DETECTION OF TUBERCULOSIS USING IMAGE ENHANCEMENT AND SEGMENTATION

A PROJECT REPORT

Submitted by

SHRAVAN KARTHICK K R L	**211418104247**
SREEMITHUN S A	**211418104259**
SURYA S	**211418104281**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MAY 2022

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"DETECTION OF TUBERCULOSIS USING IMAGE ENHANCEMENT AND SEGMENTATION"** is the bonafide work of "**SHRAVAN KARTHICK K R L (211418104247), SREEMITHUN S A (211418104259), SURYA S (211418104281)"** who carried out the project work under my supervision.

SIGNATURE	SIGNATURE
Dr.S.MURUGAVALLI M.E.,Ph.D.,	**Mr.C.THYAGARAJAN, M.E.,Ph.D.,**
HEAD OF THE DEPARTMENT	**ASSISTANT PROFESSOR**
DEPARTMENT OF CSE,	DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE	PANIMALAR ENGINEERING COLLEGE
NAZARATHPETTAI,	NAZARATHPETTAI,
POONAMALLEE,	POONAMALLEE,
CHENNAI-600 123.	CHENNAI-600 123.

Certified that the above mentioned students were examined in the End Semester project viva-voce held on ___________________.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **SHRAVAN KARTHICK K R L (211418104247), SREEMITHUN S A (211418104259), SURYA S (211418104281)** hereby declare that this project report titled "**DETECTION OF TUBERCULOSIS USING IMAGE ENHANCEMENT AND SEGMENTATION**", under the guidance of **Mr.C.THYAGARAJAN, M.E.,Ph.D.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D**. and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Coordinator Dr.N.PUGHAZENDI M.E.,Ph.D.,** and **Project Guide Mr.C.THYAGARAJAN M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

SHRAVAN KARTHICK K R L

SREEMITHUN S A

SURYA S

ABSTRACT

Tuberculosis (TB) is an airborne infectious disease and a major health threat that is deleterious in most parts of the world. Most of the diagnostic methods are time consuming as well as unreliable and they were all mostly developed in the last century. Chest radiography is used as the most common method for screening TB in a large population. The success of this method depends solely on the experience and interpretation skills of the radiologist. Convolutional neural networks (CNN) is a deep learning strategy that has gained attention and popularity due to its ability to learn midlevel as well as high-level image representations. In this work, several CNN model used as google net model were used, which classifies the chest radiographs into TB positive and TB negative classes. This project offers a comparative study on the various deep learning techniques that can process chest x-rays and are capable of TB detection. The performance of the system is measured on a publicly available dataset: Tuberculosis (TB) Chest X-ray Database.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

S. NO	ACRONYMS	EXPANSION
1.	CNN	Convolutional Neutral Network
2.	TB	Tuberculosis
3.	CAD	Computer Aided Diagnosis
4.	HU	Hounsfield Unit
5.	CT	Computerized Tomography
6.	UML	Unified Modelling Language
7.	ILSVRC	ImageNet Large – Scale Visual Recognition Challenge
8.	ReLU	Rectified Linear Unit

CHAPTER - 1

INTRODUCTION

1. INTRODUCTION

Lung cancer is one in all the foremost common cancers, accounting for over 225,000 cases, 150,000 deaths, and $12 billion in health care prices yearly within the U.S. Its additionally one in all the deadliest cancers, overall, solely revolutionary organization 17 November of individuals within the U.S. diagnosed with carcinoma survive 5 years when the diagnosing, and also the survival rate is lower in developing countries. The date of a blight refers to about abundantly it's metastasized. Stages one and a pair of talk to cancers localized to the lungs and latter stages talk to cancers that have unfold to alternative organs. Current diagnostic strategies embody biopsies and imaging, like CT scans.

Early detection of carcinoma (detection throughout the sooner stages) considerably improves the possibilities for survival, however it's additionally tougher to observe early stages of carcinoma as there square measure fewer symptoms. Our task may be a binary classification downside to observe the presence of carcinoma in patient CT scans of lungs with and while not early stage carcinoma. We have a tendency to aim to use use strategies from pc vision and deep learning, notably 2nd and 3D convolution neural networks, to make AN correct classifier. AN correct carcinoma classifier may speed up and cut back prices of carcinoma screening, granting a lot of widespread early detection and improved survival. The ambition is to assemble a computer-aided diagnosing (CAD) arrangement that takes as ascribe accommodating chest CT scans and outputs whether or not or the accommodating has carcinoma.

Chest X-rays (CXR) square measure accustomed sight abnormalities. Theradiological choices show wide variation, however in most cases they are characteristic enough to suggest the identification. the foremost common choices square measure Cavitation: looks in 5 hundredth of the patients. at intervals the cavity there may even be a bit quantity of fluid, unreal as associate air fluid level. Lymphadenopathy: fissure and mediastinal nodes ar larger than usual. Patchy, poorly printed segmental consolidation: at intervals the top and posterior segments of the upper lobes, and within the superior section of the lower lobe. Miliary tuberculosis: TB is unfold through blood vessels and looks as multiple little nodules that square measure distributed uniformly. Extension to the membrane, resulting in membrane effusion: it's associate degree accumulation of fluid at intervals the area that looks as associate degree outsized white surface. These choices may be refined and not detectable for a personal that does not have the expertise. Hence, the aim of this project is to create a Convolutional Neural Network (CNN) that classifies X-rays as TB positive or TB negative.

CHAPTER - 2

LITERATURE SURVEY

2. LITERATURE SURVEY

Roslidar Roslidar Et al [13]

The segmentation algorithm employing edge detection and second-order polynomial curve fitting techniques can effectively capture the thermograms' region of interest (ROI), thereby facilitating efficient feature extraction. The classifier was developed based on ShuffleNet by adding one block consisting of a convolutional layer with 1028 filters. The modified Shufflenet demonstrated a good fit learning with 6.1 million parameters and 22 MB size. Simulation results showed that modified ShuffleNet alone resulted in a 72% accuracy rate, but the performance excelled to a 100% accuracy rate when integrated with the proposed segmentation algorithm.

Jun Gao Et al [14]

Computer-aided detection or diagnosis (CAD) has been a promising area of research over the last two decades. Medical image analysis aims to provide a more efficient diagnostic and treatment process for the radiologists and clinicians. However, with the development of science and technology, data interpretation manually in the conventional CAD systems has gradually become a challenging task. Deep learning methods, especially convolutional neural networks (CNNs), are successfully used as tools to solve this problem. This includes applications such as breast cancer diagnosis, lung nodule detection and prostate cancer localization.

Tarun Agrawal Et al [15]

The studies performed using the Generative Adversarial Network (GAN) models for segmentation and classification on chest X-rays are also included in this study. GAN has gained the interest of the CV community as it can help with medical data scarcity. In this study, we have also included the research conducted before the popularity of deep learning models to have a clear picture of the field. Many surveys have been published, but none of them is dedicated to chest X-rays. This study will help the readers to know about the existing techniques, approaches, and their significance.

Vinayakumar Ravi Et al [16]

Most of the pretrained CNN models achieved above 99% accuracy and less than 0.005 loss with 15 epochs during the training. All 7 different types of EfficientNet (ENet)-based CNN models performed better in comparison to other models in terms of accuracy, average and macro precision, recall, F1 score. Moreover, the proposed ENet-based CNN models performed better than other existing methods such as VGG16 and ResNet-50 for TB classification tasks.

Diwan Et al [17]

The application of contemporary technologies is important to medical progress. To create accurate and specialised treatment choices for a range of ailments, extensive study performed in partnership with researchers, health care professionals, and patients is important. This study aims to identify the degree of accuracy that is acceptable in the medical sector by using deep learning on publicly available data. First, we extracted spectrogram features and labels from the annotated lung sound recordings to feed into our 2D

Convolutional Neural Network (CNN) model. In this paper, we solve the problem of medical data scarcity by identifying pulmonary diseases from chest X-Ray pictures using small volume datasets with less than a thousand samples. Several studies have been conducted on the application of deep learning to identify lung disease have been published in the literature.

Ajmal shan Et al [18]

Tuberculosis (TB) is a main global health threat. An estimated one-third of the world's population has been exposed to TB, and millions of new infections are occurring every year. Tuberculosis naturally affects the lungs it also affects the other parts of our body. It is spread through air when infectious people cough, sneeze etc. The advent of new powerful hardware and software techniques has triggered attempts to develop computer-aided diagnostic systems for TB detection in support of inexpensive mass screening in developing countries. In this paper the medical background of TB detection in conventional posterior anterior chest X-rays has been described. In the first step the chest x-rays has been given as an input. In the second step, the selected images are segmented using graph cut segmentation method. In the last step asset of features has been extracted and calculated. Lastly, the multi-support vector machine is applied to classify the extracted feature vectors as normal or abnormal lungs. If it is abnormal, provide the name of the most matching TB manifestation of both lungs.

Lokeshwaran V.B. Et al [19]

Tuberculosis (TB) is an airborne infectious disease and a major health threat that is deleterious in most parts of the world. Most of the diagnostic methods are timeconsuming as well as unreliable and they were all mostly developed in the last century. Chest radiography is used as the most common method for screening TB in a large population. The success of this method depends solely on the experience and interpretation skills of the radiologist. Convolutional neural networks (CNN) is a deep learning strategy that has gained attention and popularity due to its ability to learn midlevel as well as high-level image representations. In this work, several CNN models such as alexnet were used, which classifies the chest radiographs into TB positive and TB negative classes. This paper offers a comparative study on the various deep learning techniques that can process chest x-rays and are capable of TB detection. The performance of the system is measured on a publicly available dataset: Tuberculosis (TB) Chest X-ray Database.

P.A.Kamble Et al [20]

Tuberculosis (TB) is very dangerous and rapidly spread disease in the world. When left undiagnosed and thus untreated, mortality rates of patients with tuberculosis are high. Standard diagnostics still rely on methods developed in the last century. They are slow and often unreliable. In the investigating cases for suspected tuberculosis (TB), chest radiography is not only the key techniques of diagnosis based on the medical imaging but also the diagnostic radiology. So, Computer aided diagnosis (CAD) has been popular and many researchers are interested in this research areas and different approaches have been proposed for the TB detection and lung decease classification. In this paper we present method for detection of Tuberculosis in CXR image by using MATLAB which includes Pre Processing of Image, Segmentation and Feature extraction from that image.

CHAPTER - 3
SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In this, to induce heaps of correct results we've an inclination to divide the work into the following three stages: Image Enhancement: to reinforce the image and eliminate the noise, corruption or interference, three methods area unit used: physicist filter (has the foremost effective results), motor vehicle improvement formula, and twin tree difficult wave make over. Typical radio densities of various parts of a CT scan area unit shown in Table one. Air is sometimes is sometimes HU, internal organ tissue is sometimes is sometimes, water, blood, and different tissues area unit around zero HU, and bone is sometimes around 700 HU.

Disadvantages

- If thresholding is low, correct detection not possible.
- Database is not used.
- Alex web model has less coaching capability with less accuracy.
- Gabor filter is employed that doesn't filter external noise.

3.2 PROPOSED SYSTEM

The segmentation obtained from thresholding options a heap of noise-many voxels that were a region of internal organ tissue, significantly voxels at the sting of the internal organ, tended to fall outside the vary of internal organ tissue radio density as a result of CT scan noise. this means that our classifier will not be able to properly classify footage throughout that cancerous nodules area unit settled at the sting of the internal organ. Image Segmentation: To segregate and part the improved footage, the methods used are: Thresholding approach and Marker-Controlled Watershed Segmentation approach (which offers higher results than thresholding). Choices Extraction stage: to induce the precise choices of the improved spellbinding image exploitation Binarization and Masking Approach.

Advantages

- Noise is reduced exploitation filter.
- Segmentation is completed for eliminating high frequencies.
- CNN as a classifier it checks heaps of inputs to classify.
- By exploitation GLCM we've an inclination to induce type choices of tumour.

3.3 FEASIBILITY STUDY

3.3.1 Introduction

A Feasibility Study includes a nitty gritty evaluation of the need, value, and common sense of a proposed enterprise, such as framework development. The procedure of outlining and executing record-keeping frameworks has adequate responsibility and asset suggestions for an association. Possibility study will enable you to settle on educate and straightforward choice at urgent focuses during the developmental procedure to decide if it is operationally, economically and in fact reasonable to deliver with a specific strategy.

3.3.2 Technical feasibility

This project is desktop based application .The main technologies that are associated with project are

- MATLAB 2021

- Image Processing Tool Box
- Deep Learning Tool Box
- Data Acquisition Tool Box
- Google Net (Pre-Trained Model)

Each of the technologies are freely available and the technical skills required are manageable. Time limitations of the product and the ease of implementing using these technologies are synchronized.

This requires less bandwidth to transfer data that stored in database, because in this we don’t transfer any multimedia file. From this it is clear that our project is technically feasible.

3.3.3 Economical Feasibility

Being a desktop application it will have no cost .In this, we don’t need any server because it’s a desktop application which can be installed easily in a couple of hours and does not require an internet connection. It reduces paper and printing cost and also the usage of paper is declined. Since, it is one time investment it will become profitable. It will apply in the local scan centers and Hospitals. From these it’s clear that this project is financially feasible.

3.3.4 Source Feasibility

This project proposes an desktop application based on Matlab which helps users to easily identify the presence of tuberculosis in the image(x-ray) and also assist doctors for faster identification of tuberculosis in patients. Cheaper and easier way to find the presence of tuberculosis in patients. The Graphical user interface of our project makes the application easy to handle for any kind of age groups.

3.4 HARDWARE REQUIREMENTS

3.4.1 Laptop/PC

Laptops (or) Personal computers are used to read the detection tuberculosis using image enhancement and segmentation. Fig 3.1, discusses about the devices used for Detection of T.B.

Fig.3.1 Devices used for Detection of Tuberculosis

3.5 SOFTWARE REQUIREMENTS

3.5.1 MATLAB

MATLAB is a programming and numeric computing platform employed by several users to analyse data, develop algorithms, and build models. MATLAB apps are interactive applications written to perform technical computing tasks. Apps are enclosed in several MATLAB products. Fig 3.2, is the symbol of MATLAB. The Apps tab of the MATLAB Tool strip shows you the apps that you currently have installed. There are many ways in which to urge a lot of apps: from MATLAB File Exchange, through further MATLAB products, and by building your own.

Fig.3.2 MATLAB

CHAPTER - 4
SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 E-R DIAGRAM

An entity–relationship model (ER model) describes reticular things of interest during a specific domain of data. A basic ER model consists of entity varieties (which classify the items of interest) and specifies relationships that may exist between instances of these entity varieties. In Fig 4.1, we've well-lighted the E-R diagram for our project.

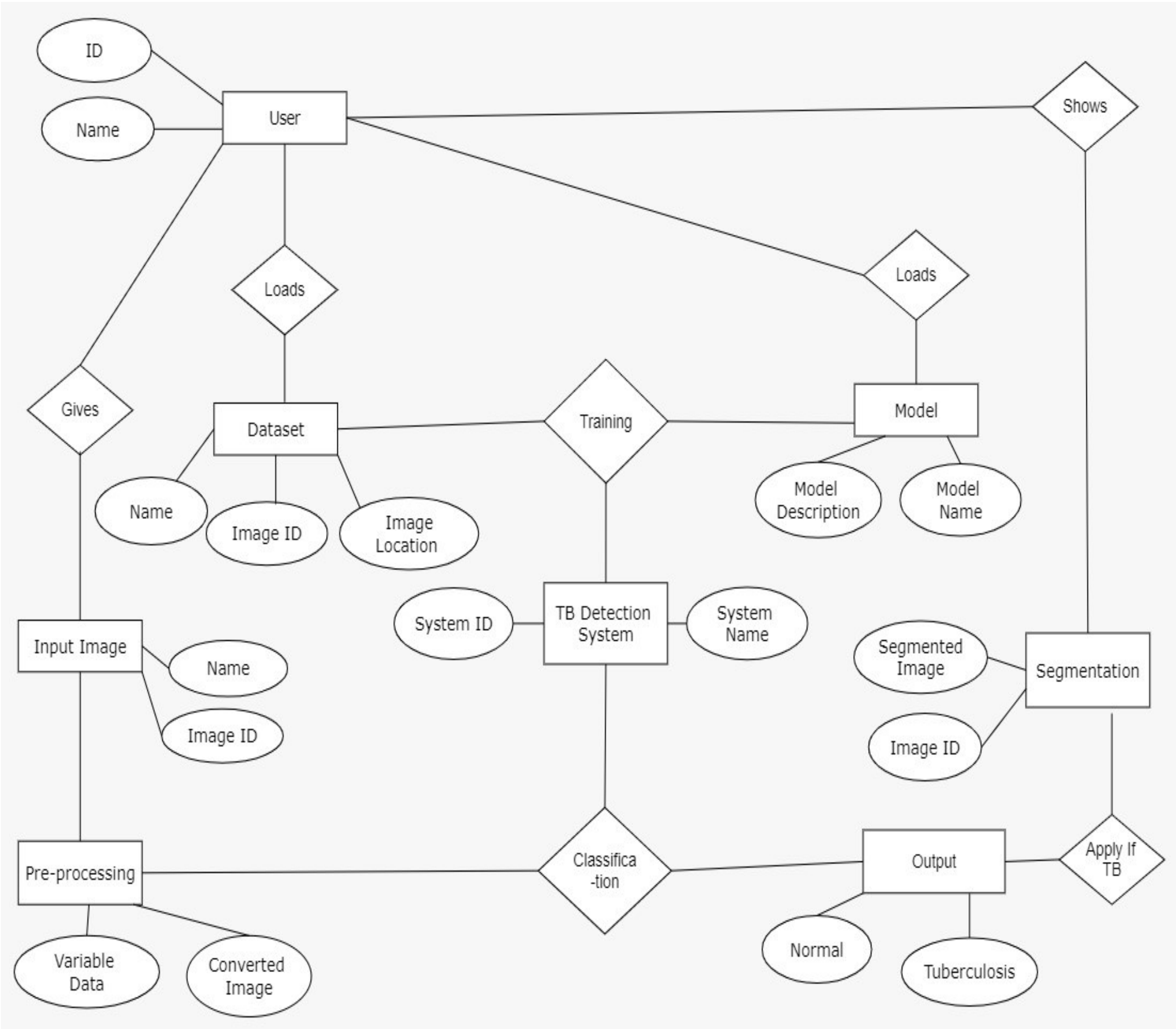


Fig.4.1 Entity-Relationship diagram

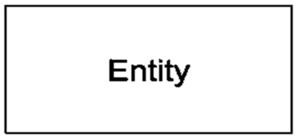
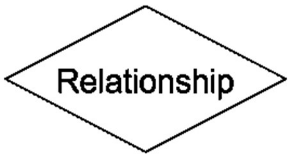
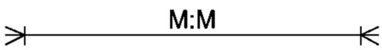
Symbol Name	Symbol	Description
Entity		An entity is represented by a rectangle which contains the entity's name.
Attribute		In the Chen notation, each attribute is represented by an oval containing attribute's name
Strong Relationship		A relationship where entity is existence-independent of other entities, and PK of Child doesn't contain PK component of Parent Entity. A strong relationship is represented by a single rhombus
One or More		It represents One or More
Many - to - Many		It represents a one through many on both sides of a relationship

Table 4.1 E-R Diagram Symbol Description

4.2 UML DIAGRAM

4.2.1 Use case diagram

Depicts the various users of the system and how they are going to use the system to meet the requirement objectives. Usecase Diagram for Detection of TB using image enhancement and segmentation contains 2 actors, some use cases and Data Source. In Fig 4.2, we have illuminated the Use Case diagram of this project.

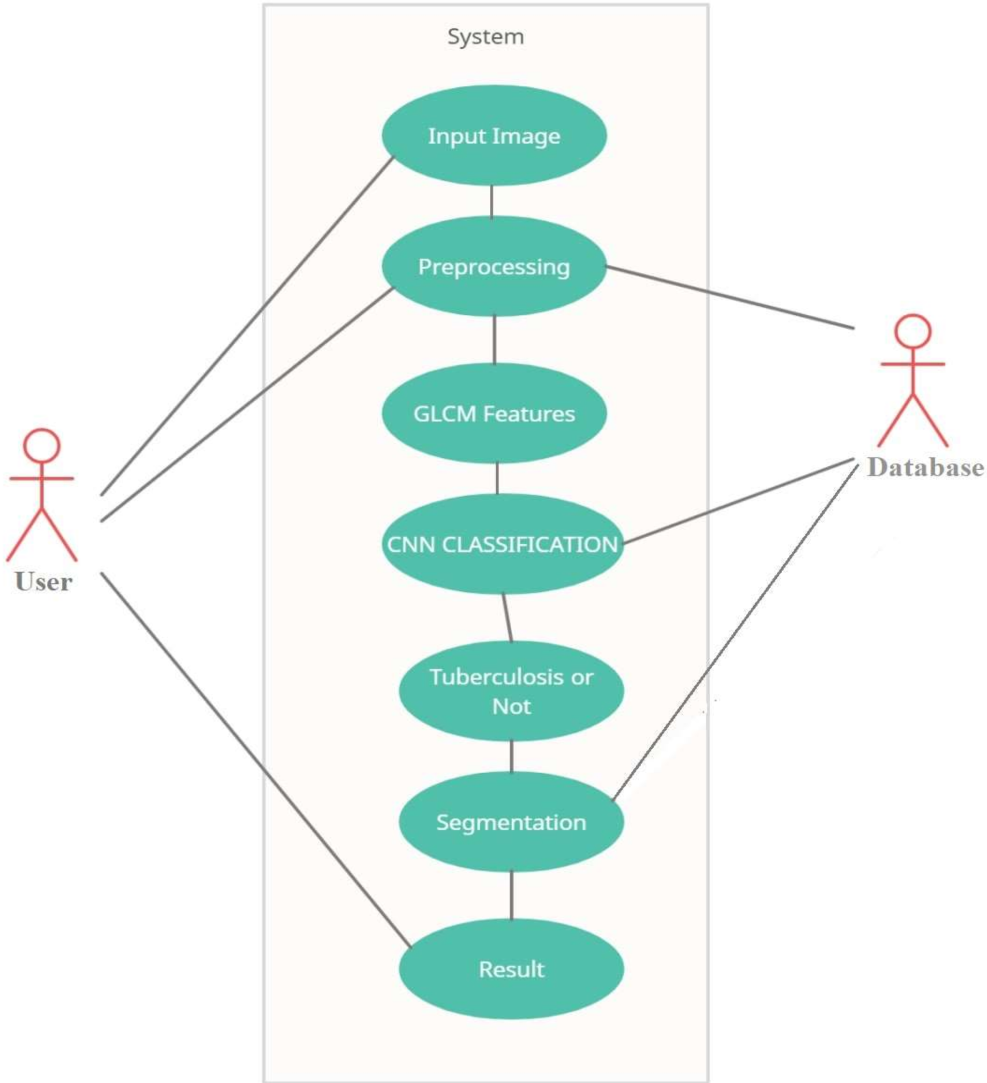


Fig.4.2 Usecase diagram

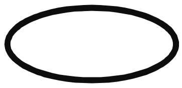
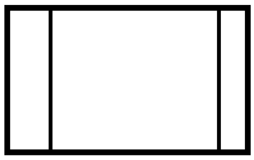
Symbol Name	Symbol	Description
Actor		Actors are the users of a system.
Usecase		Label the ovals with verbs that represent the system's functions.
Data Source		A Non-Human Actor is represented by this symbol.

Table 4.2 Usecase Diagram Symbol Description

4.2.2 Activity diagram

Activity diagram is another vital diagram in UML to explain the dynamic aspects of the system. Activity diagram is essentially a multidimensional language to represent the due one activity to a different activity. The activity is delineated as AN operation of the system. The management flow is drawn from one operation to a different. Fig 4.3, illuminates the Activity diagram of the project.

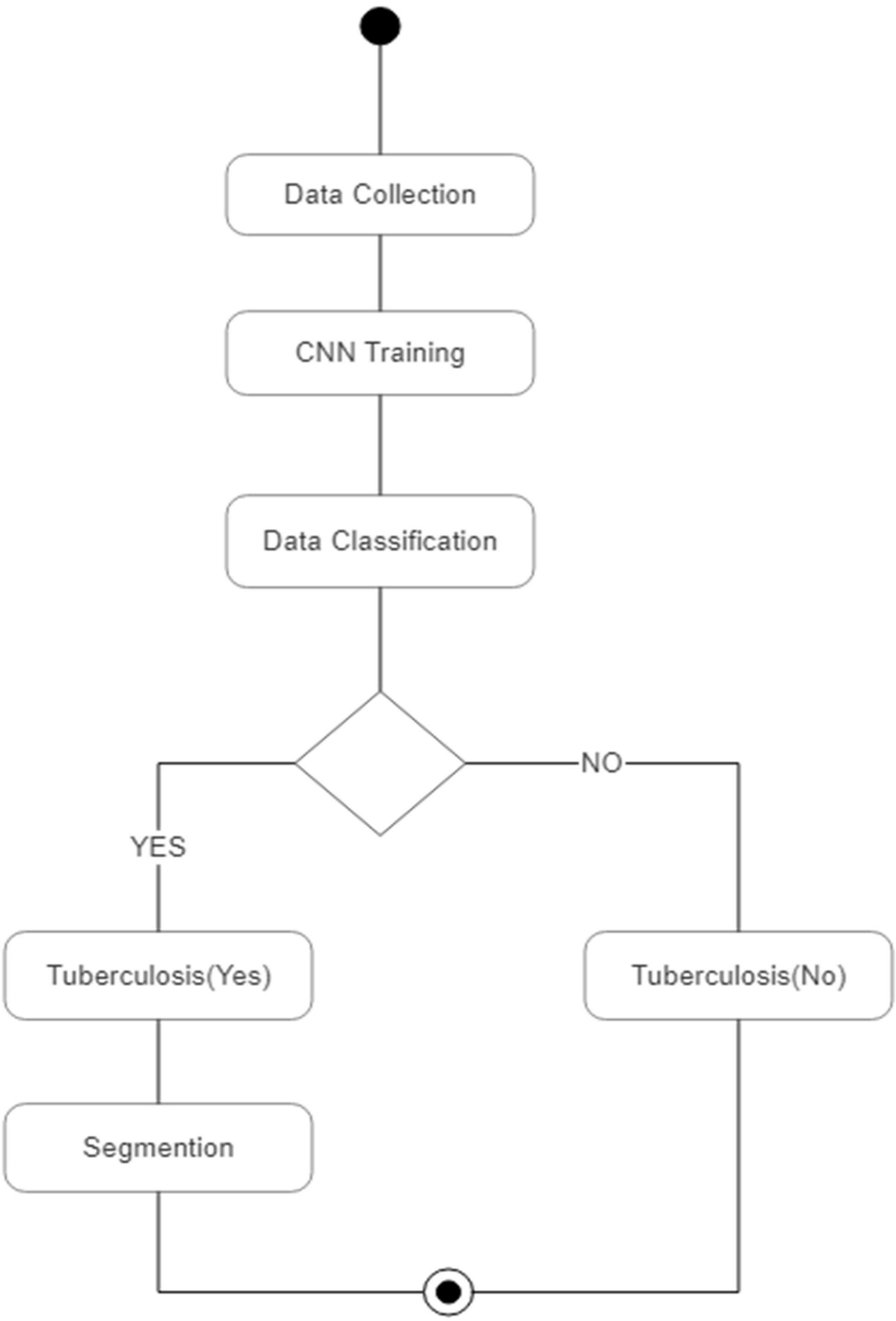


Fig.4.3 Activity diagram

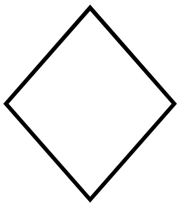
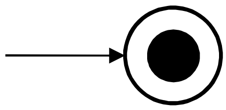
Symbol Name	Symbol	Description
Start/Initial State		A small filled circle followed by an arrow represents start point for any activity diagram.
Activity State		An action state represents the non-interruptible action of objects.
Decisions and Branching		A diamond represents a decision with alternate paths. The outgoing alternates should be labelled with a condition or guard expression. You can also label one of the paths "else."
Final State		An arrow pointing to a filled circle nested inside another circle represents the final action state.

Table 4.3 Activity Diagram Symbol Description

4.2.3 Class diagram

Class diagram, one amongst the foremost unremarkably used diagrams in object-oriented system, models the static style read for a system. The static read principally supports the practical needs of a system – the services the system ought to give to the tip users. We are going to see from our sensible expertise that legion fun comes out once modelling out system with category diagrams. A class diagram shows a collection of categories, interfaces, and collaborations and their relationships. Class diagrams involve international system description, like the system design, and detail aspects like the attributes and operations inside a category further. Fig 4.4 illuminates the category diagram of this project

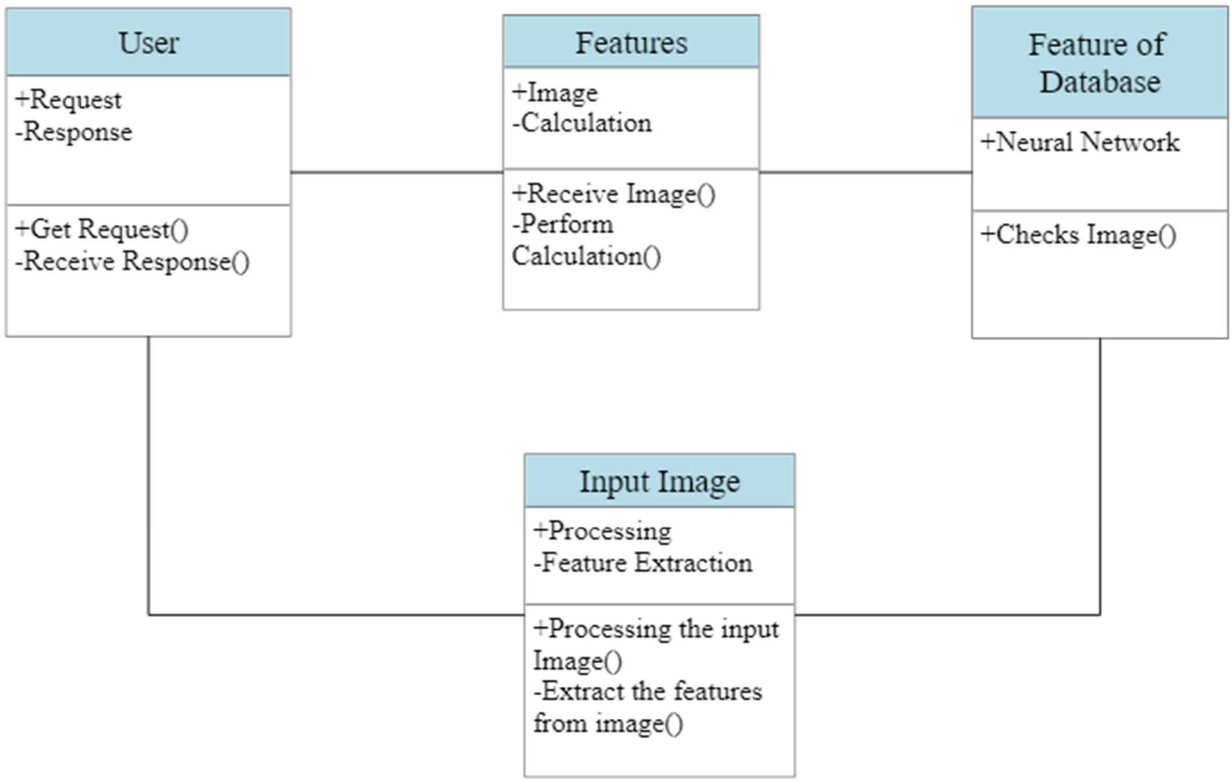


Fig.4.4 Class diagram

Symbol Name	Symbol	Description
Association		Associations are relationships between classes in a UML Class Diagram. They are represented by a solid line between classes.
Dependency	---------------->	An object of one class might use an object of another class in the code of a method. If the object is not stored in any field, then this is modeled as a dependency relationship.
Class	Class name Attributes Operations	A class represent a concept which encapsulates state (**attributes**) and behavior (**operations**). Each attribute has a type.

Table 4.4 Class Diagram Symbol Description

CHAPTER - 5

SYSTEM ARCHITECTURE

5. SYSTEM ARCHITECTURE

5.1 SYSTEM ARCHITECTURE

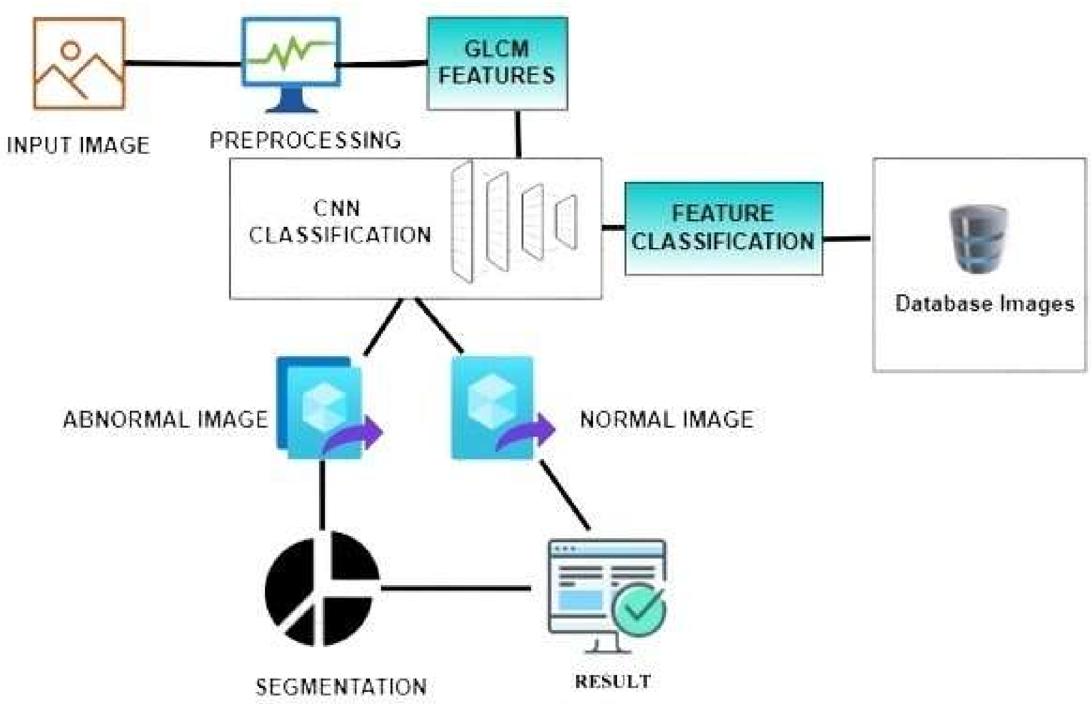


Fig.5.1 System Architecture

The system get an input image from the user. After getting the image the system pre-process the input image using medium filter, then GLCM features are added to the input image and now the image will be converted into grey scale image. Now the database image is taken for feature extraction and it is given to the CNN classification. In Fig 5.1, we have discussed the outline of project architecture.

In the CNN classification, we use google net (image processing) to train the system and will classify whether the image is normal or tuberculosis image. If the system confirms a normal image then it gives the result as normal or if the system confirms that tuberculosis is present in that image then segmented image is show in a new window and the result of that image (tuberculosis is present) is displayed.

5.2 MODULE DESCRIPTION

5.2.1 Pre-Processing

The aim of pre-processing is associate improvement of the image information that reduces unwanted distortions or enhances some image options vital for more process. Fig 5.2 shows the conversion of original image to pre-processed image. During this we have a tendency to square measure activity 3 processes they are:

- Resize
- Conversion
- Filtering

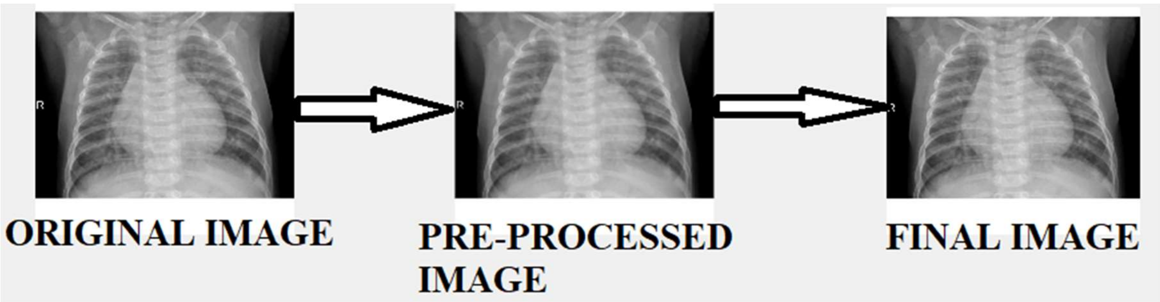


Fig.5.2 Pre-processed Image

5.2.2 Pre-Trained CNN model as Feature Extractor

Google net pre-trained CNN model is selected as the feature extractor in this proposed system. Google net has been trained on more than a million images from the ImageNet database and it can classify images into 1000 object categories with about 60 million parameters. The architecture of Google net consists of eight learned layers, five convolution layers followed by three fully connected layers. In MATLAB platform, Google net architecture consists of 26 layers: The first 23 layers are for feature extraction, whereas the last three layers are for classifying these features into 1000 classes. Google net has an image input size of 227x227x3 images with 'zerocenter' normalization. Then, the first convolution layer filters the 227x227x3 image input image with 96 kernels of size 11×11×3 with a stride of 4 pixels and followed by ReLU non-linear activation, cross channel normalization with five channel per element, and 3x3 max pooling with the stride of 2 and zero padding layer. For the second convolution layer, 256 feature kernels with a size of 5x5x48 filters 27x27x96 feature image and carry out further feature extraction. Same as the first convolution layer, second convolution layer also followed by ReLU nonlinearity activation, cross channel normalization with 5 channels per element, and a 3x3 max pooling with the stride of 2 and zero padding and output of 13x13x256 image. In Fig 5.3 and Table 5.1 we have discussed the architecture of google net.

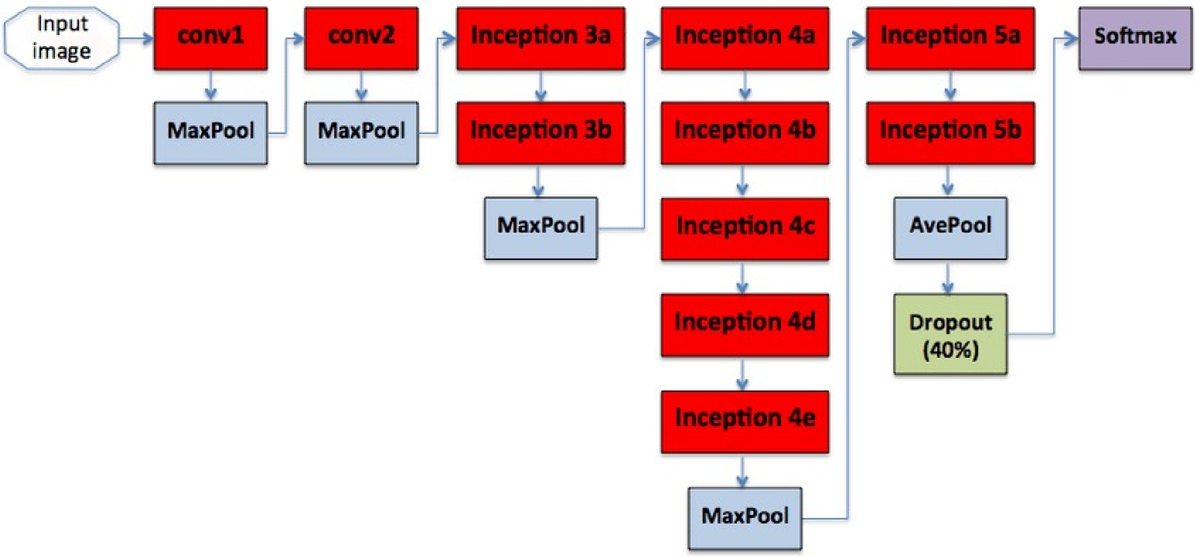


Fig.5.3 GoogleNet Architecture Classification

googlenet		ker num	size	pad	step	parameters count conv	bias	all	sub-sum		
	data (channels)	3									
	conv1/7x7_s2	64	7	3	2	9408	64	9472			
	conv1/relu_7x7										
	pool1/3x3_s2		3	0	4						
	pool1/norm1								9408	64	9472
									9.1875	0.0625	9.25
	conv2/3x3_reduce	64				4096	64	4160			
	conv2/relu_3x3_reduce										
	conv2/3x3	192	3	1	1	110592	192	110784			
	conv2/relu_3x3										
	conv2/norm2										
	pool2/3x3_s2		3	0	2				114688	256	114944
									112	0.25	112.25
	inception_3a/1x1	64				12288	64	12352			
	inception_3a/relu_1x1										
	inception_3a/3x3_reduce	96				18432	96	18528			
	inception_3a/relu_3x3_reduce										
	inception_3a/3x3	128	3	1	1	110592	128	110720			
	inception_3a/relu_3x3										
	inception_3a/5x5_reduce	16				3072	16	3088			
	inception_3a/relu_5x5_reduce										
	inception_3a/5x5	32	5	2	1	12800	32	12832			
	inception_3a/relu_5x5										
	inception_3a/pool		3	1	1						
	inception_3a/pool_proj	32				6144	32	6176			
	inception_3a/relu_pool_proj										
	inception_3a/output (Concat)	256							163328	368	163696
									159.5	0.3594	159.86
	inception_3b/1x1	128				32768	128	32896			
	inception_3b/relu_1x1										
	inception_3b/3x3_reduce	128				32768	128	32896			
	inception_3b/relu_3x3_reduce										
	inception_3b/3x3	192	3	1	1	221184	192	221376			
	inception_3b/relu_3x3										
	inception_3b/5x5_reduce	32				8192	32	8224			
	inception_3b/relu_5x5_reduce										
	inception_3b/5x5	96	5	2	1	76800	96	76896			
	inception_3b/relu_5x5										
	inception_3b/pool		3	1	1						
	inception_3b/pool_proj	64				16384	64	16448			
	inception_3b/relu_pool_proj										
	inception_3b/output (Concat)	640									
	pool3/3x3_s2		3	0	2				388096	640	388736
									379	0.625	379.63

Table 5.1 Details of GoogleNet Layers in MATLAB Platform

5.2.3 Conventional Neural Networks

A CNN contains a sequence of convolutional and max-pooling layers, activation layer and each layer has connected with its previous layer. This was classified by several fully connected layers in the next step [8]. All adjustable parameters are optimized by minimizing the misclassification by reducing the error over the training set. Each convolutional layer performs a 2D convolution with a filter of different size 3 x 3, 5 x 5, 7 x 7. The subsequent activations of the output maps are given by the total of the past convolutional responses which are gone through a nonlinear activation function. Max pooling layer was performing the dimensionality reduction. The output of a thin-layer was given by the most extreme activation over non-covering rectangular areas. Max-pooling makes location invariance and down-samples the image along every direction over a bigger neighbourhood. Filter size of convolutional and max-pooling layers are selected in such a way that a fully connected layer can combine the output into a one-dimensional vector. The last layer always be a fully connected layer which contains one output unit for all classes. Here rectification linear unit was used as the activation function. Furthermore, it was deciphered as the likelihood of a specific input image having a place with that class. Adam optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.

5.2.4 Image Classification based on CNN

Image Classification was the process of finding instances of real-world objects such as faces, buildings, and bicycles in images or videos. Thus, the working process of bone detection and classification based on CNN is shown in Figure 5.4. Image classification algorithms typically use extracted features and learning algorithms to recognize instances of an object category. It was commonly used in applications such as image retrieval, security and advanced driver assistance systems.

The CNN used in this project was implemented in MATLAB with image processing toolbar with transfer learning. All the images in the dataset were resized to 227x227 with 3 Channel before feeding as input to the network.

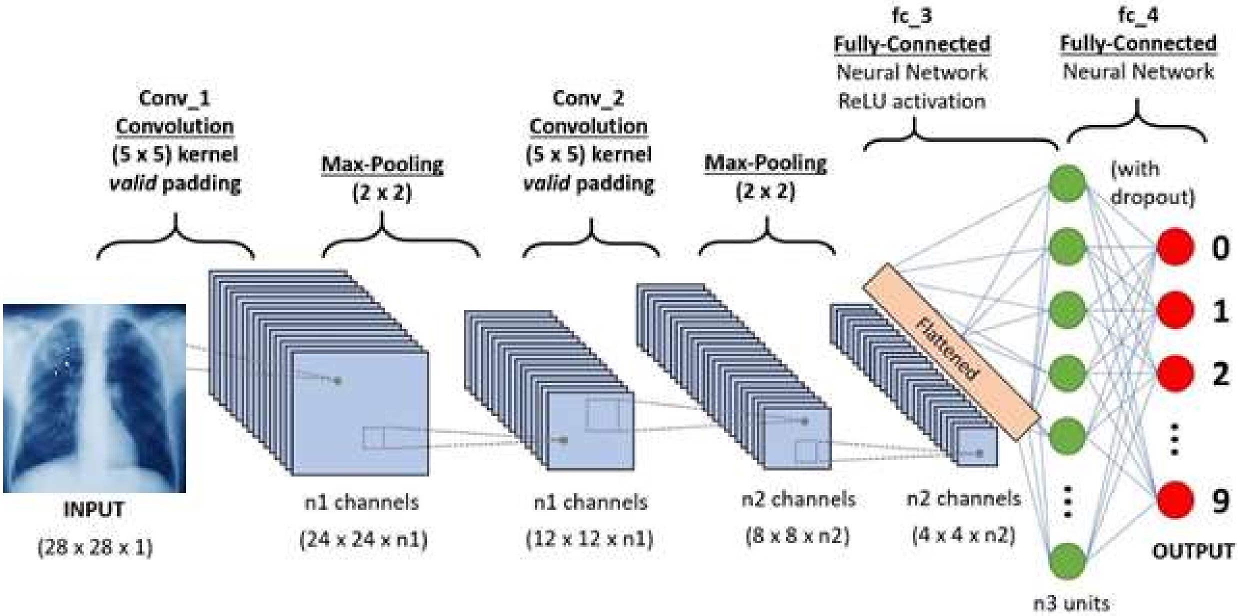


Fig.5.4 CNN Classification

5.2.5 Transfer Learning

Transfer learning is often employed in deep learning applications. You'll be able to take a pre trained network and use it as a start line to find out a brand new task. Fine-tuning a network with transfer learning is sometimes abundant quicker and easier than coaching a network with haphazardly initialized weights from scratch. During this project quickly transfer learned options to a brand new task employing a smaller variety of coaching pictures. Fig 5.5 and Fig 5.6 shows the loading of google web design.

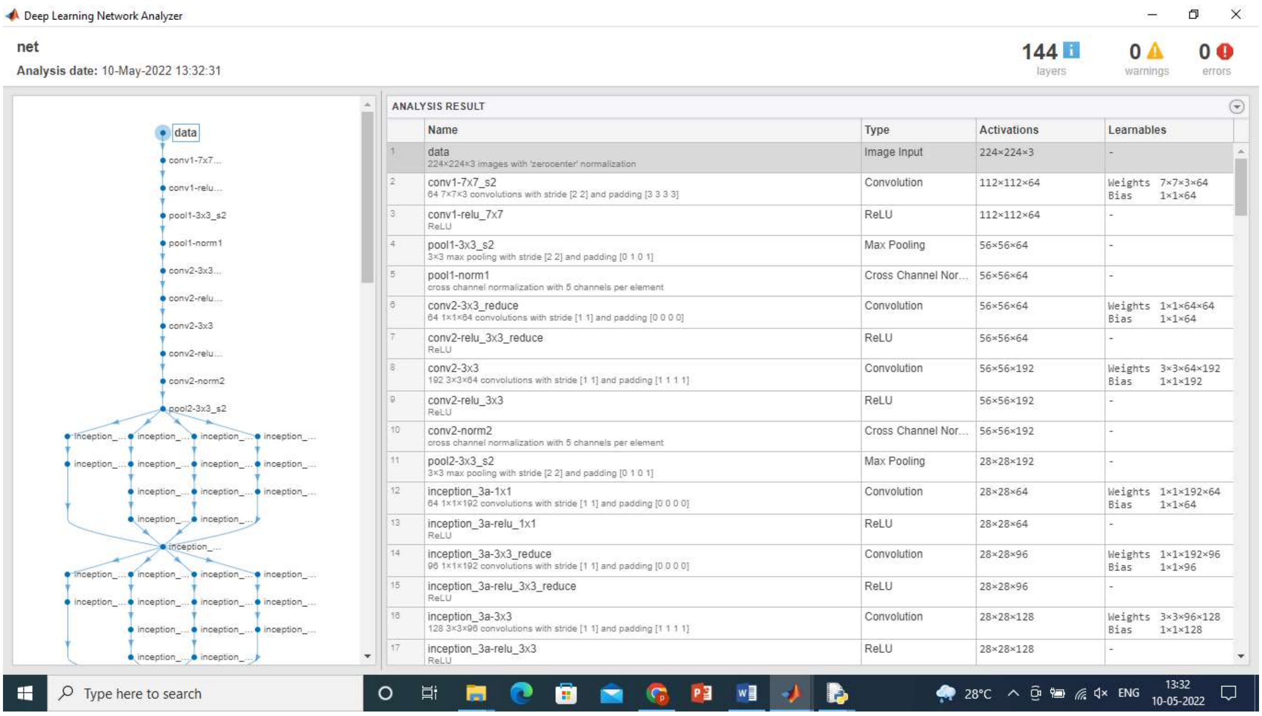


Fig.5.5 Loading of GoogleNet Architecture

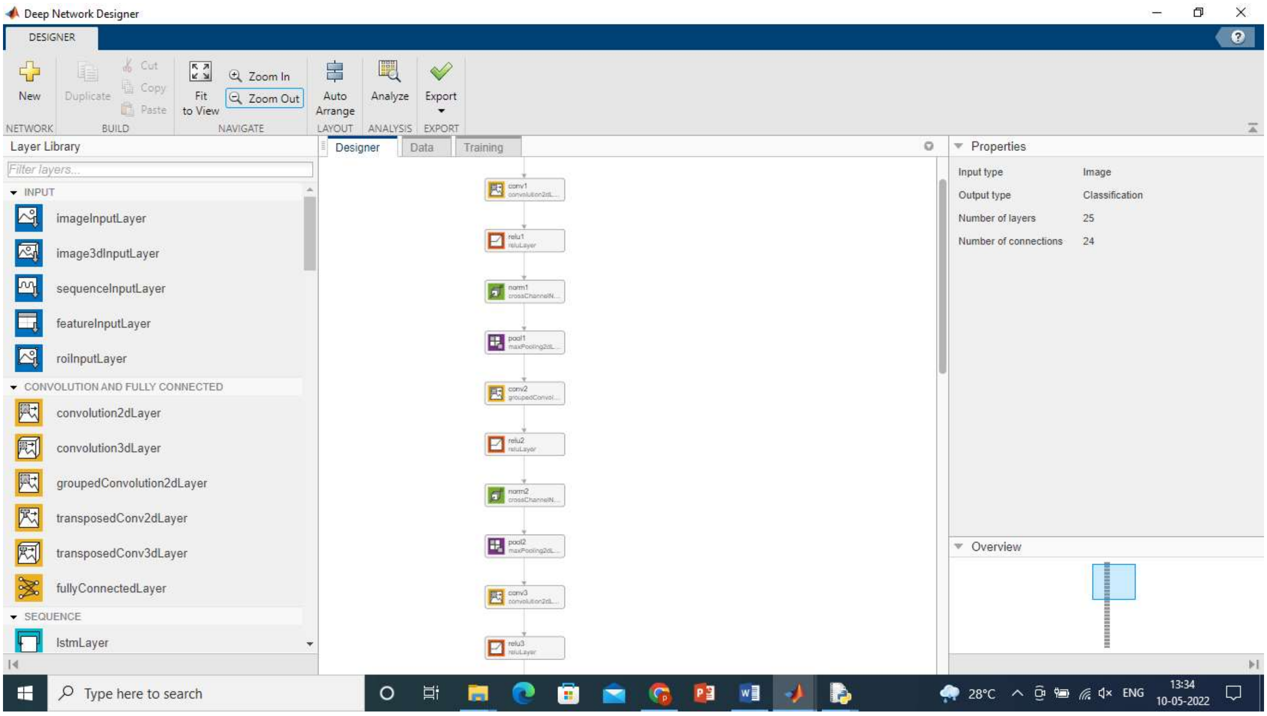


Fig.5.6 Network Analyse of GoogleNet Model

5.2.6 Training and Testing the Networks

When the network has been structured for classification application with all parameters, then it was ready for training. After each iteration, the network converges by reducing the error rate. The loop was terminating when it reached a minimum error rate. A learning rate was maintained for each network weight (parameter) and separately adapted as learning unfolds. The network weight was adjusted subsequently in each iteration from initial value based on result until it converges to a value. Weight will decide the convergence. The weight value for each image is recorded in a neural network after database loaded. Here learning rate is 0.0001. Those defined weights was further used to classify more number of datasets. In Fig 5.7 we have discussed the training of TB on CNN.

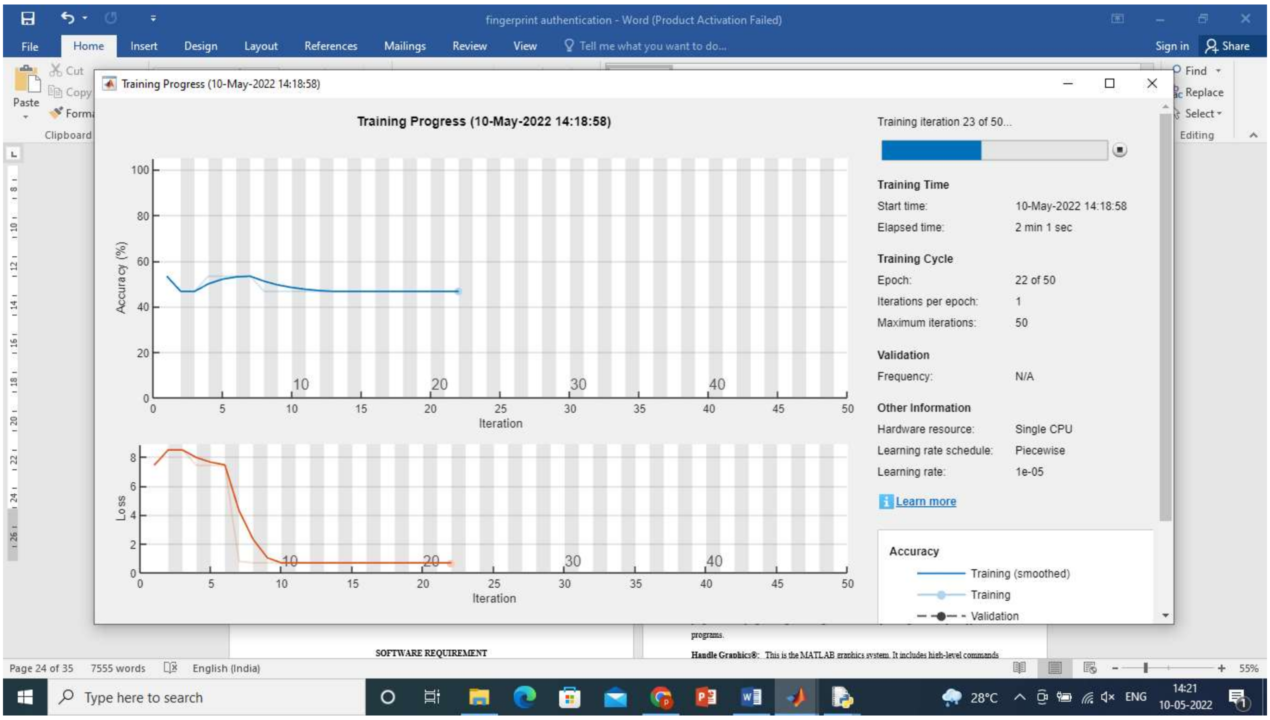


Fig.5.7 Training of Tuberculosis on CNN

The pre-trained weight which was obtained from the training phase also used in the testing phase. The input image was allowed to pass through all layers of the neural network and parameters were obtained [4]. These values were cross-checked with the pre-trained weight and identify the one which gives maximum matching with the classes' presents in the dataset. The system was considering the label to which it is closely matched.

5.2.7 Segmentation

Segmentation is to modify and alter the illustration of an image into something that is more meaningful and easier to analyse. To extract the features of the tissues to satisfy the segmented parts, first the non-specific parts are removed that are not parts of the pneumonia region. After that the filtering process is applied to eliminate the noise from the x-ray images. Fig 5.8 shows the segmented image of chest X-ray Image.

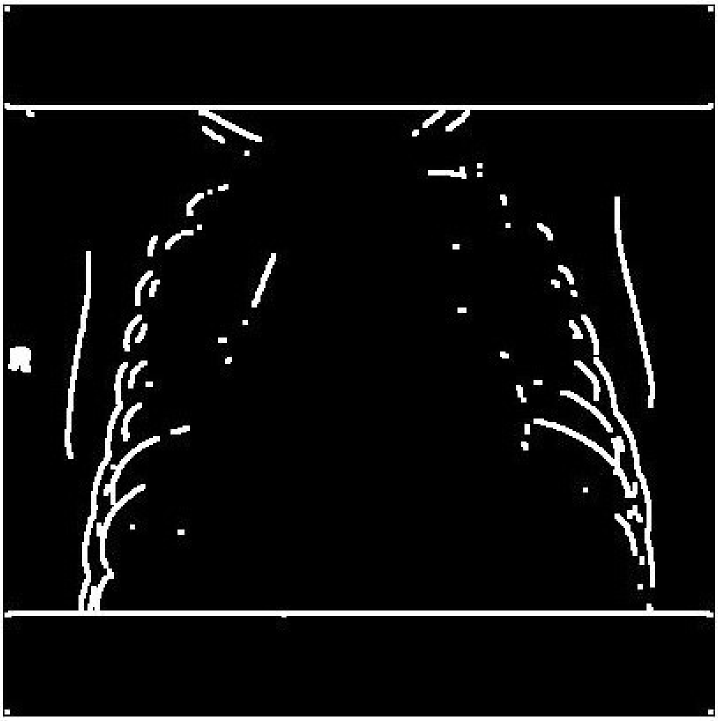


Fig.5.8 Segmented Image

5.3 ALGORITHM DESCRIPTION

BEGIN

IF LOAD DATASET Successful
Show Data Set Loaded;
ELSE
Reload Data Set;
IF LOAD MODEL Successful
Show Model in Separate Window;
ELSE
Model Not Loaded;
IF TRAINING Successful
Load Training Graph;
ELSE
Train Once More;
IF INPUT IMAGE Successful
Input Image Is Shown;
ELSE
Load Input Image;
IF PRE-PROCESS Successful

```
        Show Pre-Processed Image;

    ELSE
        Pre-Process Input Image;
    IF CLASSIFICATION Successful
        Show Image;
        IF NORMAL
            Display Normal Image;
        ELSE
            Display TB Image;
    END IF

END
```

CHAPTER - 6

SYSTEM IMPLICATIONS

6. SYSTEM IMPLICATIONS

6.1 CLIENT-SIDE CODING

gui.m

```
function varargout = gui(varargin)
% GUI MATLAB code for gui.fig
%      GUI, by itself, creates a new GUI or raises the existing
%      singleton*.
%
%      H = GUI returns the handle to a new GUI or the handle to
%      the existing singleton*.
%
%      GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI.M with the given input
arguments.
%
%      GUI('Property','Value',...) creates a new GUI or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before gui_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to gui_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
 % Edit the above text to modify the response to help gui
```

```
% Last Modified by GUIDE v2.5 28-Apr-2022 12:51:29
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
               'gui_Singleton',  gui_Singleton, ...
               'gui_OpeningFcn', @gui_OpeningFcn, ...
               'gui_OutputFcn',  @gui_OutputFcn, ...
               'gui_LayoutFcn',  [] , ...
               'gui_Callback',   []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before gui is made visible.
function gui_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to gui (see VARARGIN)
% Choose default command line output for gui
handles.output = hObject;
```

```
 a=zeros([256 256]);
axes(handles.axes1);
imshow(a);
axes(handles.axes2);
imshow(a);
axes(handles.axes3);
imshow(a);
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes gui wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = gui_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global train
matlabpath ='C:\Users\karth\Desktop\tuber matlab\'
data = fullfile(matlabpath,'dataset')
train = imageDatastore(data,
'IncludeSubfolders',true,'LabelSource','foldernames');
count = train.countEachLabel;
```

```
msgbox('Dataset Loaded Successfully')
% Update handles structure
guidata(hObject, handles);
% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global layers
disp('Pre-Trained Model Loaded...')
net = googlenet;
deepNetworkDesigner(net)
layers = [ imageInputLayer([400 400 3])
net(2:end-3)
fullyConnectedLayer(2)
softmaxLayer
classificationLayer()
]
msgbox('Pre-Trained Model Loaded Successfully')
% Update handles structure
guidata(hObject, handles);
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%training
global opt training train layers
opt = trainingOptions('adam', ...
```

```
    'InitialLearnRate', 0.001, ...
    'LearnRateSchedule', 'piecewise', ...
    'LearnRateDropFactor', 0.1, ...
    'LearnRateDropPeriod', 8, ...
    'L2Regularization', 0.004, ...
    'MaxEpochs', 10, ...
    'MiniBatchSize', 100, ...
    'Verbose', true, 'Plots','training-progress');
training = trainNetwork(train,layers,opt);
msgbox('Trained Completed')
% Update handles structure
guidata(hObject, handles);
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double
% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global inp
cd input
[file path] = uigetfile('*.bmp;*.jpg;*.jpeg','Pick an Image File');

 if isequal(file,0)
      warndlg('File not selected');
else
inp = imread(file);
cd ..
axes(handles.axes1);
imshow(inp);
img=inp;
if size(inp,3)>1
   Freg =im2gray(inp);
end
handles.img=img;
end
% Update handles structure
guidata(hObject, handles);
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
global inp J
img = im2gray(inp);
J = medfilt2(img);
axes(handles.axes2);
title('Filtered Image');
imshow(J);
% Update handles structure
guidata(hObject, handles);
% --- Executes on button press in pushbutton4.
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global training inp out
out = classify(training,inp);
axes(handles.axes3);
imshow(inp);
title(string(out));
% Update handles structure
guidata(hObject, handles);

if (string(out)=="Normal")
  msgbox(string(out));
else
  msgbox(string(out));
  [segout] = seg(inp);
  boundary = bwboundaries(im2bw(segout));
```

```
end
% out = classify(training,inp);
% yval = inp;
% accuracy = mean()
```

6.2 SERVER-SIDE CODING

Seg.m

```
function [inp,J] = seg(inp, J)
%SEG Summary of this function goes here
%   Detailed explanation goes here
% This is a program for extracting objects from an image. Written for
vehicle number plate segmentation and extraction
% Authors : Jeny Rajan, Chandrashekar P S
% U can use attached test image for testing
% input - give the image file name as input. eg :- car3.jpg

%k=input('Enter the file name','s'); % input image; color image
%im=imread(inp);
im1=rgb2gray(inp);
im1=medfilt2(im1,[3 3]); %Median filtering the image to remove noise%
BW = edge(im1,'sobel'); %finding edges
[imx,imy]=size(BW);
msk=[0 0 0 0 0;
    0 1 1 1 0;
    0 1 1 1 0;
    0 1 1 1 0;
    0 0 0 0 0;];
```

```
B=conv2(double(BW),double(msk)); %Smoothing  image to reduce the
number of connected components
L = bwlabel(B,8);% Calculating connected components
mx=max(max(L))
% There will be mx connected components.Here U can give a value between
1 and mx for L or in a loop you can extract all connected components
% If you are using the attached car image, by giving 17,18,19,22,27,28 to L
you can extract the number plate completely.
[r,c] = find(L==17);
rc = [r c];
[sx sy]=size(rc);
n1=zeros(imx,imy);
for i=1:sx
   x1=rc(i,1);
   y1=rc(i,2);
   n1(x1,y1)=255;
end % Storing the extracted image in an array
figure,imshow(inp);
figure,imshow(B);
%figure,imshow(n1,[]);
end
```

CHAPTER - 7
SYSTEM TESTING

7. TESTING

7.1 TESTING OBJECTIVES

The purpose of testing is to get errors. Testing is that the method of making an attempt to get each conceivable fault or weakness during a work product. It provides the way to envision the practicality of parts, sub-assemblies, assemblies and/or a finished product. it's the method of travail software package with the intent of guaranteeing that the package meets its necessities associate degreed user expectations and doesn't fail in an unacceptable manner. There ar varied sorts of check. every check sort addresses a selected testing demand.

7.2 TYPES OF TESTS

7.2.1 Unit Testing

Unit checking involves the look of test cases that validate that the inner program logic is functioning properly, which program inputs manufacture valid outputs. All call branches and internal code flow ought to be valid. it's the testing of individual software package units of the appliance. it's done when the completion of a private unit before integration. Unit checks perform basic tests at element level and test a selected business method, application, and/or system configuration.

7.2.2 Integration Testing

Integration tests are designed to check integrated software package parts to work out if they really run jointly program. Testing is event driven and is a lot of involved with the essential outcome of screens or fields. Integration tests demonstrate that though the parts were separately satisfaction, as shown by with success unit testing, the mixture of parts is correct and consistent.

7.2.3 Functional Testing

Functional testing give systematic demonstrations that functions tested ar on the market as such as by the business and technical necessities, system documentation, and user manuals.

Functional testing is centred on the subsequent items:

Valid Input: known categories of valid input should be accepted.

Invalid Input: known categories of invalid input should be rejected.

Functions: known functions should be exercised.

Output: known categories of application outputs should be exercised.

7.2.4 System Testing

System testing ensures that the complete integrated package meets necessities. It tests a configuration to confirm famed and sure results. Associate degree example of system checking is that the configuration orientating system integration test. System testing relies on method descriptions and flows, action predriven method links and integration points.

7.2.5 Acceptance Testing

User Acceptance Testing may be a vital section of any project and needs important participation by the tip user. It conjointly ensures that the system meets the useful demand. In table 7.1, we've got mentioned the check cases and attainable results.

7.3 TESTCASES AND RESUTS

S No	Test Cases	Expected Output	Actual Output	Status
1.	Loading dataset for Training	The dataset should be trained	Dataset is trained	PASS
2.	Loading the CNN model	The model should be loaded for training the dataset	The model has been loaded successfully	PASS
3.	Training the model	The model should be trained as per the conditions	The model has been trained	PASS
4.	Inserting the input image	The input image should be successfully uploaded for further process	The input image has been uploaded	PASS
5.	Pre-processing the input image	Pre-process the image to reduce distortion and enhance the image for further processing	Displays the pre-processed image	PASS
6.	Classification - Normal	It should display the message as “Normal”	Displays the message as “Normal”	PASS
7.	Classification - Tuberculosis	It should display the message as “Tuberculosis” and display the Segmented image	Displays the message as “Tuberculosis” and segmented image	PASS

Table 7.1 Test Cases and Possible Results

CHAPTER - 8
CONCLUSION

8. CONCLUSION

8.1 RESULTS & DISCUSSION

In this project presents experimental results and discuss the suitability of the best performing representation and model over the others. The architecture of trained model is based on the tuberculosis classification of CNN with two samples of tuberculosis and also used on chest images. After the 40 epochs our results contains 100% accuracy on figure A.5. In this figure A.2 and A.3 sample image of tuberculosis and normal classification from the Google net model. In the figure A.5 contains classification using the pre trained model. In figure A.2 and A.3 represents the output of one of the sample images.

In this testing of normal and tuberculosis sample images for the classification on pre trained model with the prediction accuracy value of normal is 1 with no loss value and prediction accuracy value of normal and tuberculosis is 0.8 with 0.2 was prediction loss.

8.2 CONCLUSION AND FUTURE ENHANCEMENT

Conclusion - In this project, transfer learning method used for X-ray tuberculosis classification on normal and abnormal detection. X ray dataset was taken from the clinical diagnostic for normal and abnormal tuberculosis. Different methodologies proposed by various researchers are considered, all of which show that image processing had a major role in chest detection, but no one touches tuberculosis classification. From the performance criteria such as accuracy, loss and this method had been recommended to increase the prognosis. Real-time application based categorization was one of the main factors in the selection of the technique. Diagnosing tuberculosis abnormalities was a complex and sensitive task to preciseness, reliability. Experiments shows the effectiveness of data augmentation, especially in the case of insufficient training data.

Future Enhancement - There are opportunities for further improvement for this project from both technical and clinical point of view. For instance, on the technical side, adding segmentation constrain to the method when it goes to abnormal condition. Also extend work for various network model for providing optimum results. Furthermore, training and testing on rigidly aligned images might provide more accurate localization. In clinical application, this proposed method will help the patients can easily understand tuberculosis with module of hardware.

APPENDICES

APPENDICES

A.1 SAMPLE SCREENSHOTS

Fig.A.1 Screenshot for Displaying Pre-Processed Image

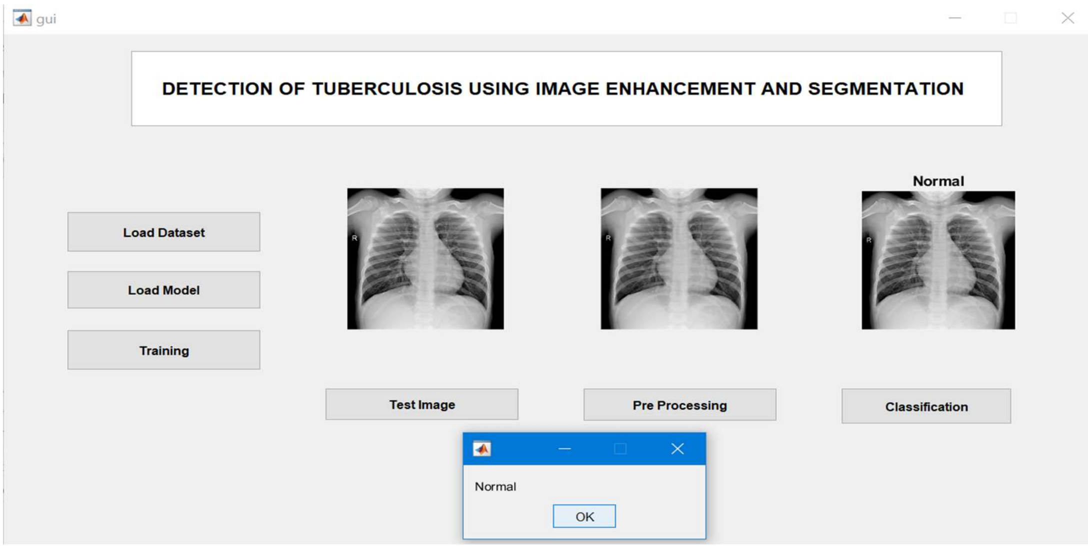


Fig.A.2 Screenshot for Displaying Normal Image

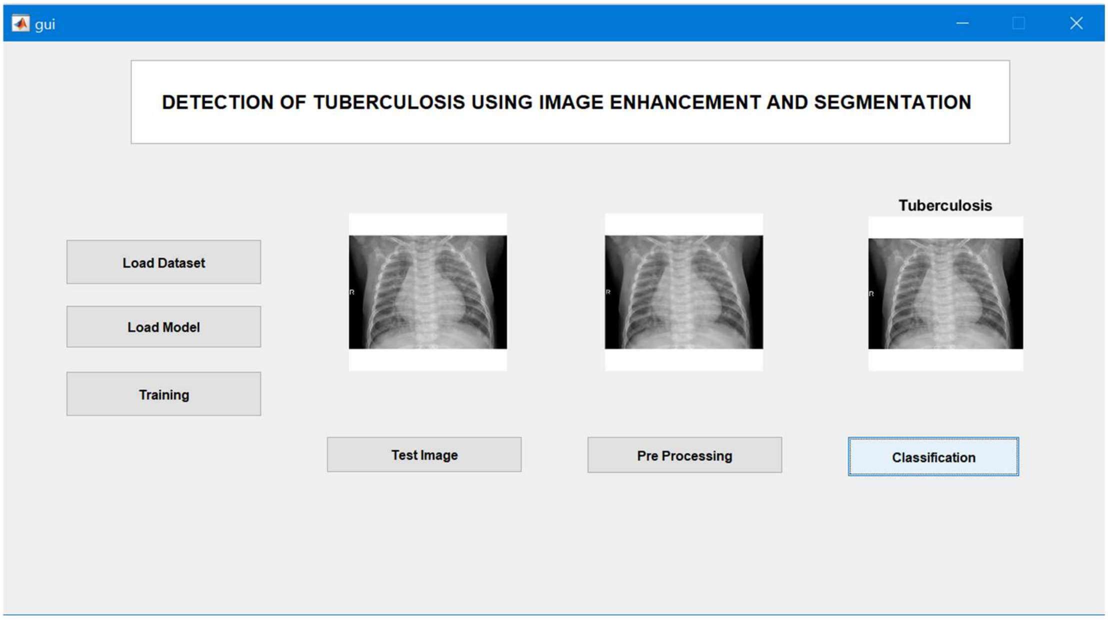


Fig.A.3 Screenshot for Displaying Abnormal Image

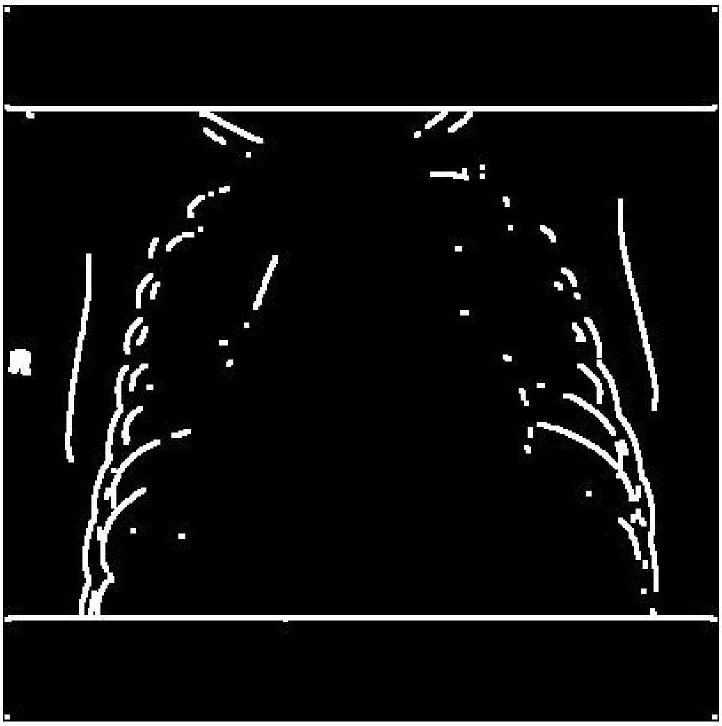


Fig.A.4 Screenshot for Displaying Segmented Image

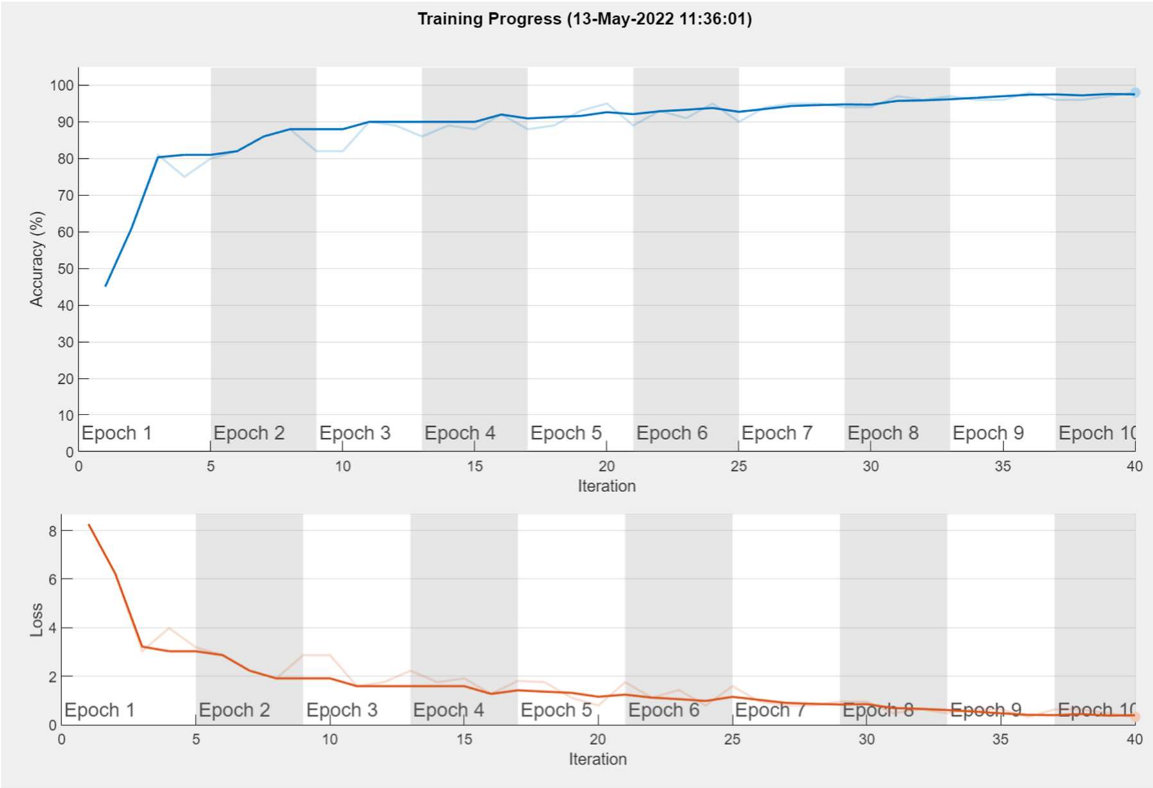


Fig.A.5 Screenshot for Displaying Accuracy Rate and Loss Rate

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] ISID," Is global TB elimination still a feasible goal by 2035? And can Oman stand the test as a pathfinder for TB elimination?",2021

[2] WHO," The top 10 causes of death",2020

[3] Ayaz M, Shaukat F, Raja G. Ensemble learning based automatic detection of tuberculosis in chest X-ray images using hybrid feature descriptors. Phys Eng Sci Med. 2021;44(1):183-194. doi:10.1007/s13246-020-00966-0

[4] Rahman, Tawsifur & Khandakar, Amith & Kadir, Muhammad & Islam, Khandakar & Islam, Khandaker & Mazhar, Rashid & Hamid, Tahir & Islam, Mohammad & Mahbub, Zaid & Ayari, Mohamed & Chowdhury, Muhammad. (2020). Reliable Tuberculosis Detection using Chest X-ray with Deep Learning, Segmentation and Visualization.

[5] M. B. Mizan, M. A. M. Hasan and S. R. Hassan, "A Comparative Study of Tuberculosis Detection Using Deep Convolutional Neural Network," 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT), 2020, pp. 157-161, doi: 10.1109/ICAICT51780.2020.9333464.

[6] Ms.SweetyBakyarani. E; Dr. H. Srimathi; Dr. P.J. Arul Leena Rose. "A Comparative Study On Performance Of Pre-Trained Convolutional Neural

Networks In Tuberculosis Detection". European Journal of Molecular & Clinical Medicine, 7, 3, 2020, 4852-4858.

[7] Kieu STH, Bade A, Hijazi MHA, Kolivand H. A Survey of Deep Learning for Lung Disease Detection on Medical Images: State-of-the-Art, Taxonomy, Issues and Future Directions. Journal of Imaging. 2020; 6(12):131. https://doi.org/10.3390/jimaging6120131

[8] Tawsifur Rahman, Amith Khandakar, Muhammad A. Kadir, Khandaker R. Islam, Khandaker F. Islam, Zaid B. Mahbub, Mohamed Arselene Ayari, Muhammad E. H. Chowdhury. (2020) "Reliable Tuberculosis Detection using Chest X-ray with Deep Learning, Segmentation and Visualization". IEEE Access, Vol. 8, pp 191586 - 191601. DOI. 10.1109/ACCESS.2020.3031384.

[9] Kandel I, Castelli M. How Deeply to Fine-Tune a Convolutional Neural Network: A Case Study Using a Histopathology Dataset. Applied Sciences. 2020; 10(10):3359. https://doi.org/10.3390/app10103359

[10] Giancarlo Zaccone, Md. Rezaul Karim, "Deep Learning with TensorFlow - Second Edition", Packt, March 2018.

[11] Zheng, Yufeng & Yang, Clifford & Merkulov, Aleksey. (2018). Breast cancer screening using convolutional neural network and follow-up digital mammography. 4. 10.1117/12.2304564.

[12] Mahmood, Ammar & Giraldo, Ana & Bennamoun, Mohammed & An, Senjian & Sohel, Ferdous & Boussaid, Farid & Hovey, Renae & Fisher, Robert & Kendrick, Gary. (2020). Automatic Hierarchical Classification of Kelps Using Deep Residual Features. Sensors. 20. 447. 10.3390/s20020447. R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev, in press.

[13] Roslidar Roslidar, Mohd Syaryadhi, Khairun Saddami, Biswajeet Pradhan, Fitri Arnia, Maimun Syukri, Khairul Munadi. BreaCNet: A high-accuracy breast thermogram classifier based on mobile convolutional neural network[J]. Mathematical Biosciences and Engineering, 2022, 19(2): 1304-1331. doi: 10.3934/mbe.2022060

[14] Jun Gao, Qian Jiang, Bo Zhou, Daozheng Chen. Convolutional neural networks for computer-aided detection or diagnosis in medical image analysis: An overview[J]. Mathematical Biosciences and Engineering, 2019, 16(6): 6536-6561. doi: 10.3934/mbe.2019326

[15] Segmentation and classification on chest radiography: a systematic survey Tarun Agrawal and Prakash Choudhary.

[16] EfficientNet-Based Convolutional Neural Networks for Tuberculosis Classification Vinayakumar Ravi, Harini Narasimhan& Tuan D. Pham.

[17] Classification of Lungs Diseases Using Machine Learning Technique Meet Diwan1, Bhargav Patel and Jaykumar Shah.

[18] An Optimal Way for Tuberculosis Detection Ajmal shan C. K. and Binoy D. L.

[19] Tuberculosis diagnosis using Deep LearningLokeshwaran V B, Monish Kumar R and Lakshman Raaj S.

[20] CXR Tuberculosis Detection Using MATLAB Image Processing Mr. P. A. Kamble, Mr. V. V. Anagire and Mr. S. N. Chamtagoudar.