Deep Neural Network models for COVID-19 diagnosis from CT-Scan, Explainability and Analysis using trained models

by

Tahsin Islam
21141076
Shahriar Absar
17101410
S.M. Ali Ijtihad Nasif
16301054
Sadman Sakib Mridul
17101157

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at BRAC University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Tahsin Islam

Tahsin Islam
21141076

Shahriar Absar

Shahriar Absar
17101410

S.M. Ali Ijtihad Nasif
16301054

Sadman Sakib Mridul
17101157

Approval

This thesis/project titled "Deep Neural Network models for COVID-19 diagnosis from CT-Scan, Explainability and Analysis using trained models" submitted by:

1. Tahsin Islam (21141076)
2. Shahriar Absar (17101410)
3. S.M. Ali Ijtihad Nasif (16301054)
4. Sadman Sakib Mridul (17101157)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 02, 2021.

Examining Committee:

Supervisor:
(Member)

MD.Saiful Islam
Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Rafeed Rahman
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

The world is going through a severe viral pandemic which is caused by COVID-19. People infected with this virus, experience severe respiratory illness. The virus spreads through particles of saliva or droplets from an infected person. There are ways of identifying COVID-19 based on the symptoms such as fever, dry cough, tiredness, but these symptoms are similar to other existing viral or respiratory infections. There is no quick approach in diagnosing if a patient is infected or not. To overcome the drawbacks mentioned, a faster diagnosis is needed which leads us to the objective of this study. we intend to construct a diagnostic approach that uses pre-existing data mostly on COVID-19, as well as take datasets from other respiratory diseases. We will apply deep learning models to the acquired datasets enabling us to obtain more accurate and efficient results. We aim to use Deep Neural Network models namely Convolutional Neural Network models (CNN) such as VGG19, Inception v3, MobileNetV2, and ResNet-50. These four models are pre-trained and they classify the CT-Scan images based on the trained learning approaches. The result of each model is compared among the models to get faster and more accurate results. This paper also proposes a "Hybrid" model which is composed of a Convolutional Neural Network (CNN) and a Support Vector Machine (SVM). The Hybrid Model is shallow and just as accurate as the pre-trained models. In light of the exactness of the result and the minimal measure of time needed for image classification, we will be able to diagnose more accurately and effectively.

Keywords: COVID-19, Respiratory Diseases, X-ray, CT-Scan, Deep Neural Network, CNN, VGG19, Inception v3, MobileNetV2, Resnet-50, Rapid approach.

Acknowledgement

To begin with, we would like to praise the Almighty Allah, for whom our thesis has been accomplished without any major hurdle.

Secondly, we would like to thank our esteemed supervisor MD. Saiful Islam, for his invaluable supervision, support, and tutelage throughout our entire research. He convincingly guided and encouraged us to think out of the box, be creative and to always do the right thing even when the road got tough. We would also like to thank MD. Rafeed Rahman for his invaluable support and guidance. Without both of their persistent help and support the goal of this project would not have been realized.

Finally, to our parents, without their kind support and prayer nothing would have been possible. Also our appreciation goes out to everyone who helped us in any manner.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that have later been used within the body of the document

$ARIMA$ Autoregressive Integrated Moving Average

$Bi - LSTM$ Bidirectional Long Short-Term Memory

CAP CT-Scans of community-acquired pneumonia

CNN Convolutional neural network

CXR chest X-ray

DNN Deep neural network

GPU Graphics Processing Unit

GRU Gated recurrent units

$LSTM$ Long Short-Term Memory

NB Naive Bayes

RF rheumatoid factor

$RT - PCR$ Reverse transcription polymerase chain reaction

SVM Support Vector Machine

SVR Support Vector Regression

Chapter 1

Introduction

1.1 Preamble

In late 2019 the world began to see the effects of the novel coronavirus (COVID-19) which is officially known as "SARS-CoV-2".The spread of this virus was so rapid that within a month of its initial report it became a global health emergency. Thousands of people died, and even with the most advanced medical treatment, it was not possible to contain the outbreak of this virus. This virus spreads from the droplets of sneeze and cough of an infected person. It damages the respiratory system of the person it infects, causing the person to suffocate due to lack of oxygen.

Though there are various ways we can prevent the virus from spreading, there are still not many ways to accurately diagnose if a person is infected or not. It generally takes 2-14 days to show the symptoms and sometimes the symptoms may not show at all. Now the way the virus is mutating and getting stronger day by day, most people may not live for 2-14 days after contracting the virus. Fever, dry cough, and tiredness are some of the symptoms of COVID-19 but based on this, the diagnosis will not be accurate, because other diseases might have the same symptoms. Moreover, the COVID-19 test centers take a lot of time to provide reports. In most cases, the patients die due to the lack of proper diagnosis and timely treatment.

So in our research, we try to introduce a diagnostic approach that detects COVID-19 and various respiratory diseases based on their pre-existing data and use deep learning models on them, so that we can obtain faster, accurate, and efficient results. We will use Deep Neural Network models namely Convolutional Neural Network models (CNN) such as VGG19 which uses deep Convolutional neural layers to improve accuracy, Inception v3 which is more computationally efficient, MobileNetV2, and Resnet-50. As they are pre-trained models, we will use them to classify the X-ray and CT-Scan images of COVID-19 and other respiratory diseases. The results of each model will be considered and compared among the pre-trained models to get a faster and more accurate result. This paper also proposes a "Hybrid" model which is composed of a convolutional neural network (CNN) and Support Vector Machine (SVM). The Hybrid Model is shallow and just as accurate as of the pre-trained models. Based on the accuracy of the outcome and the least amount of time required for image classification, we will be able to diagnose more effectively and efficiently, which will, in turn, save lives.

1.2 Motivation

More than 230 million individuals have already been diagnosed with the "COVID-19" virus since its emergence in December 2019. Over 5 million people have died as a direct result of it. The virus is spreading at an astonishing speed, claiming the lives of youngsters and old citizens alike. The rapid spread of this virus is causing a variety of respiratory issues in individuals also causes death. By the time the patients are getting their test results, it becomes too late to do anything. There are lackings in fast and accurate methods of detection, patients with other respiratory diseases like pneumonia, tuberculosis, bronchitis are also being suspected as COVID-19. This results in misdiagnosis, false treatment, and death. Radiologists all over the world have had difficulty in testing a large number of patients and delivering test results on time.

Although most of the symptoms are also similar to other respiratory illnesses, various techniques for detecting whether or not a person is infected with the virus have been identified so far. We hope to include and propose a "Hybrid Model," which is a simple and shallow model that uses both Convolutional Neural Networks and Support Vector Machines to accurately predict respiratory illness from X-ray and CT-Scans of the suspected patient in much less time than any other trained convolutional methodology.
Here we hope to contribute as follows:

- Create a "Hybrid" model for identifying the illness that is quicker and more accurate based on the patient's CT-Scans
- Provide a detailed analysis of the diseases based on the collected image datasets.

1.3 Problem Statement

In our modern world, medical science has improved significantly and with such improvements, the world has found out the means to prevent and cure many diseases throughout the years. Maximum infected individuals with COVID-19 may suffer slight respiratory illnesses. At this moment there is no solid way to cure an infected patient and it is spreading rapidly among the general population. Although no cure has been found, being thoroughly educated on the COVID-19 virus is the greatest approach to avoid and slow down the spread. Even though there are various ways we can prevent the virus from spreading, there are still not many ways to accurately diagnose a person. Even though there are ways of identifying based on the fever, dry cough, fatigue, trouble breathing or breathlessness, chest discomfort or pressure, loss of speech or movement, and so on are some of the indications. These symptoms can also be for other diseases. So based on this, the diagnosis will not be accurate. Although there are researches made in regards to quickly identifying the infected patient as this virus is completely new to everyone so as of yet there has been no faster diagnosis method found. The present form of testing, RT-PCR tests, takes a long time to perform, perhaps a week or more, and the reports are not always precise. Moreover, as COVID-19 cases are increasing at a speedy rate there has been a huge scarcity of RT-PCR kits, especially in Bangladesh. In [1] the authors discussed that due to the ongoing increase of COVID-19 cases; Bangladesh has a critical shortage of RT-PCR diagnostic kits. The government has dramatically increased testing efforts, but it appears that these efforts are insufficient because a large number of people have yet to be tested. Many suspected patients die or recover before the testing, despite their best efforts. The delays in getting the test reports are also a huge problem, these delays occur mainly due to the lack of manpower as mentioned in the article [2]. There has also been a shortage of RT-PCR kits in other countries around the world, in [3] it is discussed that laboratory-based molecular diagnostic tests are in poor supply in many parts of the world. The extent of testing in the United States is estimated to be between 3 and 3.5 million tests a week which is far less than even the most progressive predictions of the needed amount. The anticipated number of tests required ranges from 6 million per week (assuming the economy is to be partially restored) to 20 million per day or around 6% of the population. To put it in context, the present worldwide capacity for molecular testing is estimated to be between 14 million and 16 million tests per week, with less than 10 million tests performed each week. Another big issue is the inaccuracy of the present testing procedures. According to reports, RT-PCR tests are not 100 percent reliable since there is a possibility of false alarms, in which people are informed they have the virus when they don't, causing them to be segregated unnecessarily. False positives can arise as a result of the techniques utilized [4].

With all the underlying issues it is quite difficult to detect and diagnose COVID-19 with greater accuracy and a better speed with minimal delay. There has been a massive amount of deaths due to COVID-19 along with a huge loss in health in people's lives around the globe. To overcome and provide a faster and more or less automated diagnosis to the patients, we propose to use Deep Neural Network models for COVID-19 detection and diagnosis. We believe that by performing this research, we will be able to accurately identify and diagnose whether a patient has COVID-

19 or not using X-ray and CT-scan image data by applying multiple deep-learning CNN models to image datasets, such as VGG19, Inception v3, ResNet-50, and MobileNetV2 and a proposed "Hybrid" model to identify COVID-19 from similar diseases and categorize the diseases accordingly, allowing the virus to be discovered much faster and more accurately than existing methods. In this study, we introduce a diagnostic approach that uses deep learning models to diagnose COVID-19 and other respiratory disorders using pre-existing data where to obtain quicker and more effective results, the results of each model will be analyzed and compared among the pre-trained models. We will be able to identify more efficiently and accurately if the patient or person is infected or not, and which respiratory disease the patient may be suffering from, based on the accuracy of the outcome and the least amount of time necessary for image classification. Through this research, we hope to create a positive impact on the overall pandemic situation and attempt to save as many lives as possible.

1.4 Research Objective

COVID-19 has been a worldwide issue with unparalleled social and economic impacts, putting everyone's lives and livelihoods at threat in the long run. Due to the lack of vaccines for COVID-19, extensive population testing has been critical in containing the explosive increase in cases of infection. Moreover, different variants of the virus have been found in different parts of the world which also have a variety of symptoms, in many cases, the symptoms are not even found until the COVID test is positive which leads to a lot of risks. The current method of testing which is RT-PCR tests takes a lot of time, approximately a week or more than a week to give proper results, and sometimes the results are not so accurate. As the virus is mutating rapidly a large portion of the population face lung disorders where 50% to 70% of the lungs get damaged even before any reports are received. Also, with the rapid increase of COVID-19 cases, there has been a shortage of test kits. The main purpose of our research is to use various deep learning models to efficiently and precisely detect and diagnose COVID-19. By detecting the virus faster it will be easier to diagnose the disease and maintain all the procedures to cure it as soon as possible. In addition, faster and more accurate detection of COVID-19 cases will allow the authorities of the health sector to gather information about the population who are COVID-19 positive more efficiently which will, in turn, allow them to make crucial decisions about lockdowns and closing/opening of important institutions like schools, universities, banks, etc. By doing this research we want to make sure that we can perfectly identify and diagnose whether the patient has COVID-19 or not by using X-ray and CT-scan images. Our main goal is to implement various deep-learning CNN models like VGG19, Inception v3, ResNet-50, MobileNetV2 and a proposed Hybrid model on the image datasets in order to classify COVID-19 from similar diseases and categorize the diseases accordingly so that the virus can be detected a lot faster and accurately than conventional methods.

1.5 Thesis Structure

This report describes a model which has been designed to efficiently diagnose COVID-19 and other respiratory diseases from CT-scan and analysis of trained models.

First of all, the 'Introduction' chapter indicates the motivation and inspiration behind our work. The summary of our work and the main purpose of it is stated in this chapter.

Then, in the 'Background Analysis' chapter, we basically summarized some of the previous works of other researchers who worked on the topic of diagnosing COVID-19 or other respiratory diseases. These papers were studied and analyzed thoroughly in order to get and go through various idea about the different direction towards the solution of the problem. Moreover, the algorithms used on the papers are also discussed.

Following, in the 'Model Backgrounds' chapter we briefly discussed about the pre-trained models we have used namely VGG19, Inception v3, ResNet-50 and MobileNetV2. Furthermore, the workings and the layers and their attributes were also discussed here.

Next, in the 'Methodologies' chapter we discussed our collected datasets and the kind and amount of data it contains. We also briefly discussed the SMOTE analysis and processing of images. Furthermore we also discussed how we processed and utilized our pre-trained models.

Next, in the 'Proposed Model' chapter we proposed a model which is a combination of both CNN and SVM, which is shallower compared to the existing pre-trained models we implemented.

Subsequently, in the results chapter we compared our proposed Hybrid model with the pre-trained models that we took and we represented the train-accuracy, validation-accuracy, train-loss and validation-loss with graphs.

Finally, in the conclusion chapter, the conclusion was discussed along with future works.

Chapter 2

Background Analysis

2.1 Literature Review

Recently a number of research have been done regarding COVID-19 diagnosis to accelerate the rate of accurate results with the help of Machine Learning and Deep Learning algorithms. Panwar, H. et al. [5], presented a deep transfer learning algorithm, which uses various datasets of thoracic X-ray and CT-Scan to enhance the diagnosis of COVID-19. The approach depends mainly on binary image classification. Therefore, to categorize the numerous image data, which provides timely and efficient determination of COVID-19 positive cases with much less than two seconds. The authors claim that their process is speedier than RT-PCR testing. Similarities between COVID-19 patients and Pneumonia patients have been found by analyzing the patterns of the radiological images of both pneumonia and COVID-19 patients.

Singh, R. K. et al. [6], took an innovative deep learning-based approach to boost up the treatment of COVID-19 infected patients depending on their X-ray images. COVID-19, pneumonia, and normal were taken as classes that used the method of enhancing the images and also creating segmentation based on it. With this method, Naïve Bayes was used as a meta-learner to create a classification and an updated stacked model of four CNN-based learners was used. Their proposed framework introduces an efficient pruning technique that improves the model's performance and generalizability and reduces its complexity. Moreover, while dealing with small training samples; a range of cutting-edge GAN architectures were able to generate practical synthetic COVID-19 chest X-rays. Various numbers of datasets were used in their purpose of classification, segmentation, and weight initialization study. On standard datasets, their proposed approach outperforms current approaches by 98.67% precision, 0.98 Kappa, and F-1 ratings of 100, 98, and 98 the classes, respectively. This proposed approach can be used as part of a patient's analysis. And can also be considered as gold-standard for clinical laboratory works.

Alshazly, H. et al. [7], investigated the reliability of deep learning models by training them with CT-scan images of the thorax to properly exploit an automated process and also effectively identify patients infected by COVID-19. To obtain the prime results, they established refined deep network architectures to present a transfer learning technique that used precision input tuned for each deep architecture and trained them with the LAMB Optimizer. Based on two CT-scan image datasets, namely the SARS-CoV-2 and the COVID19-CT, they carried out sets of experiments thoroughly and even concluded that their models perform better as compared to the previous studies. They also used renderings of features taken from various models to add up to how deep networks represent CT scan images in the showcase area. The representations reveal thoroughly disconnected groupings representing computerized tomography images from different classes, indicating that their algorithms have learned to use exclusionary characteristics to distinguish computerized tomography scans from various instances. As explained by professional radiotherapists, their models for discovering COVID-19-correlated regions gave discriminative localization and visual explanations.

Li, L. et al. [8], in their research intended to build and analyze a completely automated system for finding COVID-19 by using thoracic CT-scan images in which they created a COVID-19 identification neural network (COV-Net). This deep learning algorithm was created to conduct the COVID-19 diagnosis by extracting visual attributes from "holographic thoracic CT- Scan images". CT-Scans of pneumonia that was obtained from outside the hospital, mostly from the general population also known as "CAP" and other non-pneumonia malformations were used to test the model's potential from August 2016 to February 2020, various data were obtained from six hospitals. The efficiency of the Diagnostic was tested using a graph. It is mentioned in the results of the research that approximately 4352 chest CT scans from 3322 patients were collected for the study. The mean age of the patient was approximately 49 years which had significantly more male subjects than females. In the separate test range, the pet-scan sensitivity and specificity for detecting COVID-19 were 90% and 96%, with an area under the receiver operating characteristic curve of 0.96 (P¡.001).

Yeşilkanat, C. M. et al. [9], used a different approach for predicting the number of instances of the occurrence of pandemics such as COVID-19 in their research paper. In their research, the efficiency of the Random Forest is also known as the "RF" machine learning algorithm. The Algorithm was evaluated in this study to determine near future case counts for 190 nations all over the world. The results were plotted against actual verified cases of COVID-19 outcomes, which were recorded between 23/01/2020 to 17/06/2020. The number of confirmed cases was divided into three sub-datasets for the random forest model which revealed that the random forest machine learning algorithm does an excellent job of predicting the number of cases in the upcoming future if such outbreak like COVID-19 ever emerges.

Shahid, F. et al. [10], discussed that it is feasible to build well-thought-out schemes in the public health system to prevent death by COVID-19 and manage patients. Their prediction models included ARIMA, SVR, LSTM, and Bi-LSTM. In this study, these models were analyzed for time series prediction of confirmed cases, fatalities, and cures in ten key COVID-19-affected countries. Mean Absolute error, root mean square error, and r2 score indices are utilized to compute model efficacy. The Bi-LSTM model surpasses recommended indices in the vast majority of instances. Bi-LSTM, LSTM, GRU, SVR, and ARIMA are the models that rank from best to worst in all cases. For deaths in China, Bi-LSTM produces the lowest MAE and RMSE values. According to the recovered cases in China. They deduced that Bi-LSTM can be used for pandemic forecasting of more conservation and restoration due to its demonstrated resilience and upgraded prediction precision. During the research of improving the detection method of COVID-19, it is quite important to study methods of comparing the rate of escalation of Covid-19 in high-risk countries as well.

Guhathakurta, S. et al. [11], came up with a method for determining whether or not a person is contaminated with COVID-19. They used a Support Vector Machine to classify the patient's health into three categories namely mild infection, no infection, and serious infection; based on the critical effect of the symptoms. Only a few of the symptoms associated with COVID-19 patients have been selected. The major frequent symptoms like fever, breathing rate, and cough are present in 90% of confirmed cases. To categorize these features/symptoms into the given classifications, they used the Support Vector Machine (SVM) classifier. They also used visual programming to compare and contrast common supervised learning models. They were able to forecast the instances with an accuracy of 87%.

Taresh, M. M. et al. [12], attempted to assess the efficacy of contemporary pertained Convolutional Neural Networks (CNNs) in the automated analysis of COVID-19 from thoracic X-rays. They employed image data from patients who had 2 respiratory illnesses and people with no illnesses in their experiments. The efficacy of AI in the timely and efficient determination of COVID-19 from thoracic X-ray images was examined, utilizing a range of pertained deep learning algorithms that were fine-tuned to optimize precision of detection to find the optimal algorithms. In their research, it is said in the results section that the maximum accuracy was achieved by VGG16 with MobileNet, scoring 98.28%. VGG16, on the other hand, surpassed all other models in COVID-19 detection. They claim that the exceptional performance of these pertained models can have a big impact in enhancing the effectiveness and reliability of COVID-19 prediction.

There have been several kinds of research regarding the computerized analysis of COVID-19 using Machine Learning and Deep Learning. Most of the pre-trained models were trained on colored images in the majority of studies. They might lead to inadequacies when fine-tuning the models on grayscale COVID-19 images. Aminu, M. et al. [13], offered the Covid-Net deep learning architecture, which needs fewer parameters. Grayscale images were accepted as inputs, and Covid-Net is acceptable for training with a small training dataset. According to them, Covid-Net exceeds other modern deep learning networks for detecting Covid-19 in experiments.

Chapter 3

Model Backgrounds

In our research, we have used Convolutional Neural Network models Such as VGG19, Inception v3, ResNet-50 and MobileNetV2 and apply and compare our dataset on them. A brief description of their architecture is given below.

3.1 VGG19

VGG19 is a VGG model type that has total of 19 layers,16 of them are convolution layers, 5 are MaxPool layers, 3 are fully-connected layers, and finally remaining 1 is the SoftMax layer. It carries over and improves mostly on ideologies out of its previous versions, and it uses the deep Convolutional neural layers in order to increase and improve accuracy. It could load a pre-trained version of the network from Image data-sets which trained on more than a million pictures. It has been pre-trained to divide the pictures into 1000 segmented categories which include different components, stationery, and a variety of animals. As an outcome, the network has learned a rich classification model for a diverse set of images. The network accepts images with a resolution of 224 by 224.It are an example of traditional Convolutional Neural Network architecture. It was founded on a study of how to strengthen such networks. The network employs small 3 x 3 filters, and is characterized by its simplicity, with just pooling layers and convolutional layers as additional components. It can be used in a variety of ways, as suitable classification architecture for a large amount of datasets, for instance. Weights are easily available in other frameworks like Keras, so they may be explored with or utilized about any purpose the user wants.

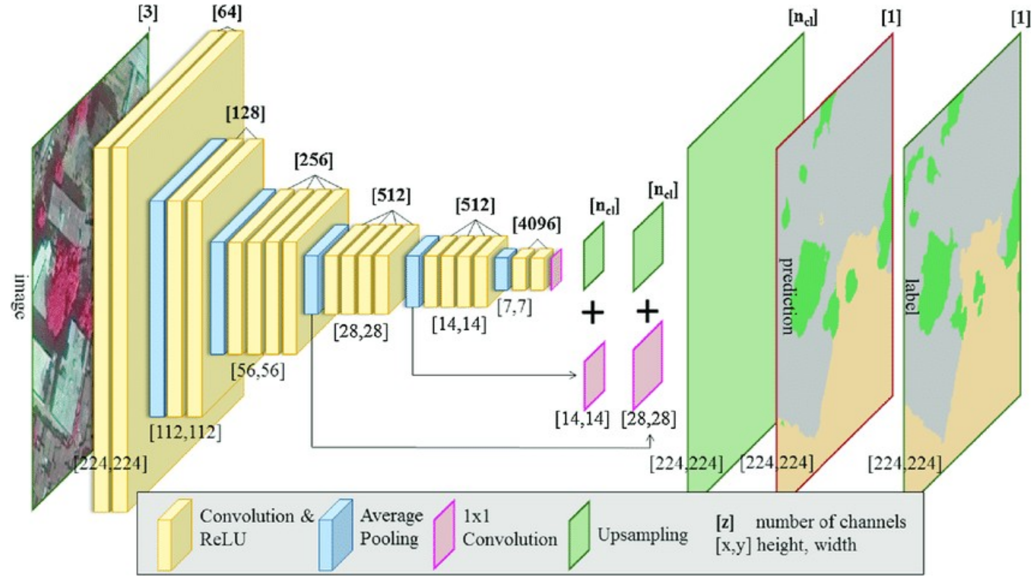

Figure 3.1: VGG19 Architecture.

3.2 Inception V3

Inception V3 is a Convolutional Neural Network architecture that incorporates Smoothing of Labels that is factorized in 7 x 7 convolutions. It also uses an auxiliary classifier to produce Information label lower down the network onward with batch normalization for layers in the side head. If Inception is compared and contrasted with VGG-Net, the networks of Inception have verified to be more computationally well- planned and significantly better structured, both in terms of the number of parameters propagated by the network and the economic cost provoked for both memory and other resources. In order to make alterations to an Inception network, it should be done extremely cautiously to make sure that the computational benefits are not lost. If an Inception v3 model is considered, various systematic approaches have been suggested to slacken the constraints for a handier model adjustment. Factorized convolutions, regularization, dimension reduction and parallelized computations are some of the approaches used. An Inception v3 architecture is progressively constructed one step followed by the other. At first, it is the use of Factorized Convolutional which aids in decreasing the computational efficiency by decreasing the number of parameters included in a network. Moreover, it keeps a check on the productivity of a network. The second step is a smaller convolution which operates by replacing bigger convolutions with smaller ones leading to more rapid training.

If Asymmetric convolutions are considered, a 3 x 3 convolution can be substituted by a 1 x 3 convolution preceded by a 3 x 1 convolution. The number of parameters will rise by a tiny amount if a convolution of 3 x 3 is substituted with a convolution of 2 x 2. A smaller CNN is deployed in the auxiliary classifier between the layers and adds losses to the main network during the training. Finally, to cut the volume of the Grid, pooling operations are employed.

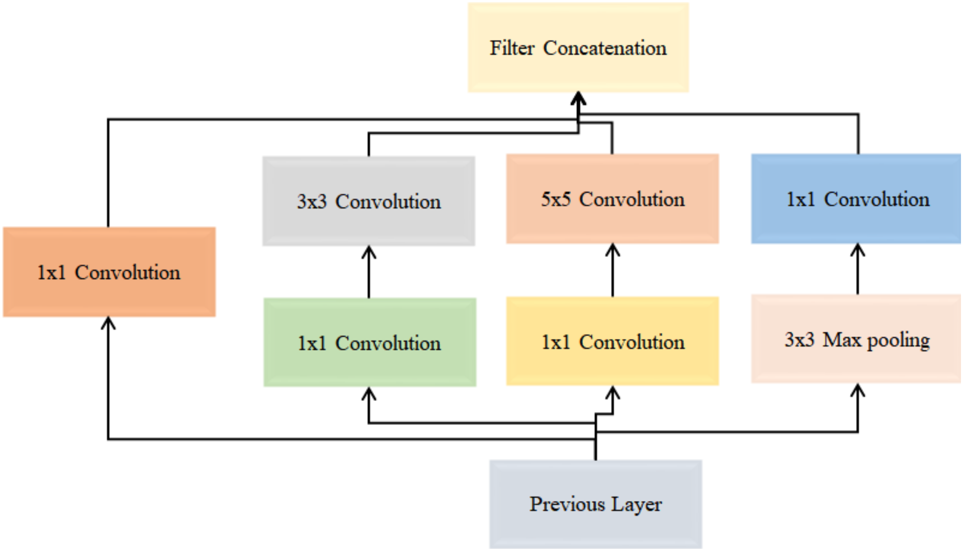

Figure 3.2: Inception V3 Architecture.

3.3 ResNet50

Residual Networks also known as Res-Nets learn residual functions depending on layer inputs instead of unreferenced functions. Residual nets permit these stacked layers to fit a residual mapping. They create a network by piling convolutions on top of each other; for example, a ResNet-50 has fifty layers composed of these blocks. Formally, we let the arrayed nonlinear levels fit a further mapping of by denoting the desired sequence as. The original mapping is reshaped into. There is empirical evidence that these connections are easier to optimize and can benefit from higher depth.

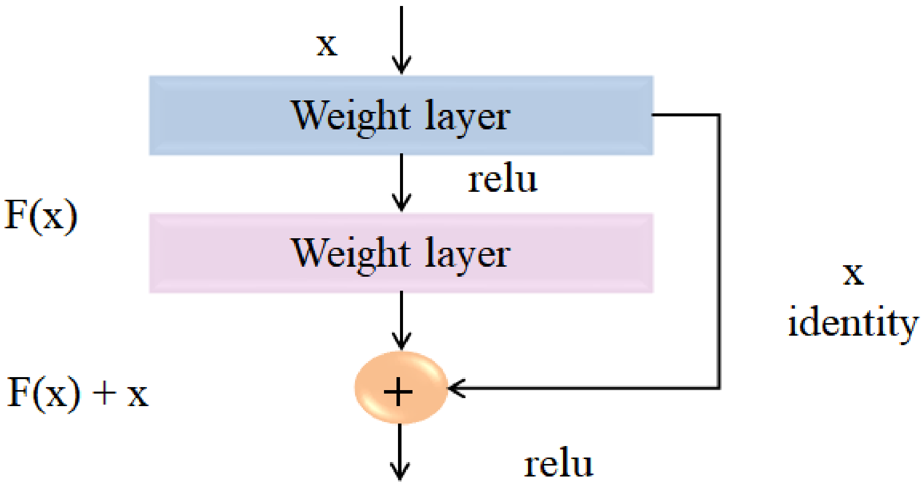

Figure 3.3: Workings of Resnet50.

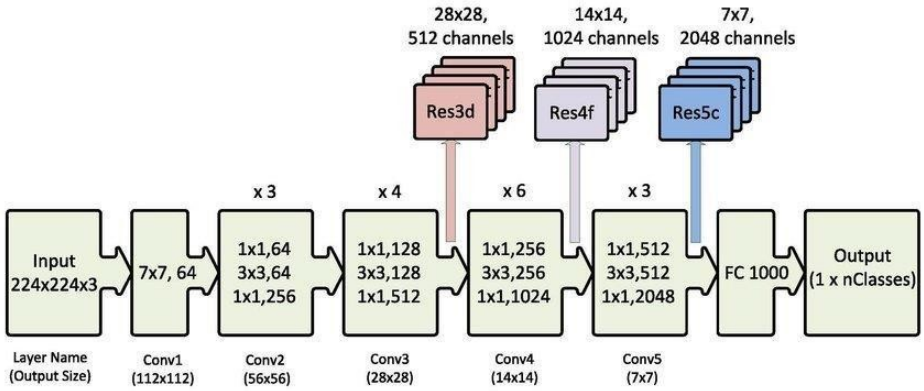

Figure 3.4: Resnet50 Architecture.

3.4 MobileNetV2

MobileNetV2 is a convolutional neural organization engineering that is improved for cell phones. It is the second form of engineering that powers numerous well-known versatile applications' picture handling usefulness. The design has likewise been installed towards systems like Tensor-Flow Lite. Versatile organizations should cautiously adjust signs of progress in PC vision and with the constraints of portable conditions, profound learning overall. Web indexes, for example, Google have been delivering updates to the Mobile Nets engineering consistently, consolidating the absolute most original thoughts in the profound learning space. It depends on an altered remaining design, with lingering associations between bottleneck layers. As a channel, the moderate upgrade layer utilizes lightweight profundity astute convolutions. The engineering of MobileNetV2 by and large contains the underlying completely functional Regarding a convolution layer with 32 channels, there are 19 leftover bottleneck layers. The essential guideline behind MobileNetV2 is that the model's moderate information sources and yields, while the internal layer exemplifies the model's ability to change from lower-level ideas, for example, pixels to more prominent descriptors like picture classes. At last, easy routes, as customary remaining associations, take into account quicker preparing and more prominent precision, and exactness.

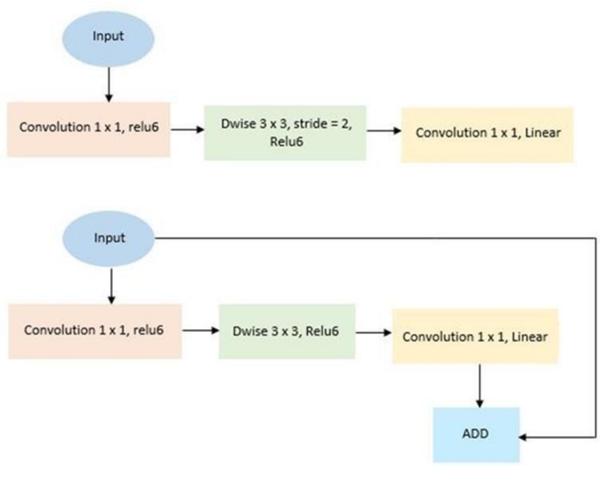

Figure 3.5: MobileNetV2 Architecture.

Chapter 4

Methodologies

In this part we are going to explain the dataset we have used and also explain about the way we used, processed and trained our datasets and implemented them on the Convoluted Neural Network (CNN) models and discuss about the outcomes we get.

4.1 Dataset description

In our research, we collected the dataset from 2 sources, namely Kaggle and GitHub. We narrowed down to the images of chest X-ray and CT-scan. Then we merged both the datasets into our final dataset that we used for training and testing by implementing different models in python programming. The final dataset is composed of 10 different classes labelled as 'Bacterial', 'Covid', 'Fungal', 'Lipoid', 'Normal', 'Other', 'Pneumonia', 'Tuberculosis', 'Unknown' and 'Viral'. The entire dataset contains 877 images and we categorized each of these images according to its respective category. The images in the dataset are in the format of either PNG or JPEG.

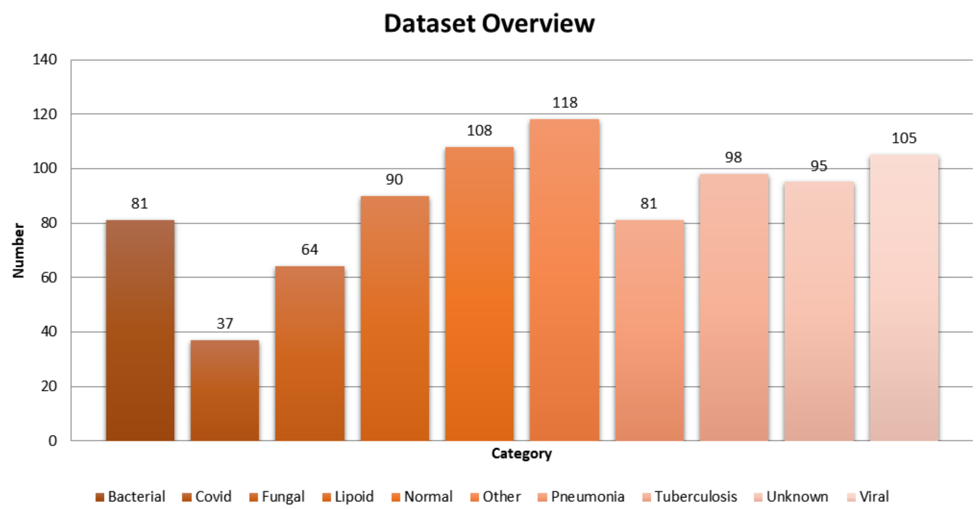

Figure 4.1: Dataset overview.

4.2 SMOTE operation

Every dataset found in the world and gathered will be somewhat imbalanced. They are imbalanced in a way that these training sets will not be divided evenly across the target classes. In this types of cases, the model being applied will be more partial to the class having a huge number of training occurrences that will lead to a downfall in the model's prediction power. Along with that, it also increases the number of false positive outcomes in case of a standard binary classification problem. In Machine Learning, SMOTE is used to handle an imbalanced dataset, where the sampling process is devoted to the training dataset only without making any modifications to the testing dataset. In python programming, Imb-learn is the library needed to execute this method; and for Data-Duplication, the K-Nearest-Neighbors algorithm is used. The artificially generated data points that are correlated with the minority class are pushed into the dataset so that both the labels are roughly of equal size. This process prohibits the model from being biased towards the superiority class and the interactivity among the target classes remains unchanged. Because of the additional data brought up by this process, the system is introduced with bias. The input records must not have any null values while using this method in particular.

4.3 Processing & model-train

We used Google Collaboratory for our python programming tasks. As we had to implement on the models mentioned above to perform transfer-learning and fine-tuning, we had to change the run-time type by choosing GPU as the hardware accelerator. Then we had to mount our notebook to the Google Drive where our dataset was previously uploaded, so that the program can execute accordingly by assessing the image files. Then we had to install Tensor Flow which is an open source platform for deep learning purposes. Later, we imported Keras which is a neural network library. Both Tensor Flow and Keras provide high level APIs for creating and training the models easily. We also have to specify the model that we will be implementing on our dataset itself. Then we set the size of the images of our dataset to 244x244. Then we used scikit- learn for the statistical purposes of each of the models we applied. For dividing the dataset into training and testing sets, we had to apply it in each of the different categories of our dataset. We assigned the test-size to 0.2, meaning that 20% for testing and 80% for training; and we assigned the random-state to 40 which just a measure of randomness (the higher the number, the higher the randomness) and the splitting of the categories into training and testing sets are done by choosing the images randomly. Then we setup our neural network by specifying each of the models as we pass the image size as input, assigning weights to ImageNet which is an image database for deep learning purposes and by assigning include-top to false so that the neural network does not get activated right away. Then we freeze the training layers. As we have mentioned before regarding the categories of our dataset where each of them are contained in separate folders which are all located in the same Google Drive link. So, for splitting the set of images into training and testing sets, it must be done to each and every folder in our dataset. Then we flatten the layers and activate the network with the Softmax function. After that, we compile the model by assigning the loss function as 'categorical cross-entropy', optimizer as 'Adam' and metrics as 'accuracy'. Categorical Cross-entropy deals with an output tensor and a target tensor. Adam is a deep learning direct training method that substitutes gradient mechanism. Accuracy is the number of correct predictions as per the model applied. For our test-data-generator, we re-scale the pixel of every image in the dataset, and for our train-data-generator, we do the same thing and also assign shear-range to 0.2, zoom-range to 0.2 And horizontal-flip to 'true'. Then for both our training and testing sets, we set up the test data-generator flow from the image directory; we also set the target size to 244x244, batch size (subset size of our training sample) to 16 and class mode to 'categorical'. Finally, we train our deep learning models by using the model-fit-generator function and assigning epochs to 20.

Chapter 5

Proposed Model

5.1 Hybrid Model

We propose to do image classification on a Hybrid model which is an SVM on the CNN (Convolutional Neural Network) itself. SVM stands for Support Vector Machine and is a linear model used to address regression and classification problems. Support Vector Machine is a conventional machine learning technique which helps in segmenting huge amounts of data. It is particularly helpful for simulation and model - based applications in a massive data environment. It is good at solving either linear or nonlinear difficulties and can also be used for a variety of tasks. SVM is a steady process: The algorithm generates a line, or hyper-plane, that separates the data into clusters. SVM is a very good algorithm for classification, be it binary or multi class classification. We used the same dataset like we did for transfer-learning and fine-tuning.

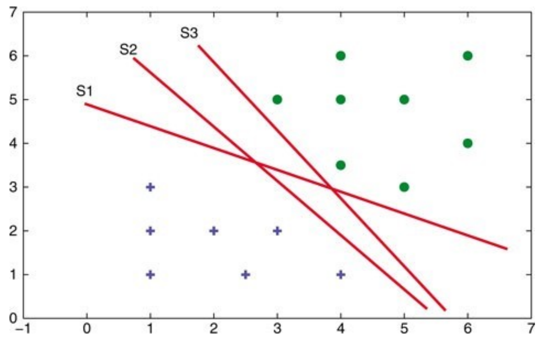

Figure 5.1: Basic classification of SVM model.

In this proposed model of ours, at first, we will import image data generator which will be used for data-augmentation techniques like augmenting the data to create more additional data in the memory. After that, we will re-scale the images in both the train and test data, and will apply image-data-generator in the training data set. Then we will set up the directories for both training and testing datasets by setting the batch-size to 32, target size to 64 x 64 and class mode to categorical as we are doing multi-classification over here.

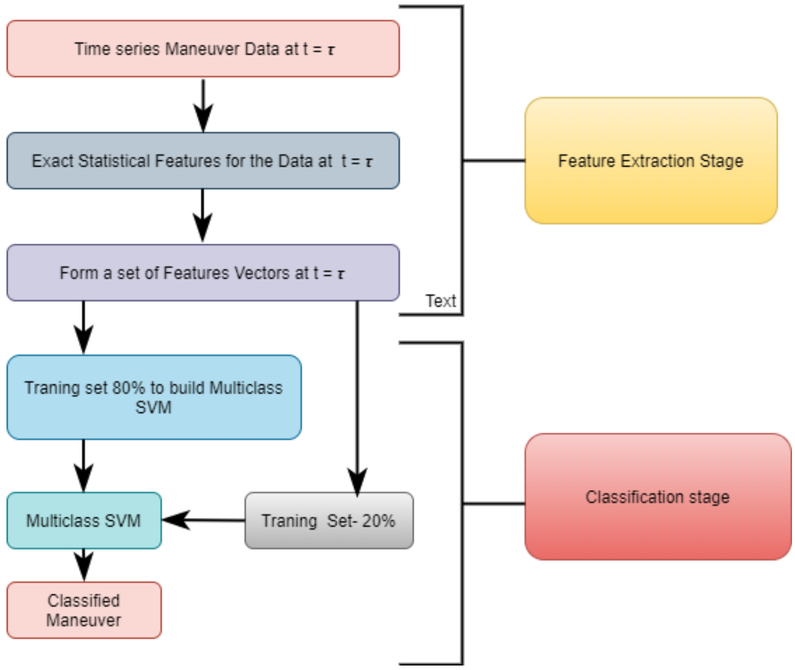

Figure 5.2: Flowchart of SVM model.

Later on, we will once again import two layers namely dense layer (to add the nodes with respect to the hidden layers) and convolutional-2D layer (for the convolutional operation) because we just want to create a plain neural network. Then we will import L2 regularize. After that we will create our layer by adding our sequential layer; a convolutional-2D layer with filter of 32, padding as same, kernel size as 3, activation function as relu, strides as 2 and input shape as 64 x 64; a max-pooling layer of pool size as 2 and strides as 3; another convolutional-2D layer (same as before) followed by another max-pooling layer (same as before) and then we flatten it. A dense layer with units of 128 and activation function as relu will be created and in our final layer, all the categories (10 according to our our dataset), L2 regularize as 0.01 and activation function as softmax. Afterwards we will compile our neural network by setting the metrics as accuracy, loss as squared hinge and optimizer as adam. Finally, we run the entire algorithm by setting up epochs to 20 and training-set to test-set.

Chapter 6

Result Analysis & Comparison

6.1 Deep Learning Models

As we have mentioned before that we have trained each of our deep learning models by assigning epochs to 20, we have obtained the following as given below:

- **Inception v3:** It took 4335 seconds (1 hour, 12minutes and 15 seconds) in total. The first epoch took 504 seconds (8 minutes and 24 seconds) where each step took 9 seconds; and the rest of the 19 epochs took an average time of 201 seconds (3 minutes and 21 seconds) where each step took 4 seconds. At the very first epoch, the initial values of train-loss, validation-loss, validation-accuracy and train-accuracy are 6.4922, 0.8518, 0.9019 and 0.6454 respectively. Then at the very last epoch, these values are 0.3464, 0.2202, 0.9806 and 0.9692 respectively. Finally, we plotted two graphs, one showing how the values of both train-loss and validation-loss changed over 20 epochs; and the other one showing how the values of both train-accuracy and validation- accuracy changed over 20 epochs.

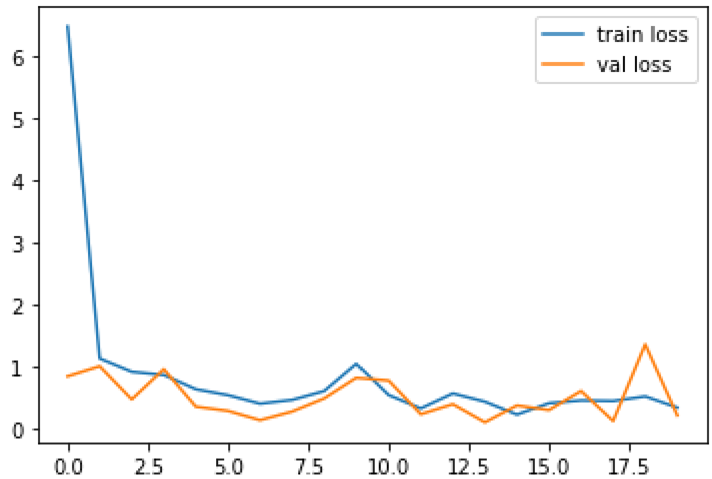

Figure 6.1: Comparison between Train-loss VS Validation-loss of Inception-V3 over 20 epochs.

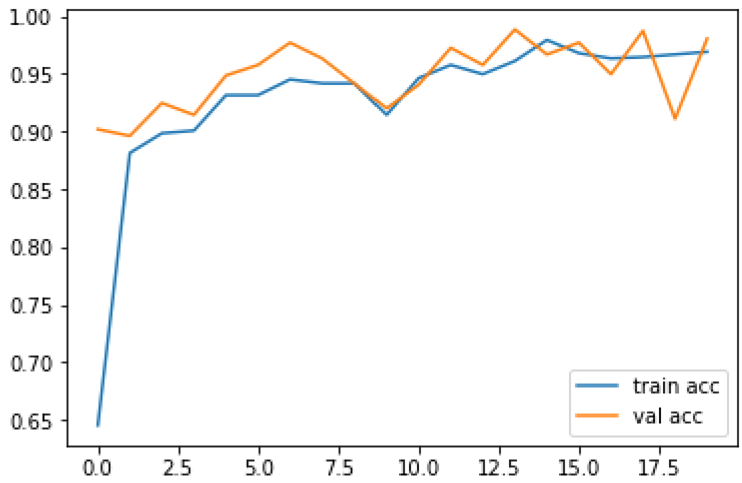

Figure 6.2: Comparison between Train-accuracy VS Validation-accuracy of Inception-V3 over 20 epochs.

- **ResNet-50:** It took 1235 seconds (20 minutes and 35 seconds) in total. The first epoch took 525 seconds (8 minutes and 45 seconds) where each step took 10 seconds; and the rest of the 19 epochs took an average time of 284 seconds (4 minutes and 44 seconds) where each step took 5 seconds. At the very first epoch, the initial values of train-loss, validation-loss, validation-accuracy and train-accuracy are 6.5914, 1.6879, 0.4390 and 0.1893 respectively. Then at the very last epoch, these values are 1.2457, 0.9659, 0.7206 and 0.6705 respectively. Finally, we plotted two graphs, one showing how the values of both train-loss and validation-loss changed over 20 epochs; and the other one showing how the values of both train-accuracy and validation-accuracy changed over 20 epochs.

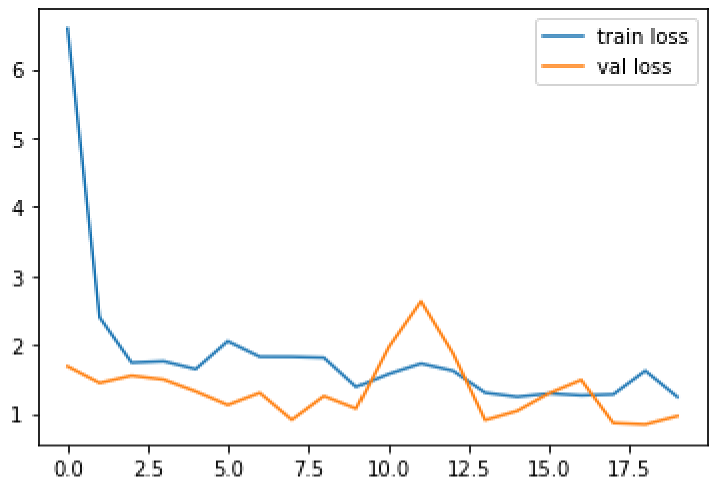

Figure 6.3: Comparison between Train-loss VS Validation-loss of ResNet-50 over 20 epochs.

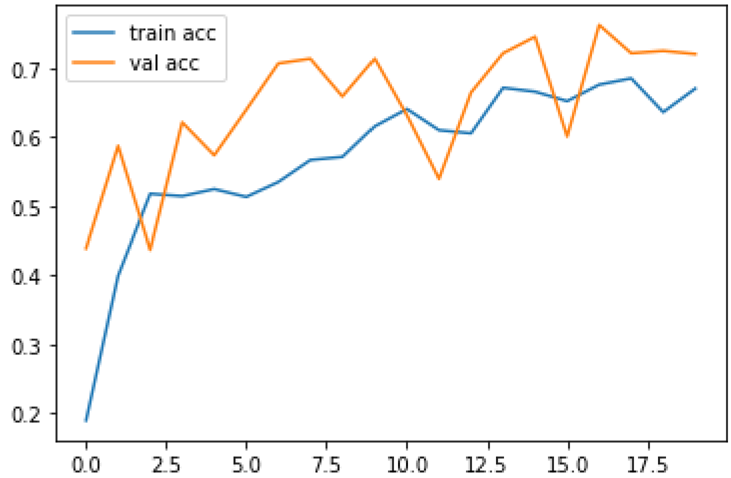

Figure 6.4: Comparison between Train-accuracy VS Validation-accuracy of ResNet-50 over 20 epochs.

- **VGG19:** It took 8614 seconds (2 hours, 23 minutes and 34 seconds) in total. The first epoch took 4677 seconds (1 hour, 17 minutes and 57 seconds) where each step took 16 seconds; and the rest of the 19 epochs took an average time of 196 seconds (3 minutes and 16 seconds) where each step took 8 seconds. At the very first epoch, the initial values of train-loss, validation-loss, validation-accuracy and train- accuracy are 0.4475, 0.2918, 0.8890 and 0.8406 respectively. Then at the very last epoch, these values are 0.1274, 0.3869, 0.9107 and 0.9685 respectively. Finally, we plotted two graphs, one showing how the values of both train-loss and validation-loss changed over 20 epochs; and the other one showing how the values of both train-accuracy and validation- accuracy changed over 20 epochs.

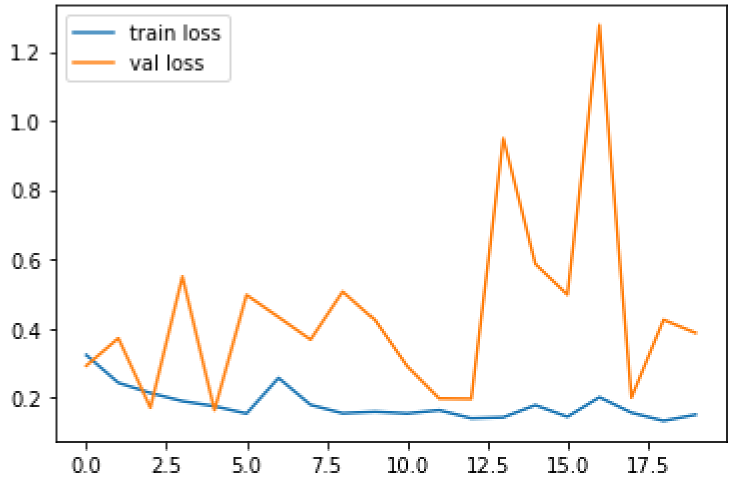

Figure 6.5: Comparison between Train-loss VS Validation-loss of VGG-19 over 20 epochs.

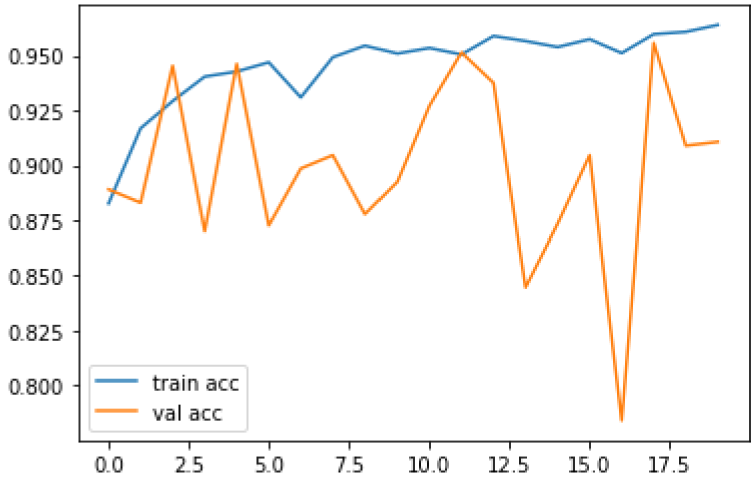

Figure 6.6: Comparison between Train-accuracy VS Validation-accuracy of VGG-19 over 20 epochs.

- **MobileNetV2:** It took 2343 seconds (39 minutes and 3 seconds) in total. The first epoch took 715 seconds (11 minutes and 55 seconds) where each step took 12 seconds; and the rest of the 19 epochs took an average time of 1615 seconds (26 minutes and 55 seconds) where each step took 5 seconds. At the very first epoch, the initial values of train-loss, validation-loss, validation-accuracy and train-accuracy are 7.4631, 0.9946, 0.9042 and 0.5730 respectively. Then at the very last epoch, these values are 0.3589, 0.3586, 0.9806 and 0.9846 respectively. Finally, we plotted two graphs, one showing how the values of both train-loss and validation-loss changed over 20 epochs; and the other one showing how the values of both train-accuracy and validation- accuracy changed over 20 epochs.

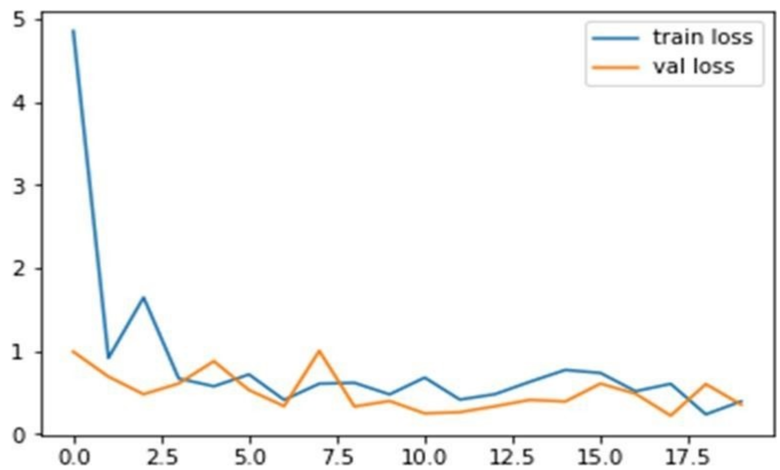

Figure 6.7: Comparison between Train-loss VS Validation-loss of MobileNet-V2 over 20 epochs.

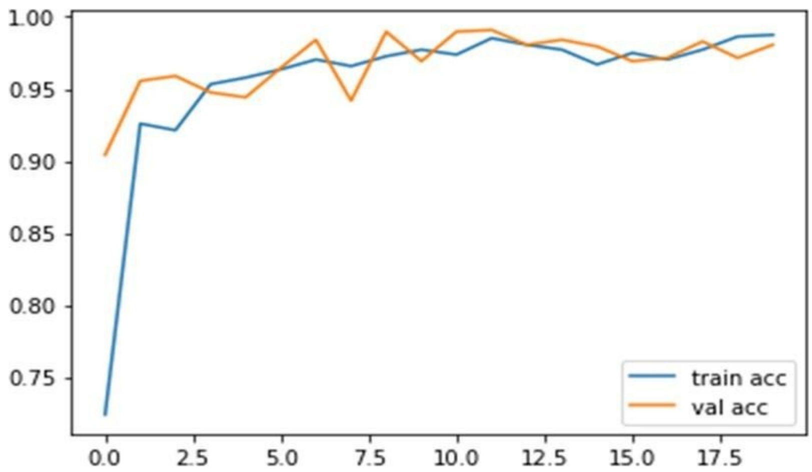

Figure 6.8: Comparison between Train-accuracy VS Validation-accuracy of MobileNet-V2 over 20 epochs.

6.2 Proposed Hybrid Model

We executed this algorithm three times and the results of each are as follows.

The first iteration took 2153 seconds (35 minutes and 53 seconds). The first epoch took 672 seconds (11 minutes and 12 seconds) and it took 23 seconds per step. The rest of the 19 epochs took an average time of 76 seconds (1 minute and 16 seconds) and it took 3 seconds per step. The initial values of train-loss, validation-loss, validation-accuracy and train-accuracy are 1.3135, 1.2733, 0.3170 and 0.1881 respectively; and the final values of those are 0.9825, 0.9767, 0.8301 and 0.8198 respectively.

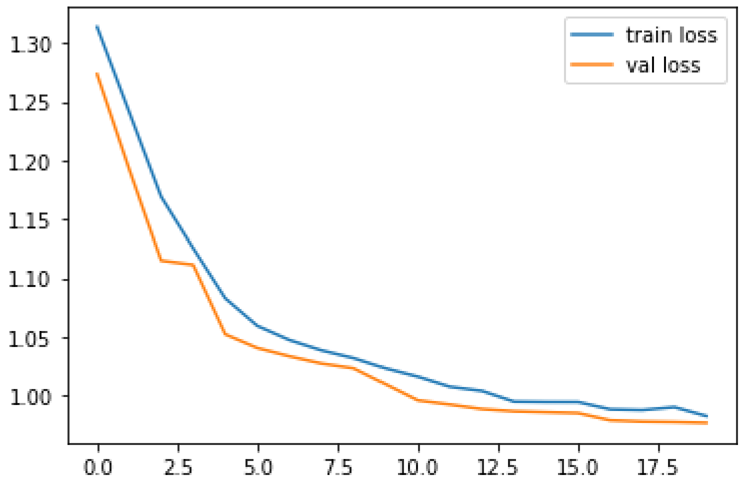

Figure 6.9: Train loss VS Validation loss.

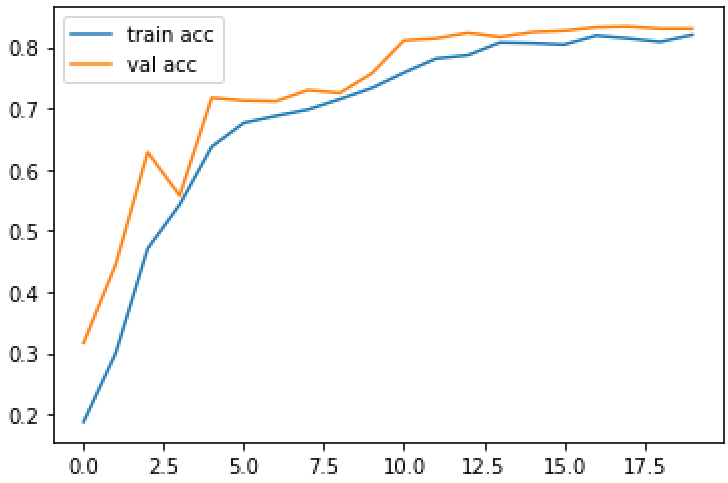

Figure 6.10: Train accuracy VS Validation accuracy.

The second iteration took 1538 seconds (25 minutes and 38 seconds). The 20 epochs took an average time of 76 seconds (1 minute and 16 seconds) and it took 3 seconds per step. The initial values of train-loss, validation-loss, validation-accuracy and train-accuracy are 0.9804, 0.9749, 0.8335 and 0.8233 respectively; and the final values of those are 0.9452, 0.9416, 0.9133 and 0.9065 respectively.

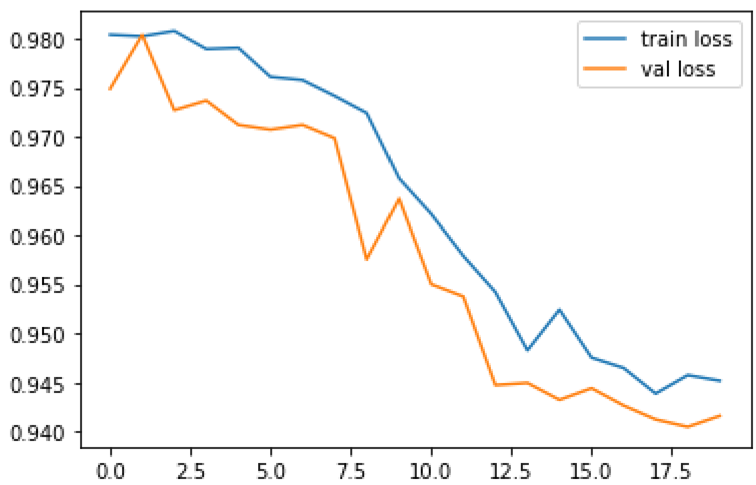

Figure 6.11: Train loss VS Validation loss.

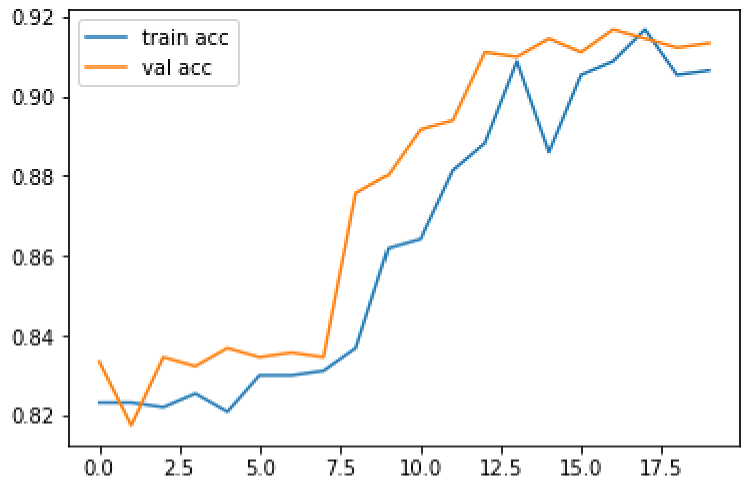

Figure 6.12: Train accuracy VS Validation accuracy.

The third iteration took 1567 seconds (26 minutes and 7 seconds). The 20 epochs took an average time of 78 seconds (1 minute and 18 seconds) and it took 3 seconds per step. The initial values of train-loss, train-accuracy, validation-loss and validation-accuracy are 0.9434, 0.9088, 0.9403 and 0.9202 respectively; and the final values of those are 0.9313, 0.9361, 0.9296 and 0.9384 respectively.

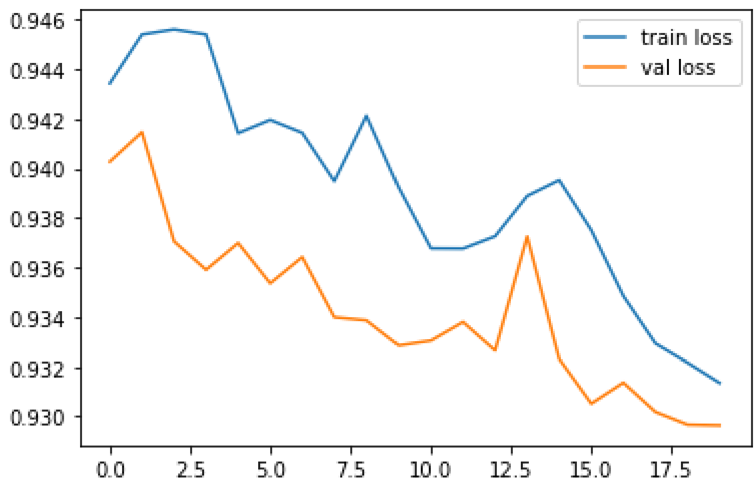

Figure 6.13: Train loss VS Validation loss.

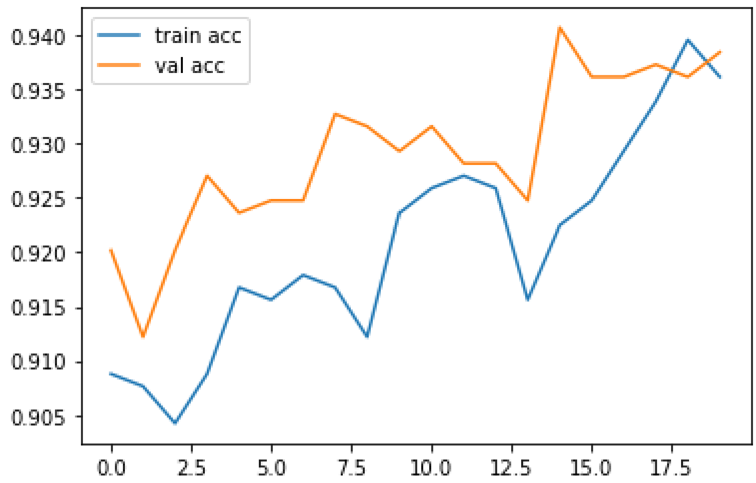

Figure 6.14: Train accuracy VS Validation accuracy.

6.3 Comparison with other trained model

In our Hybrid model, we used SVM (Support Vector Machine) which is a very good algorithm for classification; be it binary or multi-class classification. We have executed four pre-trained models and one hybrid model for transfer-learning and fine-tuning by using the same dataset for image classification. The four pre-trained models are **Inception V3, VGG19, ResNet50 and MobileNetV2.**

Comparing our hybrid model with The Inception V3 we can see that it took 4335 seconds(1hour, 12 minutes and 15 seconds), The ResNet50, total time they 1235 seconds (20 minutes and 35 seconds). The VGG19 the 24279 seconds (6 hours, 44 minutes and 39 seconds) and finally The MobileNetV2; and 3083 seconds (51 minutes and 23 seconds) respectively to be successfully executed. Each of these four models was run only once and the **Hybrid Model** we proposed was executed Three times. In the first iteration took 2153 seconds (35 minutes and 53 seconds). The second iteration took 1538 seconds (25 minutes and 38 seconds) and finally the third iteration took 1568 seconds (26 minutes and 8 seconds).

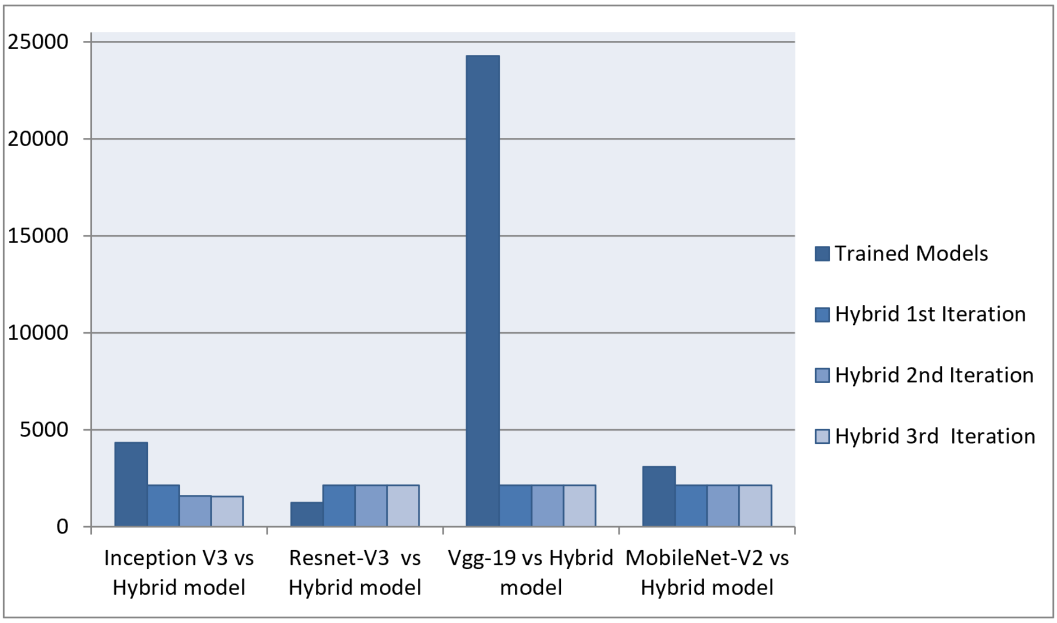

Figure 6.15: Time Comparison.

Here we can see that 3 out of 4 pre-trained model took significantly more time than our Hybrid Model and only ResNet-50 was faster and VGG19 took significantly more time than the other models. Here our Hybrid Model is a simple and shallow model which gives perfect accuracy with faster time.

To summarize, our Hybrid model has 273706 trainable parameters with 0 non-trainable parameters in total it is the lowest number of parameters among all the models we have applied; and those models had non-trainable parameters which increased the total number of parameters. This model is composed of 2 max-pooling layers, 2 dense layers and 2 convolutional 2D layers which in total is the lowest number of layers a neural network has among the applied models; and those models have some other layers and even the same layers as those in the Hybrid model more than twice. This model is shallower than the rest of the model applied. On average, the time taken to finish each epoch in our Hybrid model was the least among all the models we applied. Our Hybrid model performed well without using any boosting methods.

Moreover, the Hybrid Model was found to have the highest accuracy among all of the models used. A table showing the comparison of train-loss, validation-loss, validation-accuracy and train-accuracy on the last iteration among the all models applied in the research has been provided as follows:

Table 6.1: Performance evaluation of algorithms used (Final Iteration)

	Train-loss	**Train-accuracy**	**Validation-loss**	**Validation-accuracy**
Inception V3	0.3464	0.9692	0.2202	0.9806
VGG19	0.1274	0.9685	0.3869	0.9107
Resnet50	1.2457	0.6705	0.9659	0.7206
MobileNet V2	0.3589	0.9846	0.3586	0.9806
Hybrid	0.9313	0.9361	0.9296	0.9384

Chapter 7

Conclusion & Future Works

Science has relieved us from countless withering diseases and if given time it will be able to treat COVID-19 patients, comprehensively. This is for developing countries like Bangladesh where most of the hospitals are equipped with limited resources and hence, cannot treat COVID-19 infected patients properly and effectively. However, by using the diagnosing method on the CT-Scans and X-ray images and carefully studying the patterns in the respiratory system of COVID-19 or any other virus-infected patients, the doctors can quickly and effectively diagnose a patient while providing them with fast and accurate treatment. This can potentially help the doctor immensely during the treatment process as well as keep them safe. Moreover, there will be fewer chances of getting false positives from the test results. Also, it will allow the doctors to quickly diagnose whether it is any other respiratory disease or not by comparing the available data from previously infected patients. Furthermore, it would massively decrease the growing number of COVID-19 infected people and the death rate will come down if the virus is caught in its infancy stage. This would surely give everyone a sign of hope and will be an effective tool in ending the pandemic.

In the future, attempts will be taken to diversify the data-set with more relevant data and also increase the size. We would like to train the "Hybrid" model further for better accuracy and modify the models used for detecting other diseases. Also, we plan to use Grad-CAM for further explainability and visualization purposes.

Bibliography

[1] The Financial Express, *Covid-19 testing shortages, long wait times for results trigger concerns*, https://thefinancialexpress.com.bd/national/covid-19-testing-shortages-long-wait-times-for-results-trigger-concerns-1592972068, Accessed: 2021-10-3.

[2] V. Umarji and S. Das, *Covid-19 second wave: Delays in RT-PCR tests mainly due to manpower crunch*, https://www.business-standard.com/article/current-affairs/covid-19-second-wave-delays-in-rt-pcr-tests-mainly-due-to-manpower-crunch-121042400004_1.html, Accessed: 2021-10-3, Apr. 2021.

[3] M. Behnam, A. Dey, T. Gambell, and V. Talwar, *COVID-19: Overcoming supply shortages for diagnostic testing*, https://www.mckinsey.com/industries/life-sciences/our-insights/covid-19-overcoming-supply-shortages-for-diagnostic-testing, Accessed: 2021-10-3, Jul. 2020.

[4] D. Bardsley, *False negatives and positives: How accurate are PCR tests for covid-19?* https://www.thenationalnews.com/uae/science/false-negatives-and-positives-how-accurate-are-pcr-tests-for-covid-19-1.1113187, Accessed: 2021-10-3, Jun. 2021.

[5] H. Panwar, P. K. Gupta, M. K. Siddiqui, R. Morales-Menendez, P. Bhardwaj, and V. Singh, "A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest x-ray and CT-Scan images," en, *Chaos Solitons Fractals*, vol. 140, no. 110190, p. 110 190, 2020.

[6] R. K. Singh, R. Pandey, and R. N. Babu, "COVIDScreen: Explainable deep learning framework for differential diagnosis of COVID-19 using chest x-rays," en, *Neural Comput. Appl.*, vol. 33, no. 14, pp. 1–22, 2021.

[7] H. Alshazly, C. Linse, E. Barth, and T. Martinetz, "Explainable COVID-19 detection using chest CT scans and deep learning," en, *Sensors (Basel)*, vol. 21, no. 2, p. 455, 2021.

[8] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, J. Bai, Y. Lu, Z. Fang, Q. Song, K. Cao, D. Liu, G. Wang, Q. Xu, X. Fang, S. Zhang, J. Xia, and J. Xia, "Using artificial intelligence to detect COVID-19 and community-acquired pneumonia based on pulmonary CT: Evaluation of the diagnostic accuracy," en, *Radiology*, vol. 296, no. 2, E65–E71, 2020.

[9] C. M. YeŞİlkanat, "Spatio-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm," en, *Chaos Solitons Fractals*, vol. 140, no. 110210, p. 110 210, 2020.

[10] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM," en, *Chaos Solitons Fractals*, vol. 140, no. 110212, p. 110 212, 2020.

[11] S. Guhathakurata, S. Kundu, A. Chakraborty, and J. S. Banerjee, "A novel approach to predict COVID-19 using support vector machine," in *Data Science for COVID-19*, Elsevier, 2021, pp. 351–364.

[12] M. M. Taresh, N. Zhu, T. A. A. Ali, A. S. Hameed, and M. L. Mutar, "Transfer learning to detect COVID-19 automatically from x-ray images using convolutional neural networks," en, *Int. J. Biomed. Imaging*, vol. 2021, p. 8 828 404, 2021.

[13] M. Aminu, N. A. Ahmad, and M. H. Mohd Noor, "Covid-19 detection via deep neural network and occlusion sensitivity maps," en, *Alex. Eng. J.*, vol. 60, no. 5, pp. 4829–4855, 2021.