# logo

**A Mini Project Report On**

**Covid-19 Detector**

*Submitted in partial fulfillment of the requirement for the degree of*

**Bachelor of Engineering in**

**Computer Science and Engineering**

**Submitted by A09**

**Debabrata Maity 01FE18BCS068**

**Deepak Shinde 01FE18BCS069**

**Arun Harikant 01FE18BCS050**

**Darshan Patgar 01FE18BCS066**

*Under the guidance of*

**Dr. S.G. Kanakaraddi**

SCHOOL of COMPUTER SCIENCE & ENGINEERING,

HUBLI – 580 031 (India)



**CERTIFICATE**

This is to certify that Mini Project entitled C OVID-19 DETECTOR is a bonafied work carried out by the student team Mr. Debabrata Maity – **01FE18BCS068** , Mr. Deepak Shinde **- 01FE18BCS069**, Mr. Darshan Patgar **- 01FE18BCS066**, Mr. Arun Harikant - **01FE18BCS050** , in partial fulfillment of completion of Fifth semester B. E. in Computer Science and Engineering during the year 2020 – 2021. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said programme.

**Guide Head, SoCSE**

**Dr. S.G. Kanakaraddi Dr. Meena S. M**

**External Viva:**

**Name of the Examiners Signature with date**

**1.**

**2.**

**1. Introduction**

**1.1 Overview of the project**

Coronavirus disease 2019 (COVID-19) has widely spread all over the world. It is highly contagious and may lead to acute respiratory distress or multiple organ failure. Data science can depict certain characteristic manifestations in the lungs associated with COVID-19. Therefore, our model could serve as an effective way for early screening and diagnosis of COVID-19. Recently, artificial intelligence using deep learning technology has demonstrated great success in the medical domain due to its high capability of feature extraction.

**1.2. Motivation**

Current situation faced many problems due to COVID-19 and makes big impact over the world, still many countries is not get rid of this virus including our nation also. It turned to economic downturn and many people lose their lives over a many countries. Today it's challenging to find solution to predict or detect the COVID-19 persons who are infected by this virus before to get solutions through vaccines. It motivated us to work on COVID-19 by continuously observing a person whether he/she has infected or not. Ultimately our goal is to detect suspected person’s COVID-19 result and analysing the effect of virus on person having other external diseases.

**1.3. Objectives**

* To create required dataset and perform preprocessing
* To classify chest x-ray images into COVID-19 positive and negative
* To perform post processing/back propagation to achieve result better than SOTA result
* To generate heat map on the infected region and segment it from x-ray image

**1.4 Literature Surveys**

**Survey 1**

**Purpose:** To develop a fully automatic framework to detect COVID-19 using chest CT and evaluate its performance.

**Author:** Lin Li(BS) , Lixin Qin(PhD) , Youbing Yin(PhD from the Department of Radiology), Wuhan Huangpi People’s Hospital, Wuhan, China.

**Methodology:** A deep learning model, the COVID-19 detection neural network was developed to extract visual features from volumetric chest CT scans for the detection of COVID-19. CT scans of community-acquired pneumonia (CAP) and other non-pneumonia abnormalities were included to test the robustness of the model. The datasets were collected from six hospitals between August 2016 and February 2020. Diagnostic performance was assessed with the area under the receiver operating characteristic curve, sensitivity, and specificity.

**Constraints:** 6 hospital dataset in the span of August 2016 and February 2020.

**Key Result**

* A deep learning method was able to identify coronavirus disease 2019 on chest CT scans (area under the receiver operating characteristic curve, 0.96).
* There was overlap in the chest CT imaging findings of all viral pneumonias with other chest diseases that encourages a multidisciplinary approach to the final diagnosis used for patient treatment.

**Loop Holes or Future Scope:** We can get more accurate result by using the oxygen rate attribute from the dataset in the body using oximeter. When the pulse oximeter is placed on the fingertip or toe of the person, it beams multiple small rays of light that pass through the blood in that specific part of the body. The light sensors in the pulse oximeter measure the number of blood cells that carry oxygen and those which do not. Lower the oxygen level, confidence level of our COVID positive result of a person increases. Moreover this oxygen level may distinguish from other viral pneumonias. Another improvements to this research can be, we can categories the level of covid positive into severe or recoverable (like stage I, stage II, stage III) in consideration with other symptoms too like diabetes, high pressure.

**Survey 2:**

**Purpose:** COVID-19 outbreak was first reported in Wuhan, China and has spread to more than 50 countries. WHO declared COVID-19 as a Public Health Emergency of International Concern (PHEIC) on 30 January 2020. Naturally, a rising infectious disease involves fast spreading, endangering the health of large numbers of people, and thus requires immediate actions to prevent the disease at the community level. Therefore, CoronaTracker was born as the online platform that provides latest and reliable news development, as well as statistics and analysis on COVID-19. This paper is done by the research team in the CoronaTracker community and aims to predict and forecast COVID19 cases, deaths, and recoveries through predictive modeling. The model helps to interpret patterns of public sentiment on disseminating related health information, and assess political and economic influence of the spread of the virus.

**Author:** Fairoza Amira Binti Hamzah, Cher Han Lau, Hafeez Nazri, Ming Hong Chungj from(a The Kyoto College of Graduate Studies for Informatics, Institut Wanita Berdaya Selangor, University Malaysia Sarawk.)

**Methodology:** Both the platform and dashboard are hosted in Amazon Web Services (AWS). They provisioned AWS Relational Database Service (RDS) to host the data in MySQL table form. All the data collected and integrated using Python program running in AWS Elastic Compute Cloud (EC2) and was scheduled to automatically update every 15 minutes. The size of the database is relatively around 30GB.They developed their own micro-services and scraper in Python to fetch the data and news from the sources. Using predictive modeling, they represented confirm rate, death rate, recovery rate in graph

**Constraints:** Not identified

**Key results:** The graph and table updates after every 15 minutes.

**Loop holes or future scope**: positive rate of person’s age w.r.t time. This can show which age group is most affected.

**Survey 3:**

**Purpose:** Particle modeling of the spreading of coronavirus disease (covid19)

**Author:** INFN-TIFPA Trento institute of fundamental physics and applications, Italy. Weizmann institute of science

**Methodology:** Monte Carlo based Algorithm (To predict the virus infection rate for different population densities using the most recent epidemic data).In this survey, they tested three different time-cyclic patterns of lock down patterns. They researchers modeled this surveys a set of classical particles based on health status (susceptible to infection, infected & contiguous, Recovered/died)

**Constraints:** tested with different constraints:

* With or without moderate social distancing on days 51-200
* With or without infection from unknown sources.
* With or without start social distancing for symptomatic patients after 14 days.

**Results:** For covid19, they observed that each infection directly generates 2-4 more infections in the absence of counter measures like social distancing. This survey uses the data of total number of active cases in Sweden. They have made predictions for population density of 3500 households.

**Survey 4:**

**Purpose:** Modeling the role of respiratory droplets inCovid-19 type pandemics.

**Authors:** DugaldJ.M.Thomson,andDavidR.Barclay(The Journal of the Acoustical Society of America, Institute for Aerospace Studies,University of Toronto,Toronto,Ontario M3H5T6,Canada)

**ABSTRACT:** In this paper, we develop a first principles model that connects respiratory droplet physics with the evolution of a pandemic such as the ongoing Covid19 . The model has two parts. First, we model the growth rate of the infected population based on a reaction mechanism. The advantage of modeling the pandemic using there action mechanism is that the rate constants have sound physical interpretation.

**Methodology:** The experiments with isolate evaporating droplets were conducted in a contact-less environment of an ultrasonic levitator(tec5) to discount boundary effects, generally present in suspended, pendant, or sessile droplet setups.

**RESULTS:** To validate the model, few targeted experiments were conducted to observe isolated levitated droplets evaporating in a fixed ambient condition. Particularly, the droplets with (1%w/w) NaCl solution vaporized to shrink to 30% of its initial diameter during the first stage of evaporation. Here after, a plateau like stage is approached due to increased solute accumulation near the droplet’s surface, which inhibits the diameter from shrinking rapidly. However, shrinkage does occur (untilDs/Ds,0≈0.2) as the droplet undergoes a sol–gel transformation. The final shape of the precipitate is better observed from the micrographs presented.

**CONCLUSION:** This is the first model that utilizes the structure of a chemical reaction mechanism to connect the pandemic evolution equations with respiratory droplet lifetime by first principles modeling of the reaction rate constant. However, it must be recognized that the model assumes conditions where transmission occurs solely due to inhalation of infected respiratory droplets alongside many others amplifying assumptions.

**Survey 5:**

**Purpose:** Diagnosis of covid patient (Analysis of covid positive or negative).

**Author:** M. Rubaiyat Hossain Mondal, Subrato Bharati, Prajoy Podder, Priya Podder

**Methodology:** Feature selection and classification and regression algorithms using hemoglobin, platelets, Hematocrit, mean platelets volume attributes.

**Result:** Gives the test result for SARS-COV-2(positive/negative)

**Loop holes/future scope:**

1. To predict how the virus spreads that is the transmission possibility. To provide information about possible zones that may have clusters of the virus. This will identify possible risky zones and people may avoid these areas.

2. To develop a system that will continuously observe the conditions of a suspected person, and to automatically predict whether he/she has COVID-19 or not. This kind of warning system can help to detect cases of COVID-19 and reduce the number of undetected cases. For this, the 28 symptoms of previous patients have to be stored in a repository. For example, the images of CT scans of normal people and coronavirus affected patients can be stored for training the system, while any new patient's CT scan image can be compared with the training images to find the similarity score. The main features of these patients can be found out from the application of machine learning. With the increase in the number of patient data, the training dataset will increase making the system's prediction work better.

3. To find the basic reproductive number of this virus.

4. To find the effect on pregnancy of different stages on COVID-19 patients and their newborns.

5. To find the effect of this virus on patients having heart disease, lungs problem, kidney disease, liver disease, cancers and other diseases.

6. To find the effect of isolation or quarantine on the spread of this coronavirus. For example, we can analyze the data of China, where the spread of this disease has slowed down after applying strict quarantine measures.

7. To find the effect of travel ban and school closure on the spreading of the virus.

8. To predict the effect of different vaccinations and treatments on COVID-19 patients. This prediction can be done by analyzing datasets of other viral diseases that have vaccination/treatment attributes.

**1.5 Problem Statement**

To design and develop a system that will predict whether a person has COVID-19 or not, Categories the level of COVID-19 positive into severe or recoverable (like stage I, stage II, stage III) and find the explanation behind the used deep learning model using explainable AI.

**2. Proposed Methodology**

**2.1. Description of Proposed System**

Here, we are solving a medical application problem using machine learning and Data mining. Following is the overall structure of our proposed model. Here the x-ray image will be taken as input in the system and will be processed. Then as per system functionalities it will predict the result.

**2.2. Description of Target Users**

Any suspected covid-19 user can use this software to detect whether he/she is positive or negative using his chest x-ray.

**2.3. Advantages**

* The system will be able to predict covid-19 report rapidly.
* Easy to use, as only input x-ray image has to be put in system by the user.
* User will be able to know the probability his/her covid-19 status.
* Using this probability the user will be able to know his or her severity of stages(stage 1,2,3)

**2.4. Scope**

* Categorising the level of COVID positive into severe or recoverable (like stage I, stage II, stage III) in consideration with other symptoms too like diabetes, high pressure.
* Finding the reproductive number of viruses.
* Positive rate of person’s age w.r.t time. This can show which age group is most affected.
* Actual life span of virus within a living being and metal body with or without treatment and sanitization.
* Understanding how this virus protein interacts with our bodies.

**3. Software requirement Specification**

**3.1. Overview of SRS**

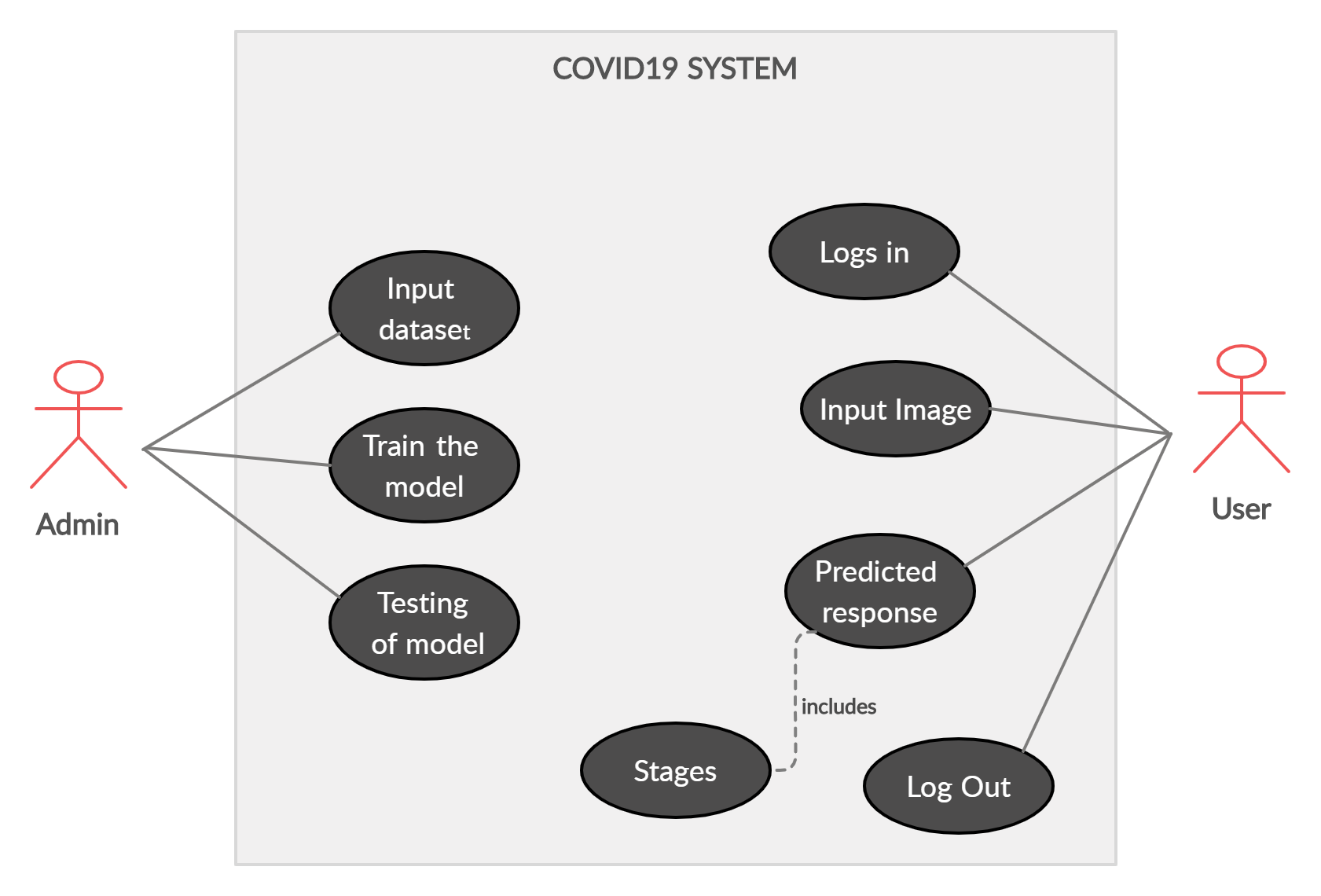
We are aiming to describe what the application will do and how it will be expected to perform. It also describes the functionality. The product needs to fulfill all stakeholders’ needs. It offers high grade definitions for the function al and non functional requirements of the application and can also include use cases that illustrate how user would interact with this system upon completion.

**3.2 Requirement Specifications**

**3.2.1. Functional Requirements**

1. System shall able to receive Input image corresponding to that gives a correct result.
2. System shall able to result Stages of COVID-19.
3. System shall generate the report to the front end/user end.

**3.2.2 Use Case Diagram**



**3.2.3. Use Case descriptions using scenarios**

The system made by us will be trained on a large input image dataset of both covid-19 positive and negative x-ray images. This will be helpful while categorizing the result into two classes. Then upon giving any x-ray input image the application will be able to predict the respective class and severity. In the other side of user, user logs in the system and will access the system then by giving proper x-ray input image the result will be predicted. And finally user can log out the system.

**3.2.4. Non Functional Requirements**

1. Authentication of user/researcher when he tries to login.
2. The system should give 85-90% accuracy.
3. The system should accept the input image less than or equal to 1MB.
4. The system should be able to respond within 3 second for the given input.
5. The system should accept the Jpg or png format image.
6. The system should be able to accept 1 input at a time rather than loading in a group by a single user.

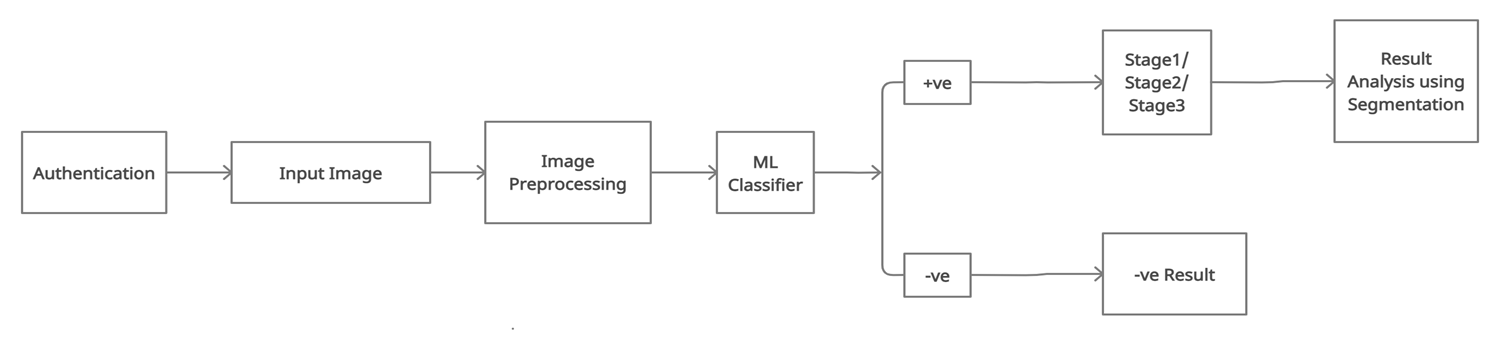
**3.3. Software and Hardware Requirement Specifications**

* Core i5 7th gen processor is required
* 64 bit Operating System
* 8GB RAM is preferred
* GPU is preferred
* Python needs to be installed
* Flask framework needs to be installed

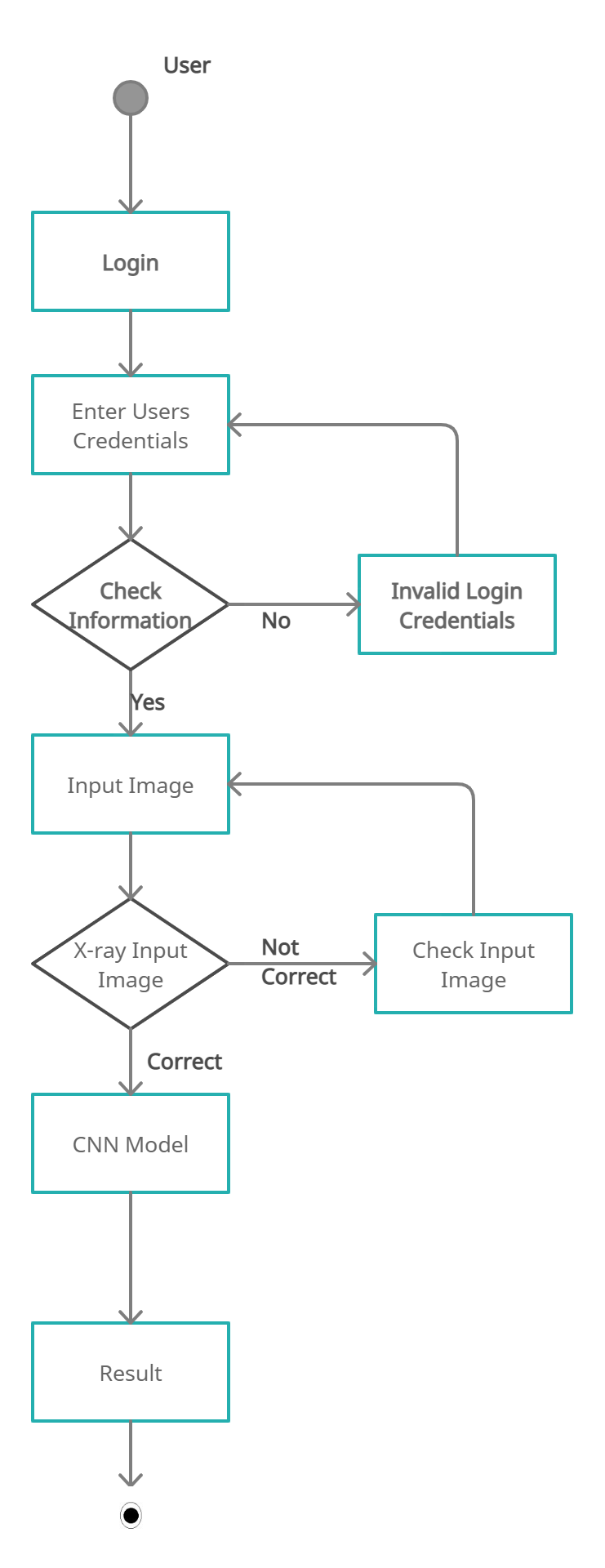
**4. System Design**

**4.1. Architecture of the System**

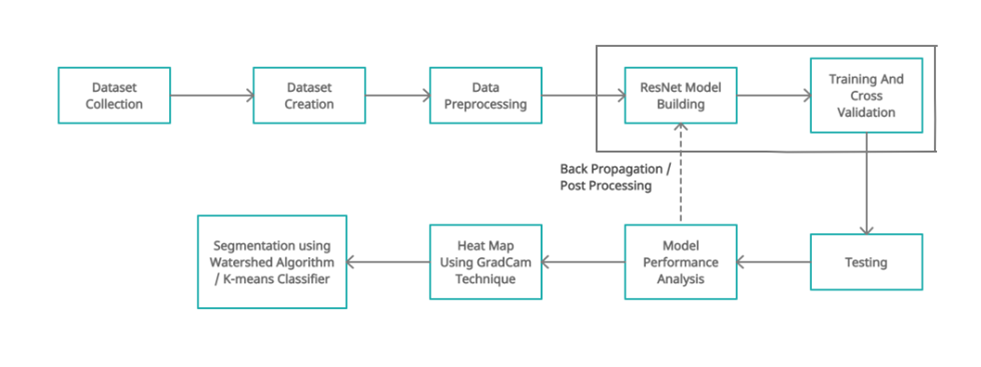
Here, we are following pipe and filter architecture design. Pipe and Filter is a simple architectural style that connects a number of components that process a stream of data, each connected to the next component in the processing pipeline via a pipe.



**4.2. Activity Diagram**

****

**4.3. Algorithmic Flowchart**

****

**4.4. Data Structure Used**

* Tuples
* List
* Array

**4.5. Dataset Description**

1. **Dataset Source / Features**

* We have collected two datasets of almost 1GB each.
* One is downloaded from github and other form kaggle.
* Using the two downloaded dataset, we created our required dataset for model building.
* Each dataset contains COVID-19 x-ray images with some patient details**.**

1. **Dataset analysis**

* Both the dataset contains various attributes of patient like name,age,sex e.t.c along with x-ray images
* Metadata.csv of first dataset contains 29 attributes and 950 tuples.
* Metadata.csv of second dataset contains 5 attributes and 5910 tuples.

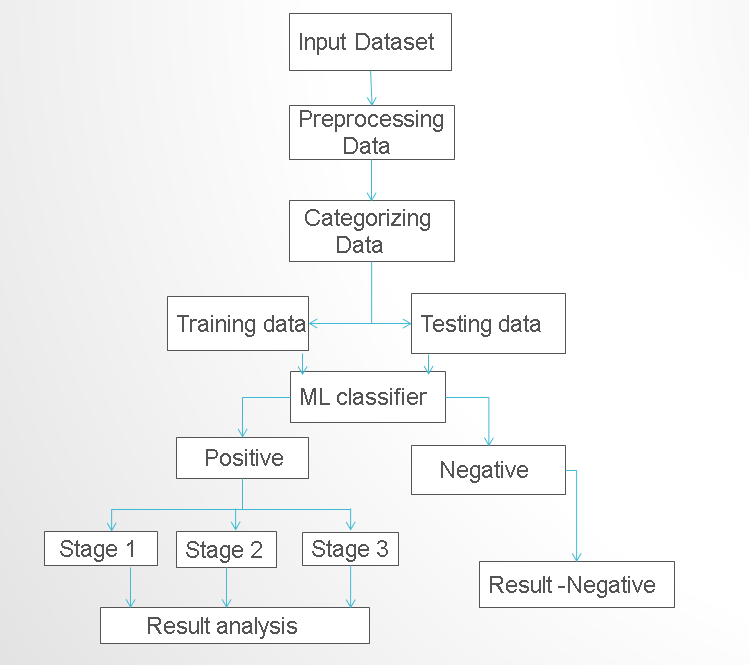
1. **Dataset Pre-Processing**

* There was need of filling some null values and removing some of the unnecessary tuples from metadata.
* Along with COVID-19 chest x-ray images, other disease images are also present in the downloaded dataset which is not necessary for us. So, only required images are extracted and pushed into our own dataset.
* 2639 x-ray images have been extracted to create our required dataset.
* All the images are resized to (100,100).
* All the images are converted to grayscale and normalized.

**5. Implementation**

**5.1. Proposed Methodology**

* Take x-ray image as input
* Preprocess image as per system requirements
* Take dataset
* Split the dataset as testing and training dataset
* Train the model on dataset
* Make classifier to classify the result
* If result is positive divide it into 3 categories as stage I, stage II, stage III and display result.
* If result is negative den display the result.

****

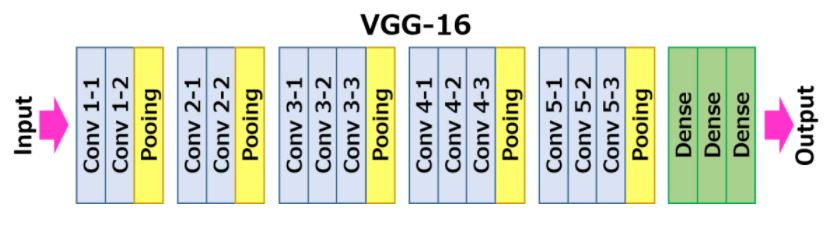
**5.2. Description of Modules**

**5.2.1. VGG16**

* The input to cov1 layer is of fixed size 224 x 224 RGB image.
* The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center).
* In one of the configurations, it also utilizes 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity).
* The convolution stride is fixed to 1 pixel; the spatial padding of conv.
* Layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3×3 conv. layers.
* Spatial pooling is carried out by five max-pooling layers, which follow some of the conv.  Layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2×2 pixel window, with stride 2.
* Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.

**Input:** X-ray Image

**Output:** Predicted class

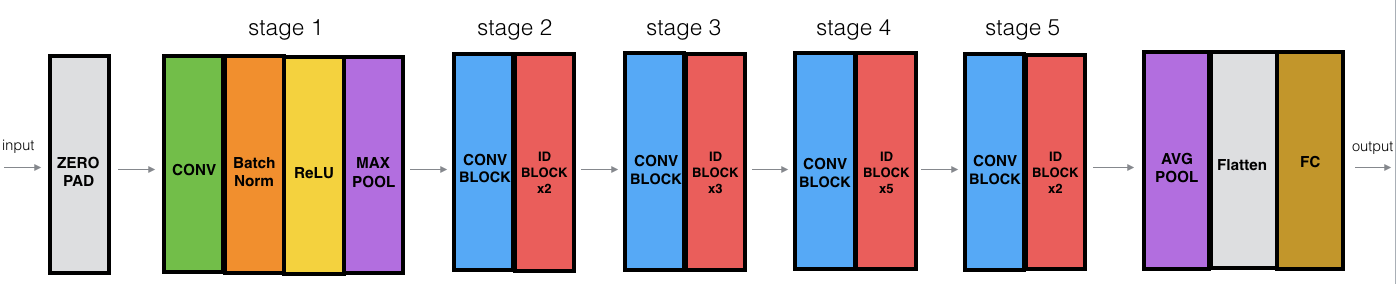
****

**5.2.2. ResNet50 Model**

* ResNet50 is a variant of [ResNet model](https://iq.opengenus.org/resnet/) which has 50 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8 x 10^9 Floating points operations.
* In residual learning, instead of trying to learn some features, we try to learn some residual.
* Residual can be simply understood as subtraction of feature learned from input of that layer. ResNet does this using shortcut connections (directly connecting input of nth layer to some (n+x)th layer.
* On this pre-trained model we have added six more convolution layers with activation function ReLU and 3 max pooling layers of kernel size (2, 2).
* In this binary classification till last dense layer is mapped with softmax activation function.
* This model has been trained and cross validated simultaneously. Model has been tested on trained model.
* Model has been evaluated through bias-variance trade off and confusion-matrix.

**Input:** X-ray Image

**Output:** Predicted class

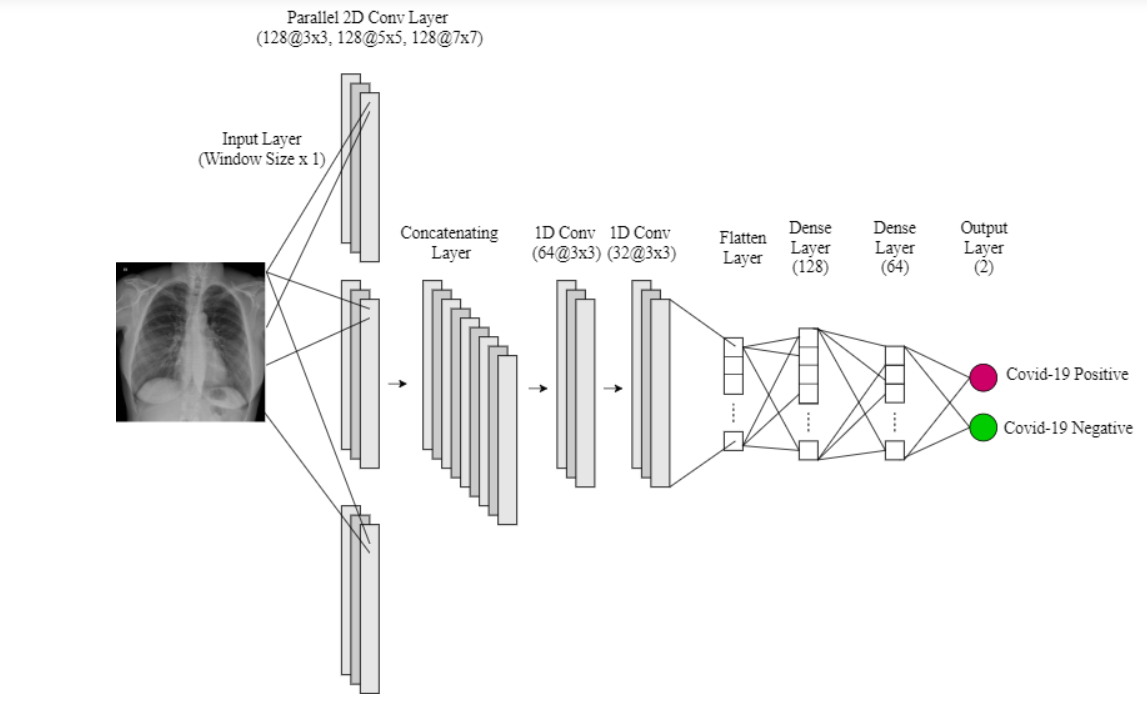


**5.2.3. Sequential Model**

* The sequential model is linear stack of layers. It is limited in that it does not allow us to create models that share layers or have multiple inputs or outputs.
* Sequential model can be generated by creating list of layers and passing through constructors or we can add layers using .add().

**Input:** X-ray Image

**Output:** Predicted class

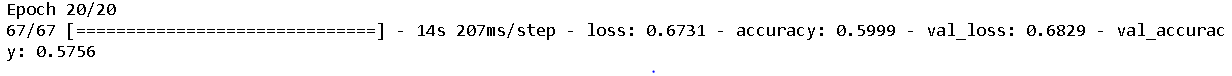
****

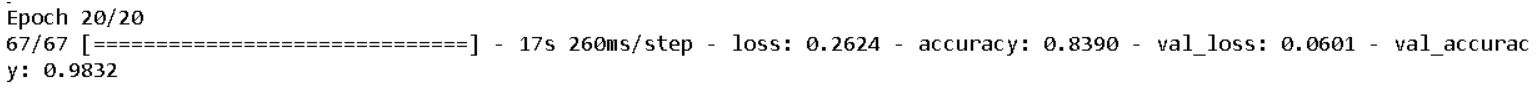
**6. Testing and Test Cases**

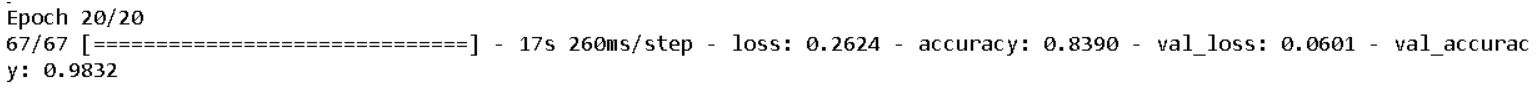
|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement id** | **Test id** | **Input description** | **Expected output** |
| **1.1** | **1.1.1** | **Input image is not an X-Ray image .** | **Display message “Input should be x-ray image”.** |
|  | **1.1.2** | **X-ray image**  **dimension< 100 \*100** | **Display message “Not recognizing an input image. Input image should be at least 100\*100 dimension”** |
| **1.2** | **1.2.1** | **Input X-ray image is covid-19 Positive.** | **Displays message**  **“Positive , corresponding stage of costiveness”** |
|  | **1.2.2** | **Input X-ray image is covid-19 Negative.** | **Display message “Negative”.** |
| **2.3** | **2.3.1** | **Input image size is greater than 1MB.** | **Display message “X-ray image should be less than or equal to 1mb** |
| **2.5** | **2.5.1** | **Input image format other than jpg or png.** | **Display message “Input X-ray image should be a png or jpg format”.** |

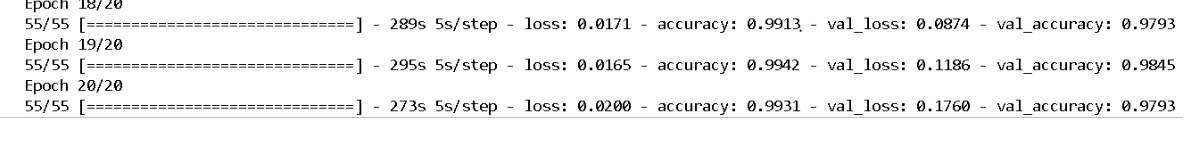
**7. Results and Discussions**

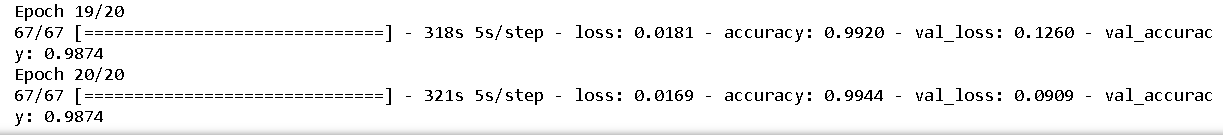
* **Sequential Model: Training Result**

****

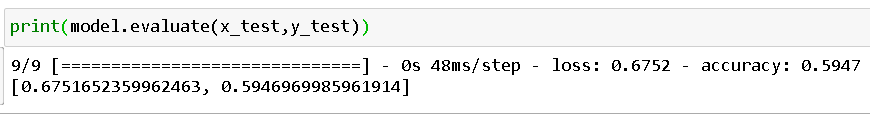
****

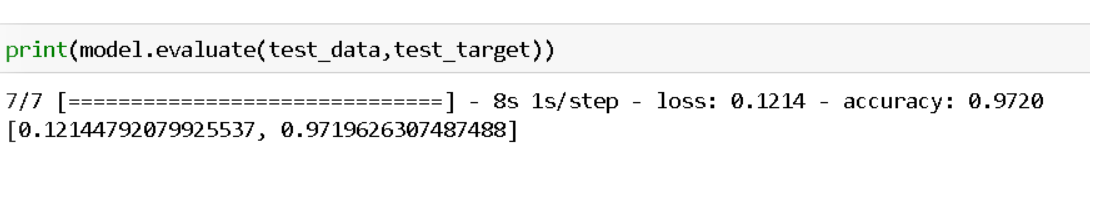
****

****

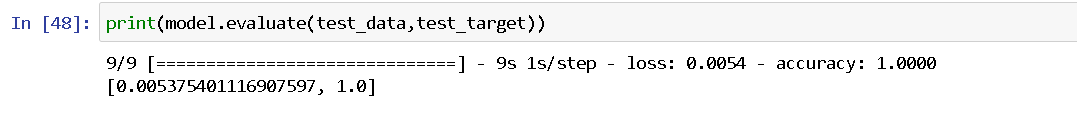
****

* **Sequential Model: Test Result**

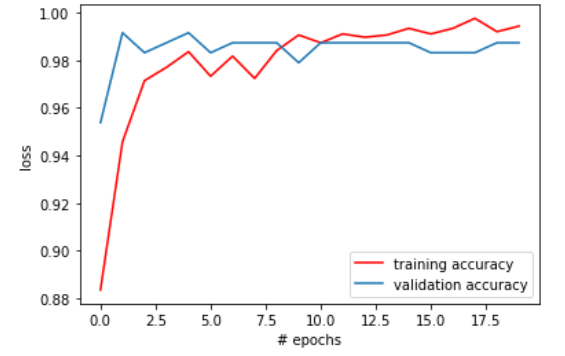
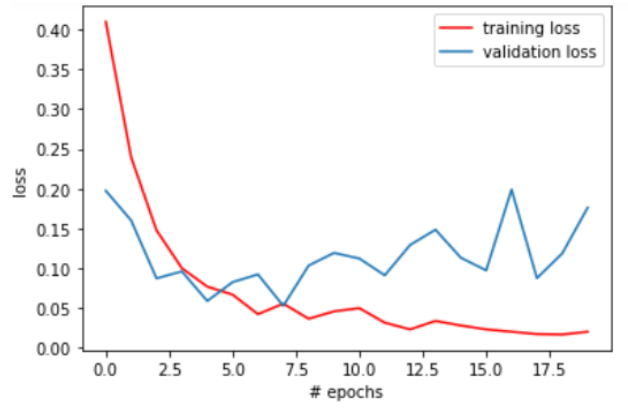
****

****

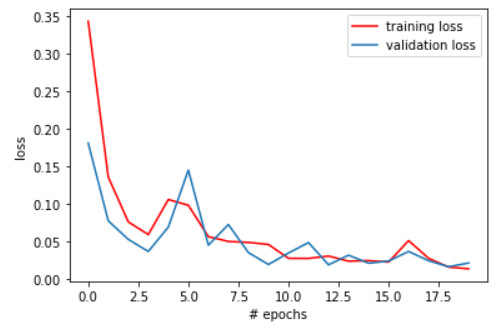
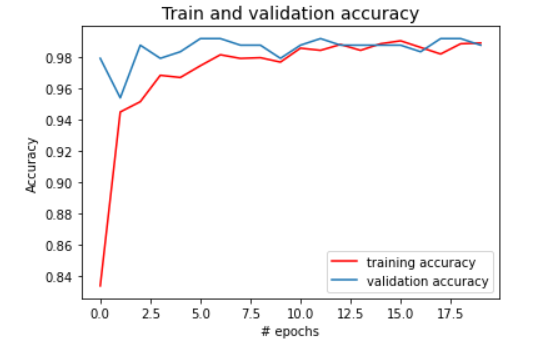
****

****

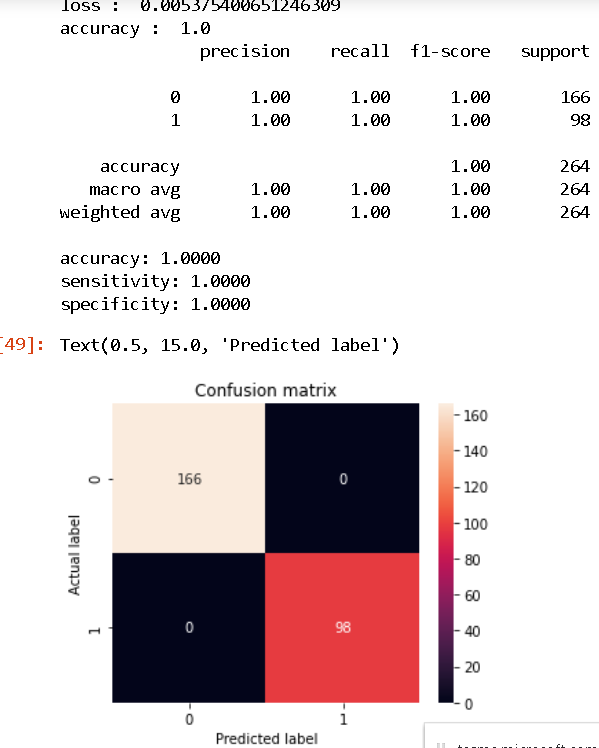
* **Sequential Model: Graphical representation 1 of accuracy and loss**

** **

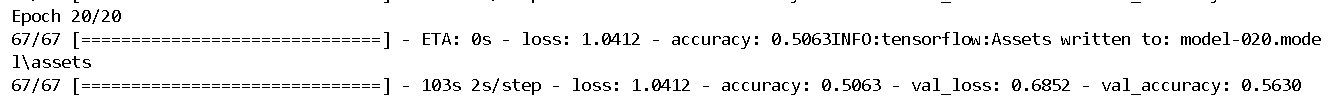
* **Sequential Model: Graphical representation 2 of accuracy and loss**

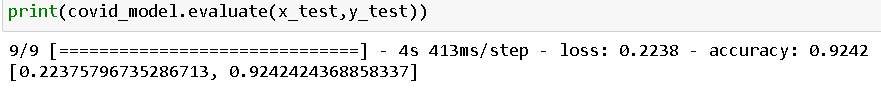
** **

* **Sequential Model: Confusion Matrix**

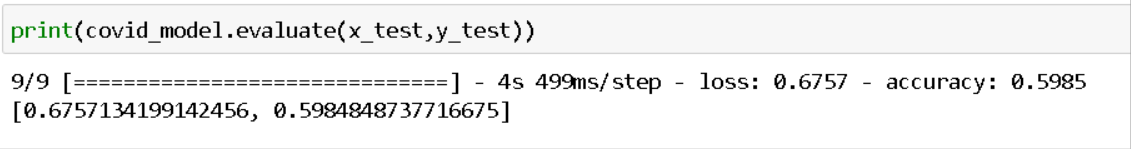
****

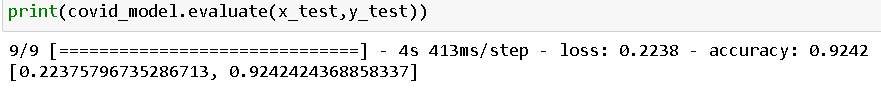
* **ResNet Model: Training Result**

****

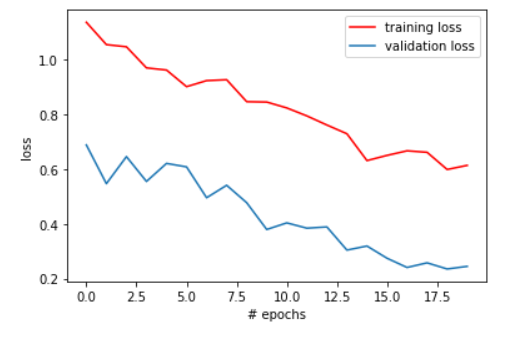
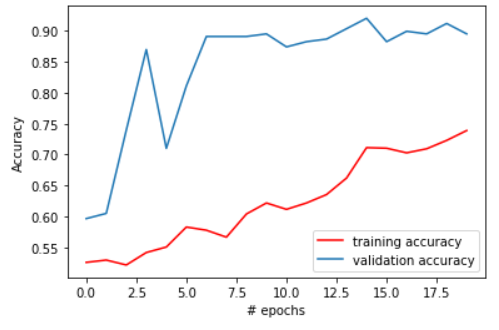
****

* **ResNet Model: Testing Result**

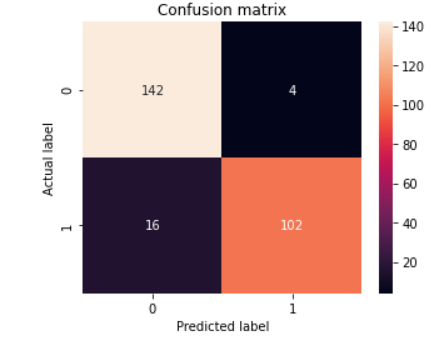
****

****

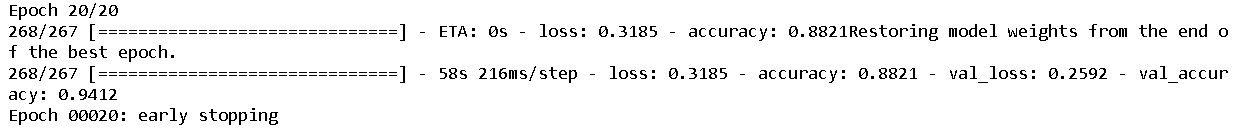
* **ResNet Model: Graphical representation of accuracy and loss**

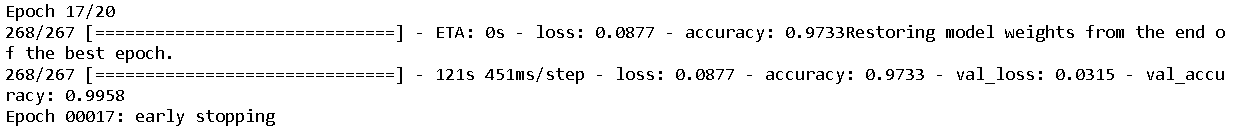
**** ****

* **ResNet Model: Confusion matrix**

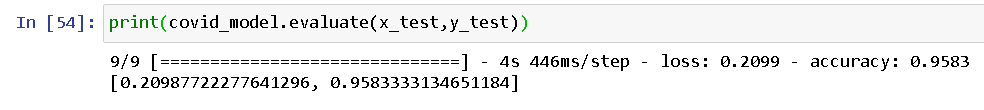
****

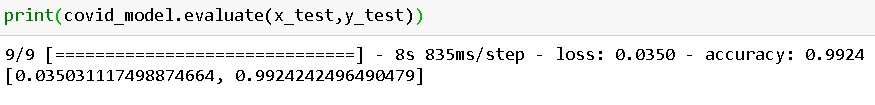
* **VGG16 Model: Training Result**

****

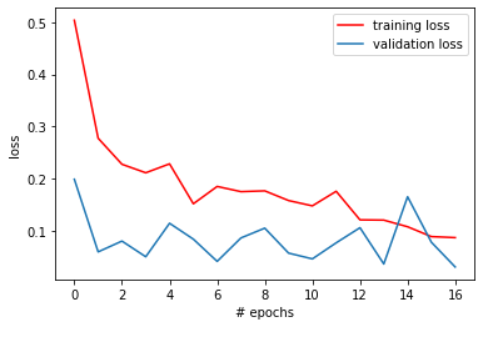
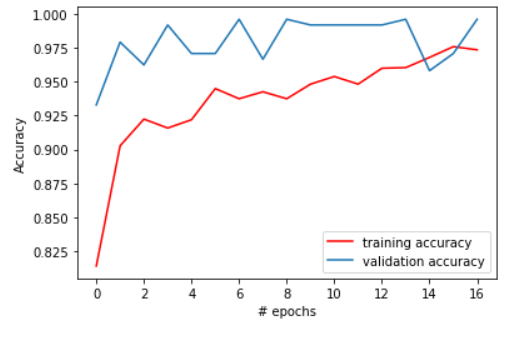
****

* **VGG16 Model: Testing Result**

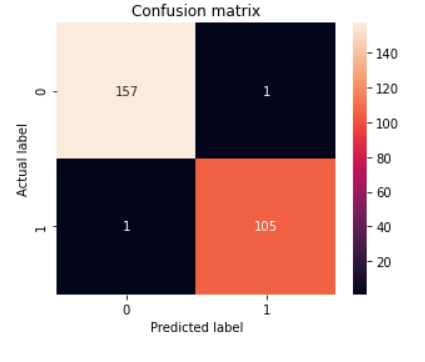
****

****

* **VGG16 Model: Graphical representation of accuracy and loss**

**** ****

* **VGG16 Model: Confusion matrix**

****

* **Novelty/Observations/Conclusions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Training Accuracy** | **Training Error** | **Testing Accuracy** | **Testing Error** |
| **Sequential model** | **99.4%** | **0.01** | **100%** | **0.005** |
| **ResNet50 model** | **72.9%** | **0.68** | **92.4%** | **0.22** |
| **VGG16 model** | **97.3** | **0.08** | **99.24%** | **0.03** |

**8. Conclusion and Future Scope**

* Built a Sequential CNN model of 17 layers.
* Obtained 99.4% accuracy on training and 100% accuracy on testing.
* Built a ResNet model of accuracy 73.8% accuracy on training and 93.5% on testing which is yet to improve.
* Using Grad cam technique for generating heat map on the infected region and will finally segment the infected region.