CHAPTER 1

INTRODUCTION

As 2019 ended, coronavirus disease, known as COVID-19, started proliferating all over the world and has created an alarming situation worldwide. The virus originated in Wuhan, a town in Eastern China, in December 2019. In 2020, it was declared by the World Health Organization (WHO) as a "Public health emergency of international concerns", and by March 2020 they classified the disease as a pandemic. The on-going COVID-19 outbreak made healthcare systems across the globe to be in the edge of the battle. Recent stats indicate that more than 660+ million confirmed cases are diagnosed globally, and the numbers keep rising. The early and auto diagnosis helps people to be precautious. One of the ways to combat this disease is the effective screening of infected patients. However, COVID-19 provides a similar pattern with diseases, such as pneumonia, and can misguide even very well-trained physicians.

Automating the diagnosis of many diseases nowadays has been based on artificial intelligence, which has proven its efficiency and high performance in automatic image classification problems through different machine learning approaches. Artificial Intelligence makes calculations and predictions based on analysing the input data, then performs tasks that require human intelligence such as speech recognition, translation, visual perception, and more whereas machine learning defines models that have the ability to learn and make decisions by using large amounts of input data examples. Deep Learning is a technique for implementing machine learning that mostly focus on the automatic feature extraction and classification of images and have shown great achievement in many applications, especially in health care . Deep learning efficiently generates models that produce more accurate results in predicting and classifying different diseases using images as in lung cancer, pneumonia, and recently COVID-19 diagnosis, without requiring any human intervention. The main reason for using deep learning is that deep learning techniques learn by creating a more abstract representation of data as the network grows deeper.

1.1 CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks (CNNs) are a type of artificial neural network designed for processing images and other grid-like data. They use specialized layers called convolutional

layers that apply filters to extract features from the input data, and then use those features to make predictions about the content of the image or data. This makes CNNs highly effective for tasks like object recognition, classification, and segmentation in fields such as computer vision and image processing.

CNN has its own advantages and disadvantages.

Advantages of CNN are-

- No human supervision.
- High accuracy at image recognition.
- Weight sharing.
- Minimise computation.
- Same knowledge across all image location.

Disadvantages of CNN are-

- A lot of training data is needed.
- CNNs tend to be much slower.
- Training process takes a long time.
- Fail to encode the position and orientation of objects.

1.2 OVERVIEW

The project aims to leverage the power of chest X-ray imaging to aid in both the diagnosis and prognosis of several lung diseases, including COVID-19, pneumonia, tuberculosis, and lung cancer. The early and automated detection of these diseases is crucial for timely and effective treatment and can help save lives.

To achieve this, a suitable Convolutional Neural Network (CNN) model is selected for the identified dataset. The CNN model is a deep learning architecture that is particularly well-suited for image analysis tasks, making it an excellent choice for this project. The model will extract features from the images, which will form the foundation for distinguishing between the different lung diseases. Finally, these extracted feature sets will pass through a classifier for prediction. The classifier will be trained on a large dataset of chest X-ray images, enabling it to accurately predict the presence of various lung diseases based on the extracted features. Overall, this project has the potential to significantly improve the early and accurate detection of lung diseases, ultimately leading to better patient outcomes.

1.3 PROBLEM STATEMENT

Deep learning has revolutionized medical image segmentation, enabling the development of powerful diagnostic tools. With the COVID-19 pandemic, early detection of the disease is crucial, and deep learning-based image classification models can help in this regard. Hence, a CNN-based image classification model is built which can categorize chest X-ray images into Normal, COVID-19, Pneumonia, Tuberculosis, and Lung Cancer. This allows healthcare professionals to quickly and accurately identify patients with respiratory diseases and provide appropriate treatment. This technology can potentially save lives and improve patient outcomes.

1.4 OBJECTIVES

The Main objective of the project is to detect Lung diseases in patients using Chest X-Ray Images through the following steps:

- Upload X-Ray Images - The system should be capable of getting X-Rays from users that will be utilized by the Model.
- Detection of Hybrid Model - The system should be able to detect the COVID-19, Pneumonia, Tuberculosis and Lung Cancer within the X – Ray images that users have uploaded.
- Display Results - The system should be able to give information that the user can appropriately understand and gain insight from it.

1.5 EXISTING SYSTEM

The existing system uses Convolutional Neural Networks (CNNs), a type of deep learning model commonly used for image classification tasks, to accurately identify lung conditions in chest X-Ray images. By training the CNN on a dataset of labelled X-Ray images, the system can distinguish between normal lungs, pneumonia-affected lungs, COVID-19-affected lungs, and lungs affected by both pneumonia and COVID-19. This can be particularly useful in identifying patients with these conditions, allowing for timely medical intervention and reducing the risk of transmission. Overall, this system represents a promising application of deep learning in healthcare, demonstrating the potential for machine learning models to assist medical professionals in diagnosis and treatment.

1.6 PROPOSED SYSTEM

The proposed system for classifying lung diseases from chest X-ray images is a valuable tool in identifying the presence of normal, COVID-19, pneumonia, tuberculosis, or lung cancer with high accuracy of 97%. The process includes four main steps: pre-processing, feature selection, feature extraction, and classification. These steps involve enhancing the quality of images, selecting important features, extracting meaningful information from images, and classifying them into appropriate categories. The additional feature of sending SMS alerts to individuals with information about the disease diagnosed and the suitable hospital is also an added benefit. This system can provide quick and accurate results, allowing for early diagnosis and timely treatment of lung diseases, which can ultimately save lives.

CHAPTER 2

LITERATURE SURVEY

[1] Kanakaprabha. S, D. Radha, "Analysis of COVID-19 and Pneumonia Detection in Chest X-Ray Images using Deep Learning", 2021 International Conference on Communication, Control and Information Sciences (ICCISc) | 978-1-6654-0295-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCISc52257.2021.9484888 - The proposed work aims to detect COVID-19 patients and Pneumonia patients from X-Rays which is one of the medical imaging modes to analyze the health of patient's lung inflammation. The model detects COVID-19 patients and Pneumonia patients on the real-world dataset of lung X-Ray images. Images are pre-processed and trained for various classifications like Normal, COVID-19 and Pneumonia. After pre-processing, the detection of the disease is done by selecting the appropriate features from the images in each of the datasets. The accuracy of detection of COVID -19 is 95 % which is a plus about this model and the current requirement of this pandemic with limited cost and computations. This model is more appropriate for medical practitioners, researchers etc.

[2] Afonso U. Fonseca et.al., "Screening of Viral Pneumonia and COVID-19 in Chest X-ray using Classical Machine Learning", 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC) | 978-1-6654-2463-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/COMPSAC51774.2021.00294 - The main goal of this work is to present a robust, lightweight, and fast technique for the automatic detection of COVID-19 from CXR images. They extracted radiomic features from CXR images and trained classical machine learning models for two different classification schemes: i) COVID-19 pneumonia vs. Normal ii) COVID-19 vs. Normal vs. Viral pneumonia. They investigated extracted features from Haar wavelet transform to obtain a set of features representative of this disease. Their best binary classification model (*CxN*) reached 100% in all metrics calculated with a set of only 18 features, while minimum accuracy was 99.61%.

[3] Mohammad Farukh Hashmi et.al., “Efficient Pneumonia Detection in Chest Xray Images Using Deep Transfer Learning”, Diagnostics2020,10,417; doi:10.3390/ diagnostics 10060417 - A novel approach based on a weighted classifier is introduced, which combines the weighted predictions from the state-of-the-art deep learning models such as ResNet18, Exceptions, InceptionV3, DenseNet121, and MobileNetV3 in an optimal way. This approach is a supervised learning approach in which the network predicts the result based on the quality of the dataset used. Transfer learning is used to fine-tune the deep learning models to obtain higher training and validation accuracy. Partial data augmentation techniques are employed to increase the training dataset in a balanced way. The proposed weighted classifier is able to outperform all the individual models. Finally, the model is evaluated, not only in terms of test accuracy, but also in the AUC score. Through this paper, the automatic detection of pneumonia in chest X-ray images using deep transfer learning techniques was proposed. The deep networks, which were used in our methodology, had more complex structures, but fewer parameters and, hence, required less computation power, but achieved higher accuracy. Transfer learning and data augmentation were used to solve the problem of overfitting, which is seen when there is insufficient training data, as in the case of medical image processing. Further, to combine different architectures efficiently, a weighted classifier was proposed.

[4] Saeed S. Alahmari et.al., “A Comprehensive Review of Deep Learning-Based Methods for COVID-19 Detection Using Chest X-Ray Images”, Received 30 August 2022, accepted 14 September 2022, date of publication 20 September 2022, date of current version 28 September 2022. Digital Object Identifier 10.1109/ACCESS.2022.3208138 - In this paper, they review deep learning approaches for COVID-19 detection using chest X-ray images. They found that the majority of deep learning approaches for COVID-19 detection use transfer learning. A discussion of the limitations and challenges of deep learning in radiography images is presented. Finally, they provide potential improvements for higher accuracy and generalizability when using deep learning models for COVID-19 detection. This review paper summarizes deep learning approaches for COVID-19 detection using chest X-ray images. Furthermore, this article summarizes the available datasets of chest X-ray images used in the reviewed approaches. Also, the article presents a discussion of the reviewed approaches and highlights directions for improvement. They also discuss the future direction of COVID-19 detection using chest X-ray images.

[5] Vinod Kumar, Anil Saini, "Detection system for lung cancer based on neural network: X-Ray validation performance", International Journal of Enhanced Research in Management & Computer Applications, ISSN: 2319-7471 Vol. 2 Issue 9, Nov.-Dec., 2013, pp: (40-47) - In this paper, the author represents Lung Cancer Detection System for finding of lung cancer by analyzing chest X-rays with the help of image processing mechanisms. This system assists radiologists with their X-ray image interpretation of lung cancer. This paper presents a neural network-based approach to detect lung cancer from raw chest X-ray images. The author uses an image processing technique to denoise, to enhance, for segmentation and edge detection in the X-ray image to extract the area, perimeter and shape of nodule. These extracted features are considered as the inputs of neural network to train and to verify whether the extracted nodule is a malignant or non-malignant. This research work concentrates on detecting nodules, early stages of cancer diseases, appearing in patient's lungs. They developed a lung cancer detection system for early detection of lung cancer by studying lung Xray images using a number of steps. The approach starts by extracting the lung regions from lung X-Ray image using several image processing techniques in MATLAB including binary image, erosion, dilation, gaussian filter and median filter. Start with binary image including of threshold technique that is used in the initial steps in the extraction process to convert X-Ray image into binary image, which is faster and user independent. After the extraction step, the region growing segmentation algorithm is applied on extracted lung regions. Then the shape of nodule is calculated using shape formula with the help of area and perimeter of nodule. Finally, the extracted features help to find the cancerous and non-cancerous candidate in X-Ray images.

[6] Stefanus Kieu Tao Hwa, Abdullah Bade, et al., " Tuberculosis detection using deep learning and contrast-enhanced canny edge detected X-Ray images.", IAES International Journal of Artificial Intelligence (IJ-AI) Vol. 9, No. 4, December 2020, pp. 713~720 ISSN: 2252-8938, DOI: 10.11591/ijai.v9.i4.pp713-720. - This paper proposes an approach for tuberculosis (TB) detection using deep learning and Contrast-Enhanced Canny edge detected x-ray images. The authors note that previous ensembles combining CNNs trained on similar features have limited the performance of the classifiers. To address this, the authors propose ensembles that combine CNNs trained on different features extracted from a different set of images, the Enhanced and Edge images. VGG16 and InceptionV3 were selected and employed as classifiers, and the results indicate that using ensembles of classifiers trained on multiple types of features extracted from various types of images improved the detection

accuracy, sensitivity, and specificity. Future work will focus on classifying chest x-ray images based on TB severity and investigating more features that would further improve classifier performance.

CHAPTER 3

SYSTEM REQUIREMENTS & SPECIFICATIONS

3.1 Hardware Requirements

- System : Pentium IV 2.4 GHz/intel i3/i4.
- Hard Disk : 40GB.
- Monitor : 15 VGAColor.
- RAM : 512 Mb Minimum

3.2 Software Requirements

- Operating system : Windows XP/ Windows 10 or More
- Software Packages : Tensorflow 1.14 , OpenCv
- Coding Language : Python.

CHAPTER 4

SYSTEM DESIGN

4.1 Design Overview

Design overview explains the architecture that would be used for developing a software product. It is an overview of an entire system, identifying the main components that would be developed for the product and their interfaces.

4.2 CNN System Architecture

A CNN is a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data. It is another type of neural network that can uncover key information in both time series and image data. For this reason, it is highly valuable for image-related tasks, such as image recognition, object classification and pattern recognition. To identify patterns within an image, a CNN leverages principles from linear algebra, such as matrix multiplication. A deep learning CNN consists of three layers: a convolutional layer, a pooling layer and a fully connected (FC) layer. The convolutional layer is the first layer while the FC layer is the last. From the convolutional layer to the FC layer, the complexity of the CNN increases. It is this increasing complexity that allows the CNN to successively identify larger portions and more complex features of an image until it finally identifies the object in its entirety.

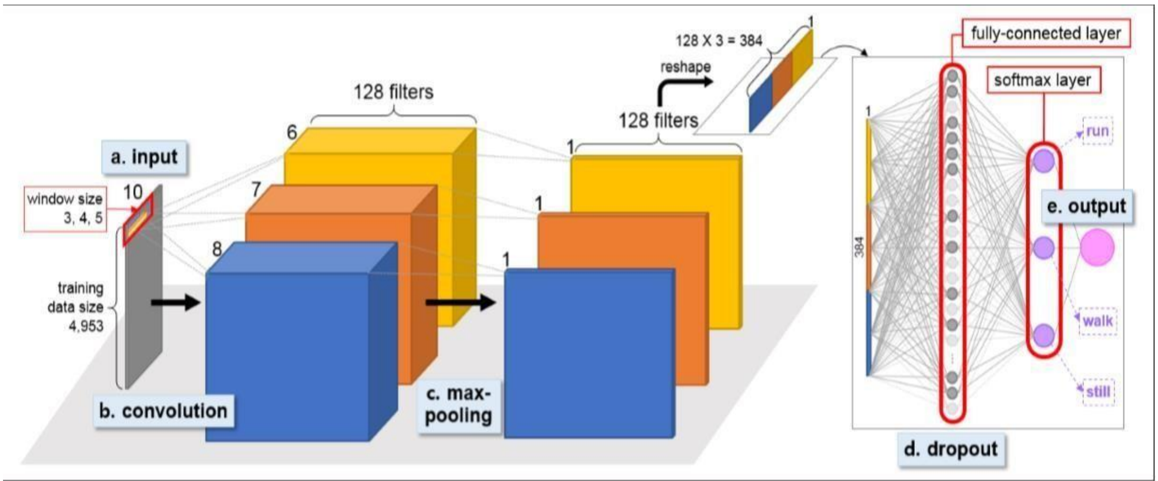

Fig 4.2.1 Convolutional Neural Network General Architecture

A CNN is composed of several kinds of layers are:

Convolutional layer: creates a feature map to predict the class probabilities for each feature by applying a filter that scans the whole image, few pixels at a time.

Pooling layer (down-sampling): scales down the amount of information the convolutional layer generated for each feature and maintains the most essential information (the process of the convolutional and pooling layers usually repeats several times).

Fully connected input layer: flattens the outputs generated by previous layers to turn them into a single vector that can be used as an input for the next layer.

Fully connected layer: Applies weights over the input generated by the feature analysis to predict an accurate label.

4.3 Flowcharts and Diagrams

4.3.1 Data flow Diagram

A dataflow outline is a tool for referring to knowledge progression from one module to the next module. This graph gives the data of each module's info and yield. The map has no power flow and there are no circles at the same time.

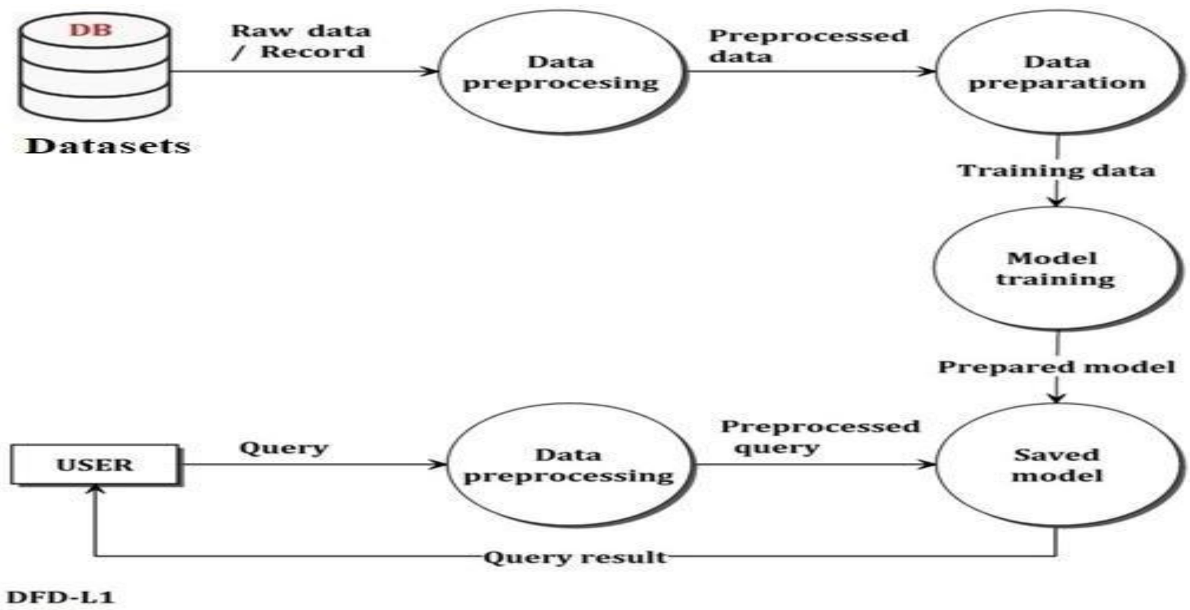

Fig 4.3.1 Data Flow Diagram

4.3.2 Use Case Diagram

Use case diagram is the boundary, which defines the system of interest in relation to the world around it. The actors, usually individuals involved with the system defined according to their roles. The use cases, which are the specific roles played by the actors within and around the system.

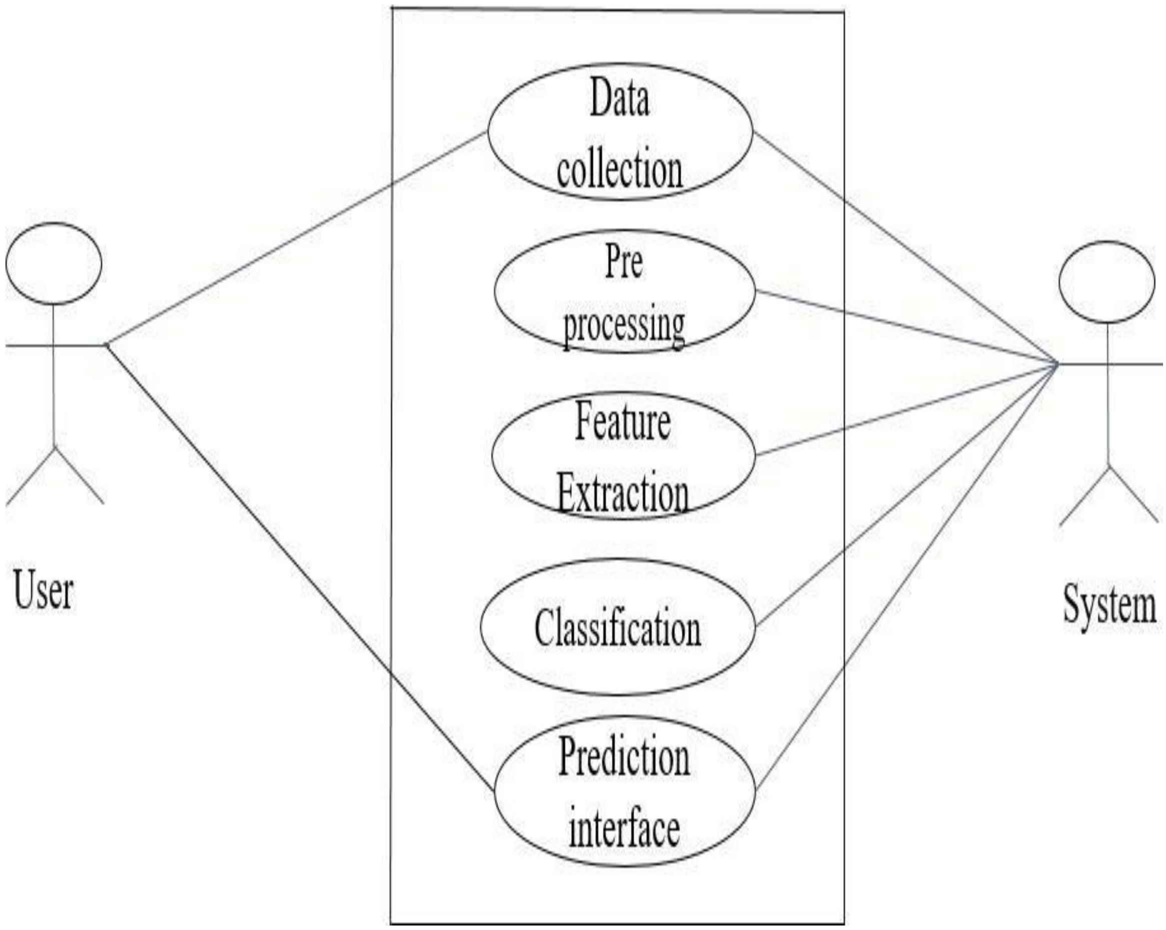

Fig 4.3.2 Use Case Diagram

4.3.3 Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place.

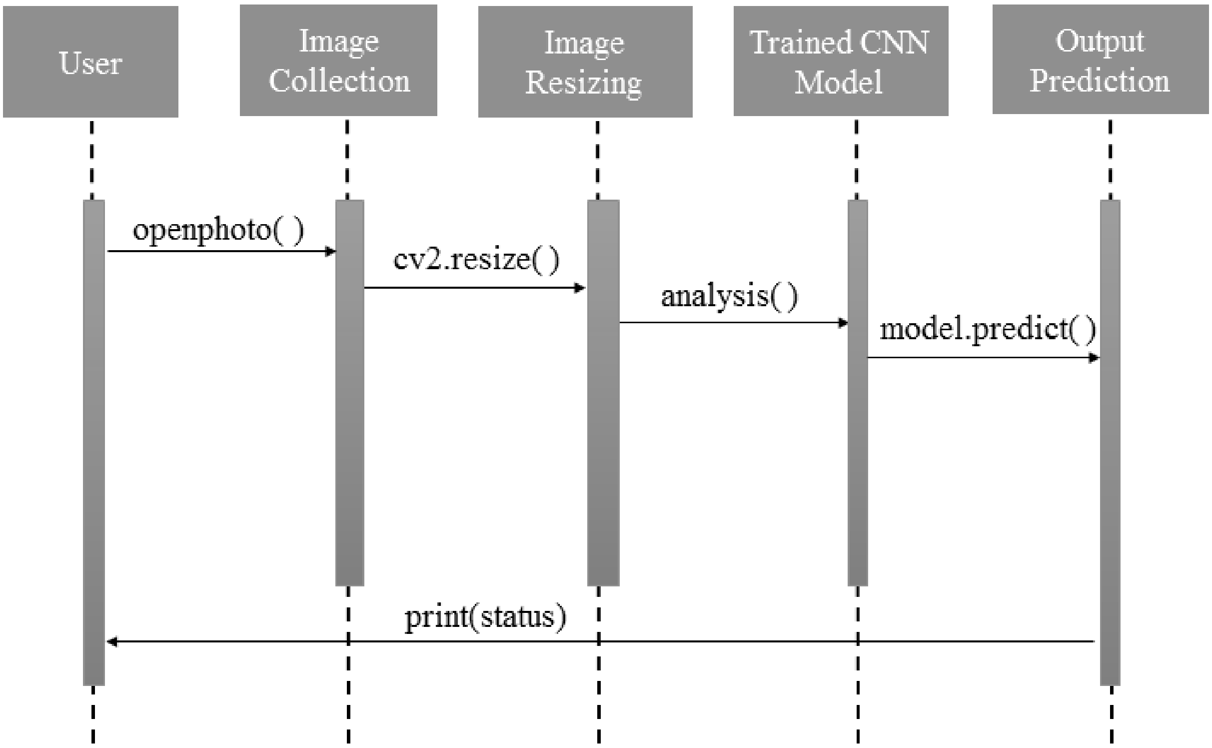

Fig 4.3.3 Sequence Diagram

4.3.4 Data Flow Diagram for Pre-processing

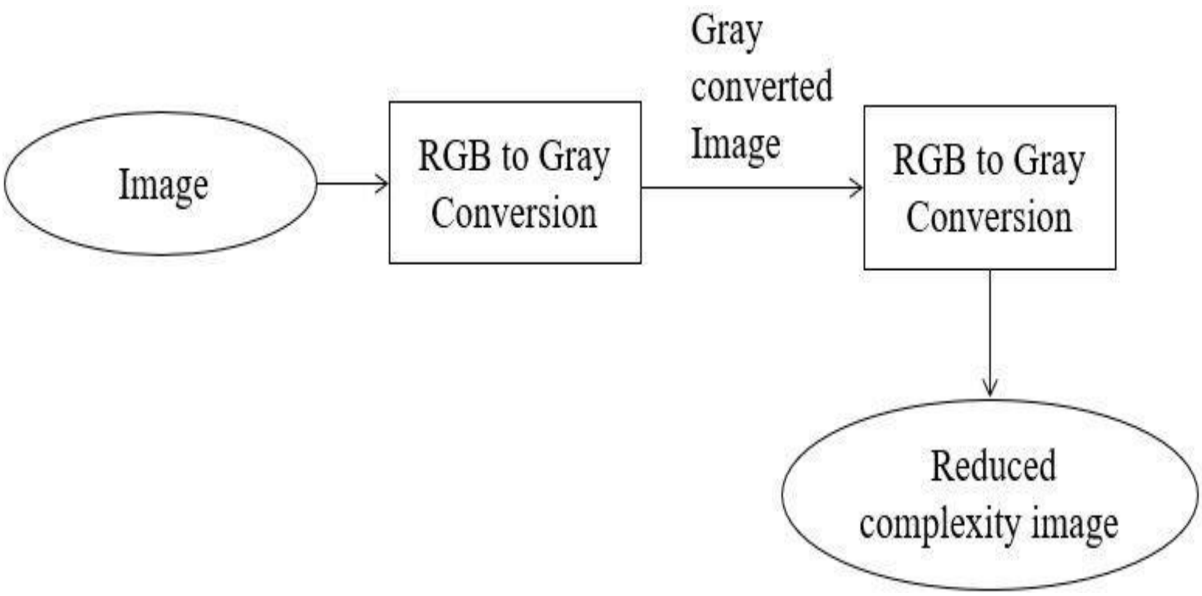

Fig 4.3.4 Data Flow Diagram for Pre-processing

4.3.5 Data Flow Diagram for Identification

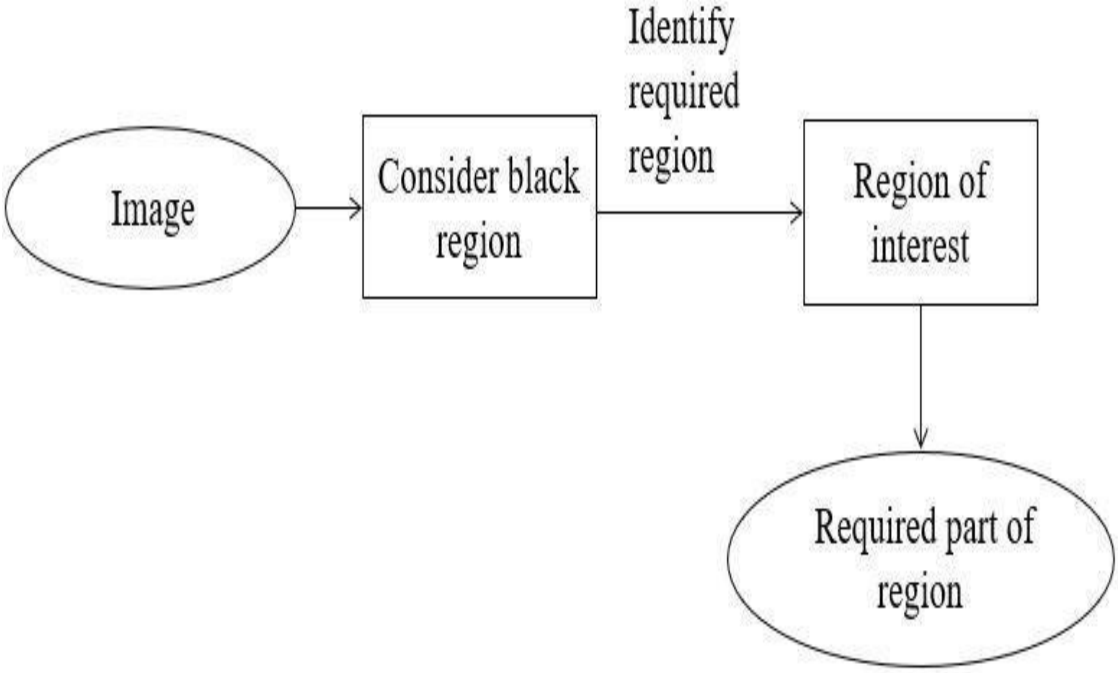

Fig 4.3.5 Data Flow Diagram for Identification

4.3.6 Data Flow Diagram for Feature Extraction

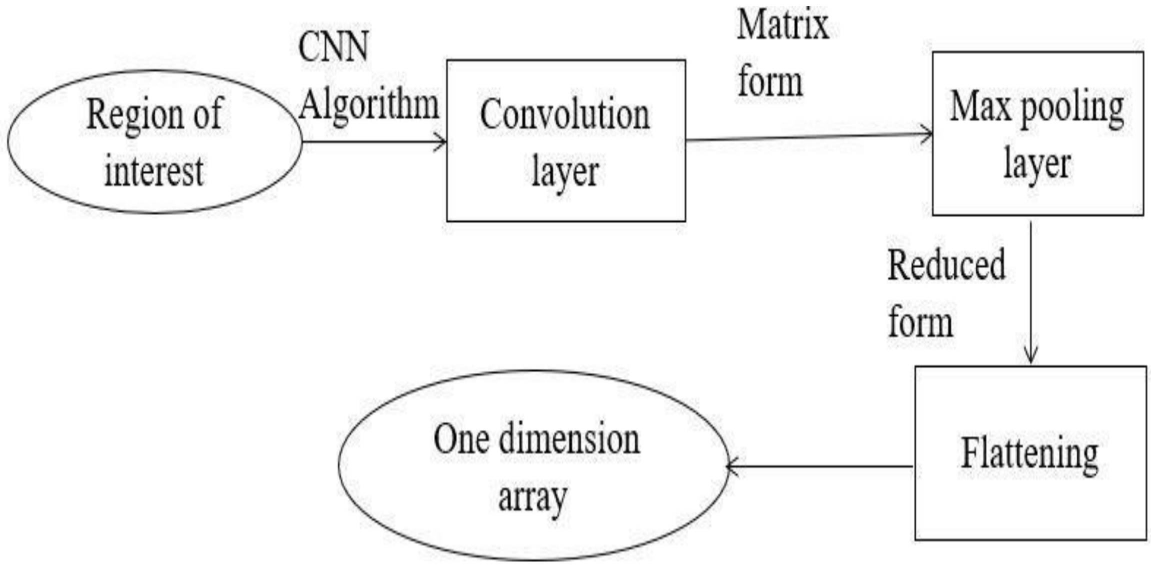

Fig 4.3.6 Data Flow Diagram for Feature Extraction

4.3.7 Data Flow Diagram for Classification and Detection

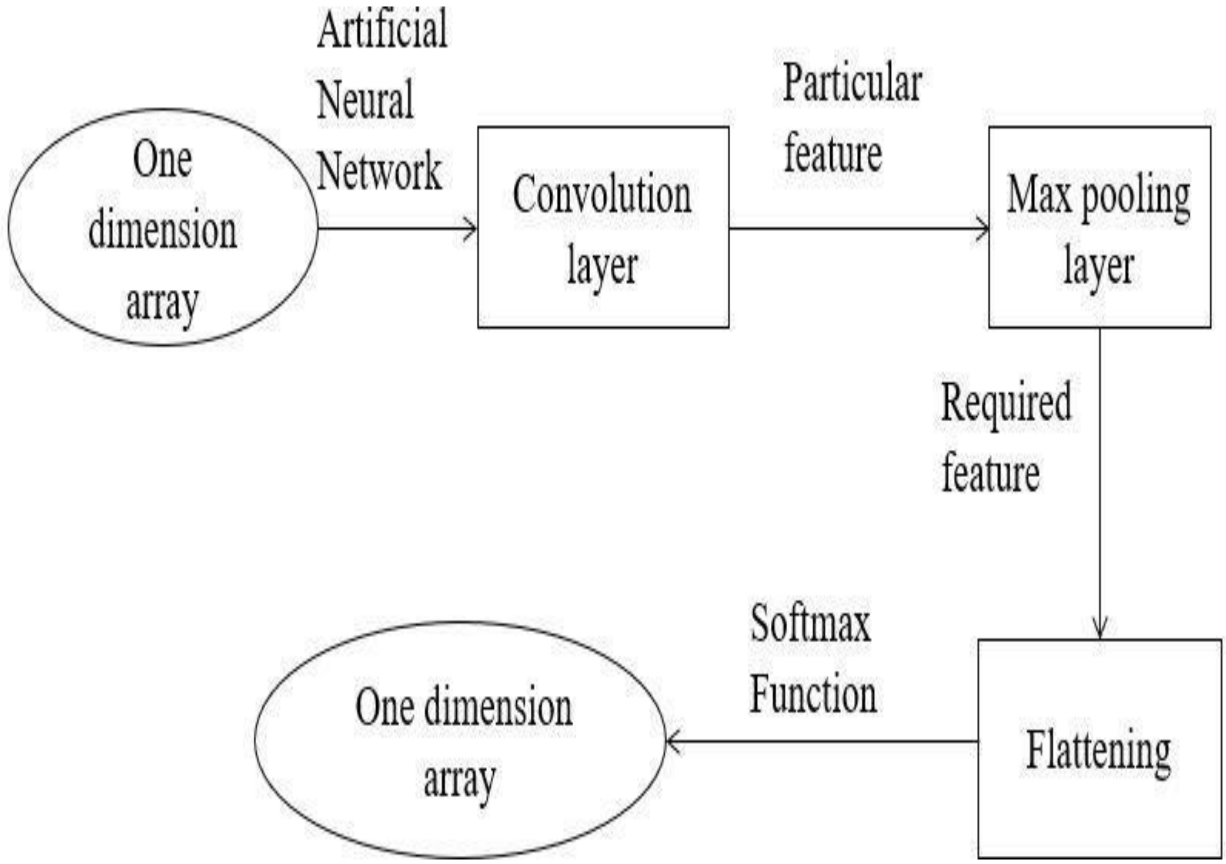

Fig 4.3.7 Data Flow Diagram for Classification and Detection

CHAPTER 5

IMPLEMENTATION

Implementation is the process of converting a new system design into an operational one. It is the key stage in achieving a successful new system. It must therefore be carefully planned and controlled.

5.1 Module specification:

Module Specification is the way to improve the structure design by breaking down the system into modules and solving it as independent task. By doing so the complexity is reduced and the modules can be tested independently. The number of modules for our project is, namely collection, preprocessing, segmentation, feature extraction, training and classification. So each phase signify the functionalities provided by the proposed system. In the data pre-processing phase noise removal using median filtering is done.

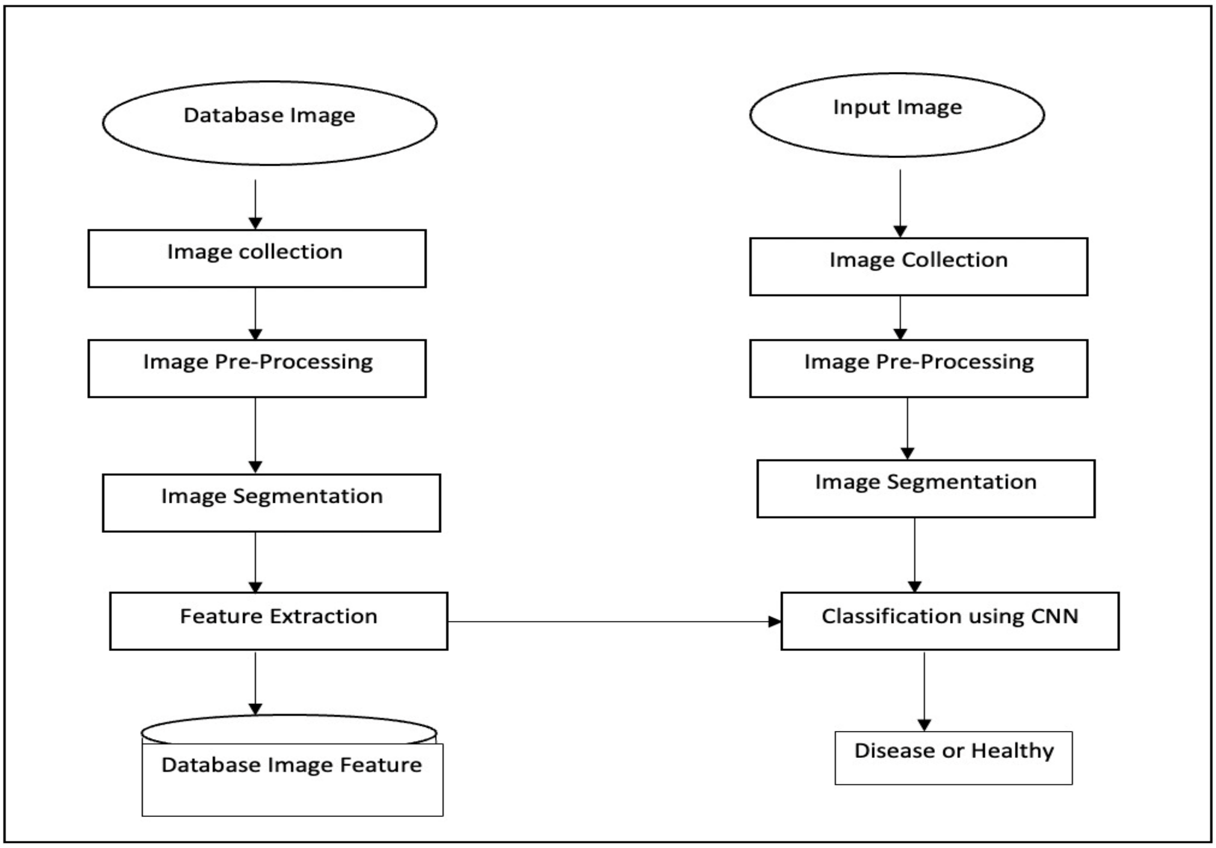

Fig 5.1.1 System Architecture

The System design mainly consists of

1. Image Collection
2. Image Preprocessing
3. Image Segmentation
4. Feature Extraction
5. Training
6. Classification

1. Image Collection:

The dataset that we have used in this project is from Kaggle website.

2. Image Preprocessing:

The goal of pre-processing is an improvement of image data that reduces unwanted distortions and enhances some image features important for further image processing. Image pre-processing involves three main things a) Grayscale conversion b) Noise removal c) Image enhancement.

a) Grayscale Conversion: Grayscale image contains only brightness information. Each pixel value in a grayscale image corresponds to an amount or quantity of light. The brightness graduation can be differentiated in grayscale image. Grayscale image measures only light intensity 8 bit image will have brightness variation from 0 to 255 where ‘0’ represents black and ‘255’ represent white. In grayscale conversion color image is converted into grayscale image shows. Grayscale images are easier and faster to process than colored images. All image processing technique are applied on grayscale image.

b) Noise Removal: The objective of noise removal is to detect and remove unwanted noise from digital image. The difficulty is in deciding which features of an image are real and which are caused by noise. Noise is random variations in pixel values. We are using median filter to remove unwanted noise. Median filter is nonlinear filter, it leaves edges invariant. Median filter is implemented by sliding window of odd length. Each sample value is sorted by magnitude, the center most value is median of sample within the window, is a filter output.

c) Image Enhancement: The objective of image enhancement is to process an image to increase visibility of feature of interest. Here contrast enhancement is used to get better quality result.

3. Image Segmentation:

Image segmentation are of many types such as clustering, threshold, neural network based and edge based. In this implementation we are using the clustering algorithm called mean shift clustering for image segmentation. This algorithm uses the sliding window method for converging to the Centre of maximum dense area. This algorithm makes use of many sliding windows to converge the maximum dense region. Mean shift clustering Algorithm This algorithm is mainly used for detecting highly dense region.

4 Feature Extraction:

There are many features of an image mainly color, texture, and shape. Here we are considering three features that are color histogram, Texture which resembles color, shape, and texture.

5. Training :

Training dataset was created from images of known Cancer stages. Classifiers are trained on the created training dataset. A testing dataset is placed in a temporary folder. Predicted results from the test case, Plots classifiers graphs and add feature-sets to test case file, to make image processing models more accurate

6. Classification:

The binary classifier which makes use of the hyper-plane which is also called as the decision boundary between two of the classes is called as Convolution Neural Network. Some of the problems are pattern recognition like texture classification makes use of CNN. Mapping of nonlinear input data to the linear data provides good classification in high dimensional space in CNN. The marginal distance is maximized between different classes by CNN. Different Kernels are used to divide the classes. CNN is basically a binary classifier that determines hyper plane in dividing two classes. The boundary is maximized between the hyperplane and two classes. The samples that are nearest to the margin will be selected in determining the hyperplane is called support vectors.

5.2 CNN Algorithm Explanation

The invention of the CNN in 1994 by Yann Le Cun is what propelled the field of Artificial Intelligence and Deep learning to its former glory. The first neural network named LeNet5 had a very less validation accuracy of 42% since then we have come a long way in this field. Nowadays almost every giant technology firms rely on CNN for more efficient performance. The idea to detect diseases in mulberry leaf incorporates the use of CNN before we dive into the “functionality and working of CNN” concept, we must have a basic idea on how the human brain recognizes an object in spite of its varying attributes from one another. Our brain has a complex layer of neurons , each layer holds some information about the object and all the features of the object are extracted by the neurons and stored in our memory, next time when we see the same object the brain matches the stored features to recognize the object, but one can easily mistake it as a simple “IF-THEN” function, yes it is to some extent but it has an extra feature that gives it an edge over other algorithms that is Self-Learning, although it cannot match a human brain but still it can give it a tough competition . Image is processed using the Basic CNN to detect the diseases in leaves. The data training in our CNN model has to satisfy following constraints:

1) No missing values in dataset.

2) The dataset must distinctly be divided into training and testing sets, either the training or the testing set shouldn’t contain any irrelevant data out of our model domain in case of an image dataset all the images must be of the same size, one uneven distribution of image size in our dataset can decrease the efficiency of our neural network.

3) The images should be converted into black and white format before feeding it into the convolution layer because reading images in RGB would involve a 3-D numPy matrix which will reduce the execution time of our model by a considerable amount. Any kind of corrupted or blurred images should also be trimmed from the database before feeding it into the neural network. This is all about data pre-processing rules, let us dive right into the working of the convolution neural network.

5.3 Convolutional Neural Network layers

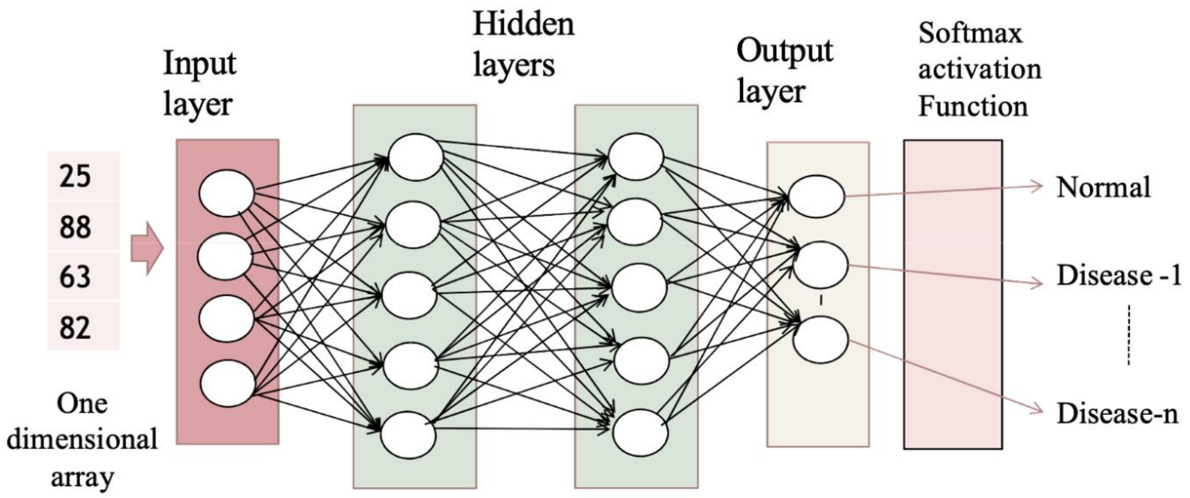

Fig 5.3.1 Convolutional Neural Network Layers

A. Convolution layer

This layer involves scanning the whole image for patterns and formulating it in the form of a 3x3 matrix. This convolved feature matrix of the image is known as Kernel. Each value in the kernel is known as a weight vector.

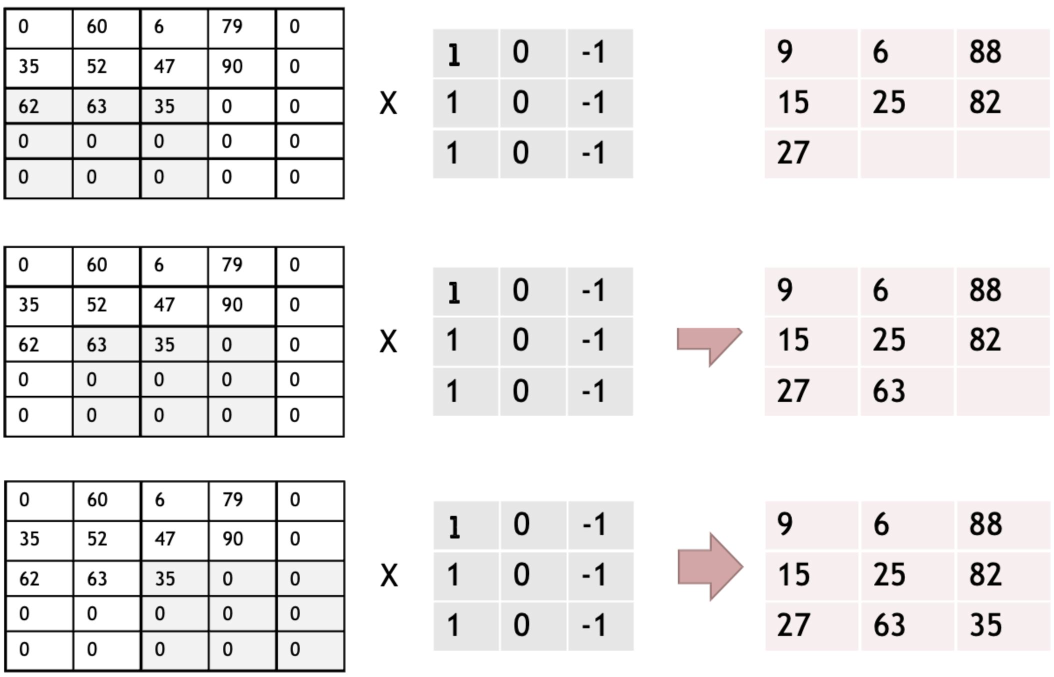

Fig 5.3.2 Convolutional Output: A 3*3 matrix

B. Pooling layer

After the convolution comes to the pooling here the image matrix is broken down into the sets of 4 rectangular segments which are non-overlapping.

There are two types of pooling, Max pooling and average pooling. Max pooling gives the maximum value in the relative matrix region which is taken. Average pooling gives the average value in relative matrix region. The main advantage of the pooling layer is that it increases computer performance and decreases over- fitting chances.

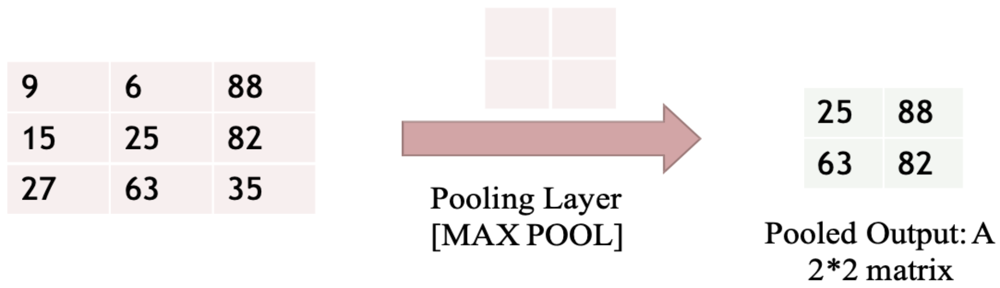

Figure 5.3.3 Pooling Layer Output

C. Activation layer

It is the part of the Convolutional Neural Networks where the values are Normalized that is, they are fitted in a certain range. The used convolutional function is ReLU which allows only the positive values and then rejects the negative values. It is the function of low computational cost.

CHAPTER 6

CODE

```
from flask import Flask, render_template, url_for, request

import sqlite3

import shutil

import os

import sys

import cv2  # working with, mainly resizing, images

import numpy as np  # dealing with arrays

import os  # dealing with directories

from random import shuffle  # mixing up or currently ordered data that might lead our
network astray in training.

from tqdm import \

tqdm  # a nice pretty percentage bar for tasks. Thanks to viewer Daniel BA1/4hler for this
suggestion

import tflearn

from tflearn.layers.conv import conv_2d, max_pool_2d

from tflearn.layers.core import input_data, dropout, fully_connected

from tflearn.layers.estimator import regression

import tensorflow as tf

import matplotlib.pyplot as plt

from twilio.rest import Client

account_sid="AC96ee88c73a7f5e022ba6400a2a009f74"

auth_token="adbc485e319a34834a91ed6eb6e67958"

client=Client(account_sid,auth_token)
```

```
app = Flask(_name_)

@app.route('/')

  def index():

  return render_template('index.html')

  @app.route('/userlog', methods=['GET', 'POST'])

  def userlog():

 if request.method == 'POST':

connection = sqlite3.connect('user_data.db')

cursor = connection.cursor()

name = request.form['name']

password = request.form['password']

query = "SELECT name, password FROM user WHERE name = '"+name+"' AND password= '"+password+"'"

cursor.execute(query)

result = cursor.fetchall()

if len(result) == 0:

return render_template('index.html', msg='Sorry, Incorrect Credentials Provided,  Try Again')

else:

return render_template('userlog.html')

return render_template('index.html')
```

CHAPTER 7

TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system to identify any gaps, errors, or missing requirements in contrary to the actual requirements. System testing of a software or hardware is a testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing fails within the scope of black-box testing, and such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, of all the integrated software components that have passed integration testing and the software system itself integrated with any applicable software systems.

The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together. System testing is a more limited type of testing. It seeks to detect defects both within the inter-assemblages and within the system. System testing is performed on the entire system in the context of a Functional Requirement Specification (FRS) and / or a System Requirement Specification (SRS). System testing tests not only the design, but also the behaviour and even the believed expectation of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirement specification. Before applying methods to design effective test cases, a software engineer must understand the basic principle that guides software testing. All the tests should be traceable to customer requirements.

Types of testing

Software testing methods and traditionally divided into two: white-box and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

a) **White-box testing** (also known as clear box testing, glass box testing, transparent box testing and structural testing, by seeing the source code) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.

While white-box testing can be applied at the unit, integration, and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between sub systems during a system-level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

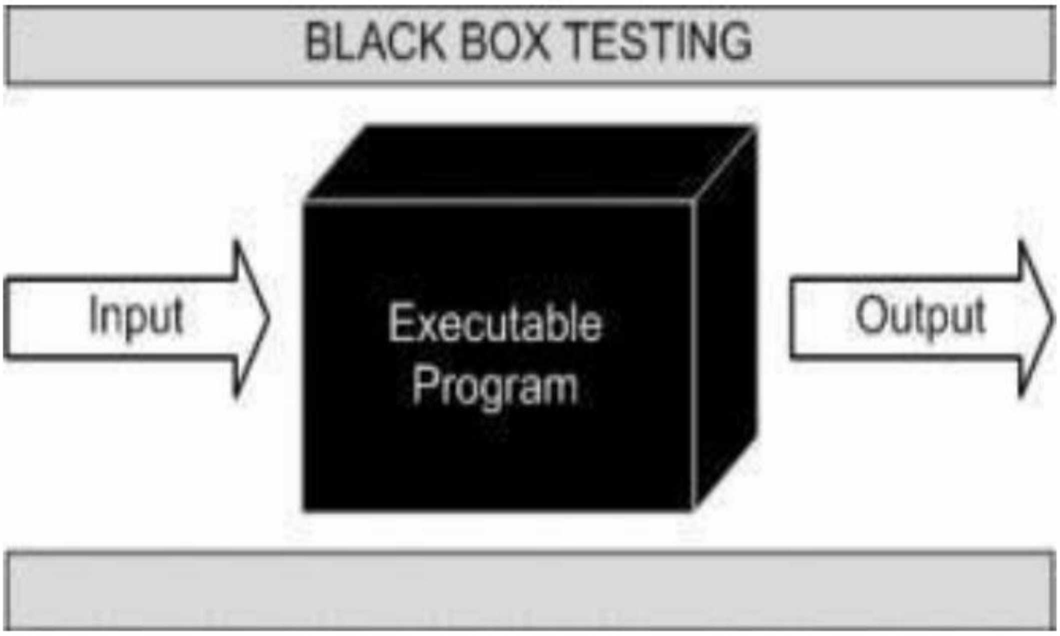

Fig 7.1 Black box Testing

b) **Black box testing**: The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon. This project has been tested under different circumstances, which includes different types such as Unit testing, Integration testing and System testing that are described below.

Levels of Testing:

There are different levels during the process of testing. Levels of testing include different methodologies that can be used while conducting software testing. The main levels of software testing are:

Functional Testing: This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. There are five steps that are involved while testing an application for functionality.

- The determination of the functionality that the intended application is meant to perform.
- The creation of test data based on the specifications of the application.
- The output based on the test data and the specifications of the application.
- The writing of test scenarios and the execution of testcases.
- The comparison of actual and expected results based on the executed testcases.

Non-functional Testing: This section is based upon testing an application from its non-functional attributes. Non functional testing involves testing software from the requirements which are non- functional in nature but important such as performance, security, user interface, etc.

Unit Testing

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. During the development process itself all the syntax errors etc. got rooted out. For this developed test case that result in executing every instruction in the program or module i.e. every path through program was tested. Test cases are data chosen at random to check every possible branch after all the loops.

Unit Testing Test Case 1

S1# Test Case :-	UTC-1
Name of Test :-	Uploading image
Iteams being tested :-	Tested for uploading different image
Sample Input :-	Upload Sample image
Expected output :-	Image should upload properly
Actual output :-	Upload successful
Remarks :-	Pass

Unit Testing Test Case 2

S1 # Test Case:	**UTC-2**
Name of Test	Detecting X-Ray Lung images
Items being tested	Test for different X-Ray Lung images
Sample Input	Tested for different images of X-Ray Lung images
Expected output	X-Ray Lung image should be displayed
Actual output	Should Display disease or Healthy
Remarks	Predicted result

Integration Testing:

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. It occurs after unit testing and before validation testing. Integration testing can be done in two ways: Bottom-up integration testing and Top-down integration testing.

1.**Bottom-up Integration**: This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.

2.Top-down Integration: In this testing, the highest-level modules are tested first and progressively, lower-level modules are tested thereafter. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic actual situations.

Functional Testing Test Case 1

S1 # Test Case:	**ITC**
Name of Test	Working of choose file option
Items being tested	User convenience to access images stored
Sample Input	Click and select the image
Expected output	Should open selected Image
Actual output	Selected open selected image
Remarks	Pass.

Functional Testing Test Case 2

S1 # Test Case:	**ITC-2**
Name of Test	Working of Segmentation and Displaying X-Ray Lung Image
Items being tested	Selecting different images and verifying X-Ray Lung Image
Sample Input	Click and select the image
Expected output	Should show X-Ray Lung and predict disease or healthy
Actual output	Image Segmented, X-Ray Lung image detected should be displayed
Remarks	Pass.

System testing:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. System testing is important because of the following reasons:

• System testing is the first step in the Software Development Life Cycle, where the application is tested.

• The application is tested thoroughly to verify that it meets the functional and technical specifications.

• The application is tested in an environment that is very close to the production environment where the application will be deployed.

• System testing enables us to test, verify, and validate both the business requirements as well as the application architecture.

System Testing is shown in the below tables

S1 # Test Case:	**STC-1**
Name of Test	System testing in various versions of OS
Items being tested	OS compatibility.
Sample Input	Execute the program in windows XP/Windows-10
Expected output	Performance is better in windows-10
Actual output	Same as expected output performance is better in windows-10
Remarks	Pass.

CHAPTER 8

RESULTS AND SNAPSHOTS

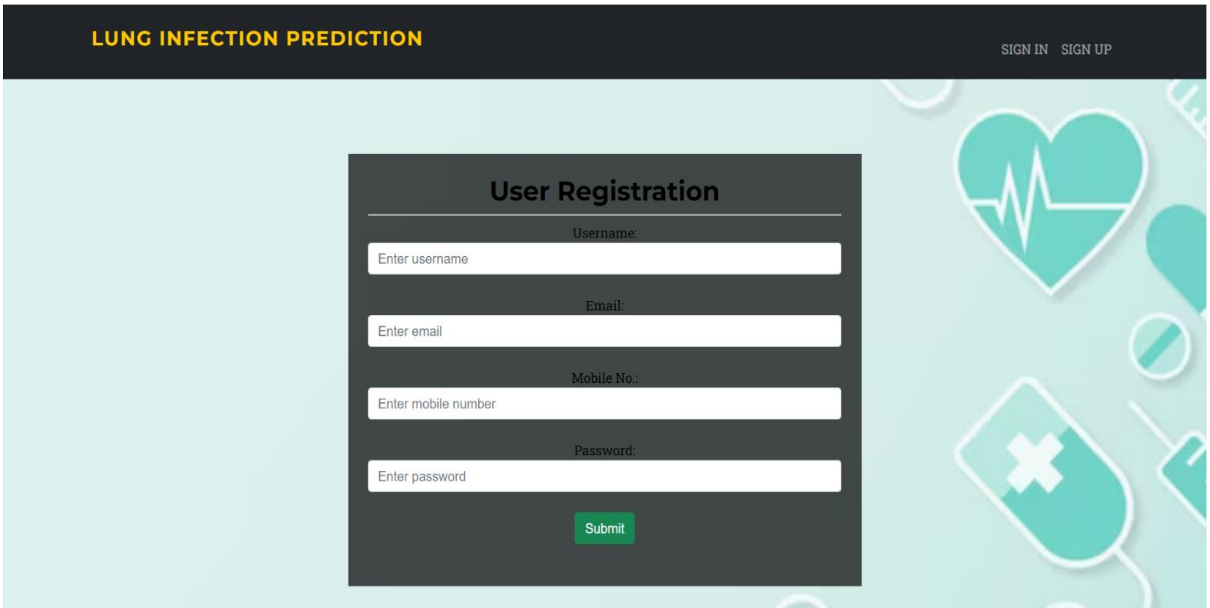

Figure 8.1: **User Registration Page**

The figure 8.1 shows the snapshot of User Registration Page where the users can register into the website by giving the necessary credentials like name, email, mobile number and password.

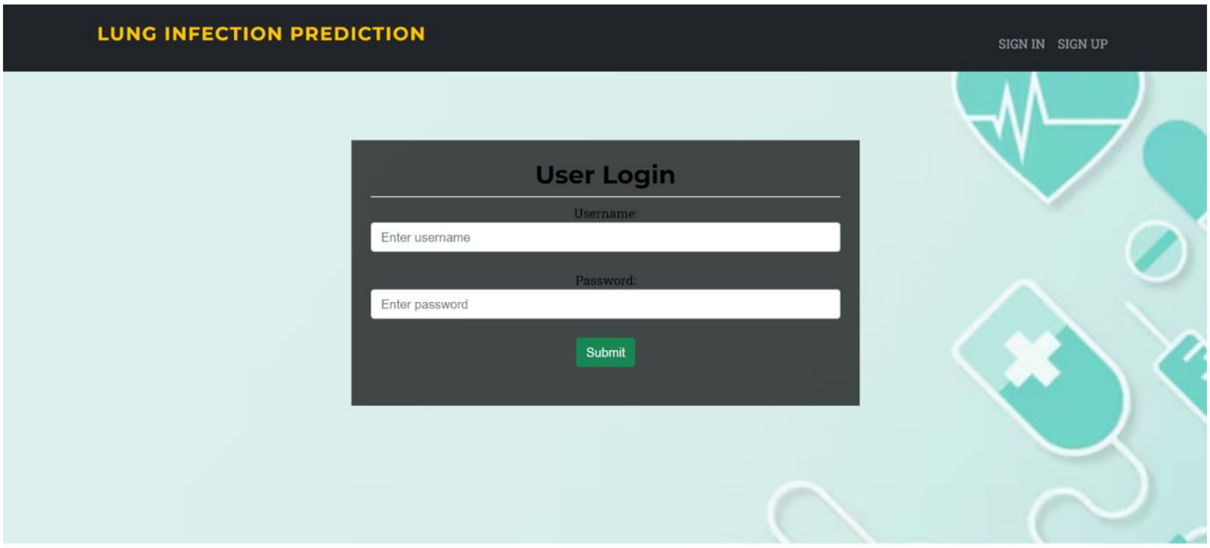

Figure 8.2: **User Login Page**

The figure 8.2 shows the snapshot of user login page where the user can login into the website by giving name and password.

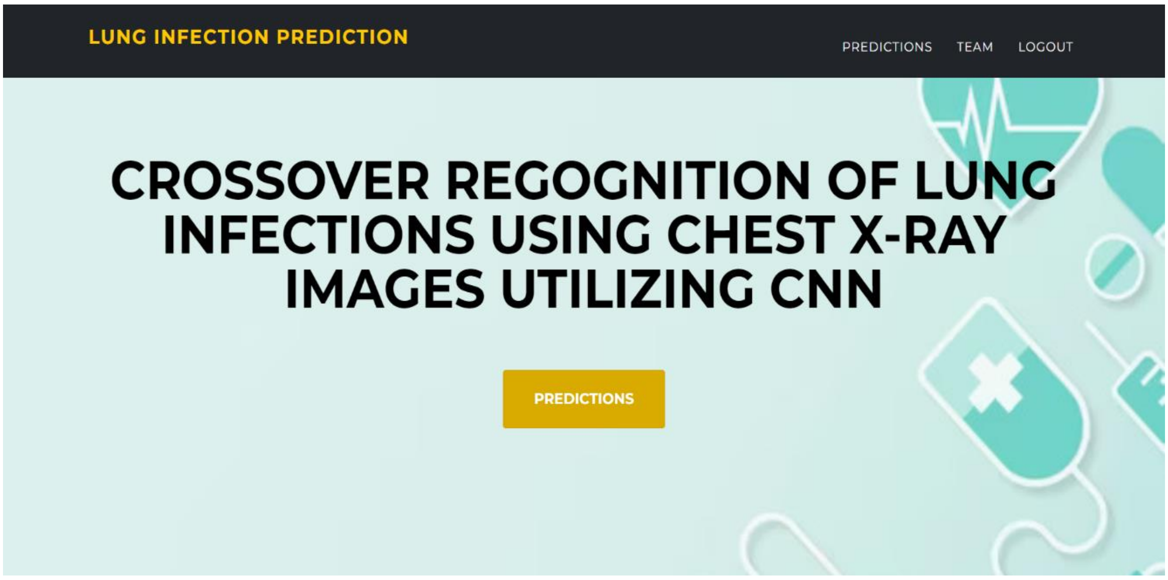

Figure 8.3: Home Page

The figure 8.3 shows the snapshot of Home Page where its shows the title of the project and nav bar consisting of team, predictions and logout.

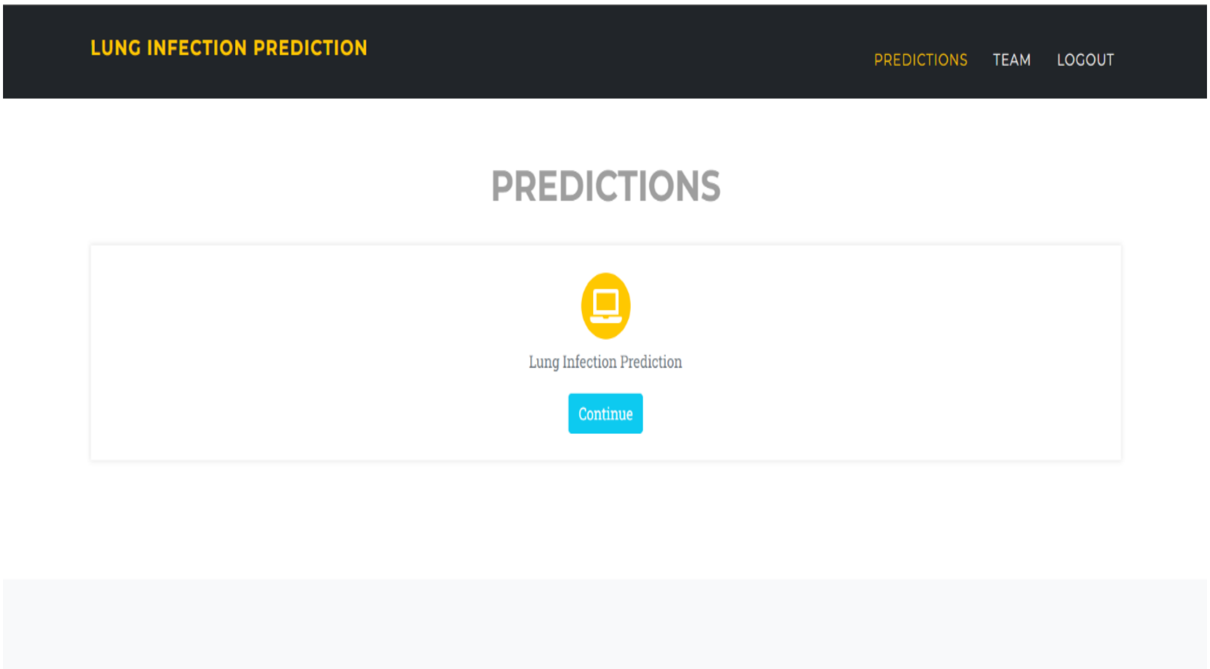

Figure 8.4: Predictions Page

The figure 8.4 shows the lung infections predictions page

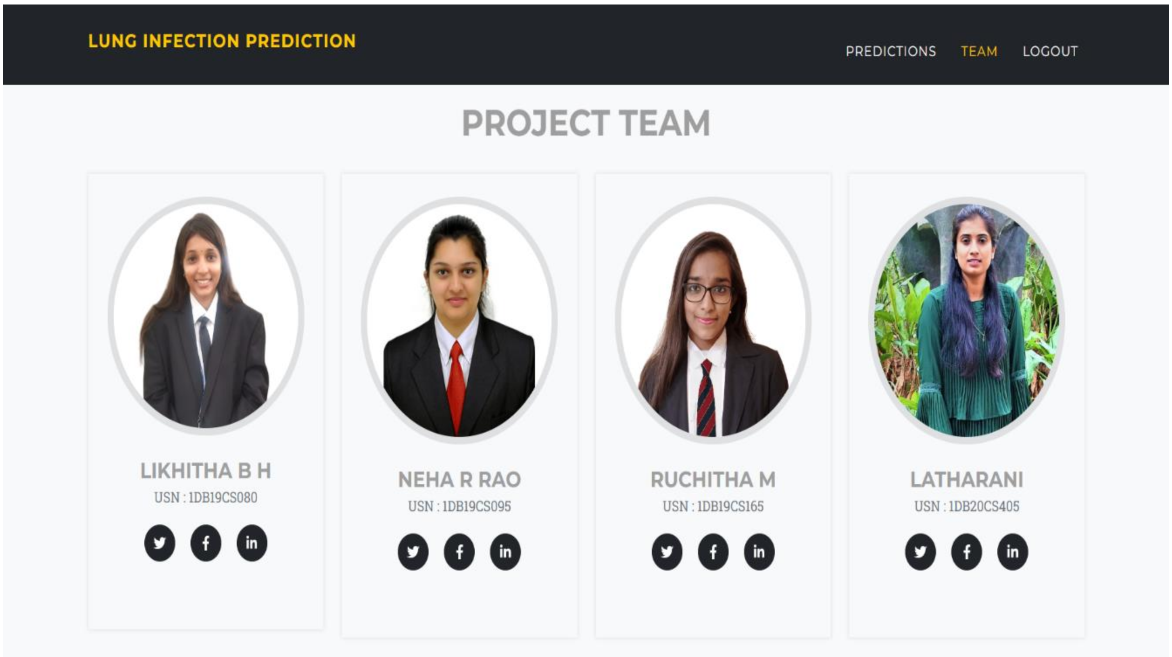

Figure 8.5: Project team Page

The figure 8.5 shows the snapshot of project team members and their details.

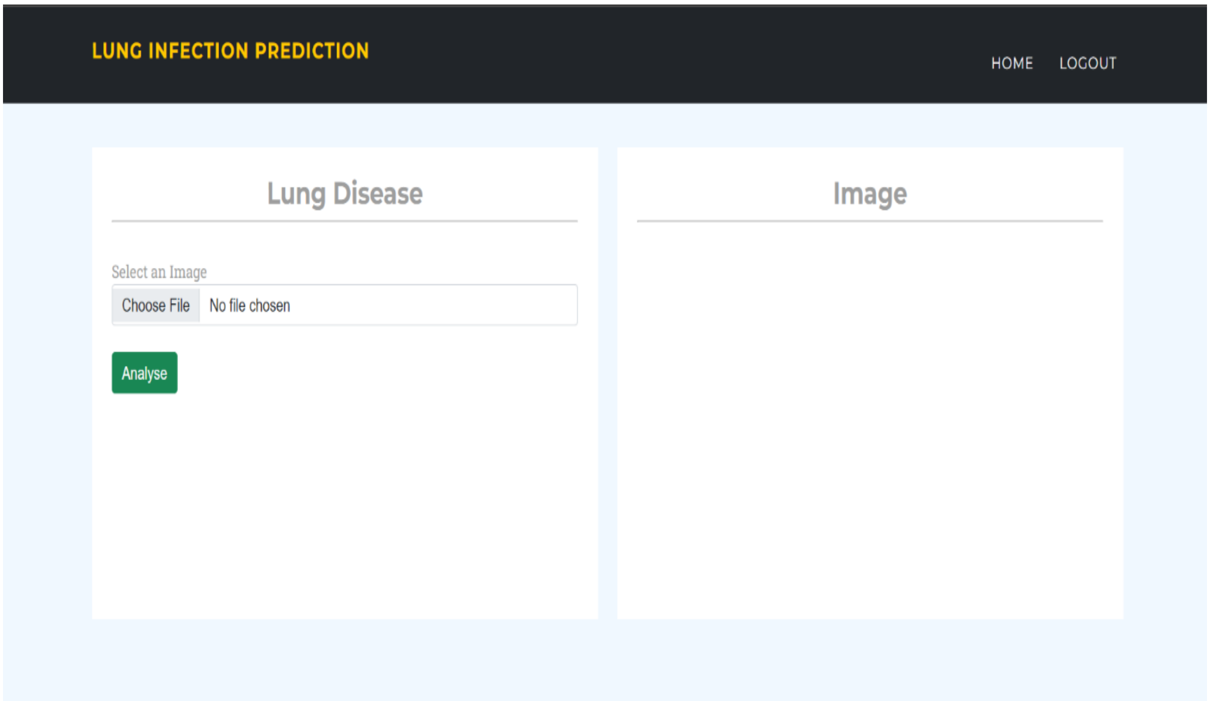

Figure 8.6: Result Page Template

The figure 8.6 shows the result page template which will take image as input, analyses the image and then display the results

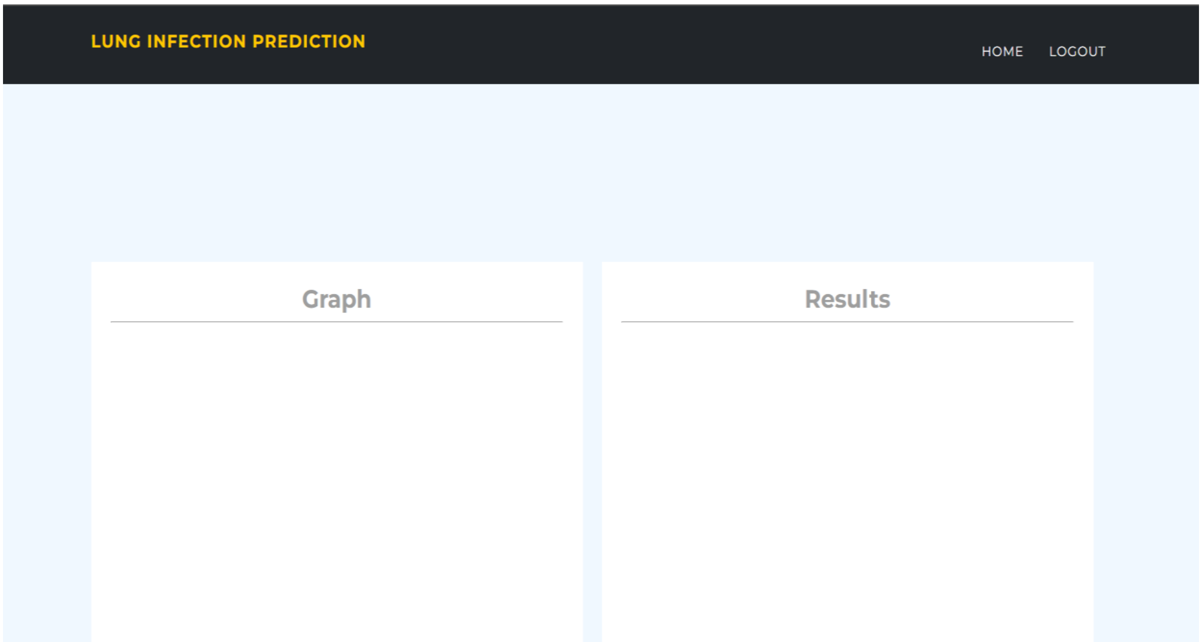

Figure 8.7: Results Page Template

The figure 8.7 shows the result page which will show the results and graph.

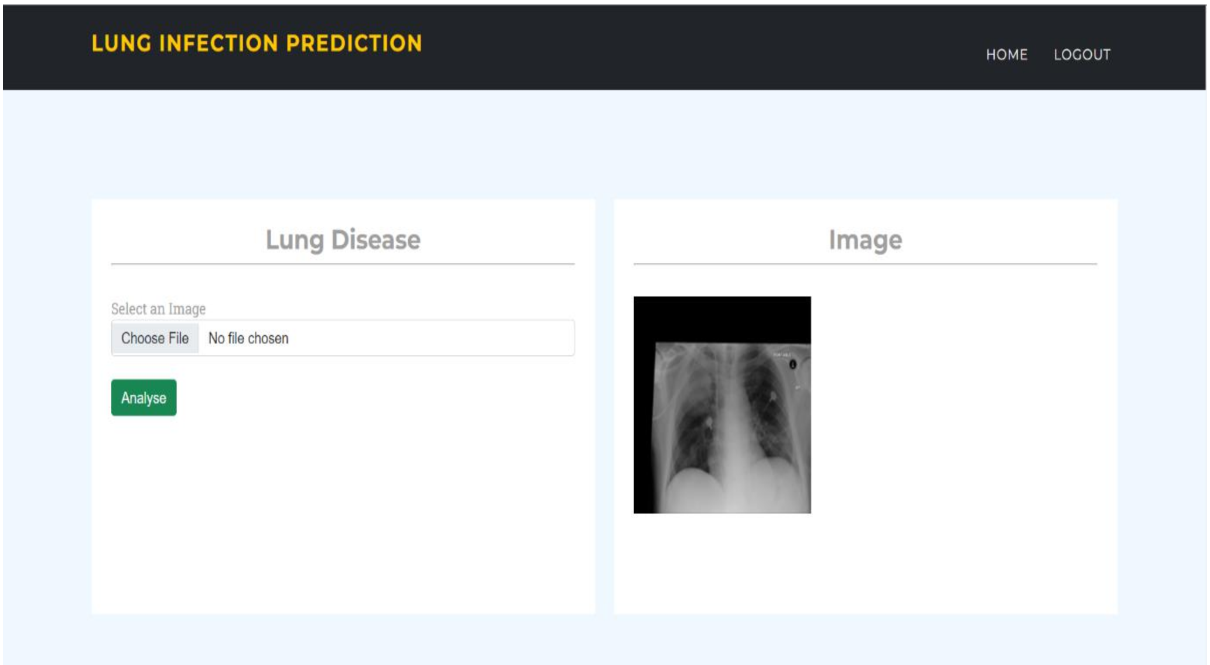

Figure 8.8: Result Display Page for Lung Cancer Infection Prediction

The figure 8.8 shows the result display page which will take lung X-Ray image as input and displays the input image.

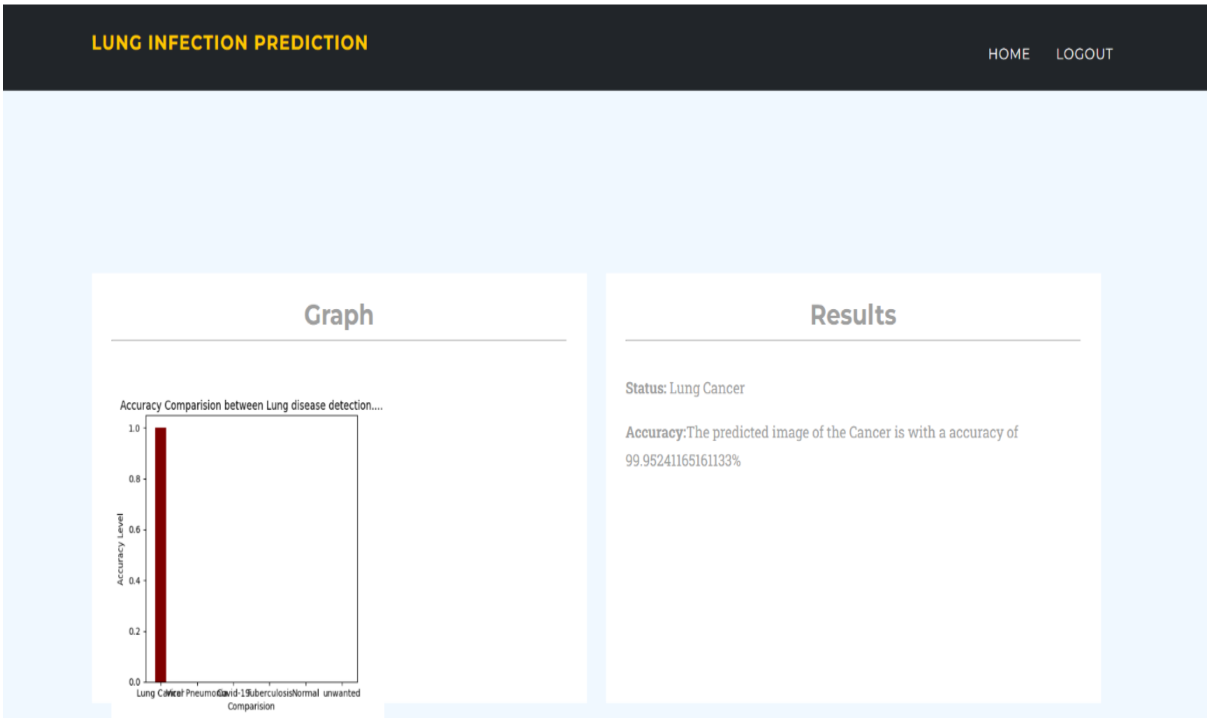

Figure 8.9: Result Display Page for Lung Cancer Infection Prediction

The figure 8.9 shows the result display page which displays the result with graph and accuracy for lung cancer infection

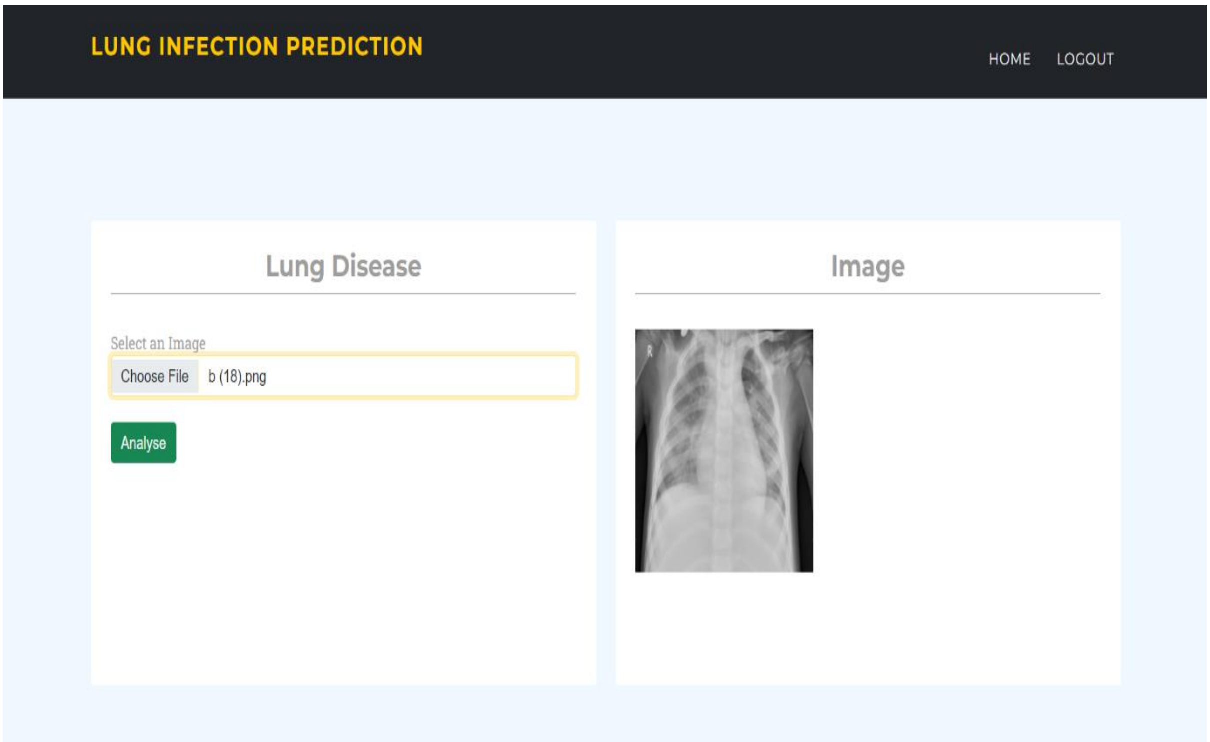

Figure 8.10: Result Display Page for Pneumonia Infection Prediction

The figure 8.10 shows the result display page which will take Pneumonia image as input and displays the input image.

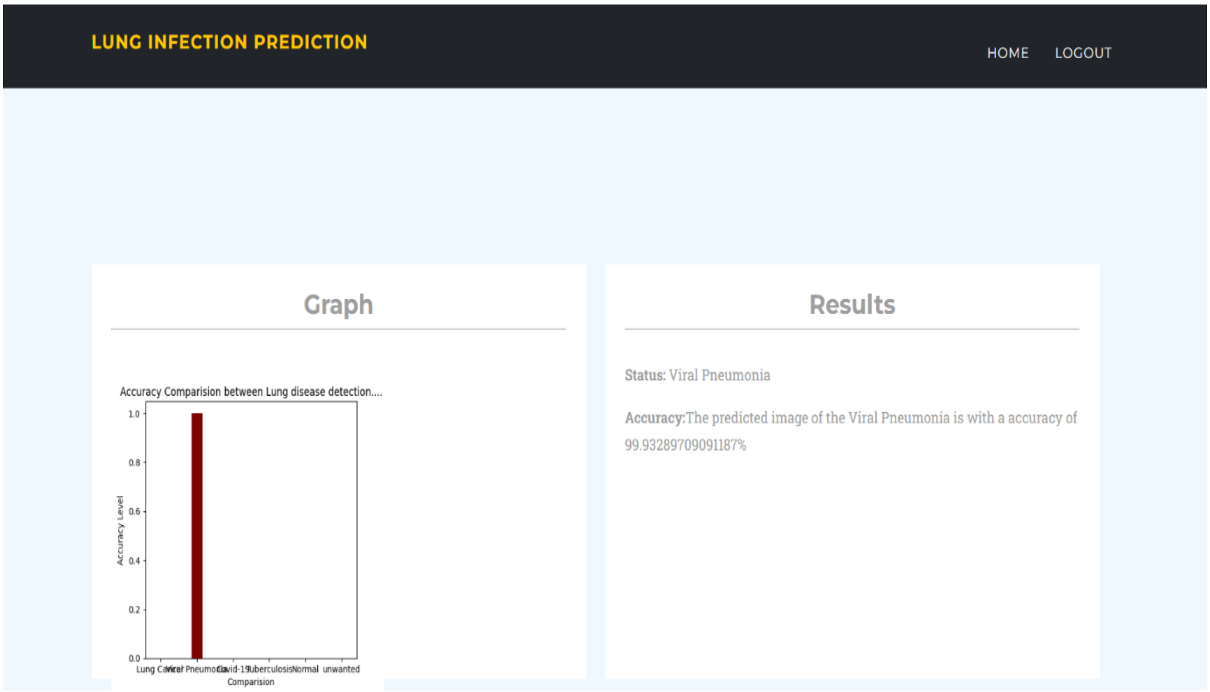

Figure 8.11: Result Display Page for Viral Pneumonia Infection Prediction

The figure 8.11 shows the result display page which displays the result with graph and accuracy for Viral Pneumonia Infection

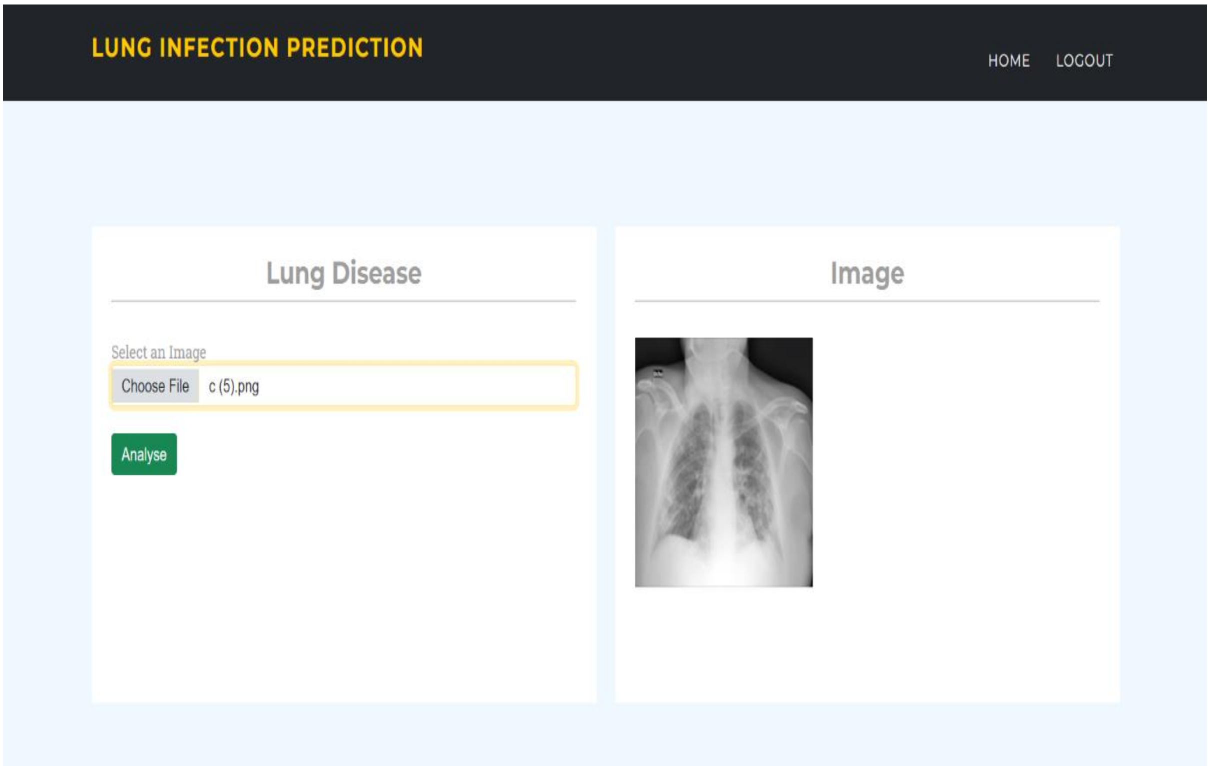

Figure 8.12: Result Display Page for Covid – 19 Prediction

The figure 8.12 shows the result display page which will take Covid-19 as input and displays the input image.

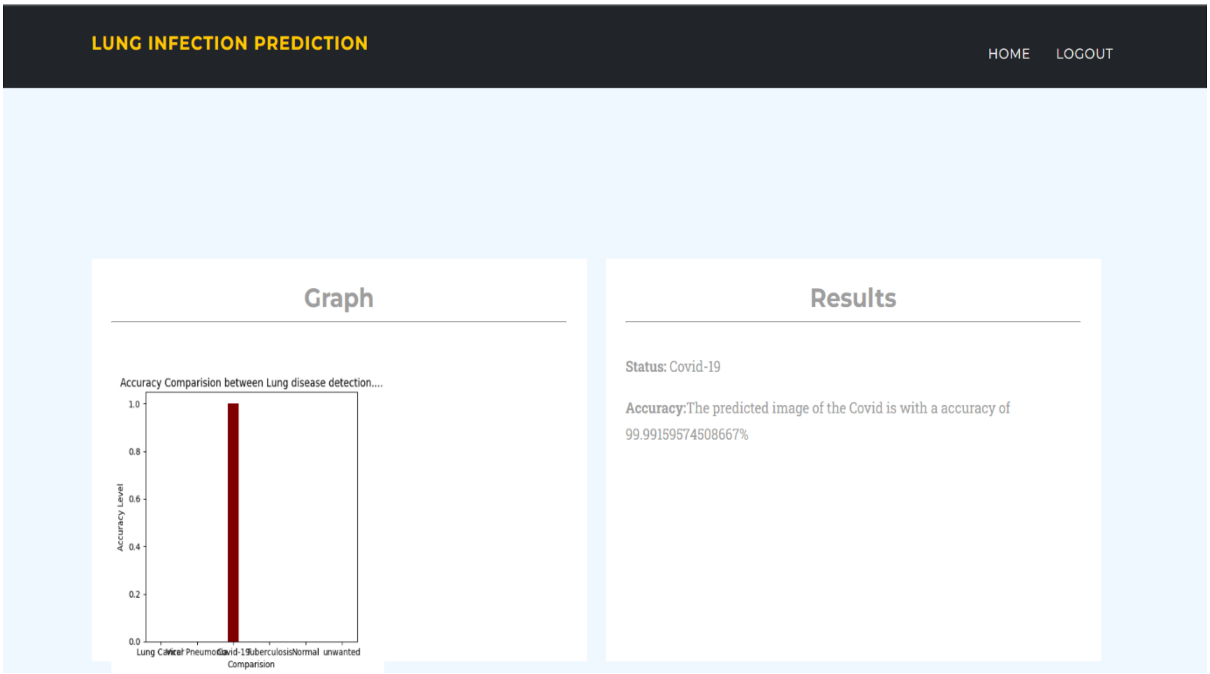

Figure 8.13: Result Display Page for Covid-19 Infection Prediction

The figure 8.13 shows the result display page which displays the result with accuracy and graph for Covid-19

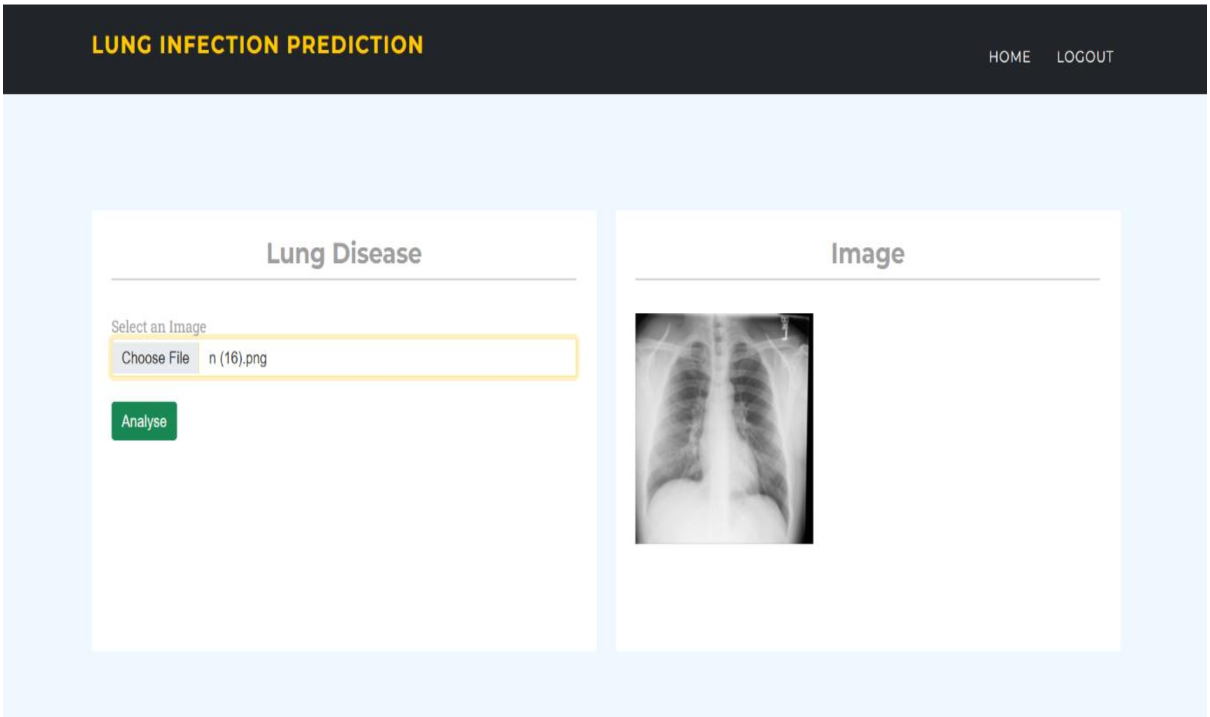

Figure 8.14: Result Display Page for Normal Prediction

The figure 8.14 shows the result display page which will take Normal Image as input and displays the input image.

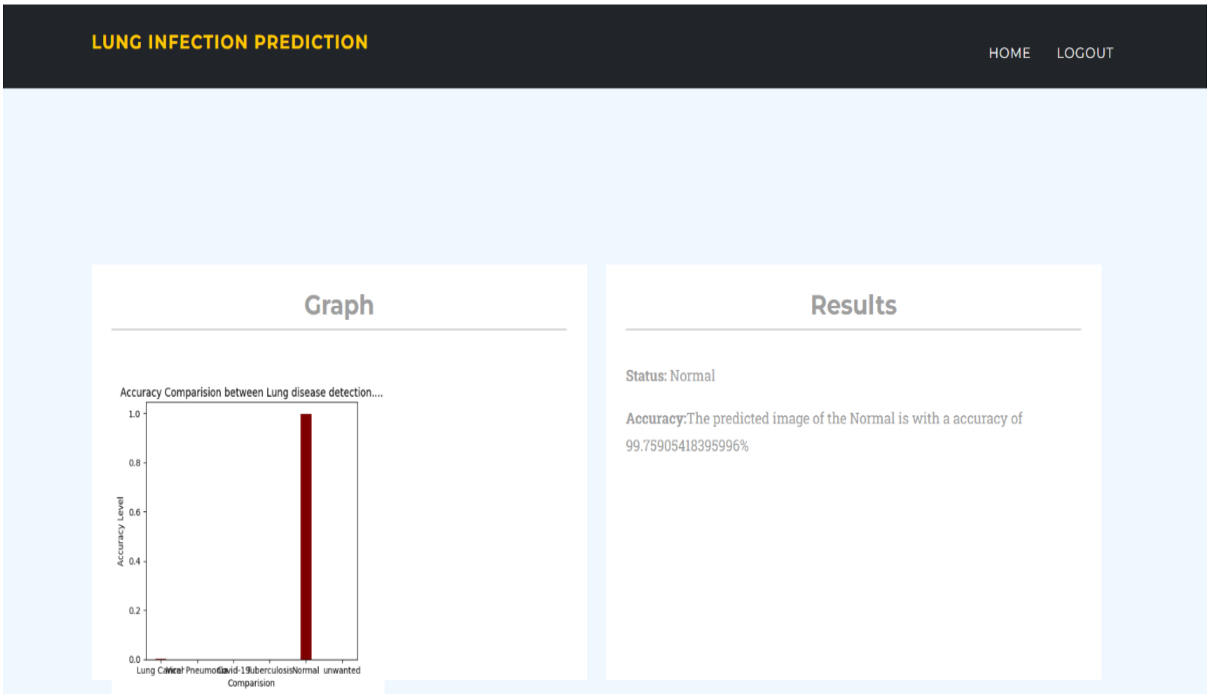

Figure 8.15: Result Display Page for Normal Prediction

The figure 8.15 shows the result display page which displays the result with accuracy and graph for Normal

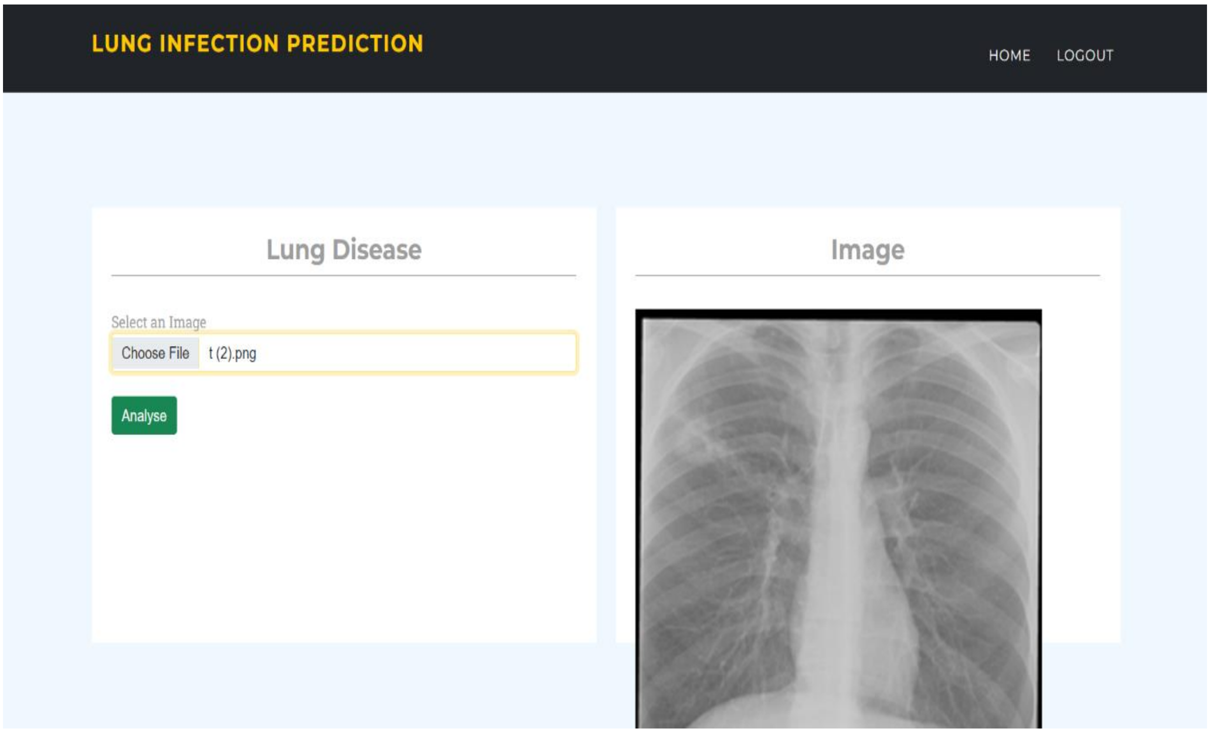

Figure 8.16: Result Display Page for Tuberculosis Prediction

The figure 8.16 shows the result display page which will take Tuberculosis Image as input and displays the input image.

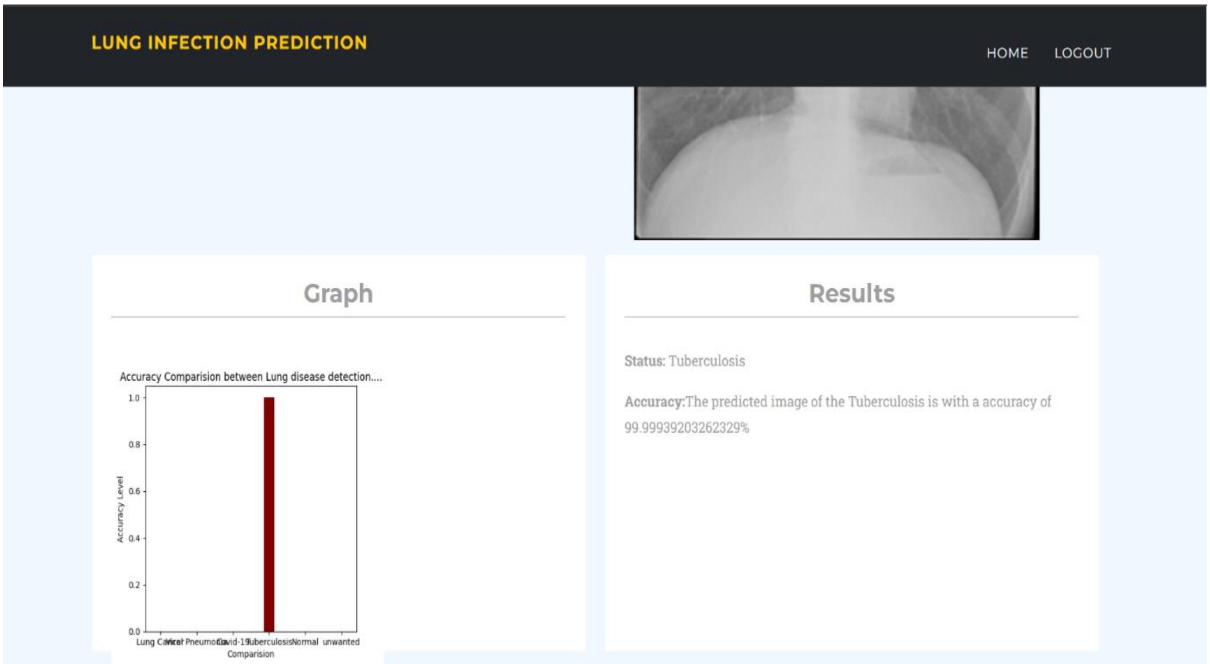

Figure 8.17: Result Display Page for Tuberculosis Prediction

The figure 8.17 shows the result display page which displays the result for with accuracy and graph Tuberculosis

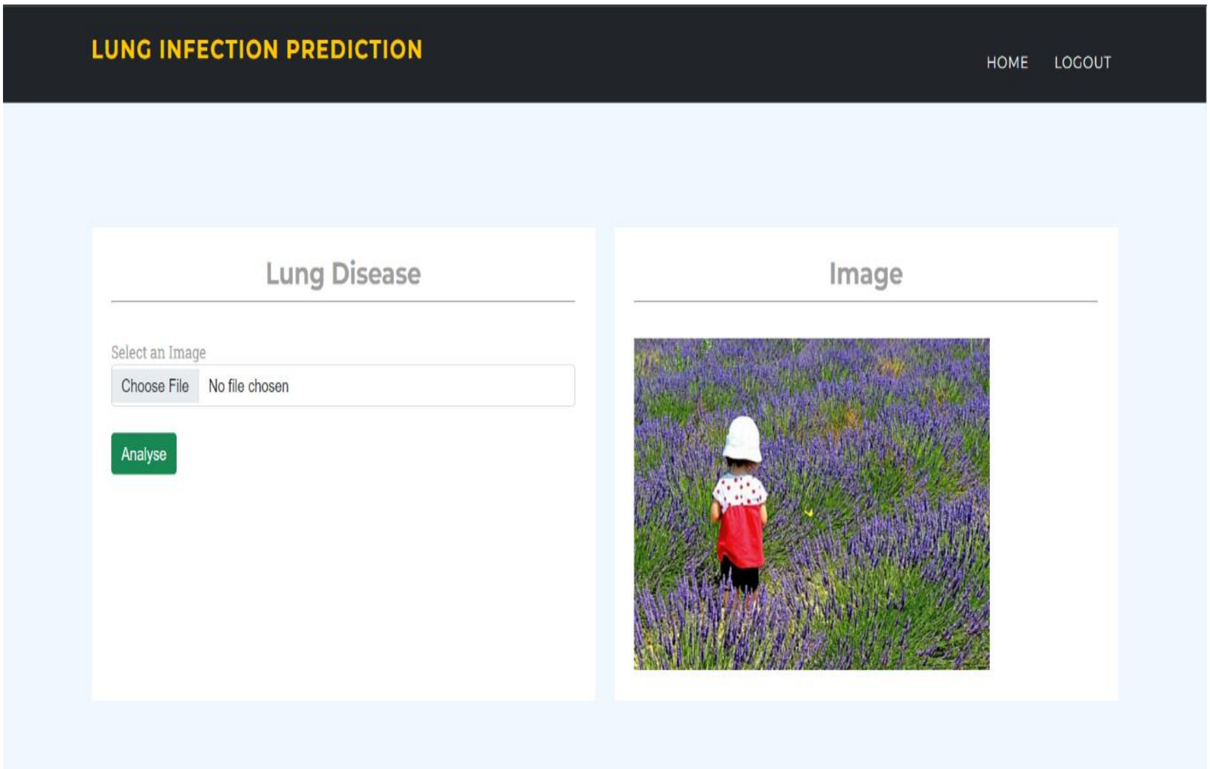

Figure 8.18: Result Display Page for Unwanted Image

The figure 8.18 shows the result display page which will take Unwanted Image as input and displays the input image.

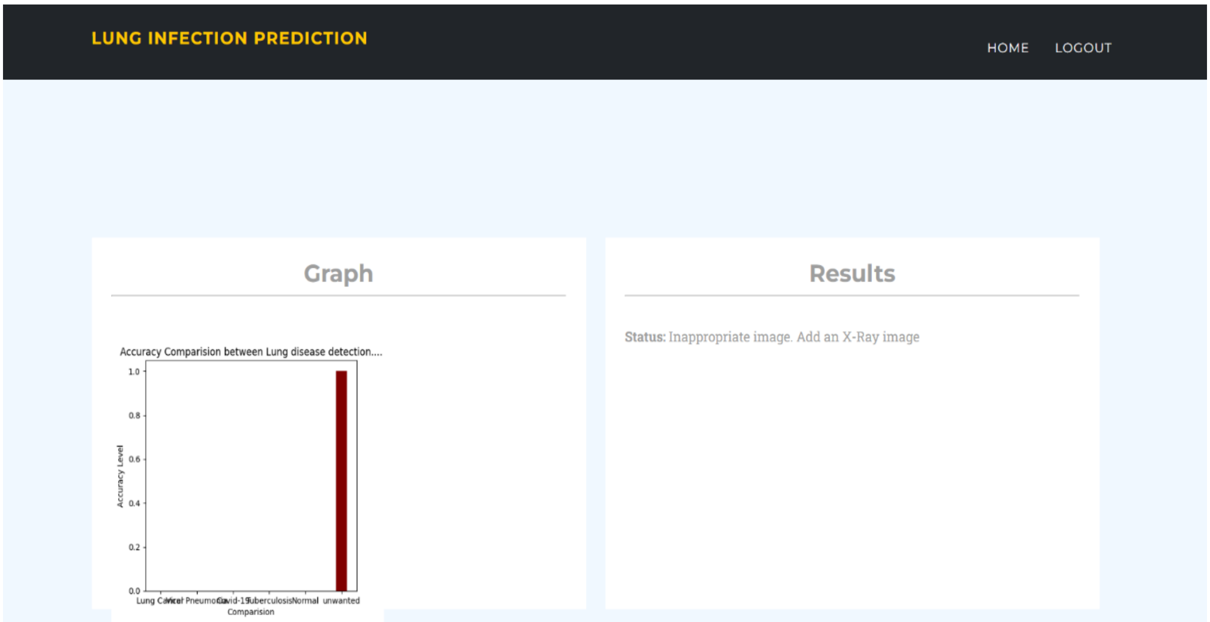

Figure 8.19: Result Display Page for Unwanted Image Prediction

The figure 8.19 shows the result display page which displays the result with accuracy and graph for unwanted image

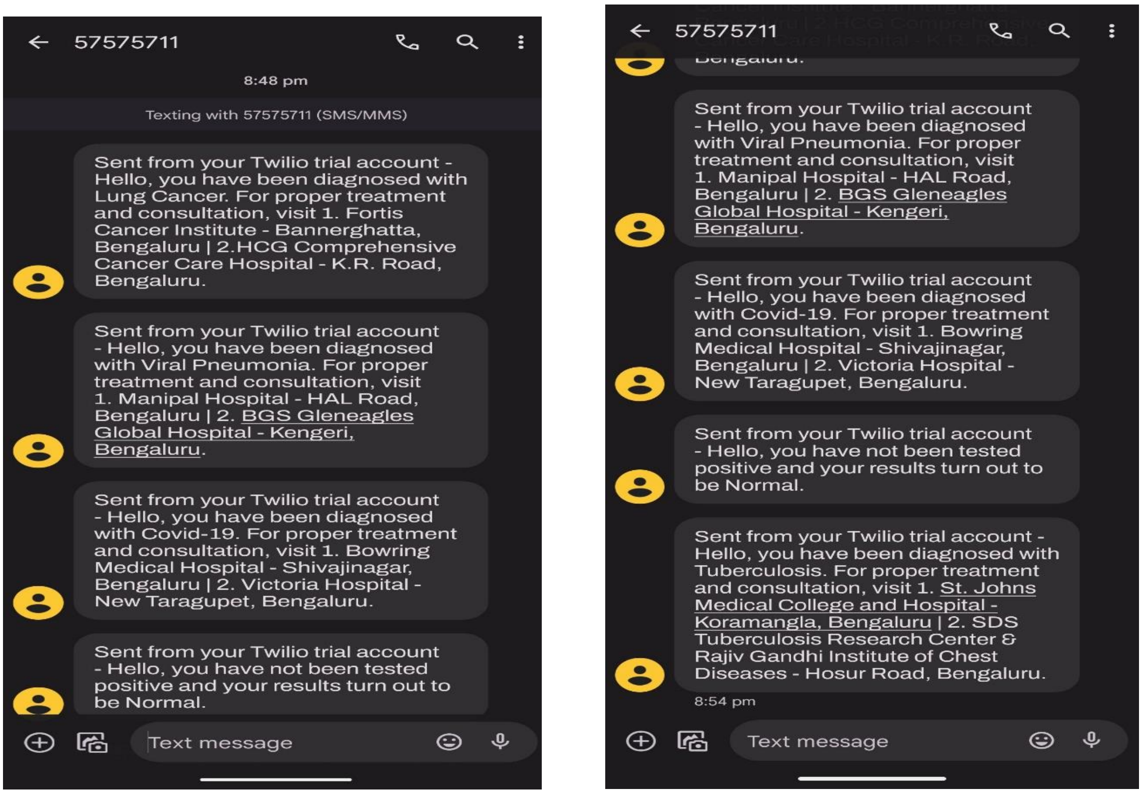

Figure 8.20: Messages Received Page

The figure 8.20 shows the message received page about the type of disease and the recommended hospital

CHAPTER 9

CONCLUSION

The proposed work presents a CNN model for detecting different lung diseases from X-ray images, including Normal lungs, COVID-19 Infected lungs, Pneumonia Infected lungs, Tuberculosis infected lungs, and lung cancer-infected lungs. With an accuracy of around 97%, the model offers a valuable tool for medical practitioners and researchers, as it can provide a quick and cost-effective diagnosis of various lung diseases.

Furthermore, the model's low computational requirements make it ideal for use in resource-limited settings. This technology has the potential to revolutionize the field of medical imaging and aid in the fight against the COVID-19 pandemic, as well as other lung diseases.

The proposed work can be extended by combining multiple models into an ensemble, as this has been shown to improve accuracy even further. Additionally, incorporating additional features, such as patient demographics or clinical history, can help tailor the diagnosis to individual patients and further improve the model's accuracy.

In conclusion, the proposed work presents a promising solution for the detection of lung diseases from X-ray images, with the potential to improve medical diagnosis and treatment. The model's high accuracy, low cost, and computational requirements make it a valuable tool for medical practitioners, especially in resource-limited settings. Future research can further improve the model's accuracy and expand its application to a wider range of lung diseases.

REFERENCES

[1] *Kanakaprabha. S, D. Radha, "Analysis of COVID-19 and Pneumonia Detection in Chest X-Ray Images using Deep Learning", 2021 International Conference on Communication, Control and Information Sciences (ICCISc) | 978-1-6654-0295-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCISc52257.2021.9484888.*

[2] *Afonso U. Fonseca ,Gabriel S. Vieira , et al., "Screening of Viral Pneumonia and COVID-19 in Chest X-ray using Classical Machine Learning", 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC) | 978-1-6654-2463-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/COMPSAC51774.2021.00294.*

[3] *Mohammad Farukh Hashmi , Satyarth Katiyar, , et al.,"Efficient Pneumonia Detection in Chest Xray Images Using Deep Transfer Learning", Diagnostics 2020, 10, 417; doi:10.3390/diagnostics1006041.*

[4] *SAEED S. ALAHMARI 1, (Member, IEEE), et al., "A Comprehensive Review of Deep Learning-Based Methods for COVID-19 Detection Using Chest X-Ray Images", Received 30 August 2022, accepted 14 September 2022, date of publication 20 September 2022, date of current version 28 September 2022.*

[5] *Vinod Kumar, Anil Saini, "Detection system for lung cancer based on neural network: X-Ray validation performance", International Journal of Enhanced Research in Management & Computer Applications, ISSN: 2319-7471Vol. 2 Issue 9, Nov.-Dec. 2013, pp: (40-47).*

[6] *Stefanus Kieu Tao Hwa, Abdullah Bade, et al., " Tuberculosis detection using deep learning and contrast-enhanced canny edge detected X-Ray images.", IAES International Journal of Artificial Intelligence (IJ-AI) Vol. 9, No. 4, December 2020, pp. 713~720 ISSN: 2252-8938, DOI: 10.11591/ijai.v9.i4.pp713-720.*

[7] *Prabhjot Kaur, Shilpi Harnal, et al., "A Hybrid Convolutional Neural Network Model for Diagnosis of COVID-19 Using Chest X-ray Images", Int. J. Environ. Res. Public Health 2021, 18, 12191. https://doi.org/10.3390/ijerph182212191*

[8] *Hassaan Malik1,2· Tayyaba Anees3 · Mui-zzud-din4, "BDCNet: multi-classification convolutional neural network model for classification of COVID-19, pneumonia, and lung cancer from chest radiographs", Received: 16 May 2021 / Accepted: 7 December 2021 / Published online: 18 January 2022 © The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021.*

DETAILS OF THE STUDENT

BATCH 2019-2023

NAME:	Likhitha B H	Neha R Rao	Ruchitha M	Latharani
USN:	1DB19CS080	1DB19CS095	1DB19CS165	1DB19CS405
PHONE NUMBER:	8861448268	9535687203	8088203082	7899393852
EMAIL ID:	likhithabh8@gmail.com	neha.250901@gmail.com	ruchitha.manohar@gmail.com	latharani2305@gmail.com
PLACED-IN:	Mphasis Quinnox Agile Point Vtiger	Quinnox	Mphasis	
PERMANENT ADDRESS:	#24 1st Cross Road 1st Main Road AzadNagar Chamrajpet Bangalore-560018	No.103,Gajamukha residency, off Mysore road, Byatarayanapura , Banglore - 560040	#57/58, 9th Cross, Manjunath Nagar, Magadi Road, Behind Prasanna Theatre (GT WORLD MALL), Bengaluru - 560023	Muganapura hutridurga hobli kunigal taluk tumkur district