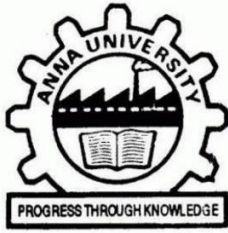


LUNG CANCER DETECTION



MINI PROJECT REPORT

Submitted by

MOHANA ACHUTHAN.B (510620205030)
SUHAIL.S (510620205047)

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

C. ABDUL HAKEEM COLLEGE OF ENGINEERING

AND TECHNOLOGY, MELVISHARAM-632509

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2023.

BONAFIDE CERTIFICATE

Certified that this project report “**LUNG CANCER DETECTION** ” is the Bonafide work of **MOHANA ACHUTHAN B (510620205030)** and **SUHAIL S (510620205047)** who carried out the project work under my supervision.

SIGNATURE

Dr.S.UMAMAHESWARI Ph.D.,
HEAD OF THE DEPARTMENT,
ASSOCIATE PROFESSOR,
Department of Information
Technology,

C.Abdul Hakeem College
of Engineering And Technology,
Melvisharam – 632 509

SIGNATURE

MR.T.S.KARTHICK M.E.,
SUPERVISOR
ASSOCIATE PROFESSOR,
Department of Information
Technology,

C.Abdul Hakeem College
of Engineering and Technology,
Melvisharam – 632 509

Submitted for the University viva voce Examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGMENT

At the outset, we would like to express our gratitude to our beloved and respected Almighty, for his support and blessings to accomplish the project. We would like to express our thanks to our Honourable Chairman **Dr. S. Ziauddin Ahemed SAHEB B.A.**, and to our beloved Correspondent **Dr. V. M. ABDUL LATHEEF SAHEB B.E., FIE.**, for their encouragement and guidance.

We wish to express our sincere thanks to our respected principal **Dr. M. Sasi Kumar, Ph.D.**, who has been kind enough to us in all aspects.

I also take this opportunity to express my sincere thanks to our head of the department **Dr.S.Umamaheswari, Ph.D** and our dean-placement **Mr. K.Mohammed Abdul Kader**, who has provided all the needful help for me in executing the project successfully.

I wish to express my thanks to our head of the Department, **Dr. S.Umamaheswari, Ph.D.**, for her encouragement and support to complete this project. I express my heartfelt gratitude to our guide,

MR.T.S.KarthickM.E., Associate Professor, Department of Information Technology for his priceless guidance and motivation which helped me to bring this project to a perfect shape who encouraged me in each and every step of this project to complete it successfully.

Finally, our sincere thanks to our lovely parents and friends who had been the constant source of our strength throughout our life.

ABSTRACT

Cancer is one of the main reason for loss of human life across the world. All the medical practitioners and researchers are dealing with the demanding situations to fight against cancer. Based on the report in 2019 from American Cancer Society, 96,480 deaths are anticipated due to skin cancers, 142,670 deaths are from lung cancers, 42,260 deaths are from breast cancers, 31,620 deaths are from prostate cancers, and 17,760 deaths are from mind cancers. Initial detection of most cancers has the pinnacle precedence for saving the lives. So it is important to detect the lung cancer as to give proper medical treatments. This project proposed a lung cancer detection using Deep Learning based on CNN Algorithm. Deep learning is used for detection and it is an efficient method used for image processing and pattern recognition. For Lung Cancer detection, Kaggle datasets taken from DICOM files of lung cancer patients are used for detection. Since the data provided by the CT scans of patients involves complicated pre-processing methods in order to get into the form that is usable to apply deep learning methodologies to it. CNN is used to classify and detect nodules using these datasets. In this project, we apply deep learning algorithms to identify lung cancer and its presence without the need for several consultations from different doctors. Lung nodules are predicted on this data using the nodule detector by running on the CT scans as grids.

உலகில் மனித உயிர்கள் பலியாவதற்கு புற்றுநாய் முக்கிய காரணமாகும். அனைத்து மருத்துவ பயிற்சியாளர்கள் மற்றும் ஆராய்ச்சியாளர்கள் புற்றுநாய்க்கு எதிராக நபாராட நவண்டிய சூழ்நிலைகளை எனகயாளுகின்றனர். அமமரிக்கன் நகன்சர் மசானசட்டியின் 2019 அறிக்கையின் அடிப்படையில், நதால் புற்றுநாயால் 96,480 இறப்புகள் எதிர்பார்க்கப்படுகின்றன, 142,670 இறப்புகள் நுனரயீரல் புற்றுநாயால், 42,260 இறப்புகள் மார்பக புற்றுநாயால், 31,620 இறப்புகள் புநராஸ்நட்ட புற்றுநாயால், மற்றும் 17,760 இறப்புகள் மை புற்றுநாயால் ஏற்படுகின்றன. மபரும்பாலாை புற்றுநாய்களின் ஆரம்பக் கண்டறிதல் உயிர்களைக் காப்பாற்றுவதற்காை உச்ச முன்னுரினமனயக் மகாண்டுள்ளது. எைநவ நுனரயீரல் புற்றுநானயக் கண்டறிந்து முனறயாை மருத்துவ சிகிச்சகளை வழங்குவது அவசியம். இந்த திட்டம் சிஎன்என் அல்காரிதம் அடிப்படையிலாை ஆழமாை கற்றனலப் பயன்படுத்தி நுனரயீரல் புற்றுநானயக் கண்டறிய முன்மமாழிந்தது. ஆழமாை கற்றல் கண்டறிதலுக்குப் பயன்படுத்தப்படுகிறது, நமலும் இது படச் மசயலாக்கம் மற்றும் வடிவ அங்கீகாரத்திற்குப் பயன்படுத்தப்படும் திறனமயாை முனறயாகும். நுனரயீரல் புற்றுநானயக் கண்டறிவதற்கு, நுனரயீரல் புற்றுநாயாளிகளின் DICOM

நகாப்புகளிலிருந்து எடுக்கப்பட்ட Kaggle தரவுத்மதாகுப்புகள் கண்டறிவதற்குப் பயன்படுத்தப்படுகின்றன. நாயாளிகளின் CT ஸ்நகன் மூலம் வழங்கப்படும் தரவு சிக்கலாை முன் மசயலாக்க முனறகளை உள்ளடக்கியிருப்பதால், அதற்கு ஆழ்ந்த கற்றல்

முனறகளைப் பயன்படுத்துவதற்குப் பயன்படும் படிவத்னதப் மபறலாம். இந்த தரவுத்மதாகுப்புகளைப் பயன்படுத்தி முடிச்சகளை வனகப்படுத்தவும் கண்டறியவும் CNN பயன்படுகிறது. இந்தத் திட்டத்தில், பல்நவறு மருத்துவர்களிடமிருந்து பல ஆநலாசனைகள் நதனவயில்லாமல் நுனரயீரல் புற்றுநானயயும் அதன் இருப்னபயும் கண்டறிய ஆழமாை கற்றல் வழிமுனறகளைப் பயன்படுத்துகிறதாம். CT ஸ்நகன்களை கட்டங்களாக இயக்குவதன் மூலம் நாடூல் டிமடக்டனரப் பயன்படுத்தி இந்தத் தரவில் நுனரயீரல் முடிச்சுகள் கணிக்கப்படுகின்றன

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
i	ABSTRACT	4
1.	INTRODUCTION 1.1 INTRODUCTION 1.2 SCOPE OF THE PROJECT 1.3 OBJECTIVES OF THE PROJECT	8
2.	SYSTEM REQUIREMENTS AND SPECIFICATION 2.1 SYSTEM ANALYSIS 2.2 EXISTING SYSTEM 2.2.1 DISADVANTAGES 2.3 PROPOSED SYSTEM 2.3.1 ADVANTAGES 2.4 SYSTEM SPECIFICATION 2.4.1 HARDWARE SPECIFICATION 2.4.2 SOFTWARE SPECIFICATION 2.5 SOFTWARE DESCRIPTION	10

3.	SYSTEM DESIGN 3.1 ARCHITECTURE DIAGRAM 3.2 USE CASE DIAGRAM 3.3 CLASS DIAGRAM 3.4 SEQUENCE DIAGRAM 3.5 COLLABORATION DIAGRAM	13
4.	MODULES	18
5.	SYSTEM TESTING	22
6.	CONCLUSION AND FUTURE WORKS 6.1 CONCLUSION 6.2 FUTURE SCOPE	28
7.	APPENDICES 7.1 APPENDIX-1 SAMPLE CODE 7.2 APPENDIX-2 SCREENSHOTS	29
8.	BIBLIOGRAPHY	47

Chapter1

INTRODUCTION

1.1Introduction

Lung cancer is also called lung carcinoma. This is a kind of harmful lung tumor considered as unrestrained cell growth in lung tissues. This development spread outside the lung by the method of metastasis into neighboring tissue or additional parts of the body. It is one of the leading reasons for death worldwide and it is one of the deadliest cancers. There are several stages of lung cancer. Stages of a cancer refers to how extensively cancerous cells has metastasized. Stages 1 and 2 refers ti cancers that are localized on lungs and later stages refers to cancers that have spread to other organs. Early detection of lung cancer significantly improves the chances for survival, but it is also more difficult to detect early stages of lung cancer as there are fewer symptoms. We will be using a binary classification problem to detect the presence of lung cancer in patient CT scans of lungs with and without early stage lung cancer. The aim is to use methods from computer vision and deep learning to detect lung cancer. Although many methods have been proposed for image classification, few of them focus on the classification of lung cancer type from CT images without biopsy. We use deep learning to identify lung cancer from CT images of patients. CNN are used to build an accurate classifier. Dataset for detection is extracted from the kaggle. An accurate lung cancer classifier could speed up and reduce costs of lung cancer screening, allowing for more widespread early detection and improved survival.

1.2 Scope of the Project

The Scope of this project is to detect the cancerous cells by using Deep Learning algorithms and to predict that the person is suffering from cancer or not. And also to reduce the death counts of the people who are killed by lung cancer all over the world.

1.3 Objective of project

An algorithm is required to solve a problem on a computer. An algorithm is a set of instructions that must be followed in order to convert input to output. For instance, one may create a searching algorithm. The input is the list of numbers and a value, and the output is an element searched and get from a list of numbers. There may be several algorithms for the same task, and we may be interested in finding the most efficient one by two parameters one is space complexity and the other is time complexity, needing the fewest instructions or memory, or both.

Deep learning is an algorithm that impersonates the network of neurons during a brain. It is a subset of machine learning which is named as deep learning and this uses deep neural network. A deep neural network delivers progressive accurateness in numerous tasks, from object recognition to speech recognition, and will acquire mechanically, without predefined information expressly coded by the application programmer. Deep learning approach is an influential technique to obtain better prediction of an actionable output. Deep learning outperforms in pattern identification and knowledge based forecast. Combining both will enhance the performance with extraordinary leads on factors such as innovation, management and production

Chapter 2

SYSTEM REQUIRMENTS AND SPECIFICATIONS

2.1 system analysis

The feasibility system of the project is assessed in this phase, and a business proposal is presented, along with a very generic project plan and some cost estimates. A feasibility system of the proposed system is to be carried out during system analysis. This is to ensure that the planned system will not cause the organization any problems. Understanding the system's primary requirements is necessary for feasibility analysis

2.2 Existing System

In the existing system which is a manual process that patient needs to visit the hospital for prediction. First the patient needs to go to the hospital and get an appointment from the doctor to discuss about the cancer. In the discussion the patient needs to explain the symptoms of cancer. Then the doctor needs to study the case and patient needs to wait for an hours and then take CT scan and after evaluation the patient can know that whether they are having cancer or not. By these process lots of time and money is wasted.

2.2.1 Disadvantages

Although, CT scan imaging is best imaging technique in medical field, it is difficult for doctors to interpret and identify the cancer from CT scan images. It is found that some has low accuracy and some has higher accuracy but not nearer to 50 percentage. Therefore, our research targets to increase the accuracy towards 80 percentage. Not all the time the patient gets an exact output from the CT scan it will

not clearly show the nodules of the lungs which is very important for detection and some times it gives a wrong output.

2.3 Proposed System

By these detection system patient can easily know whether they are affected by cancer or not by an application. Therefore computer aided diagnosis can be helpful for doctors to identify the cancerous cells accurately. Many computer aided techniques using image processing and machine learning has been researched and implemented. The main aim of this research is to evaluate the various computeraided techniques, analyzing the current best technique and finding out their limitation and drawbacks and finally proposing the new model with improvements in the current best model. The method used was that lung cancer detection techniques were sorted and listed on the basis of their detection accuracy. The techniques were analyzed on each step and overall limitation, drawbacks were pointed out.

2.3.1 ADVANTAGES

1. This system can detect lung cancer by CT scan images more accurately.
2. It will decrease the human error which leads to death of valuable lives.
3. Time period for evaluation of exam for doctors is reduced by this system.
4. Using our detection method cancer is detected by detailed analysis of the nodules of the cancer.

2.4 System Specification

2.4.1 Hardware Specification

- Processor -Intel i5 or higher

- RAM -8GB or higher
- Storage-512GB or higher
- Connectivity - WIFI

2.4.2 Software Specification

- Anaconda
- Visual studio code

2.5 SOFTWARE DESCRIPTION

Anaconda Prompt

Anaconda prompt is an IDE through which we can work with python in a secure manner. It is a type of command line interface which explicitly deals with the ML(Machine Learning) modules. And navigator is available in all the Windows, Linux and Mac OS. The anaconda prompt has many number of IDE's which make the coding easier. The UI can also be implemented in python. Standard Used: ISO/IEC 27001

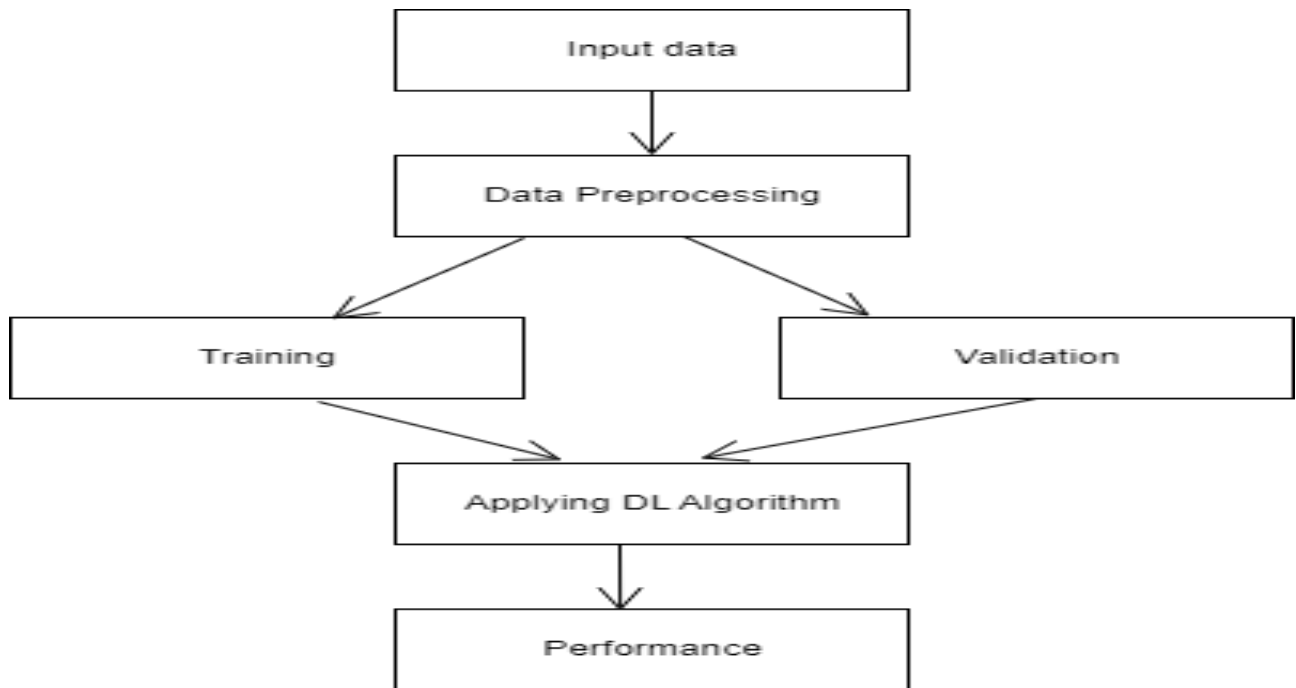
Visual studio code

Visual Studio Code also commonly referred to as VS Code. It is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and mac OS. It has many Features which includes support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. It can be used with a variety of programming languages including Java, JavaScript, Go, Node.js, Python, C++, C, Rust and Fortran.

CHAPTER – 3

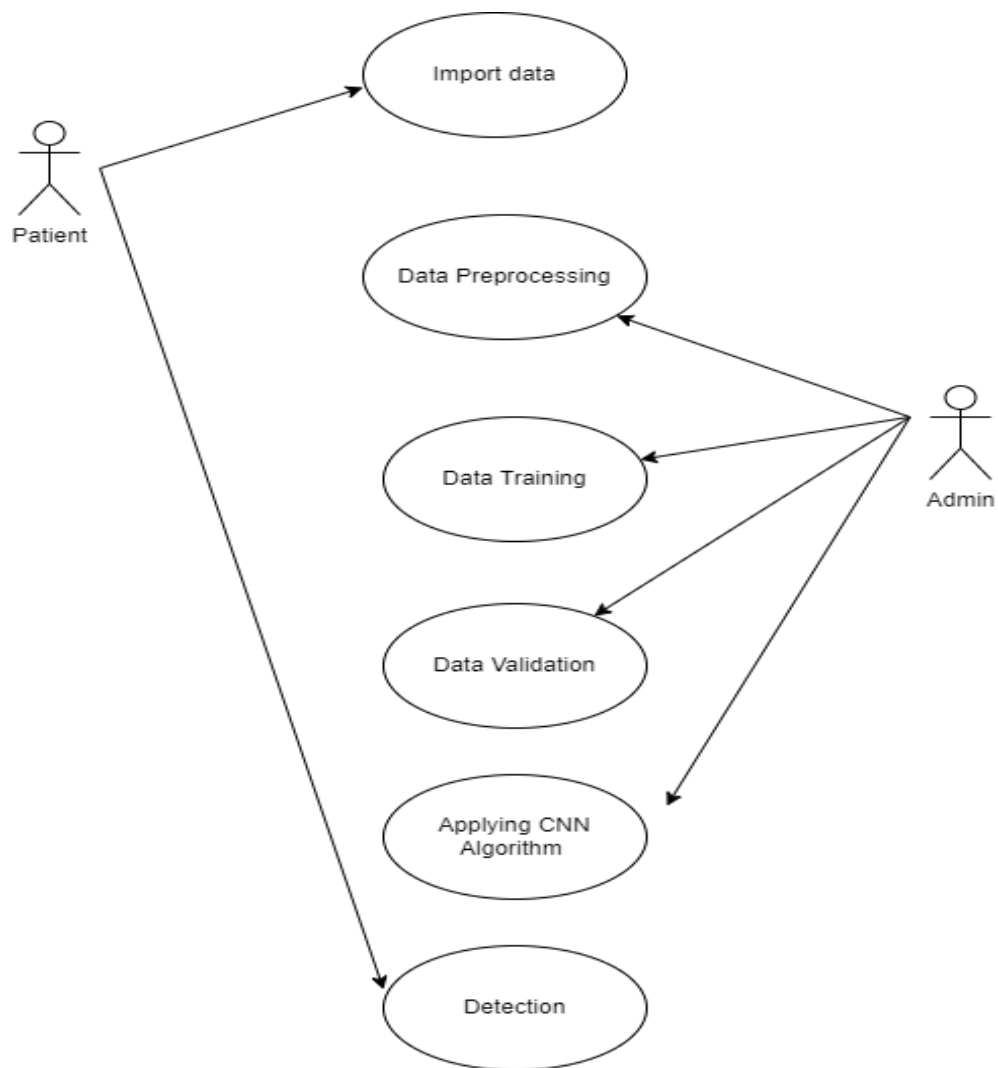
SYSTEM DESIGN

3.1 ARCHITECTURE DIAGRAM



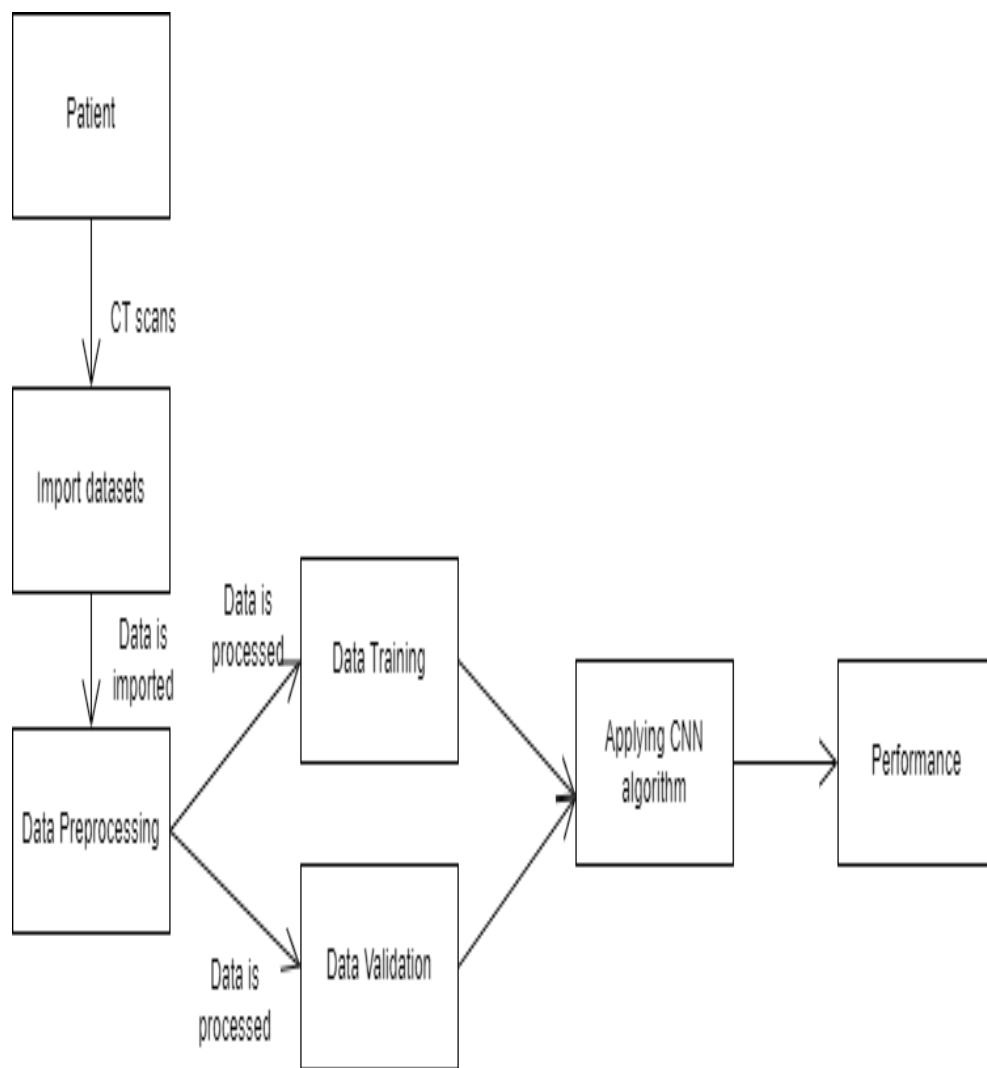
A systems architecture is the conceptual model that specifies a systems structure, behaviour and other perspectives. A formal description and representation of a system is arranged in a way that facilitates reasoning about the system's structures and behaviours are called as architecture diagram. Here the data is collected and preprocessed and the data is trained and validated for variables and after that we apply deep learning algorithms to predict that the patient is having cancer or not. A system architecture is made up of system components and subsystems that will collaborate to construct the overall system.

3.2 Use Case Diagram



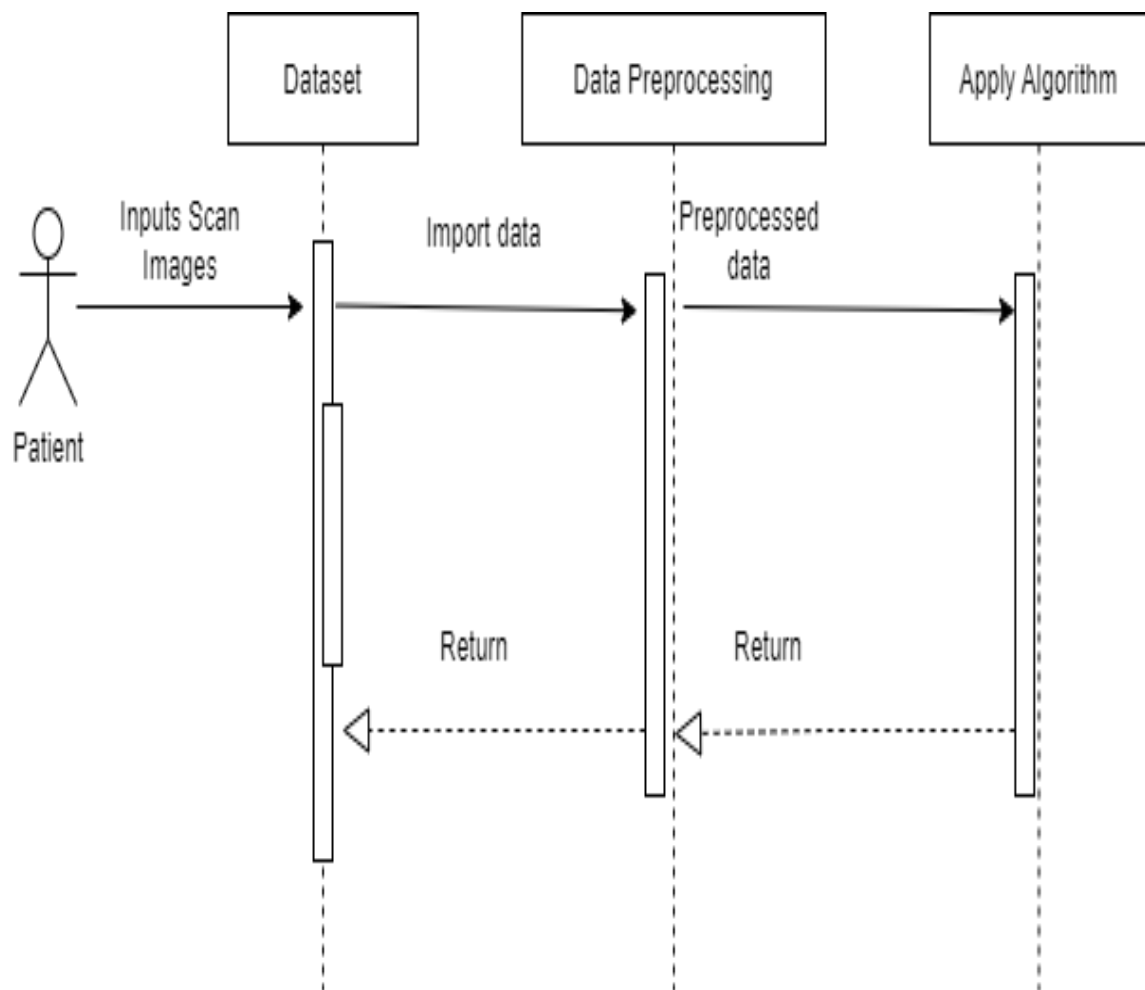
In the UML, a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. The purpose of a use case diagram in UML is to demonstrate the different ways that a user might interact with a system.

3.3 Class Diagram



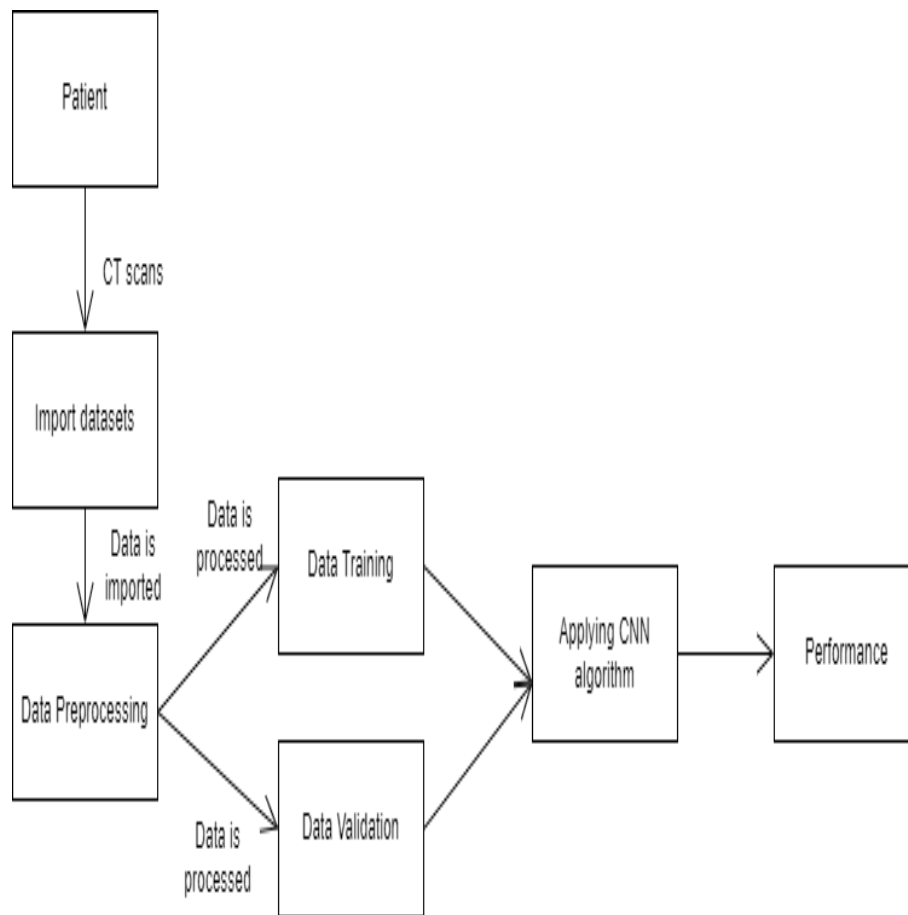
In the UML, a class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling.

3.4 Sequence Diagram



In the UML, a sequence diagram or system sequence diagram (SSD) shows process interactions arranged in time sequence in the field of software engineering. It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality. Sequence diagrams are typically associated with use case realizations in the 4+1 architectural view model of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

3.5 Collaboration diagram



In the UML, the collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

CHAPTER – 4

MODULES DESIGN AND METHODOLOGIES

► MODULE 1

Importing libraries and collecting the datasets and importing it.

► MODULE 2

Data Pre processing is performed to convert the raw data into the understandable format.

► MODULE 3

Data Training is performed to train the model based on the dataset and then predict on new data.

► MODULE 4

Data is validated according to the datasets and the result is produced in GUI for user.

MODULE 1

Step 1 : Importing all the required libraries.

Pandas: This library helps to load the data frame in a 2D array format and has multiple functions to perform analysis tasks in one go.

Matplotlib: This library is used to draw visualizations.

Sklearn: This module contains multiple libraries having pre-implemented functions to perform tasks from data preprocessing to model development and evaluation.

Tenserflow: This is an open-source library that is used for Machine Learning and Artificial intelligence and provides a range of functions to achieve complex functionalities with single lines of code. Step 2 : Collecting the datasets from Kaggle.

Step 3: Importing the data from external sources

MODULE 2

Step 1 :

Dataset is collected from the images of dicom files.

Step 2 :

Average is calculated.

Step 3 :

Check if data labels are available in csv or not.

Step 4 :

Check the results of data preprocessing.

MODULE 3

Step 1 :

Take the numpy data and load it.

Step 2 :

Data is trained from the numpy data.

Step 3 :

Apply the CNN algorithm.

Step 4 :

Prediction is performed and the actual results will be displayed.

MODULE 4

Step 1 :

Data is validated.

Step 2 :

Outputs from the data validation is displayed on the screen according to the datasets and data training.

Step 3 :

Outputs are displayed in GUI Screen.

4.2 Module Description

4.2.1 Import libraries and datasets

Step1:Importing all the required libraries like Pandas,Matplotlib,Sklearn,Tensorflow,Cv.

Step 2 : Collecting the datasets from Kaggle.

Step 3: Importing the data from external sources

4.2.2 Apply Data Preprocessing

Step 1 : Dataset is collected from the images of dicom files.

Step 2 : Average is calculated.

Step 3 : Check if data labels are available in csv or not. Step 4 :

Check the results of data preprocessing.

4.2.3 Apply Data Training

Step 1 : Take the numpy data and load it.

Step 2 : Data is trained from the numpy data.

Step 3 : Apply the CNN algorithm.

Step 4 : Prediction is performed and the actual results will be displayed

CHAPTER – 5

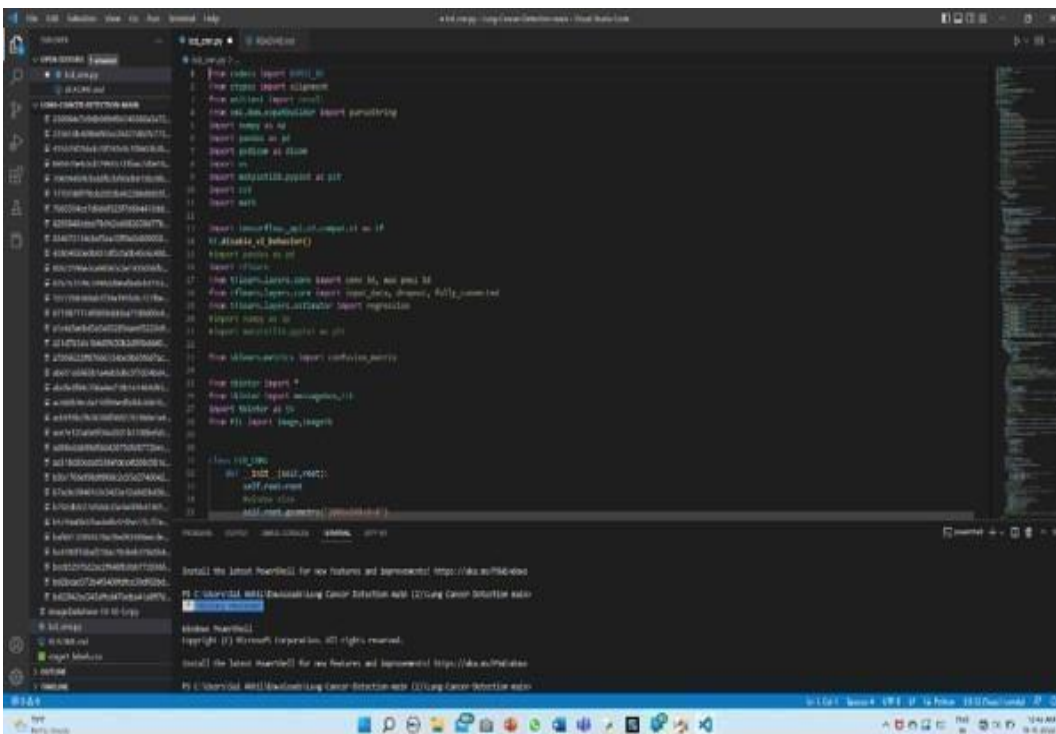
TESTING

5.1 Testing

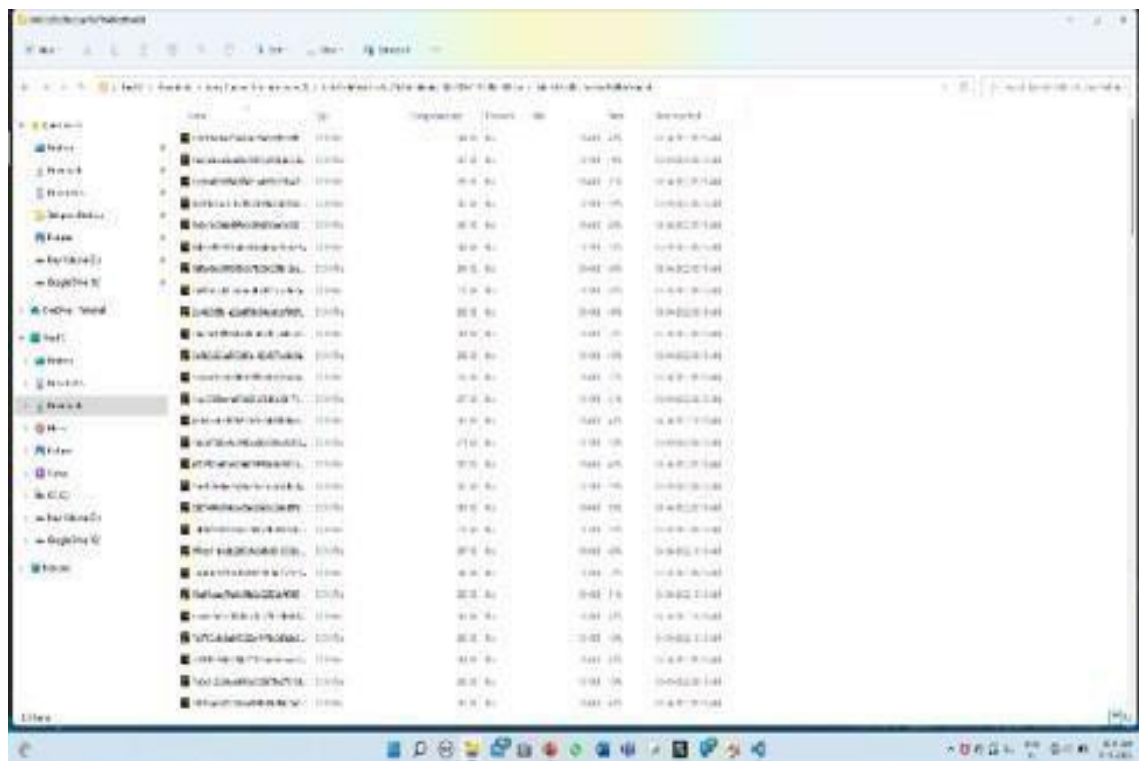
Types of Testing

- **Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and or system configuration. Input



Test result



Name	Duration	Passed	Failed	Skipped	Reported
Test Case 1	00:00:01.123	1	0	0	00:00:01.123
Test Case 2	00:00:01.456	1	0	0	00:00:01.456
Test Case 3	00:00:01.789	1	0	0	00:00:01.789
Test Case 4	00:00:02.012	1	0	0	00:00:02.012
Test Case 5	00:00:02.345	1	0	0	00:00:02.345
Test Case 6	00:00:02.678	1	0	0	00:00:02.678
Test Case 7	00:00:02.901	1	0	0	00:00:02.901
Test Case 8	00:00:03.234	1	0	0	00:00:03.234
Test Case 9	00:00:03.567	1	0	0	00:00:03.567
Test Case 10	00:00:03.890	1	0	0	00:00:03.890
Test Case 11	00:00:04.123	1	0	0	00:00:04.123
Test Case 12	00:00:04.456	1	0	0	00:00:04.456
Test Case 13	00:00:04.789	1	0	0	00:00:04.789
Test Case 14	00:00:05.012	1	0	0	00:00:05.012
Test Case 15	00:00:05.345	1	0	0	00:00:05.345
Test Case 16	00:00:05.678	1	0	0	00:00:05.678
Test Case 17	00:00:05.901	1	0	0	00:00:05.901
Test Case 18	00:00:06.234	1	0	0	00:00:06.234
Test Case 19	00:00:06.567	1	0	0	00:00:06.567
Test Case 20	00:00:06.890	1	0	0	00:00:06.890
Test Case 21	00:00:07.123	1	0	0	00:00:07.123
Test Case 22	00:00:07.456	1	0	0	00:00:07.456
Test Case 23	00:00:07.789	1	0	0	00:00:07.789
Test Case 24	00:00:08.012	1	0	0	00:00:08.012
Test Case 25	00:00:08.345	1	0	0	00:00:08.345
Test Case 26	00:00:08.678	1	0	0	00:00:08.678
Test Case 27	00:00:08.901	1	0	0	00:00:08.901
Test Case 28	00:00:09.234	1	0	0	00:00:09.234
Test Case 29	00:00:09.567	1	0	0	00:00:09.567
Test Case 30	00:00:09.890	1	0	0	00:00:09.890

Figure 5.3: Unit Test Result

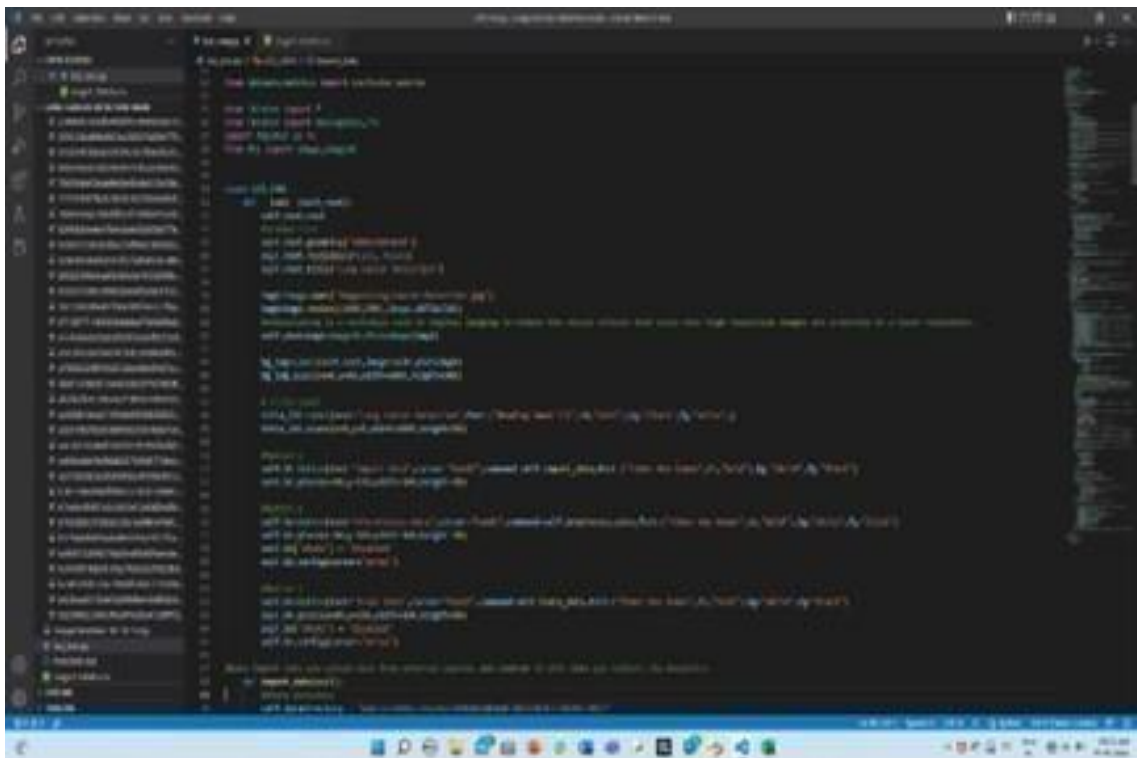
• Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Input

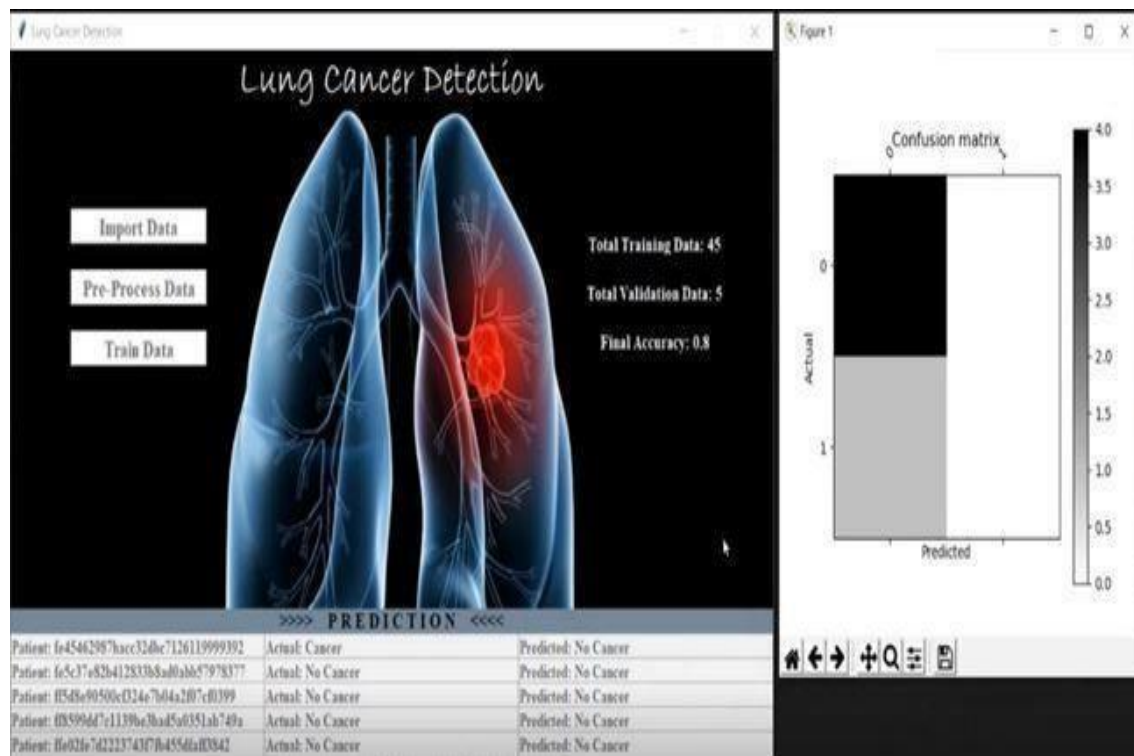
requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested. System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer.

Input



This testing, completely evaluates how the various components of an application interact together and various errors are detected when sent as an input to testing. Test

Result



The Test results are produced when we apply data preprocessing, training and validation and after CNN algorithm we get an prediction of results in the confusion matrix. In the Machine Learning, this confusion matrix is a useful method which is used to define the performance of the classification algorithms and measure the accuracy and precision.

CHAPTER-6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

As we all know that, technology advances at a rapid pace, entering all aspects of fields. This detection system does not lag behind the competition in this regard. In this project, Data was collected and trained using deep learning algorithm and the predicted data has been tested. To analyze the prediction efficiently. Also a confusion matrix is generated in order to provide a better view for the doctors and patients. It showed high performance based on chest radiographs. This detection method saves time and effort, and gives accurate results because we have used CNN algorithm, it differentiates the images and gives the reliable result according to the datasets and images. This method aims to build an algorithm to classify and detect the cancerous cells and non cancerous cells. This will aid in the improvement of the detection , and increase the performance and the reduction of costs.

6.2 Future work

Later we can create an application for the detection of all types of cancers so that the patients can easily know that they are affected by any cancer and what type of cancer and details about cancer or not affected. And also we can create an option to predict that in future whether there are chances of getting cancer by their previous reports and this can easily be done by uploading their scan images without visiting an doctor and waiting for longer time. Artificial intelligence is expected to increase the precision of medical care by improving disease prediction, diagnosis, treatment, and outcome.

CHAPTER – 7

APPENDICES

7.1 APPENDIX 1 - SAMPLE CODE

```
from codecs import BOM32_BE
from ctypes import alignment
from unittest import result
from xml.dom.expatbuilder import parseString
import numpy as np
import pandas as pd
import pydicom as dicom
import os
import matplotlib.pyplot as plt
import cv2
import math

import tensorflow._api.v2.compat.v1 as tf
tf.disable_v2_behavior()
import pandas as pd
import tflearn
from tflearn.layers.conv import conv_3d ,
maxpool_3d
from tflearn.layers.core import input_data ,
dropout,fully_connected
from tflearn.layers . estimator import regression
import numpy as np
import matplotlib.pyplot as plt
```

```
from sklearn.metrics import confusion_matrix
```

```
from tkinter import *
```

```
from tkinter import messagebox , ttk
```

```
import tkinter as tk
```

```
from PIL import Image , ImageTk
```

```
class LCD_CNN:
```

```
    def __init__(self,root):
```

```
        self.root = root
```

```
        self.root.geometry("1006x500+0+0")
```

```
        self.root.resizable (False , False)
```

```
        self.root.title("Lung Cancer Detection")
```

```
img4=Image.open("Images\Lung-Cancer-Detection.jpg")
```

```
img4=img4.resize((1006,500),Image.ANTIALIAS)
```

```
    self.photoimg4  
=ImageTk.PhotoImage(img4)
```

```
    bg_img=Label(self.root,  
image=self.photoimg4)
```

```
bg_img.place(x=0,y=50,width
=1006,height=500)
```

```
title_lbl=Label(text="Lung Cancer
Detection",font=("Bradley Hand
ITC",30,"bold"),bg="black",fg ="white",)
title_lbl.place(x=0,y =0,width =1006,height
=50)
```

```
self.b1=Button(text="Import Data
",cursor="hand2",command=self.import_data,fon
t=("Times New
Roman",15,"bold"),bg="white",fg =" black")
self.b1.place(x=80,y=130,width=180,height
=30)
```

```
self.b2=Button(text="Pre-Process
Data",cursor =" hand2 " , command=
self.preprocess_data,font=( "Times New
Roman",15," bold") , bg=" white " , fg =" black
")
self.b2.place (x=80,y=180 , width
=180,height =30)
self.b2 ["state"]="disabled"
self.b2.config (cursor="arrow")
```

```

        self.b3= Button(text="Train Data",cursor="
hand2",command=self.train_data,font=("Times
New Roman",15,"bold"),bg="white",fg
="black")

        self.b3.place(x=80,y=230,width =180 ,
height =30)

        self.b3 ["state"]= "disabled "
        self.b3.config(cursor="arrow")
def import_data(self):
    self.dataDirectory = 'sampleimages/'
    self.lungPatients = os.listdir (self.
dataDirectory)

    self.labels = pd . read_csv('stage1_labels.
csv ' , index_col =0)

    self.size = 10

    self.NoSlices = 5

    messagebox . showinfo("Import Data " ,
"Data Imported Successfully!")

    self.b1["state"] = "disabled"
    self.b1.config(cursor=" arrow " )
    self.b2["state"]="normal"
    self.b2.config(cursor="hand2")

```



```
def preprocess_data(self):
```

```
def chunks(l, n):
```

```
    count=0
```

```
    for i in range(0,len (l),n):
```

```
        if (count < self.NoSlices):
```

```
            yield l[i:i+n]
```

```
            count=count + 1
```

```
def mean (l):
```

```
    return sum (l)/len(l)
```

```
def dataProcessing (patient, labels_df , size  
=10, noslices =5, visualize = False):
```

```
    label =
```

```
labels_df._get_value(patient,'cancer')
```

```
    path = self.dataDirectory + patient
```

```
    slices = [dicom . read_file( path+ '/' +s) for  
s in os .listdir(path)]
```

```
    slices.sort( key=lambda x : int  
(x.ImagePositionPatient[2]))
```

```
    new_slices = []
```

```

        slices =
[ cv2.resize(np.array(each_slice.pixel_array) ,
(size,size) ) for each_slice in slices ]

```

```

        chunk_sizes = math.floor(len(slices) /
noslices)
        for slice_chunk in chunks(slices ,
chunk_sizes) :
            slice_chunk=list(map(mean,
zip(*slice_chunk)))
            new_slices.append(slice_chunk)

```

```

        if label == 1:
            label = np.array([0,1])
        elif label == 0:
            label = np.array([1,0])
        return np.array(new_slices),label

```

```

imageData = []
for num, patient in
enumerate(self.lungPatients):
    if num % 50 == 0:
        print('Saved -',num)
    try:
        img_data,label=dataProcessing(patient,
self.labels, size=self.size, noslices=self.NoSlices)

```

```

        imageData.append([img_data,
label,patient])
    except KeyError as e:
        print('Data is unlabeled')

    np.save('imageDataNew-{}-{}-
{}.npy'.format(self.size, self.size, self.NoSlices),
imageData)

    messagebox.showinfo("Pre-ProcessData" ,
    "DataPre-Processing Done Successfully!")

    self.b2["state"]="disabled"
    self.b2.config(cursor="arrow")
    self.b3["state"]="normal"
    self.b3.config(cursor="hand2")

    def train_data(self):

        imageData=np.load('imageDataNew-10-10-
5.npy',allow_pickle=True)
        trainingData = imageData[0:45]
        validationData = imageData[5:50]

training_data=Label(text="TotalTrainingData: "+

```

```
str(len(trainingData)),font=("Times New  
Roman",13,"bold"),bg="black",fg="white",)
```

```
training_data.place(x=750,y=150,width=200,height=18)
```

```
validation_data=Label(text="Total  
Validation Data:  
"+str(len(validationData)),font=("Times New  
Roman",13,"bold"),bg="black",fg="white",)
```

```
validation_data.place(x=750,y=190,width=200,height=18)
```

```
x = tf.placeholder('float')
```

```
y = tf.placeholder('float')
```

```
size = 10
```

```
keep_rate=0.8
```

```
NoSlices=5
```

```
def convolution3d(x, W):
```

```
    return tf.nn.conv3d(x, W, strides=[1, 1, 1,  
1, 1], padding='SAME')
```

```
def maxpooling3d(x):
```

```

        return tf.nn.max_pool3d(x, ksize=[1, 2, 2,
2, 1],strides=[1, 2, 2, 2, 1],padding='SAME')

```

```

def cnn(x):
    x = tf.reshape(x,shape=[-1, size, size,
NoSlices, 1])
    convolution1 = tf.nn.relu(

convolution3d(x,tf.Variable(tf.random_normal([3
, 3, 3, 1,
32]))) + tf.Variable(tf.random_normal([32])))

convolution1 = maxpooling3d(convolution1)
    convolution2 = tf.nn.relu(

convolution3d(convolution1,tf.Variable(tf.rando
m_normal([3, 3, 3, 32, 64]))) + tf.Variable(
    tf.random_normal([64])))

convolution2 = maxpooling3d(convolution2)
    convolution3 = tf.nn.relu(

convolution3d(convolution2,tf.Variable(tf.rando
m_normal([3, 3, 3, 64, 128]))) + tf.Variable(
    tf.random_normal([128])))
    convolution3 =
maxpooling3d(convolution3)

```

```

convolution4 = tf.nn.relu(

convolution3d(convolution3,tf.Variable(tf.rando
m_normal([3, 3, 3, 128, 256]))) + tf.Variable(
    tf.random_normal([256])))
convolution4 =
maxpooling3d(convolution4)
convolution5 = tf.nn.relu(

convolution3d(convolution4,tf.Variable(tf.rando
m_normal([3, 3, 3, 256, 512]))) + tf.Variable(
    tf.random_normal([512])))
convolution5 =
maxpooling3d(convolution4)
fullyconnected =
tf.reshape(convolution5,[-1, 256])
fullyconnected = tf.nn.relu(
    tf.matmul(fullyconnected,
tf.Variable(tf.random_normal([256,
256]))) + tf.Variable(tf.random_normal([256])))
fullyconnected =
tf.nn.dropout(fullyconnected,keep_rate)

output=tf.matmul(fullyconnected,tf.Variable(tf.ra
ndom_normal([256,2]))) + tf.Variable(tf.random_n
ormal([2]))
return output

```

```

def network(x):
    prediction = cnn(x)
    cost =
tf.reduce_mean(tf.nn.softmax_cross_entropy_wit
h_logits(logits=prediction, labels=y))

optimizer=tf.train.AdamOptimizer(learning_rate
=1e-3).minimize(cost)
    epochs=100
    with tf.Session() as session:

session.run(tf.global_variables_initializer())
        for epoch in range(epochs):
            epoch_loss = 0
            for data in trainingData:
                try:
                    X = data[0]
                    Y = data[1]
                    _, c = session.run([optimizer,
cost],feed_dict={x: X, y: Y})
                    epoch_loss += c
                except Exception as e:
                    pass

            correct = tf.equal(tf.argmax(prediction,
1),tf.argmax(y, 1))

```

```
accuracy=tf.reduce_mean(tf.cast(correct,'float'))  
    print('Epoch',epoch+1,'completed out  
of',epochs,'loss:',epoch_loss)
```

```
print('Accuracy:',accuracy.eval({x:[i[0]for i in  
validationData],y:[i[1]for i in validationData]}))  
    x1=accuracy.eval({x:[i[0]for i in  
validationData],y:[i[1]for i in validationData]}))
```

```
    final_accuracy=Label(text="Final  
Accuracy:"  
+str(x1),font=("TimesNewRoman",13,"bold"),bg  
="black",fg="white",)
```

```
final_accuracy.place(x=750,y=230,width=200,height=18)
```

```
patients=[]  
actual=[]  
predicted=[]
```

```
finalprediction=tf.argmax(prediction,1)  
actualprediction=tf.argmax(y,1)  
for i in range(len(validationData)):
```

```
patients.append(validationData[i][2])
```



```

        for i in finalprediction.eval({x:[i[0]for i
in validationData],y:[i[1]for i in
validationData]}):
            if(i==1):
                predicted.append("Cancer")
            else:
                predicted.append("NoCancer")
        for i in actualprediction.eval({x:[i[0]for
i in validationData],y:[i[1]for i in
validationData]}):
            if(i==1):
                actual.append("Cancer")
            else:
                actual.append("NoCancer")
        for i in range(len(patients)):

print("-----")
print("-----")
        print("Patient:",patients[i])
        print("Actual:",actual[i])
        print("Predicted:",predicted[i])

print("-----")
print("-----")

y_actual = pd.Series(

```

```

        (actualprediction.eval( {x:[i[0]for i
in validationData],y:[i[1]for i in
validationData]})),
        name='Actual')
    y_predicted=pd.Series(
        (finalprediction.eval( {x:[i[0]for i
in validationData],y:[i[1]for i in
validationData]})),
        name='Predicted')

```

```

df_confusion=pd.crosstab(y_actual,y_predicted).
reindex(columns=[0,1],index=[0,1],fillvalue=0)
    print('Confusion Matrix:\n')
    print(df_confusion)

```

```

    prediction_label=Label(text=">>>>
PREDICTION
<<<<",font=("TimesNewRoman",14,"bold"),bg=
"#778899",fg="black",)

```

```

prediction_label.place(x=0,y=458,width=1006,height=20)

```

```

    result1=[]

```

```

    for i in range(len(validationData)):

```

```

        result1.append(patients[i])
        if(y_actual[i]==1):
            result1.append("Cancer")
        else:
            result1.append("NoCancer")

        if(y_predicted[i] == 1):
            result1.append("Cancer")
        else:
            result1.append("No Cancer")

    total_rows=int(len(patients))

    total_columns=int(len(result1)/len(patients))

    heading=["Patient:", "Actual:", "Predicted:"]

    self.root.geometry("1006x"+str(500+(len(patients)*20)-20)+"0+0")
    self.root.resizable(False,False)

    for i in range(total_rows):
        for j in range(total_columns):

```

```

        self.e = Entry(root,
width=42,fg='black',font=('TimesNewRoman',12,
'bold'))

        self.e.grid(row=i,column=j)

self.e.place(x=(j*335),y=(478+i*20))
        self.e.insert(END, heading[j] +
result1[j + i*3])
        self.e["state"]="disabled"
        self.e.config(cursor="arrow")

self.b3["state"]="disabled"
self.b3.config(cursor="arrow")

messagebox.showinfo("TrainData" ,
"Model Trained Successfully!")

def plotconfusionmatrix(df_confusion,
title='Confusion matrix',cmap=plt.cm.gray_r):\

plt.matshow(df_confusion,cmap=cmap)

plt.colorbar()

tick_marks=np.arange(len(df_confusion.columns
))

```

```
plt.title(title)
```

```
plt.xticks(tick_marks,df_confusion.columns,rotation=45)
```

```
plt.yticks(tick_marks,df_confusion.index)
```

```
plt.ylabel(df_confusion.index.name)
```

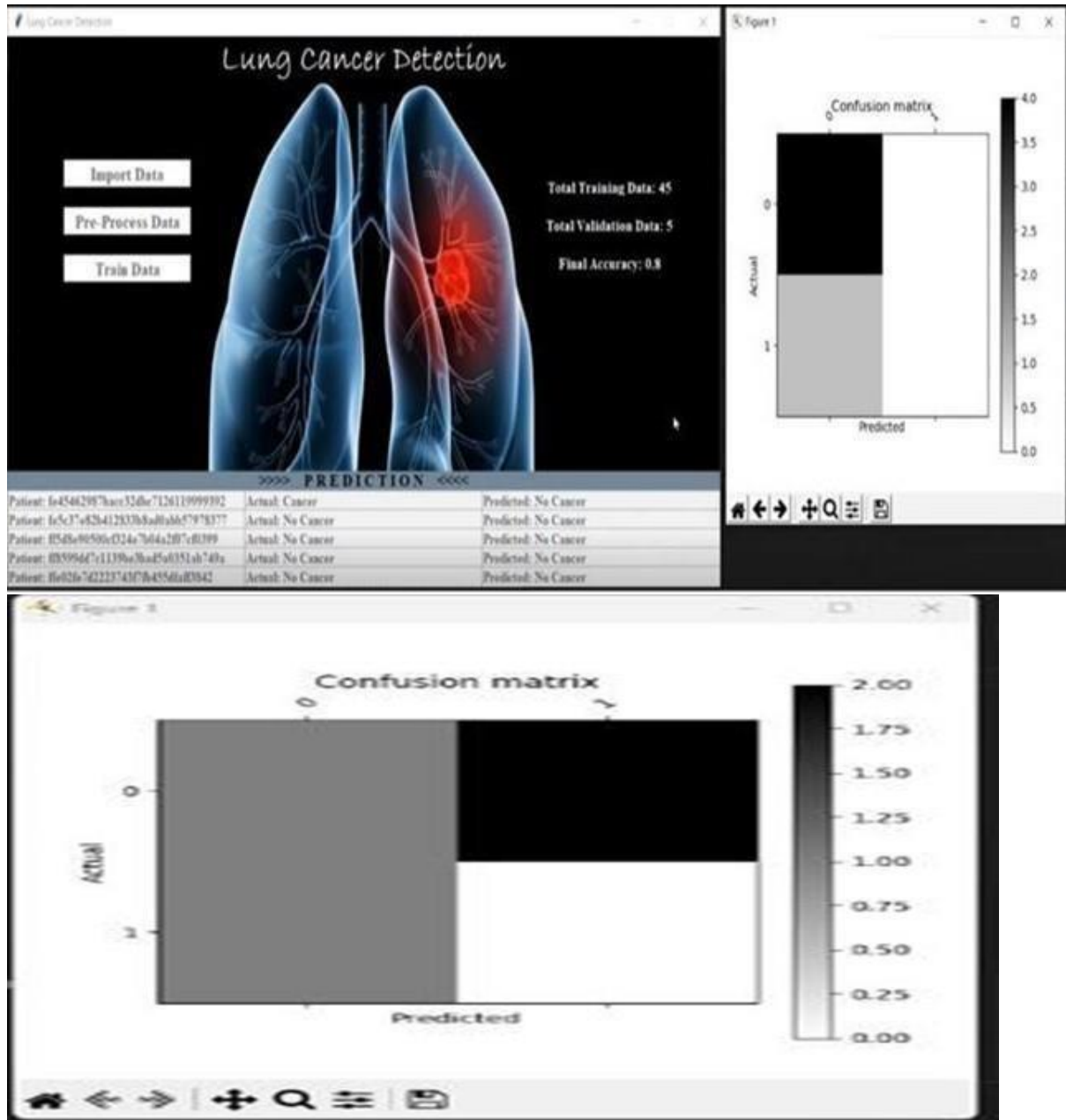
```
plt.xlabel(df_confusion.columns.name)  
plt.show()
```

```
plot_confusion_matrix(df_confusion)
```

```
network(x)
```

```
if __name__=="__main__":  
    root=Tk()  
    obj=LCD_CNN(root)  
    root.mainloop()
```

7.2 APPENDIX-2 SCREENSHOTS



BIBLIOGRAPHY

- [1] Narendra Mohan. "Tumor Detection From Brain MRI Using Modified Sea Lion Optimization Based Kernel Extreme Learning Algorithm", International Journal of Engineering Trends and Technology 68.9,84-100,2020.
- [2] AA. Borkowski, MM. Bui, LB. Thomas, CP. Wilson, LA. DeLand, SM. Mastorides. "Lung and Colon Cancer Histopathological Image Dataset."(LC25000). 1912.12142v1 [eess.IV], 2019.
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton et al., "ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems." 25, 10.1145/3065386,2021.
- [4] S. Sasikala, M. Bharathi, B. R. Sowmiya et al., "Lung Cancer Detection and Classification Using Deep CNN.", 2018.
- [5] L. G. Collins, C. Haines, R. Perkel , R. E. Enck et al., "Lung cancer: diagnosis and management." American family physician vol. 75, 1 : 56-63,2020.
- [6] T. Atsushi, T. Tetsuya, K. Yuka, and F. Hiroshi et al., "Automated Classification of Lung Cancer Types from Cytological Images Using Deep Convolutional Neural Networks" BioMed Research International. 1-6. 10.1155/2017/4067832,2018.