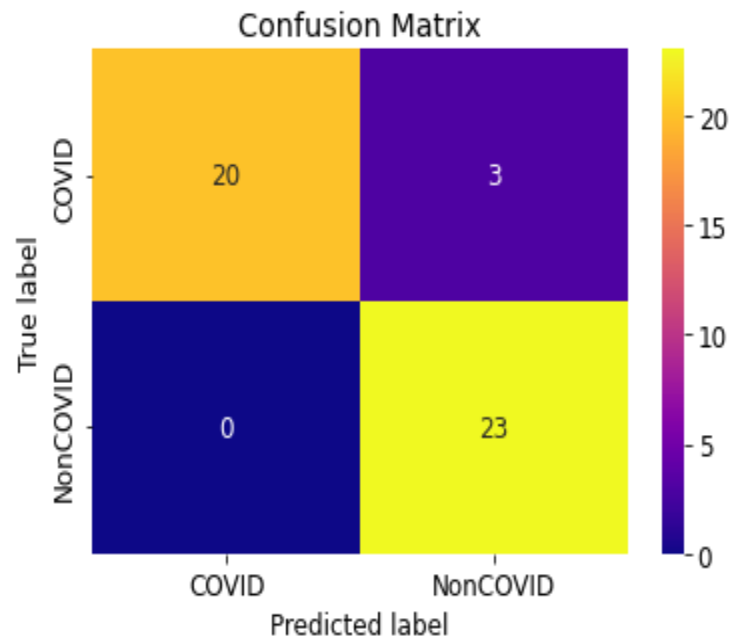


Figure-5.4(d): Confusion Matrix without Normalization

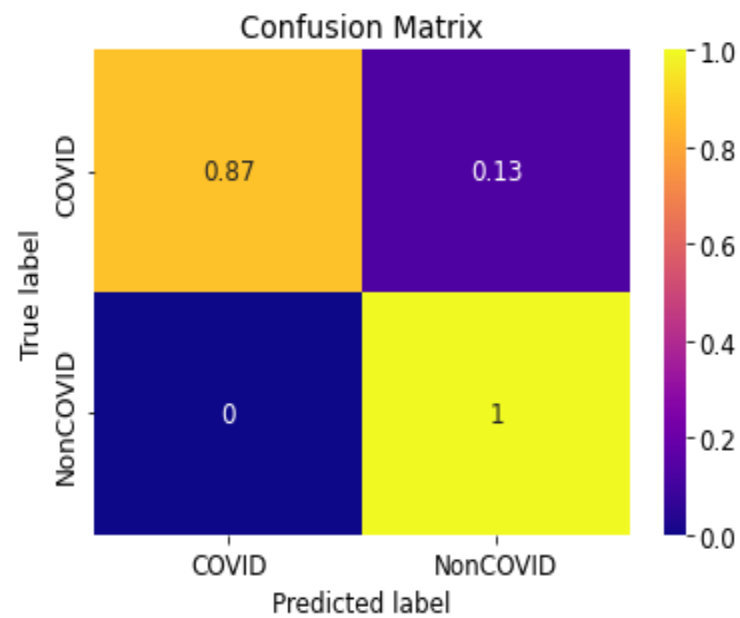


Figure-5.4(e): Confusion Matrix with Normalized Values

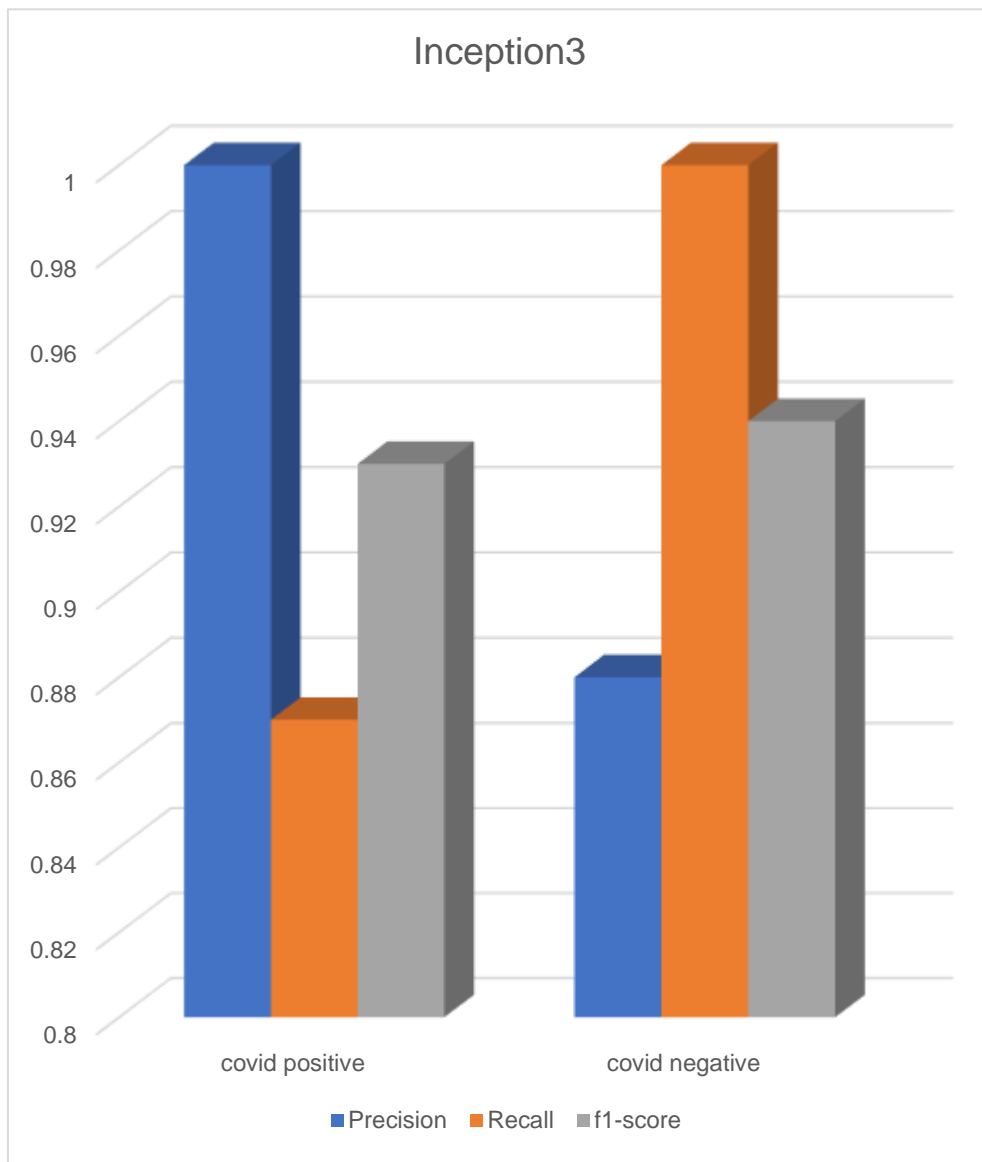


Figure-5.4(f): Class activation report for Inception3

5.5 GoogleNet

Now we are going to show the loss curve and accuracy curve for our Google Net model:

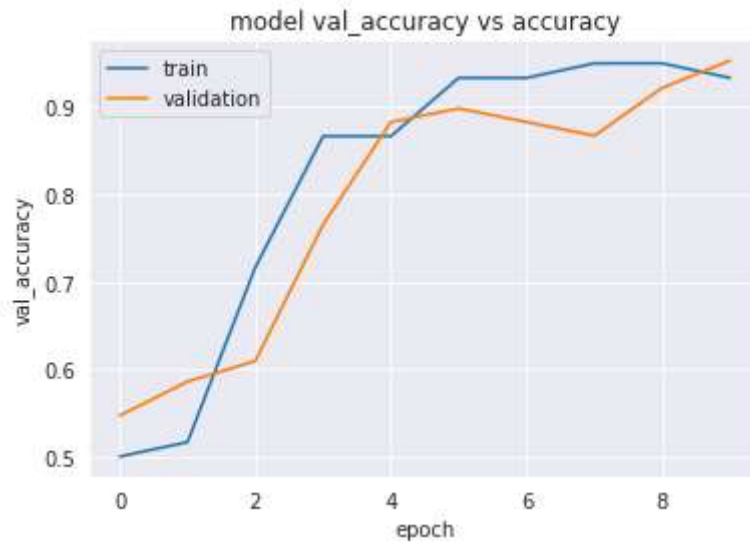


Figure-5.5 (a): Here we can see the difference between train accuracy and validation accuracy for our given dataset.

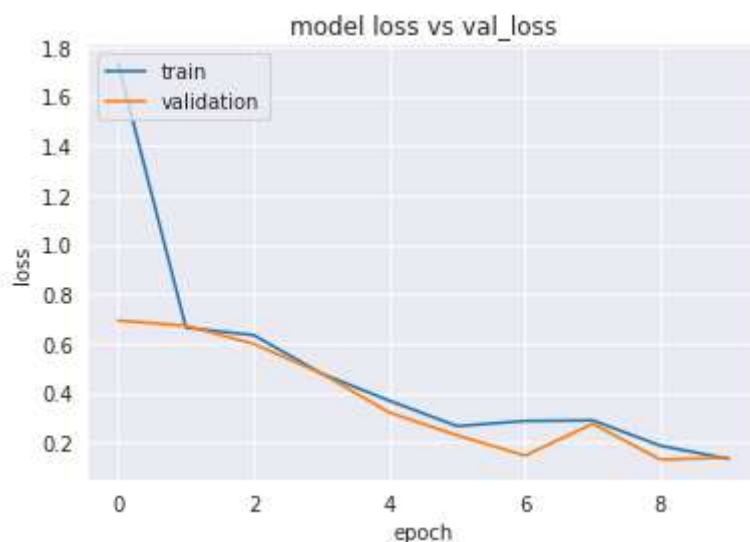


Figure-5.5 (b): Here we can see the difference between train loss and validation loss for our given dataset.

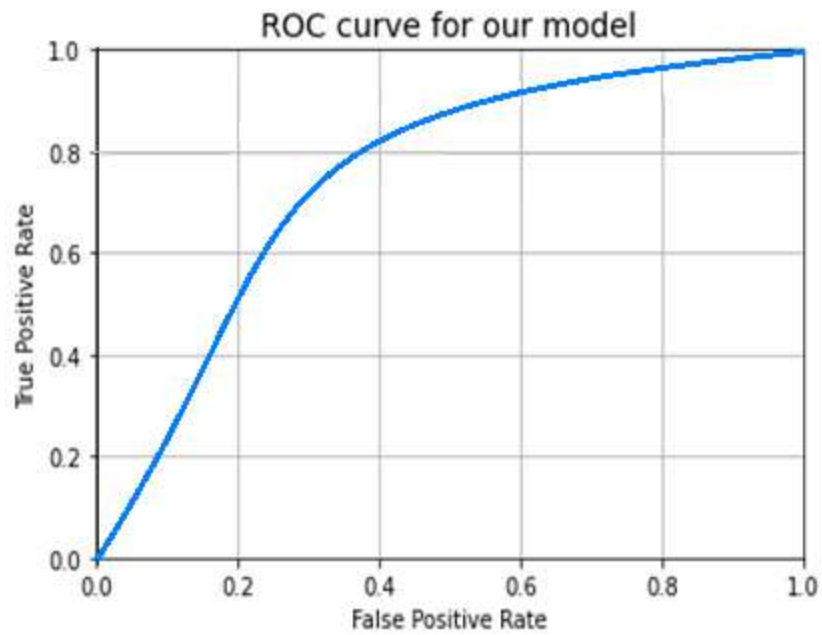


Figure-5.5(c): Here we can see the ROC curve for Inception v3 model where x axis denotes false positive rate and y axis denote true positive rate.

Now we are going to show the classification report for our GoogleNet model:

	Precision	Recall	f1-score	No of test images
COVID+	0.95	0.88	0.91	23
COVID-	0.95	0.88	0.91	23
Accuracy			0.91	46
Macro avg	0.95	0.88	0.91	46
Weighted avg	0.95	0.88	0.91	46

Table-5.5: Classification report for our GoogleNet model.

Confusion matrix for GoogleNet model is given below

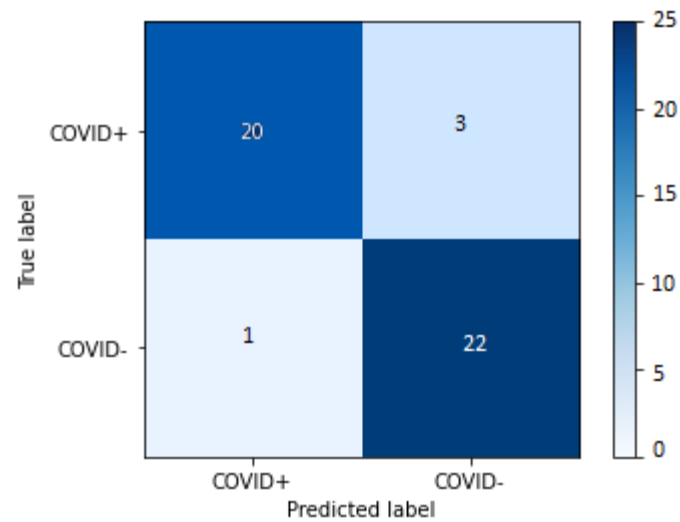


Figure-5.5(d): Confusion Matrix without Normalization

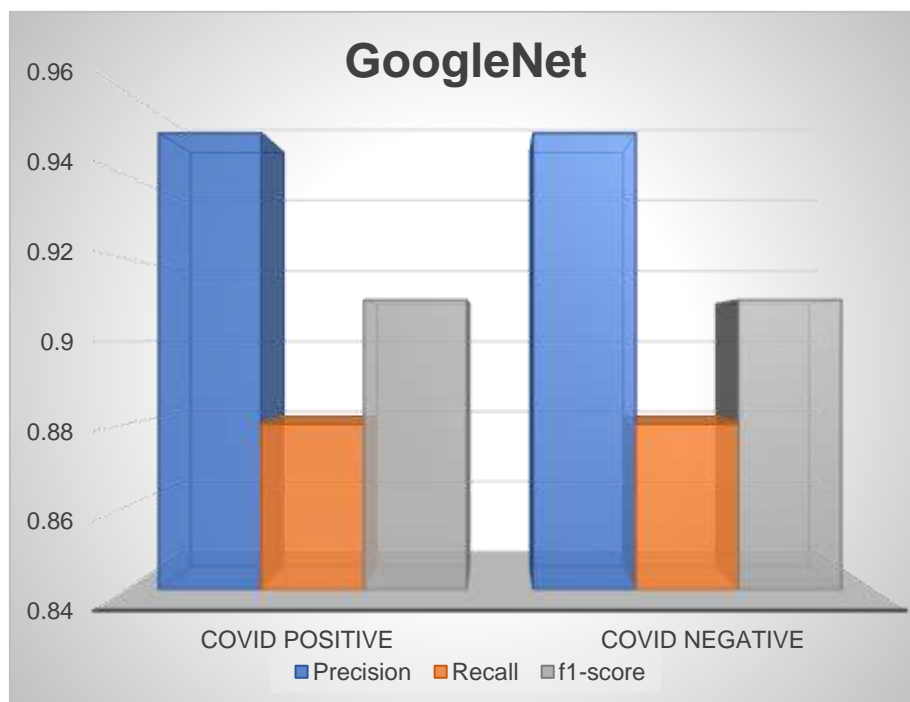


Figure-5.5(e): Classification report for GoogleNet

5.6 AlexNet

Now we are going to show the loss curve and accuracy curve for our AlexNet model:

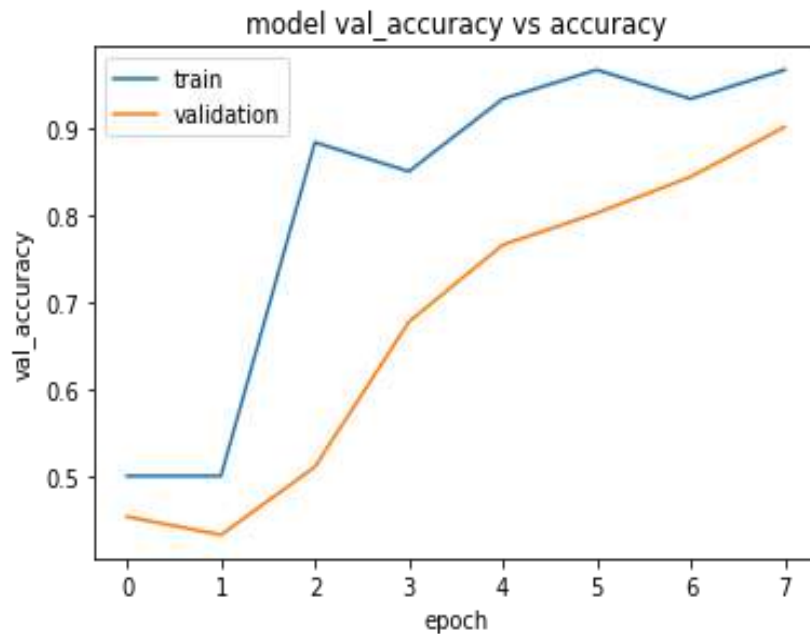


Figure-5.6 (a): Here we can see the difference between train accuracy and validation accuracy for our given dataset.

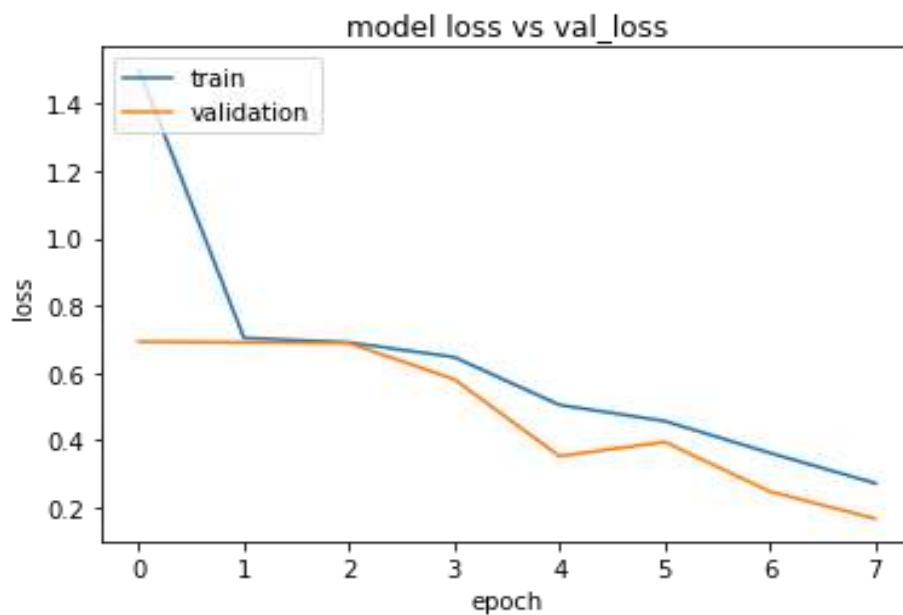


Figure-5.6 (b): Here we can see the difference between train loss and validation loss for our given dataset.

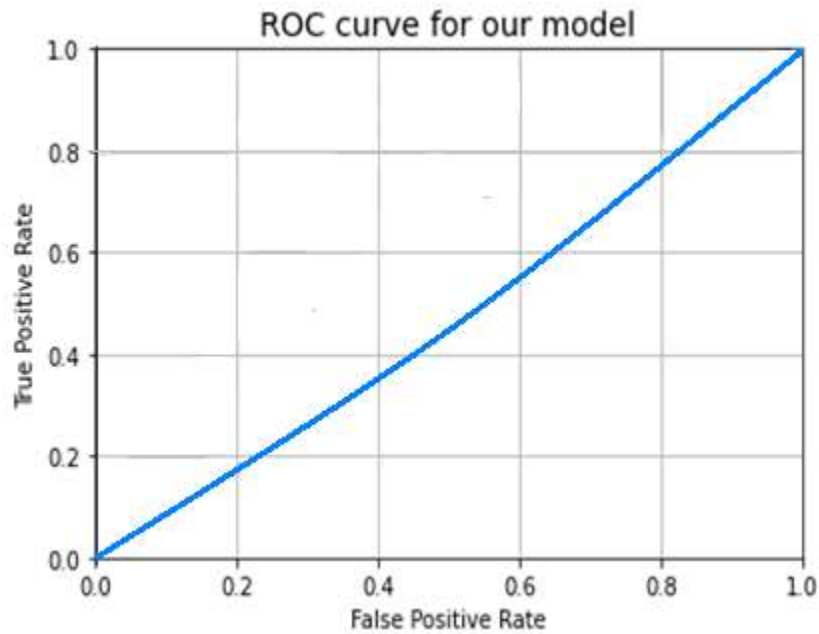


Figure-5.6(c): Here we can see the ROC curve for AlexNet model where x axis denotes false positive rate and y axis denote true positive rate.

Now we are going to show the classification report for our AlexNet model:

	Precision	Recall	f1-score	Support
COVID+	0.50	0.71	0.66	30
COVID-	0.50	0.70	0.65	15
Accuracy			0.66	45
Macro avg	0.25	0.71	0.66	45
Weighted avg	0.25	0.71	0.66	45

Table-5.6: Classification report for our AlexNet model.

Confusion matrix for AlexNet model is given below

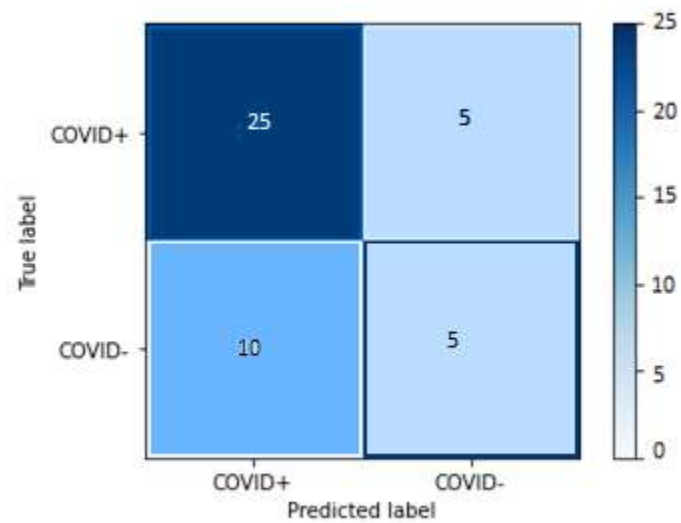


Figure-5.6(d): Confusion Matrix without Normalization

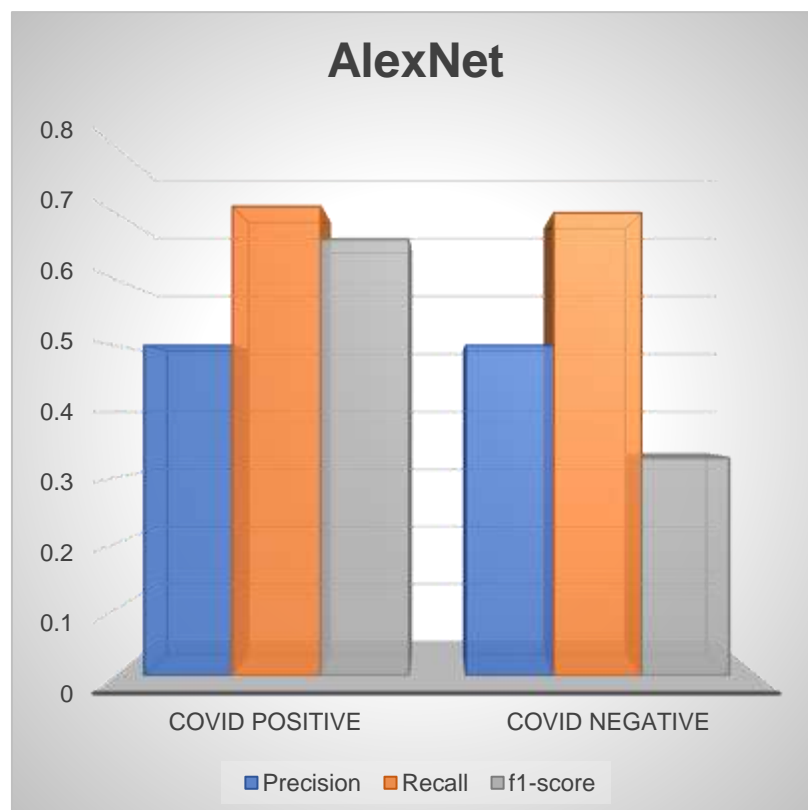


Figure-5.6(e): Classification report for AlexNet

5.7 DenseNet

Now we are going to show the loss curve and accuracy curve for our DenseNet model:

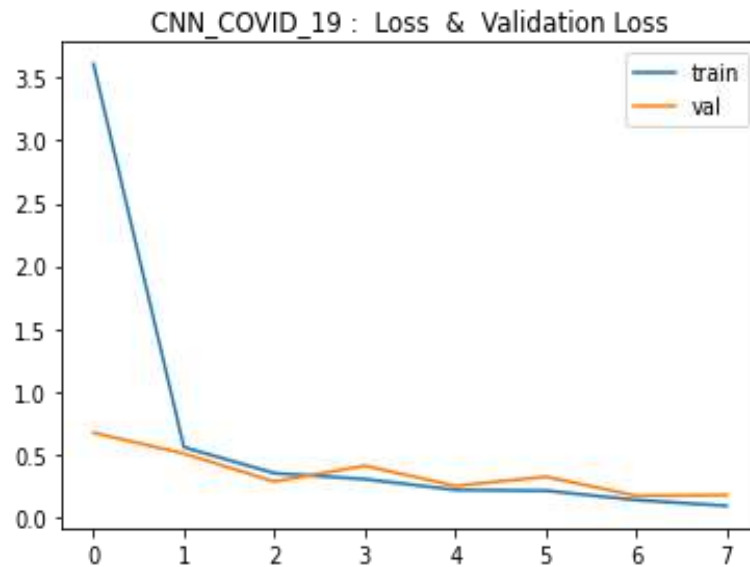


Figure-5.7 (a): Here we can see the difference between train loss and validation loss for our given dataset.

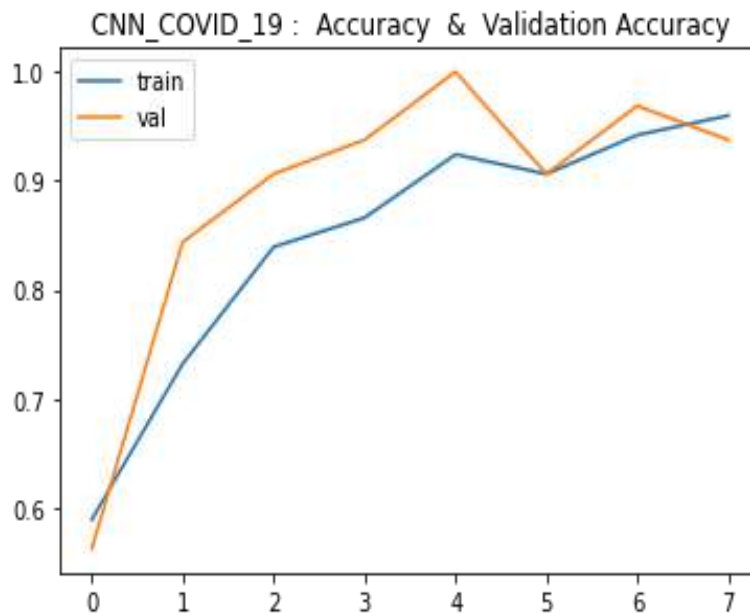


Figure-5.7 (b): Here we can see the difference between train accuracy and validation accuracy for our given dataset.

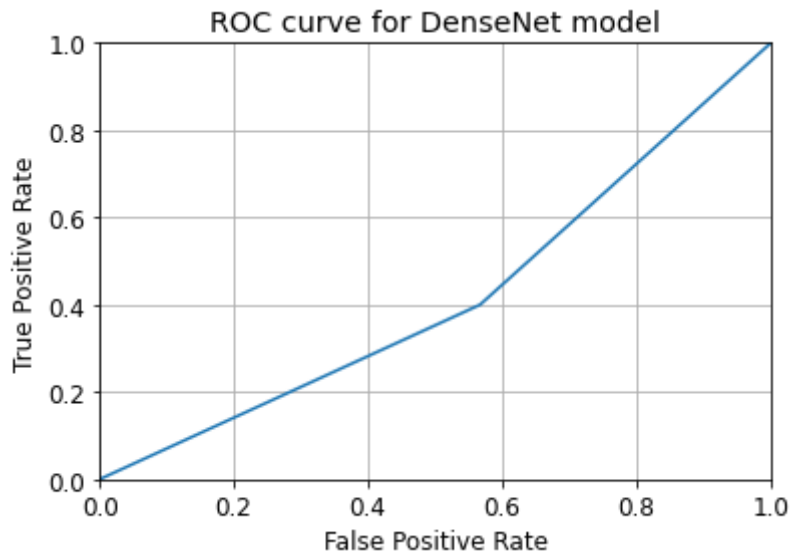


Figure-5.7(c): Here we can see the ROC curve for DenseNet model where x axis denotes false positive rate and y axis denote true positive rate.

Now we are going to show the classification report for our DenseNet model:

	Precision	Recall	f1-score	No of test images
COVID+	0.42	0.43	0.43	30
COVID-	0.41	0.40	0.41	30
Accuracy			0.42	60
Macro avg	0.42	0.42	0.42	60
Weighted avg	0.42	0.42	0.42	60

Table-5.7: Classification report for our DenseNet model.

Confusion matrix for DenseNet model is given below

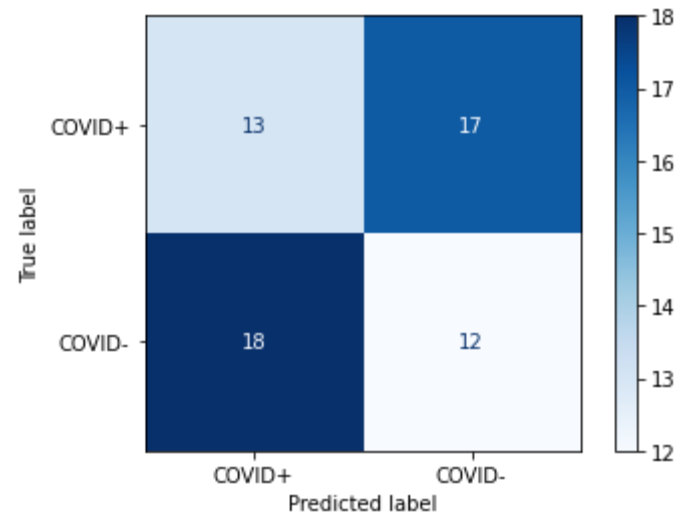


Figure-5.7(d): Confusion Matrix without Normalization

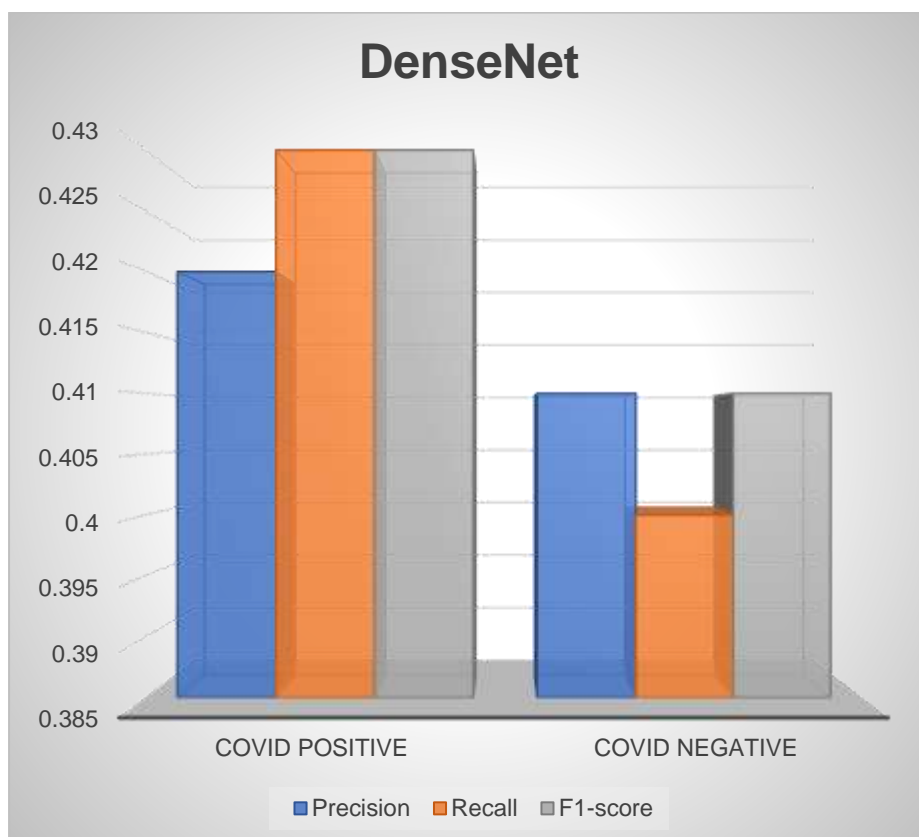


Figure-5.8(e): Visual representation of classification report of DenseNet.

5.8 Accuracy, Sensitivity and Specificity in all our proposed model

As we already showed how we calculated accuracy, sensitivity and specificity in all our proposed model in methodology now here we are just going to showed all the value in a tabular form for better understanding.

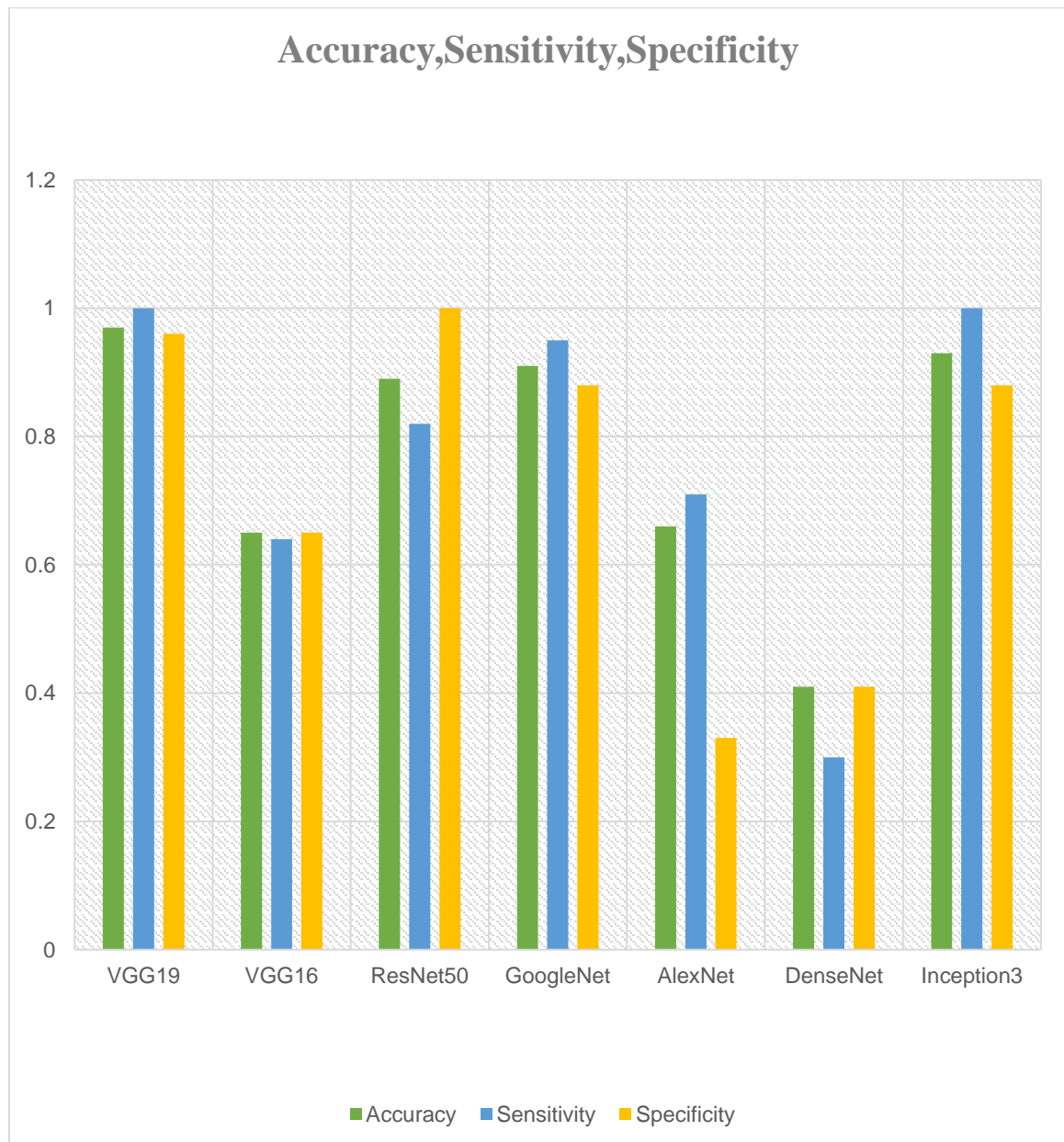


Figure-5.8: Visual representation of accuracy, sensitivity and specificity of different architecture

5.8.1 Final result

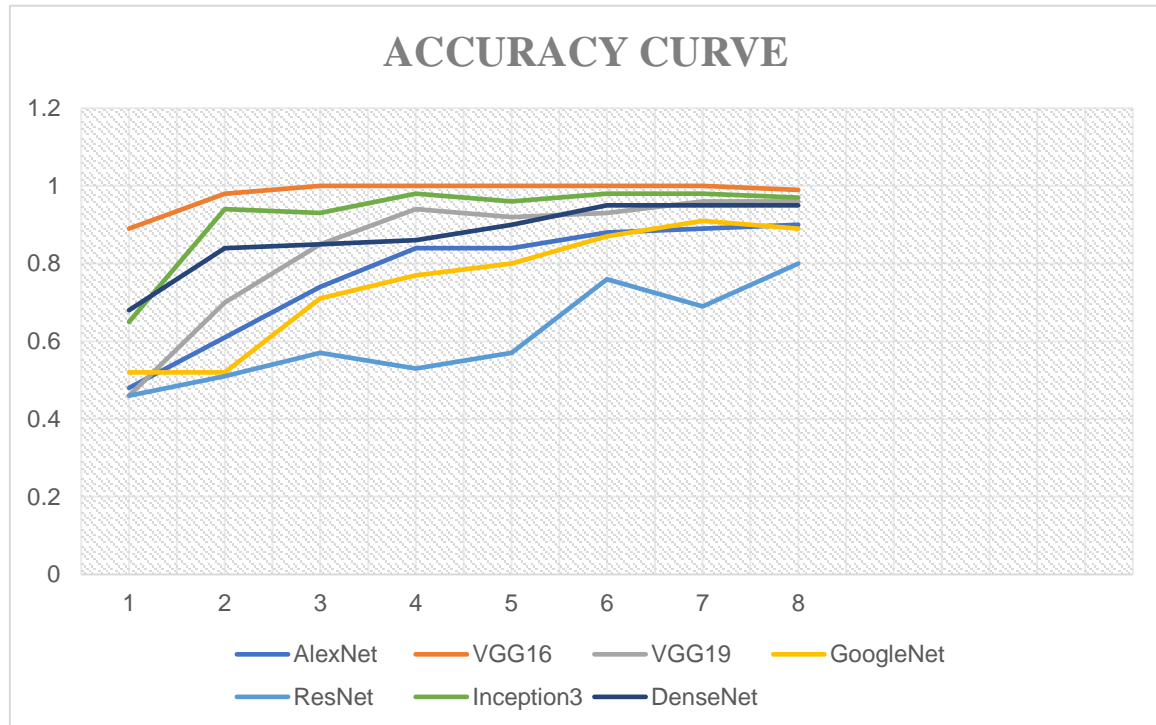


Figure-5.9(a) : Accuracy curve of different architecture.

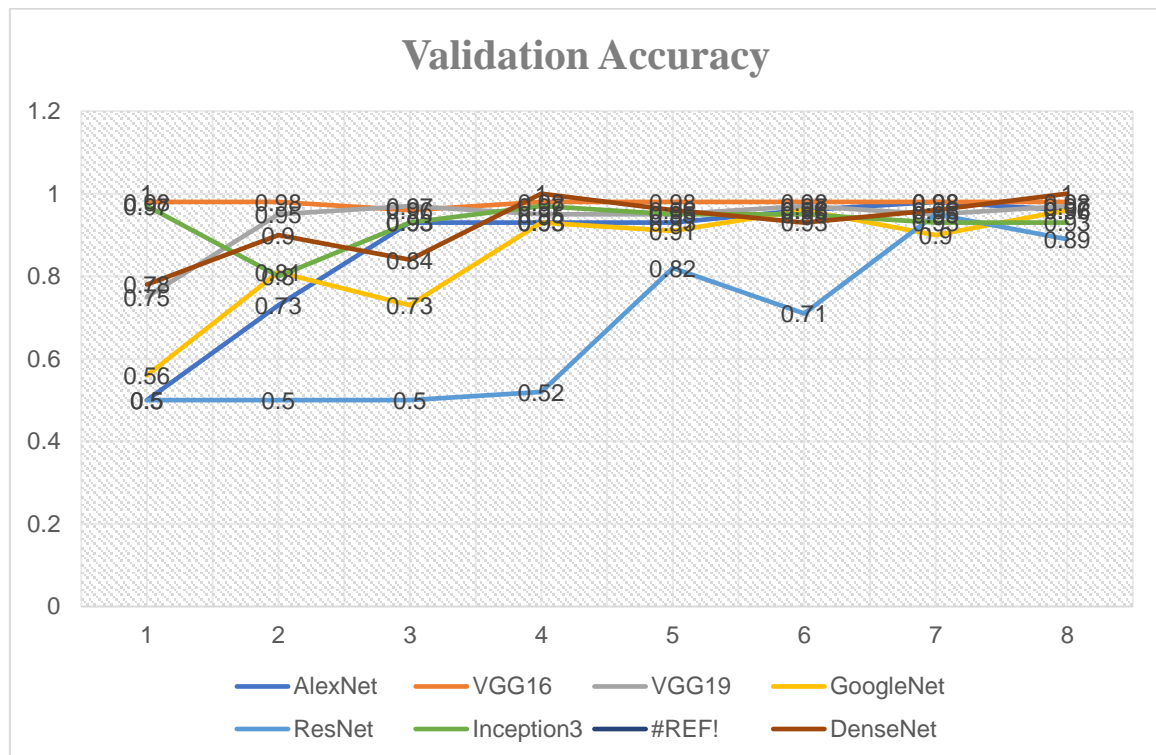


Figure-5.9(c) : Validation accuracy curve of different architecture.

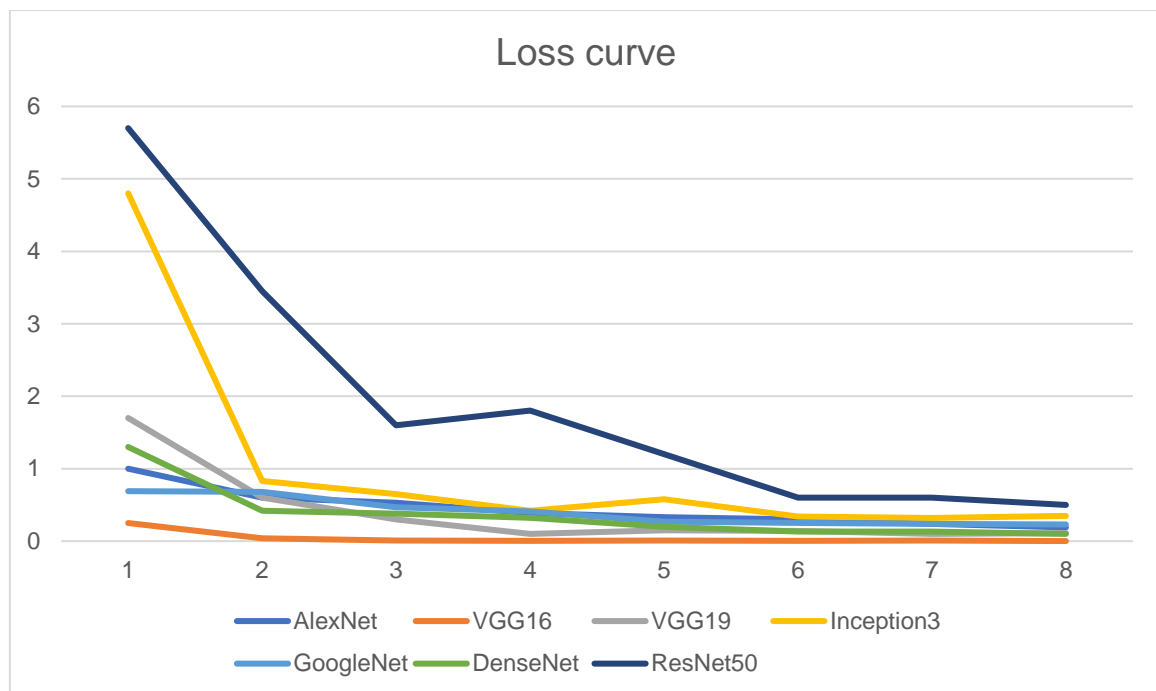


Figure-5.9(c) : Loss curve of different architecture.

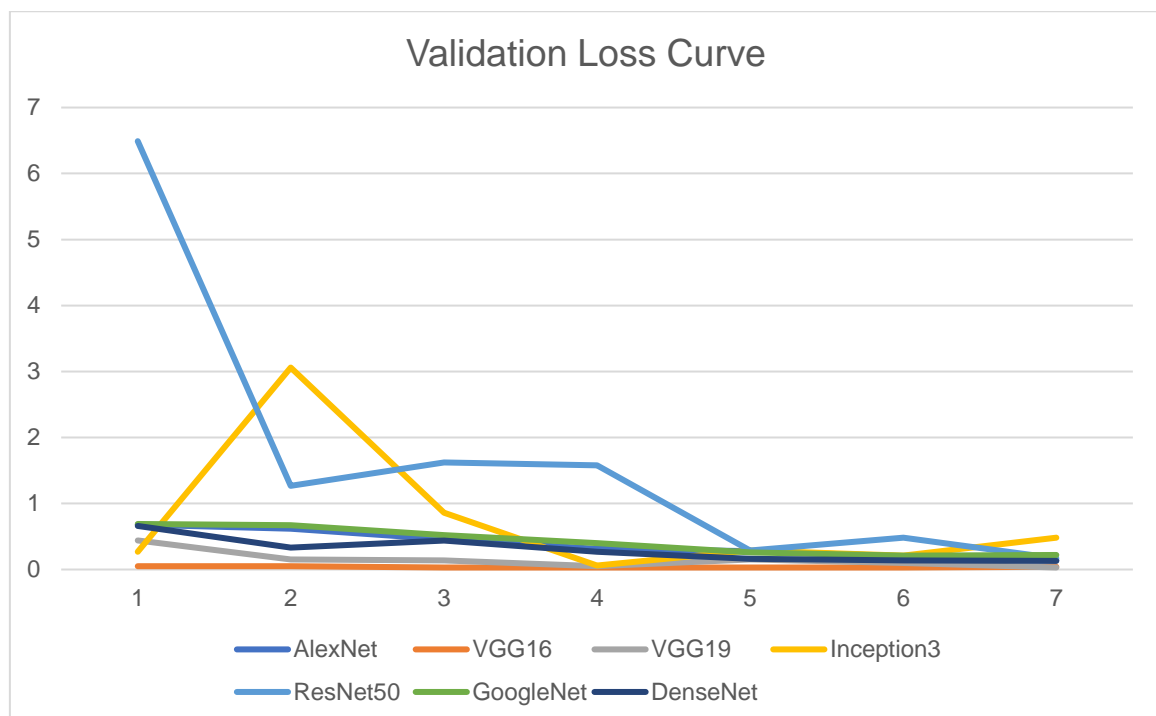


Figure-5.9(d) : Validation Loss curve of different architecture

Chapter-6

Conclusion and future work

Among all of the architecture we find VGG19 the best. Its Accuracy, sensitivity, specificity is highest among all the architecture. Not only that but also its validation accuracy is 98% and validation loss is 6%

6.1 Recommendation and Future work

We are going to upgrade the purpose of the study by attempting to predict the seriousness of COVID-19 and find the infected regions within the chest. We have already compared all CNN models to decide which one is the most excellent for such an utilized case. But we think we can still make it better if we can combine two or more architectures and then make a model out of that. If we have a dataset which contains only the X-ray image of covid patients from Bangladesh then we can make a model of our country. Maybe we have our dataset but only for doctors. We dream to have a dataset to which access will be also given to the engineers, after all, they are the ones behind making all bio-electronic devices for the medical sector.

6.2 Advantages of Deep Learning

There are so many advantages of Deep Learning. Only few of them are given below:

1. Without any further human intervention, deep learning algorithms may produce new features from a small set of features in the training dataset. Deep learning can thus execute complicated tasks that frequently necessitate substantial feature engineering.
2. Deep learning's capacity to work with unstructured data is one of its most appealing features. This is especially important in the business world because the vast bulk of company data is unstructured. The most prevalent data forms used by organizations are text, pictures, and voice. Because traditional machine learning algorithms are

limited in their ability to evaluate unstructured data, this wealth of knowledge is frequently overlooked. And this is where deep learning has the most potential.

3. Deep neural networks' numerous layers make it easier for models to learn complicated characteristics and do more expensive computational tasks, such as completing several complex operations at the same time. It outperforms machine learning in tasks involving unstructured information, such as machine perception (the capacity to understand pictures, sounds, and video as a person would).
4. Deep learning models may be taught at scale using parallel and distributed techniques. Simultaneous techniques, but at the other hand, may be dispersed across numerous systems/computers to finish the learning in under a day.

6.3 Limitation of Deep Learning

Since the models are not scalable or rotation invariant, they can readily misclassify photos with atypical item poses. Some of the most common disadvantages are given below:

1. Deep learning methods require enormous datasets for training, which is a major drawback. To illustrate, in order to achieve desirable results, data formulating numerous dialects, demographics, and time frames are required for a speech recognition algorithm.
2. The deep learning model becomes increasingly data demanding as the architecture grows larger, requiring more data to give valid results. In such cases, recycling data may not be the best option, and data augmentation may be beneficial to some extent, but having more data is always the best option.
3. Due to complicated data models, training deep learning models is an incredibly costly endeavor.
4. Whereas the model might excel at mapping inputs to outputs, it may struggle to comprehend the contextual of the data they're dealing with.
5. It moreover lacks broad understanding and the ability to integrate diverse domains. Deep learning algorithms effectively address the opacity or black box

problem, which makes it difficult to analyze or comprehend how they make judgments.

6. It also makes it difficult for users to comprehend why some components fail. Deep learning architectures excel at tasks such as image classification and sequence prediction. Like GANs, they can even produce data that matches another's pattern. They do not, however, applicable to all supervised learning problems.

Reference

- [1]https://en.wikipedia.org/wiki/Convolutional_neural_network#:~:text=A%20CNN%20architecture%20is%20formed,of%20layers%20are%20commonly%20used.
- [2]https://en.wikipedia.org/wiki/Convolutional_neural_network#:~:text=A%20CNN%20architecture%20is%20formed,of%20layers%20are%20commonly%20used.
- [3]<https://www.upgrad.com/blog/basic-cnn-architecture/>
- [4]Centers for Disease Control and Prevention; Symptoms of Coronavirus. [cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html](https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html). Accessed 6 June 2020
- [5]Ai T, Yang Z, Hou H, Zhan C, Chen C, Lv W, Tao Q, Sun Z, Xia L. Correlation of chest ct and rt-pcr testing in coronavirus disease, (2019) (covid-19) in china: a report of 1014 cases. *Radiology*. 2020;296:200642. doi: 10.1148/radiol.2020200642. [PMC free article] [PubMed] [CrossRef] [Google Scholar]
- [6]<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8490848/>
- [7]Kong W, Agarwal PP. Chest imaging appearance of covid-19 infection. *Radiol Cardiothorac Imaging*. 2020;2(1):e200028. doi: 10.1148/ryct.2020200028. [PMC free article] [PubMed] [CrossRef] [Google Scholar]
- [8]Shi H, Han X, Jiang N, Cao Y, Alwalid O, Gu J, Fan Y, Zheng C. Radiological findings from 81 patients with covid-19 pneumonia in wuhan, china: a descriptive study. *Lancet Infect Dis*. 2020;20:425. doi: 10.1016/S1473-3099(20)30086-4. [PMC free article] [PubMed] [CrossRef] [Google Scholar]
- [9]Chan JF, Yuan S, Kok KH, To KK, Chu H, Yang J, Xing F, Liu J, Yip CC, Poon RW, Tsoi HW, et al. A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: a study of a family cluster. *Lancet*. 2020;395(10223):514–523. doi: 10.1016/S0140-6736(20)30154-9. [PMC free article] [PubMed] [CrossRef] [Google Scholar]
- [10]Santosh KC. Ai-driven tools for coronavirus outbreak: need of active learning and cross-population train/test models on multitudinal/multimodal data. *J Med Syst*. 2020;44(5):1–5. doi: 10.1007/s10916-020-01562-1. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

- [11]Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR. Automated detection of covid-19 cases using deep neural networks with x-ray images. *Comput Biol Med.* 2020;121:103792. doi: 10.1016/j.compbio.2020.103792. [PMC free article] [PubMed] [CrossRef] [Google Scholar]
- [12]Abbas A, Abdelsamea MM, Gaber MM. Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network. *Appl Intell.* 2021;51:854–864. doi: 10.1007/s10489-020-01829-7. [CrossRef] [Google Scholar]
- [13]Majeed T, Rashid R, Ali D, Asaad A. Issues associated with deploying cnn transfer learning to detect covid-19 from chest x-rays. *Phys Eng Sci Med.* 2020;43(4):1289–1303. doi: 10.1007/s13246-020-00934-8. [PMC free article] [PubMed] [CrossRef] [Google Scholar]
- [14]Loey M, Smarandache F, Khalifa NEM. Within the lack of chest covid-19 x-ray dataset: a novel detection model based on gan and deep transfer learning. *Symmetry.* 2020;12(4):651. doi: 10.3390/sym12040651. [CrossRef] [Google Scholar]
- [15]Cohen JP, Morrison P, Dao L (2020) Covid-19 image data collection. *arXiv:2003.11597*
- [16] Zhang Y, Niu S, Qiu Z, Wei Y, Zhao P, Yao J, Huang J, Wu Q, Tan M (2020) Covid-da: deep domain adaptation from typical pneumonia to covid-19. *arXiv preprint arXiv:2005.01577*,
- [17]https://en.wikipedia.org/wiki/COVID-19_pandemic_in_Bangladesh
- [18]<https://www.who.int/director-general/speeches/detail/who-director-general-s-remarks-at-the-media-briefing-on-2019-ncov-on-11-february-2020>
- [19]<https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020>
- [20]I. D. Apostolopoulos and T. A. Mpesiana, “Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks,” *Physical and Engineering Sciences in Medicine*, vol. 43, no. 2, pp. 635–640, 2020.
- [21] West, Jeremy; Ventura, Dan; Warnick, Sean (2007). "Spring Research Presentation: A Theoretical Foundation for Inductive Transfer". Brigham Young University, College of Physical and Mathematical Sciences. Archived from the original on 2007-08-01. Retrieved 2007-08-05.

- [22]Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR. Automated detection of covid-19 cases using deep neural networks with x-ray images. *Comput Biol Med.* 2020;121:103792. doi: 10.1016/j.compbimed.2020.10379
- [23]Abbas A, Abdelsamea MM, Gaber MM. Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network. *Appl Intell.* 2021;51:854–864. doi: 10.1007/s10489-020-01829-7.
- [24]Majeed T, Rashid R, Ali D, Asaad A. Issues associated with deploying cnn transfer learning to detect covid-19 from chest x-rays. *Phys Eng Sci Med.* 2020;43(4):1289–1303. doi: 10.1007/s13246-020-00934-8.
- [25]Cohen JP, Morrison P, Dao L (2020) Covid-19 image data collection.
- [26]<https://paperswithcode.com/method/inception-v3#:~:text=Inception%2Dv3%20is%20a%20convolutional,use%20of%20batch%20normalization%20for>
- [27]<https://paperswithcode.com/method/densenet>
- [28]<https://www.sciencedirect.com/science/article/pii/S1568494620308504>
- [29]<https://www.sciencedirect.com/science/article/pii/S1361841520301584>
- [30]<https://ieeexplore.ieee.org/abstract/document/9430229>
- [31]<https://ieeexplore.ieee.org/abstract/document/9167243>
- [32]W. Zhao, “Research on the deep learning of the small sample data based on transfer learning,” *AIP Conf. Proc.*, vol. 1864, no.1, pp. 020018, Aug. 2017.
- [33]C. M. J. M. Dourado Jr, S. P. P. da Silva, R. V. M. da Nóbrega, A. C. da S. Barros, P. P. R. Filho, and V. H. C. de Albuquerque, “Deep learning IoT system for online stroke detection in skull computed tomography images,” *Comput. Netw.*, vol. 152, pp.25–39, Apr. 2019.
- [34]J. D. C. Rodrigues, P. P. R. Filho, E. Peixoto Jr, A. Kumar, and V. H. C. de Albuquerque, “Classification of EEG signals to detect alcoholism using machine learning techniques,” *Pattern Recognit Lett*, vol.125, pp.140–149, Jul. 2019.
- [35]<https://en.wikipedia.org/wiki/AlexNet>
- [36]<https://www.geeksforgeeks.org/understanding-googlenet-model-cnn-architecture/>

[37]<https://www.mygreatlearning.com/blog/resnet/>

[38]https://www.researchgate.net/profile/Rasber-Rashid/publication/341511596_Covid-19_Detection_using_CNN_Transfer_Learning_from_X-ray_Images/links/5eccda45458515626ccc73d6/Covid-19-Detection-using-CNN-Transfer-Learning-from-X-ray-Images.pdf