

TUBERCULOSIS DETECTION USING MACHINE LEARNING

A MINI-PROJECT REPORT

Submitted by

SANJAY A

(412620104033)

GOWTHAM M

(412620104301)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



**SRI VENKATESWARAA COLLEGE OF TECHNOLOGY,
ORAGADAM INDUSTRIAL CORRIDOR, VADAKKAL VILLAGE,**

TAMIL NADU - 602105

ANNA UNIVERSITY :: CHENNAI 600 025

MAY 2023

BONAFIDE CERTIFICATE

Certified that the mini-project work entitled “**TUBERCULOSIS DETECTION
USING MACHINE LEARNING**” is a bonafide work carried out by

SANJAY A

(412620104033)

GOWTHAM M

(412620104301)

The report has been approved as it satisfies the academic requirements in respect of mini-project work prescribed for the course.

.....
Mrs. BRINDHA DEVI R
Internal Guide

.....
Mr. RATHINA SABAAPATHI P
Head of the Department

Submitted for the Anna University Project Viva-Voce Examination held on

.....

Internal Examiner

External Examiner

DECLARATION

I **Sanjay A** (Reg.No.412620104033) hereby declare that the Project Report entitled “**TUBER CULOSIS DETECTION USING MACHINE LEARNING**” done by me under the guidance of **Mrs. Brindha Devi R ,** at **Sri Venkateswaraa College of Technology** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science And Engineering.

Date :

Place :

.....

Signature of the Candidate

ACKNOWLEDGEMENT

We wish to express our sincere thanks to our beloved chairman **Mr.B.Haribabu** for providingimmense facilities in our institution.

We proudly render our thanks to our Principal **Dr.N.Janaki Manohar**, for the facilities and the encouragement given by him to the progress and completion of our project.

We Proudly render our immense gratitude and sincere thanks to our Head of the Departmentof Computer Science And Engineering **Mr.Rathina Sabaapathi P** for effective leadership, encouragement and guidance in the project.

We highly indebted to provide our heart full thanks to our project coordinator and supervisor **Mrs.Brindha Devi R** for her variable ideas and suggestion during execution of our project workand for continuous encouragement in conveying us for making many constructive comments for the successful completion of our project.

We wish to acknowledge the help received from various department and various individuals during the preparation and editing stages of the manuscript.

ABSTRACT

THE PROPOSED DATA ANALYSIS FRAMEWORK FOR IDENTIFYING A RELIABLE GENE SIGNATURE FOR TB DIAGNOSIS USING DEEP LEARNING AND IMAGE AUGMENTATION TECHNIQUES APPEARS TO BE A PROMISING APPROACH. TB IS A MAJOR GLOBAL HEALTH CHALLENGE, AND TIMELY DIAGNOSIS IS CRITICAL FOR EFFECTIVE TREATMENT AND PREVENTION OF TRANSMISSION. WHILE PREVIOUS ATTEMPTS TO IDENTIFY A TRANSCRIPTIONAL SIGNATURE FOR TB DIAGNOSIS HAVE FACED CHALLENGES DUE TO POPULATION DIVERSITY, THIS APPROACH AIMS TO DIRECTLY INCORPORATE VARIOUS PUBLICLY AVAILABLE GENE EXPRESSION ARRAY DATASETS TO IMPROVE ACCURACY. IN ADDITION TO USING DEEP LEARNING FOR GENE SIGNATURE IDENTIFICATION, THE STUDY ALSO EXPLORES THE USE OF CONVOLUTIONAL NEURAL NETWORKS (CNNs) FOR TB DETECTION FROM CHEST RADIOGRAPHY. THIS IS AN EXCITING DEVELOPMENT, AS DEEP LEARNING HAS SHOWN PROMISE IN MEDICAL PICTURE ANALYSIS, AND COULD PROVIDE A MORE EFFICIENT AND COST-EFFECTIVE WAY TO DIAGNOSE TB IN AREAS WHERE RESOURCES AND TRAINED MEDICAL PROFESSIONALS ARE LIMITED.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PG NO
	ABSTRACT	
1.	INTRODUCTION	1
	1.1 OVERVIEW	1
	1.2 OBJECTIVE	4
	1.3 LITERATURE SURVEY	5
2.	SYSTEM ANALYSIS	9
	2.1 EXISTING SYSTEM	9
	2.1.1 DISADVANTAGES	9
	2.2 PROPOSED SYSTEM	10
	2.2.1 ADVANTAGES	12
3.	SYSTEM REQUIREMENTS	13
	3.1 INTRODUCTION	13
	3.2 HARDWARE REQUIREMENTS	13
	3.3 SOFTWARE REQUIREMENTS	15
4.	SYSTEM DESIGN	17
	4.1 SYSTEM ARCHITECTURE	17
	4.2 DATA FLOW DIAGRAM	19
	4.3 CLASS DIAGRAM	20
	4.4 SEQUENCE DIAGRAM	21
	4.5 ACTIVITY DIAGRAM	22

5.	SYSTEM IMPLEMENTATION	23
	5.1 LIST OF MODULES	23
	5.2 MODULE DESCRIPTIONS	23
6.	TESTING	26
	6.1 FUNCTIONAL TESTING	26
	6.1.1 INPUT	26
	6.1.2 TEST RESULT	27
	6.2 SYSTEM TESTING	27
7.	RESULTS AND DISCUSSION	28
	7.1 RESULTS	28
	7.2 DISCUSSION	29
8.	CONCLUSION AND FUTURE ENHANCEMENT	30
	8.1 CONCLUSION	30
	8.2 FUTURE ENHANCEMENT	30
	ANNEXURE	
	APPENDIX I	31
	REFERENCES	39

LIST OF FIGURES

4.1 Tuberculosis Detection Architecture	17
4.2 Data Flow Diagram.	18
4.3 Class Diagram	19
4.4 Sequence Diagram	20
4.5 Activity Diagram	21
5.1 Input Design	23
5.2 Output Design	23

LIST OF ABBREVIATIONS

CXR	Chest X-ray
CNN	Convolutional Neural Networks
HEF	High-Frequency Emphasis Filtering
SDDs	Software Design Descriptions
SRS	System Requirements Specifications
STD	Software Test Documentation
SVVP	Software Verification and Validation
SIFT	Scale Invariant Feature Transform
TB	Tuberculosis
UM	Unsharp Masking

CHAPTER – 1

INTRODUCTION

1.1 OVER VIEW

Tuberculosis (TB) is a major public health concern and a leading cause of death worldwide. Early detection and treatment are critical to controlling the spread of the disease and improving patient outcomes. Machine learning (ML) and convolutional neural networks (CNNs) are promising tools for improving TB detection and diagnosis. CNNs are a type of deep learning algorithm that can analyze images and identify patterns that indicate the presence of TB. CNNs are particularly useful in analyzing chest X-rays, which are a common imaging study used in TB diagnosis. In a chest X-ray, TB can present as nodules, cavities, or infiltrates. However, detecting these patterns can be challenging, particularly for less experienced radiologists. By using a CNN algorithm, a computer can learn to analyze chest X-rays and identify patterns that may be indicative of TB. The algorithm can be trained using a large dataset of chest X-rays, some of which contain TB and some of which do not. The algorithm can then be tested using new chest X-rays to determine its accuracy in detecting TB. One advantage of using a CNN algorithm for TB detection is its ability to learn from large amounts of data. CNNs can process large datasets of chest X-rays quickly and accurately, improving the accuracy of TB diagnosis and reducing the risk of misdiagnosis. ML is a subset of artificial intelligence (AI) that involves teaching computers to recognize patterns in data and make predictions or

decisions based on those patterns. ML algorithms can be trained using large

datasets of clinical, imaging, and laboratory data to identify patterns that indicate the presence of TB. These patterns can then be used to develop algorithms that can accurately detect TB in patients. One of the main advantages of ML in TB detection is its ability to analyze large amounts of data quickly and accurately. ML algorithms can process a wide range of data, including patient demographics, clinical signs and symptoms, laboratory results, and imaging studies. By analyzing these data points in combination, ML algorithms can identify patterns that may not be immediately apparent to human clinicians. This can help to improve the accuracy of TB diagnosis and reduce the risk of misdiagnosis. ML algorithms can also be used to analyze imaging studies, such as chest X-rays or CT scans, to detect signs of TB. For example, ML algorithms can be trained to identify patterns in chest X-rays that indicate the presence of TB, such as the presence of nodules, cavities, or infiltrates. By comparing new imaging studies to a database of known TB cases, ML algorithms can accurately detect TB in patients. In addition, CNN algorithms can be customized to detect specific patterns or features associated with TB, improving their accuracy in TB detection. Another advantage of using a CNN algorithm for TB detection is its ability to improve the efficiency of TB screening and diagnosis. Chest X-rays are commonly used in TB screening, particularly in resource limited settings. However, interpreting chest X-rays can be time-consuming and require significant expertise. By using a CNN algorithm, chest X-rays can be quickly and accurately analyzed, allowing healthcare providers to prioritize patients for further testing and treatment. ML and CNN algorithms can also be used to predict the likelihood of TB in patients based on their clinical, demographic, and laboratory data. For example, ML algorithms can be used to identify

patients who are at high risk of developing TB, based on factors such as their age, gender, and underlying health conditions. This can help healthcare providers to prioritize TB screening and testing in high-risk populations, leading to earlier detection and treatment of the disease. Despite the potential benefits of using ML and CNN algorithms for TB detection, there are also challenges associated with their use. One of the main challenges is the need for high-quality data to train the algorithms. This requires large datasets that are both comprehensive and accurate, which may be difficult to obtain in some settings. In addition, CNN algorithms can be complex and difficult to interpret, which may make it challenging for healthcare providers to understand and use the results of CNN-based TB detection algorithms. One advantage of using a CNN algorithm for TB detection is its ability to learn from large amounts of data. CNNs can process large datasets of chest X-rays quickly and accurately, improving the accuracy of TB diagnosis and reducing the risk of misdiagnosis. In addition, CNN algorithms can be customized to detect specific patterns or features associated with TB, improving their accuracy in TB detection. Another advantage of using a CNN algorithm for TB detection is its ability to improve the efficiency of TB screening and diagnosis. Chest X-rays are commonly used in TB screening, particularly in resource limited settings. However, interpreting chest X-rays can be time-consuming and require significant expertise. By using a CNN algorithm, chest X-rays can be quickly and accurately analyzed, allowing healthcare providers to prioritize patients for further testing and treatment. ML and CNN algorithms can also be used to predict the likelihood of TB in patients based on their clinical, demographic, and laboratory data. For example, ML algorithms can be used to identify patients who are at high risk of developing TB, based on factors

such as their age, gender, and underlying health conditions. This can help healthcare providers to prioritize TB screening and testing in high-risk populations, leading to earlier detection and treatment of the disease.

In conclusion, ML and CNN algorithms are promising tools for improving TB detection and diagnosis. By analyzing chest X-rays, CNN algorithms can identify patterns that may be indicative of TB, improving the accuracy of TB diagnosis and reducing the risk of misdiagnosis. In addition, ML algorithms can be used to predict the likelihood of TB in patients and optimize treatment plans for better patient outcomes. However, challenges associated with data quality and algorithm complexity must be addressed to fully realize the potential of ML and CNN algorithms in TB detection and treatment.

1.2 OBJECTIVE OF THE PROJECT

The main aim of our project is to detect the tuberculosis Detection to an accurate system for diagnosing tuberculosis through the use of machine learning techniques. This involves collecting and preparing a high-quality dataset of clinical data or medical imaging, identifying relevant features, selecting and training a suitable ML algorithm or deep learning architecture, and optimizing the model's performance. The ultimate goal is to develop a reliable and efficient system that can help in the early detection and treatment of TB, leading to better patient outcomes and public health benefits.

1.3 LITERATURE REVIEW

[1] D.R Nayak et al., (2021) implemented an automated system that screens chest x-rays for detection of TB and other lung diseases in which an intensity mask, a lung model mask, and a Log Gabor mask are used for lung segmentation. The deep neural network achieved excellent performance in automatically delineating the boundaries of lung parenchyma with extensive pathological conditions on non-contrast chest CT images.

[2] E.Paul,P.Gowsalya et al., (2021) implemented chest radiographs using a combination of textural, focal, and shape abnormality analysis. It proposes a segmentation of the lung so that the subsequent analysis is done only in the delimited area and then a clavicle segmentation was made. Automated delineation of anatomical structures in chest radiographs is difficult due to superimposition of multiple structures. In this work an automated technique to segment the clavicles in posterior-anterior chest radiographs is presented in which three methods are combined.

[3] H. Shouno, and S. Kido et al., (2021) proposed CNN architectures, dataset characteristics and transfer learning, the primary intention is to automatically determine the quality of an image in accordance with a good assessment of human quality. Remarkable progress has been made in image recognition, primarily due to the availability of large-scale annotated datasets and deep convolutional neural networks (CNNs). CNNs enable learning data-driven, highly representative, hierarchical image features from sufficient training data. However, obtaining datasets as comprehensively annotated as ImageNet in the medical imaging domain remains a challenge.

[4] K. He, X. Zhang, S. Ren, and J. Sun et al., (2020) proposed a system for obtaining a set of image key points using SIFT algorithm which returns a set of key points with their attributes, in this work we use the coordinates, radius angle. Another important characteristic of these features is that the relative positions between them in the original scene shouldn't change from one image to another. For example, if only the four corners of a door were used as features, they would work regardless of the door's position; but if points in the frame were also used, the recognition would fail if the door is opened or closed. Similarly, features located in articulated or flexible objects would typically not work if any change in their internal geometry happens between two images in the set being processed.

[5] S. Faridi, M. A. Zia, Z. Javed, I. Mumtaz, and S. al., (2019) proposed the output from a convolutional layer passes through an activation function which performs some alteration on the results .A variety of attacks are regularly attempted at network infrastructure. With the increasing development of artificial intelligence algorithms, it has become effective to prevent network intrusion for more than two decades. Deep learning methods can achieve high accuracy with a low false alarm rate to detect network intrusions.

[6] M. Sarıgul, B. M Ozyildirim, and M Avci,et al., (2018) implemented two features help VGG achieve a much higher accuracy compared to pre-existing Convent's such as Alex Net that uses large filters up to a size of 11. Squeeze Net is one of several new CNNs that we have discovered while broadly exploring the design space of CNN architectures. We hope that Squeeze Net will inspire the reader to consider and explore

the broad range of possibilities in the design space of CNN architectures and to perform that exploration in a more systematic manner.

[7] M. van Cleeff, L. E. K. Ndugga, H. Meme, J. Odhiambo, and P. Klatser, et al., (2018) proposed a TensorFlow system for large-scale machine learning, presented at the Proceedings of Operating Systems Design and Implementation. TensorFlow is a machine learning system that operates at large scale and in heterogeneous environments. Tensor-Flow uses dataflow graphs to represent computation, shared state, and the operations that mutate that state. It maps the nodes of a dataflow graph across many machines in a cluster, and within a machine across multiple computational devices, including multicore CPUs, general-purpose GPUs, and custom-designed ASICs known as Tensor Processing Units (TPUs).

[8] P. Sadaphal, J. Rao, G. W. Comstock, and M. F. Beg et al., (2018) implemented the application of artificial neural networks for automated analysis of cystoscopy images a review of the current status and future prospects. An essential requirement for training neural networks is to have a sufficient amount of annotated training data. Manual annotation is time-consuming; for instance, in image segmentation each object must be traced exactly. Ideally, the training data and the test data should come from the same distribution.

[9] R. Ramya and P. S. Babu, et al., (2017) proposed automatic detection of tuberculosis related abnormalities in chest X-ray images using hierarchical feature extraction scheme. We propose an automatic technique for detection of abnormal CXR images containing one or more pathologies like pleural effusion, infiltration, the features are used in two-level of hierarchy to categorize healthy and unhealthy group

CHAPTER 2

SYSTEM ANALYSIS

2.1 Existing System

To differentiate between TB and healthy individuals, several study teams used CXR pictures and a common machine learning technique. The objective of this essay, on the other hand, is to comprehend the issue in greater detail. to evaluate recent research and papers and consider strategies to improve our deep learning model's precision. Use a pre-existing dataset to compare our efforts, then assess their model. implementing a deep learning approach. With 82.09 percent accuracy, Chest X-Ray images could be divided into groups with and without tuberculosis. By adjusting the settings of deep layered CNN, it has been shown that deep machine learning techniques can be used to diagnose tuberculosis. Transfer learning was used in the context of deep learning to identify tuberculosis using pretrained models and their ensembles. Additionally, they demonstrated an interactive visualization programmer for TB patients.

2.1.1 Dis Advantages

- ✓ False-negative results: One of the major disadvantages of tuberculosis detection is that the tests may not always detect the presence of the disease accurately. False-negative results can occur due to a variety of factors, including testing at an early stage of infection, poor quality samples, or errors in laboratory testing
- ✓ False-positive results: Similarly, false-positive results can also occur, which can lead to unnecessary treatment and cause anxiety and stress for the patient.

- ✓ Cost: Some of the tests for tuberculosis detection can be expensive, which may limit their accessibility in low-income countries or for individuals without access to adequate healthcare
- ✓ Time-consuming: Some tests for tuberculosis detection can take several days to produce results, which can delay diagnosis and treatment.
- ✓ Invasive procedures: Some tests for tuberculosis detection require invasive procedures, such as bronchoscopy or lung biopsy, which can be uncomfortable, risky, and expensive

2.2 Proposed System

The goal of the proposal is to evaluate in the most straightforward manner the ability of CNN architecture, which has been pre-trained on the ImageNet dataset, to extract features that are pertinent to the classification of radiographic pictures. Following preprocessing, the image with the segmented lungs is scaled to fit the requirements of each CNN's input layer before being spread throughout the network. That has a significant drawback. The CNN input layer requires the lung images to be scaled, which may result in the loss of a lot of information that could be crucial for diagnosing tuberculosis symptoms. A sliding window technique, where each window is categorized separately, would be an easy solution to solve this issue. They combine point segmentation and classification in two distinct procedures that are assessed by an expert system in the end and enable diagnosis. Convolutional neural network was used to classify the

segmented images. Their findings indicate that the plan is better than the current systems. The foundation of how neural networks work is hidden layers. The hidden layers in the neural network go through a series of changes through the network's neurons if it receives a picture as a single vector. Numerous neurons make up each hidden layer, and each neuron in one layer communicates with the neurons in the next. However, unlike between neurons in separate layers, there is no connection between neurons in the same layer.

In this application process simply gathering the data and analyzing with the free tool like google collab .This process doesn't cost greater. So this is inexpensive .The World Health Organization is currently developing guidelines on screening for tuberculosis disease to inform national screening strategies. This process is complicated by significant gaps in knowledge regarding mass screening. This study aimed to assess feasibility, uptake, yield, treatment outcomes, and costs of adding an active tuberculosis case-finding program to an existing mobile HIV testing service.

2.2.1 Advantages

- ✓ Early detection: One of the main advantages of TB detection is that it can lead to early diagnosis and treatment, which can improve outcomes for the patient and prevent the spread of the disease.
- ✓ Improved treatment: TB detection can help healthcare providers identify the most appropriate treatment regimen for the patient, which can improve treatment outcomes and reduce the risk of drug resistance.

- ✓ Public health benefits: TB detection can also benefit public health by identifying individuals with active TB who may be at risk of spreading the disease to others. This can help prevent further transmission and limit the spread of TB in the community.
- ✓ Contact tracing: TB detection can facilitate contact tracing, which involves identifying individuals who may have been exposed to TB and testing them for the disease. This can help prevent the spread of TB within families, workplaces, and other settings.
- ✓ Monitoring treatment: TB detection can also help healthcare providers monitor the effectiveness of treatment and adjust the treatment regimen as needed. This can improve treatment outcomes and reduce the risk of relapse

CHAPTER – 3

SYSTEM REQUIREMENTS

3.1 Introduction

The system requirements for tuberculosis detection using machine learning would depend on various factors such as the complexity of the machine learning model, the size of the dataset, the hardware and software infrastructure, and the specific use case. However, some general system requirements for a tuberculosis detection system using machine learning might include

3.2 Hardware Requirements

A powerful computer with a high-end processor, sufficient RAM and storage capacity, and a dedicated graphics processing unit (GPU) to perform computationally intensive tasks

- ✓ **Software:** A machine learning framework such as TensorFlow, PyTorch, or Scikit-learn to build and train the models. Additional software packages for data preprocessing, feature extraction, and visualization may also be necessary.

- ✓ **Data:** Large amounts of high-quality data, including medical images, patient records, and other relevant data sources. The data should be pre-processed and annotated to ensure consistency and accuracy.
- ✓ **Model Development:** Development of a machine learning model with high accuracy, sensitivity and specificity to detect tuberculosis, which should be able to process and analyze large amounts of data efficiently
- ✓ **User Interface:** A user-friendly interface for the end-users to interact with the system, and view and interpret the results
- ✓ **Security and Privacy:** A secure system to ensure the confidentiality and integrity of the patient data and other sensitive information.
- ✓ **Regulatory Compliance:** Compliance with relevant regulations and standards, such as the Health Insurance Portability and Accountability Act (HIPAA) and General Data Protection Regulation (GDPR) for data privacy and protection.

It is important to note that the specific requirements may vary depending on the context and intended use of the tuberculosis detection system using machine learning

3.3 Software Requirements

A software system for tuberculosis detection using machine learning typically involves the following components:

- ✓ **Data Collection and Preprocessing:** The first step is to collect and preprocess the data. This involves gathering medical images, patient records, and other relevant data sources. The data may be preprocessed to correct for artifacts, normalize the data, and remove noise.

Feature Extraction: The next step is to extract relevant features from the data. Machine learning algorithms typically work best with numerical or categorical data, so the raw data may need to be converted to a suitable format. Feature extraction involves identifying the relevant attributes of the data that are likely to contribute to the detection of tuberculosis.

- ✓ **Machine Learning Model:** A machine learning model is then developed and trained on the preprocessed data. The model may use different algorithms such as Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), Random Forests, or Decision Trees. The model is typically trained on a subset of the data, and the performance is evaluated using metrics such as accuracy, sensitivity, and specificity.
- ✓ **Deployment:** Once the model is trained, it can be deployed to detect tuberculosis in new data. The deployment may involve integrating the machine learning model into a larger software system or developing a standalone application.
- ✓ **User Interface:** A user interface is designed to allow the end-users to interact with the system. The interface may display the results of

the machine learning model and allow users to input new data for analysis.

- ✓ **Security and Privacy:** The system should include security and privacy measures to protect patient data and ensure regulatory compliance. This may involve encrypting data, implementing access controls, and ensuring compliance with relevant regulations and standards.

Overall, a software system for tuberculosis detection using machine learning involves gathering and preprocessing data, extracting relevant features, training a machine learning model, deploying the model, and providing a user-friendly interface for end-users

CHAPTER – 4

SYSTEM DESIGN

4.1 System Architecture

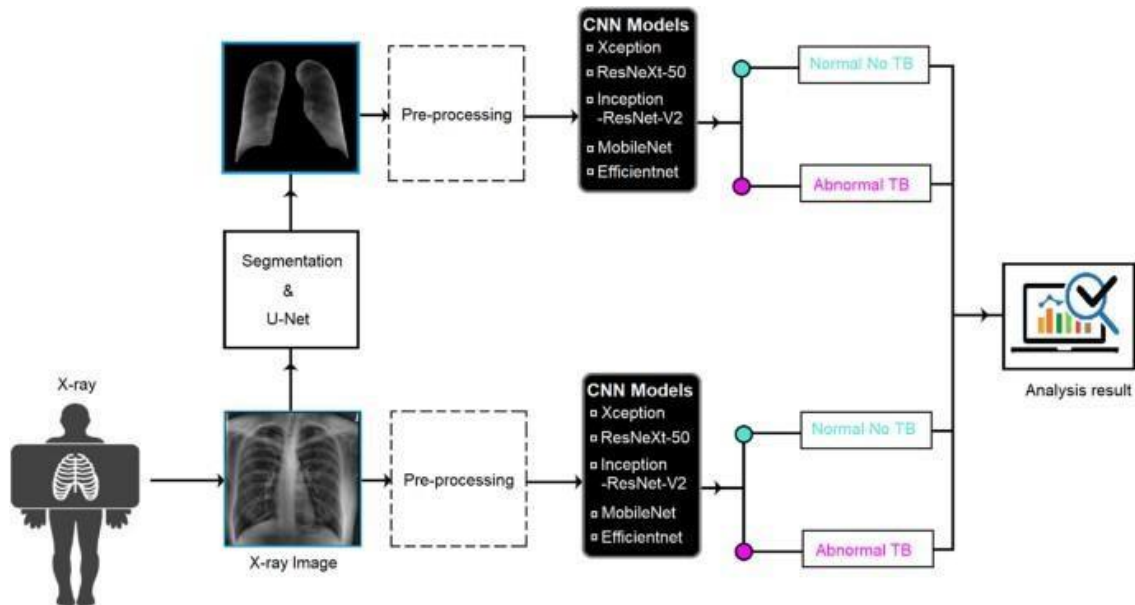


Figure 4.1: Tuberculosis Detection Architecture

✓ Represents the convolution layer is the core building block of the CNN. It carries the main portion of the network's computational load. This layer performs a dot product between two matrices, where one matrix is the set of learnable parameters otherwise known as a kernel, and the other matrix is the restricted portion of the receptive field. The kernel is spatially smaller than an image but is more in depth. Unlike standard object detection task encapsulating bounding box information, the localization of TB manifestations is achieved primarily via class activation mapping;

essentially, this entails the production of an attention map over the image to spotlight all the detected suspicious areas instead of predicting edges of bounding boxes.

4.2 Data Flow Diagram



Figure 4.2: Data Flow Diagram

✓ Figure 4.2 represents the input of CNN is assumed to be in the form of an image. This allows us to encode certain properties in the network. In our work, since the input is a grayscale image, the array of pixels will be height \times width \times depth (e.g., $224 \times 224 \times 1$). For generating useful features from the images data, specific parameters were modified in the Conv layers throughout the training. Concurrently, the collection of learned parameters

in the FC layers classify the extracted features in the target classes (normal image or TB image).

4.3 Class Diagram

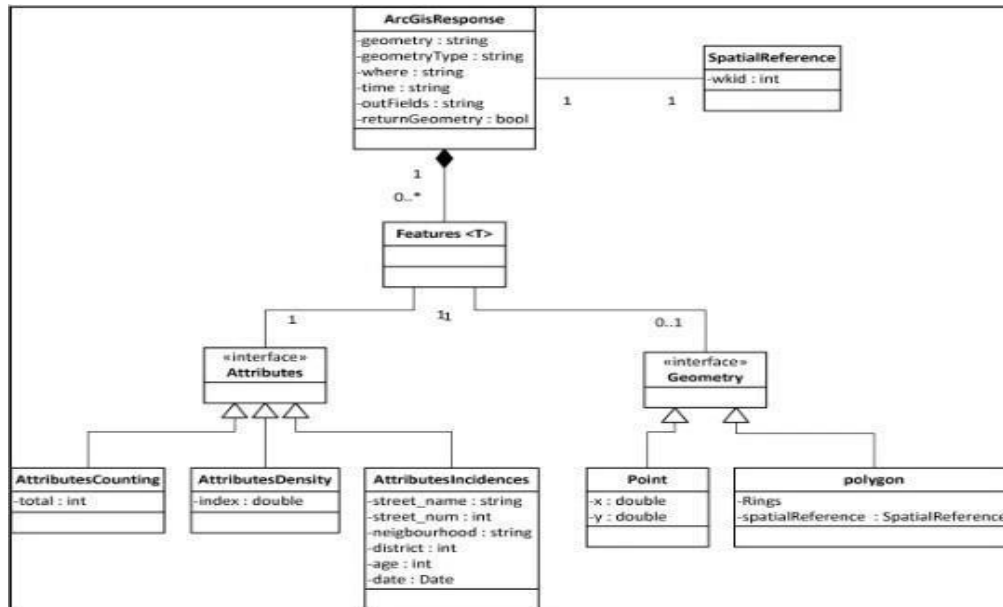


Figure 4.3: Class Diagram

✓ Figure 4.3 shows the static view of an application. The describes the types of objects in the system and the different types of relationships that exist among them. This modeling method can run with almost all Object-Oriented Methods. A class can refer to another class. A class can have its objects or may inherit from other classes. Illustrates data models for even very complex information systems It provides an overview of how the application is structured before studying the actual code. This can easily reduce the maintenance time

4.4 Sequence Diagram

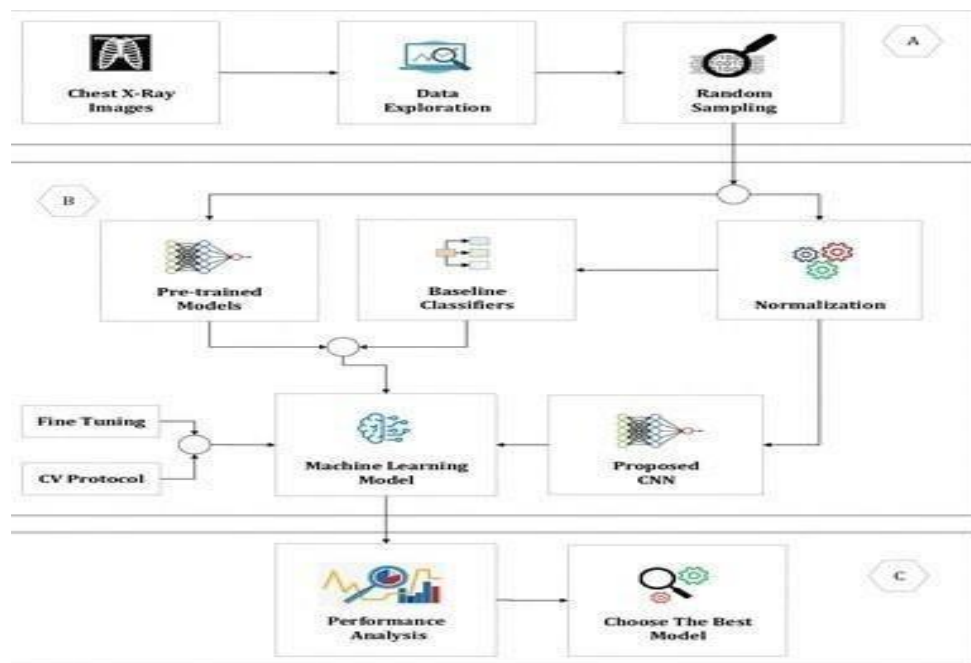


Figure 4.4: Sequence Diagram

✓ Figure 4.4 shows the sequence Diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

4.5 Activity Diagram

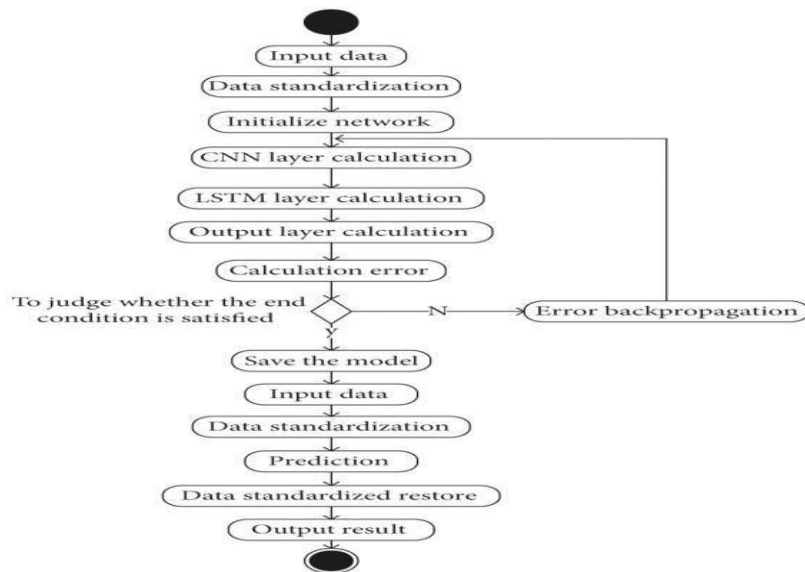


Figure 4.5: Activity Diagram

Figure 4.5 represents the input data. input the data required for CNN-LSTM training. Data standardization: as there is a large gap in the input data, in order to train the model better, the standardization method is adopted to standardize the input data, LSTM layer calculation the output data of the CNN layer are calculated through the LSTM layer, and the output value is obtained. Output layer calculation: the output value of the LSTM layer is input into the full connection layer to get the output value. Calculation error the output value calculated by the output layer is compared with the real value of this group of data, and the corresponding error is obtained. To judge whether the end condition is satisfied

CHAPTER – 5

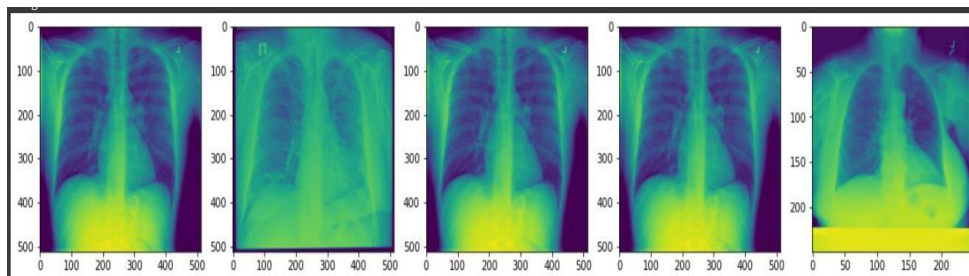
SYSTEM IMPLEMENTATIONS

5.1 List Of Modules

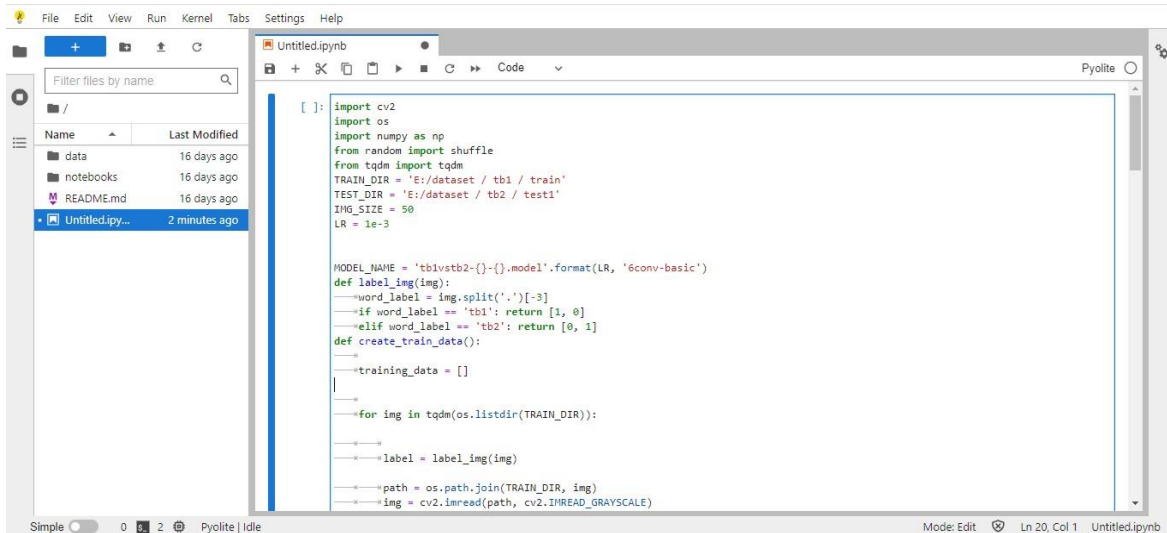
- ✓ **MODULE 1** : Collection of Chest images from various sources.
- ✓ **MODULE 2** : By using CNN ALGORITHM.
- ✓ **MODULE 3** : Testing and training the data sets.
- ✓ **MODULE 4** : Preparing the report

5.2 MODULE DESCRIPTION

5.2.1 : COLLECTION OF CHEST IMAGES FROM VARIOUS SOURCES.



5.2.2 : Processing of data for chest images using CNN ALGORITHM. And coding for algorithm using python



```
[ ]: import cv2
import os
import numpy as np
from random import shuffle
from tqdm import tqdm
TRAIN_DIR = 'E:/dataset / tb1 / train'
TEST_DIR = 'E:/dataset / tb2 / test1'
IMG_SIZE = 50
LR = 1e-3

MODEL_NAME = 'tb1vstb2-{}-{}.model'.format(LR, '6conv-basic')

def label_img(img):
    word_label = img.split('.')[0]
    if word_label == 'tb1': return [1, 0]
    elif word_label == 'tb2': return [0, 1]
def create_train_data():
    training_data = []
    for img in tqdm(os.listdir(TRAIN_DIR)):
        label = label_img(img)
        path = os.path.join(TRAIN_DIR, img)
        img = cv2.imread(path, cv2.IMREAD_GRAYSCALE)
```

5.2.3 : Testing and training the data sets.

Layer	Type	Input Size	No. of Filters	Filter Size	Padding	Stride	Output Size
1	Conv+ReLU	227x227x3	96	11x11x3	0	4	55x55x96
2	Max Pool	55x55x96	96	3x3	0	2	27x27x96
3	LR Norm	27x27x96	-	-	-	-	27x27x96
4	Conv+ReLU	27x27x96	256	5x5x96	2	1	27x27x256
5	Max Pool	27x27x256	256	3x3	0	2	13x13x256
6	LR Norm	13x13x256	-	-	-	-	13x13x256
7	Conv+ReLU	13x13x256	384	3x3x256	1	1	13x13x384
8	Conv+ReLU	13x13x384	384	3x3x384	1	1	13x13x384
9	Conv+ReLU	13x13x384	256	3x3x384	1	1	13x13x256
10	Max Pool	13x13x256	256	3x3	0	2	6x6x256
11	Fully Connected	6x6x256	-	-	-	-	1x4096
12	Dropout	-	-	Keep Probability = 0.5	-	-	-
13	Fully Connected	1x4096	-	-	-	-	1x4096
14	Dropout	-	-	Keep Probability = 0.5	-	-	-
15	Fully Connected	1x4096	-	-	-	-	1x1000

5.2.4 Preparing the report .

- ✓ Tuberculosis (TB) detection module is a computer-based system designed to aid in the detection and diagnosis of tuberculosis. The

module typically consists of two main components: image acquisition and image analysis.

- ✓ The image acquisition component involves the collection of medical images, such as chest X-rays or computed tomography (CT) scans, from patients suspected of having tuberculosis. These images are then stored in a database for further analysis.

Overall, the tuberculosis detection module is designed to provide a fast and accurate method for identifying tuberculosis in patients, allowing for earlier detection and treatment of the disease

CHAPTER - 6

TESTING

6.1 Functional Testing: Functional testing is a type of testing that seeks to establish whether each application feature works as per the software requirements. Each function is compared to the corresponding requirement to ascertain whether its output is consistent with the end user's expectations. The testing is done by providing sample inputs, capturing resulting outputs, and verifying that actual outputs are the same as expected outputs.

6.1.1 Input

```
1: predicted = model.predict(np.array(Xtest))
2: model.evaluate(Xtest, np.array(ytest))
3 predicted
```

6.1.2 Test result

```
array([[3.9285904e-01,  6.0714096e-01], [1.0673884e-01,  8.9326113e-01],
[1.2404387e-12,  1.0000000e+00], ..., [2.7577824e-08,  1.0000000e+00],
[6.5884673e-11,  1.0000000e+00], [4.8934150e-01,  5.1065850e01]], dtype=float32)
```

6.2 SYSTEM TESTING

The 224x224 pixels in three dimensions were used for the resizing of the images (RGB). Below Figure depicts an example of one of these photographs. In order to train the VGG16 algorithm, these photographs were utilized to train the algorithm. VGG is a Convnet that placed first and second, respectively, in the ImageNet ILSVRC-2014 competition for localization and classification, in which it performed exceptionally well. It takes use of convolutional filters with a window size of 3*3 pixels, which are extremely small. The depth of the Convnet has also been increased when compared to earlier systems, and it now has between 16 and 19 layers, depending on the system. The image comparison between original TB CXR, enhanced Unsharp Masking (UM) , and enhanced High-Frequency Emphasis Filtering (HEF) images.

CHAPTER – 7

RESULTS AND DISCUSSION

7.1 RESULTS

- ✓ The results of tuberculosis detection module can vary depending on various factors such as the quality of medical images, the accuracy of the analysis algorithms used, and the prevalence of the disease in the population being screened.
- ✓ In general, a well-designed tuberculosis detection module can achieve high levels of sensitivity and specificity. Sensitivity refers to the ability of the module to correctly identify patients who have tuberculosis, while specificity refers to the ability to correctly identify patients who do not have tuberculosis.
- ✓ Studies have shown that some tuberculosis detection modules can achieve sensitivity and specificity rates of over 90%, which is considered to be very good. However, it is important to note that no tuberculosis detection module is 100% accurate, and some false positive and false negative results may occur.

7.2 DISCUSSION

Tuberculosis detection is critical for controlling the spread of the disease and initiating treatment for those who are infected. Traditional TB detection methods include sputum microscopy and culture, which can be time-consuming and may not be suitable for all patients, especially those with extra-pulmonary TB. In recent years, computer-aided detection systems, including tuberculosis detection modules, have emerged as a promising approach for TB diagnosis. These systems can analyze medical images, such as chest X-rays and CT scans, to aid in the detection of TB. The use of tuberculosis detection modules has several advantages over traditional TB detection methods. They can provide a fast and accurate diagnosis, reducing the time required for TB detection and treatment initiation. They can also help detect TB in patients who may not show any clinical symptoms, reducing the risk of transmission to others.

CHAPTER – 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 Conclusion

The CNN that was trained using Visual Geometry Group (VGG Net) as feature extractor and the CNN build and trained from scratch have similar performance. The effect of different types of data augmentation on the performance of the proposed architecture can also be tested as proven in the previous research. Future improvements could be evaluated in terms of using different configurations of architecture in the model and also different hardware platforms. The experiments showed that the accuracy of the proposed idea is very competitive.

8.2 Future Enhancements

Annotating the chest regions for abnormalities and detecting them Generating clinical readings from the CXRs Building Visualization pipeline more robust to different changes in network architectures. To improve recognition rate of final classification process hybrid algorithms like Artificial Neural Network, Bayes classifier, Fuzzy Logic can also be used. Mobile application can be developed which is handy and easy to use.

ANNEXURE

APPENDIX 1

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os
import cv2
from pathlib import Path
import seaborn as sns
import matplotlib.pyplot as plt
from skimage.io import imread

# Define path to the data directory
data_dir = Path(r'C:\Users\kuchi\Downloads\tb-
project\TB_Chest_Radiography_Database')
data_dir

# Get the path to the normal and pneumonia sub-directories
normal_cases_dir = data_dir / 'Normal'
Tuberculosis_cases_dir = data_dir / 'Tuberculosis'

# Get the list of all the images
normal_cases = normal_cases_dir.glob('*.png')
Tuberculosis_cases = Tuberculosis_cases_dir.glob('*.png')
```

```
# An empty list. We will insert the data into this list in (img_path, label) format
train_data = []
```

```
# Go through all the normal cases. The label for these cases will be 0
for img in normal_cases:
    train_data.append((img,0))
```

```
# Go through all the pneumonia cases. The label for these cases will be 1
for img in Tuberculosis_cases:
    train_data.append((img, 1))
```

```
# Get a pandas dataframe from the data we have in our list
train_data = pd.DataFrame(train_data, columns=['image', 'label'],index=None)
```

```
# Shuffle the data
train_data = train_data.sample(frac=1.).reset_index(drop=True)
```

```
# How the dataframe looks like?
train_data.head()
```

```
train_data.shape
```

```
# Get the counts for each class
cases_count = train_data['label'].value_counts()
print(cases_count)
```



```

# Plot the results
plt.figure(figsize=(10,8))
sns.barplot(x=cases_count.index, y= cases_count.values)
plt.title('Number of cases', fontsize=14)
plt.xlabel('Case type', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.xticks(range(len(cases_count.index)), ['Normal(0)', 'Tuberculosis(1)'])
plt.show()

# Get few samples for both the classes
Tuberculosis_samples = (train_data[train_data['label']==1]['image'].iloc[:5]).tolist()
normal_samples = (train_data[train_data['label']==0]['image'].iloc[:5]).tolist()

# Concat the data in a single list and del the above two list
samples = Tuberculosis_samples + normal_samples
del Tuberculosis_samples, normal_samples

# Plot the data
f, ax = plt.subplots(2,5, figsize=(30,10))
for i in range(10):
    img = imread(samples[i])
    ax[i//5, i%5].imshow(img, cmap='gray')
    if i<5:
        ax[i//5, i%5].set_title("Tuberculosis")
    else:
        ax[i//5, i%5].set_title("Normal")
    ax[i//5, i%5].axis('off')

```

```

        ax[i//5, i%5].set_aspect('auto')
plt.show()

from tqdm import tqdm
train_normal = data_dir / 'Normal'
train_Tuberculosis = data_dir / 'Tuberculosis'

# Get the list of all the images
normal_cases = normal_cases_dir.glob('*.*png')
Tuberculosis_cases = Tuberculosis_cases_dir.glob('*.*png')
train_data = []
train_labels = []
from keras.utils.np_utils import to_categorical

for img in tqdm(normal_cases):
    img = cv2.imread(str(img))
    img = cv2.resize(img, (28,28))
    if img.shape[2] == 1:
        img = np.dstack([img, img, img])
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = np.array(img)
    img = img/255
    label = 'normal'
    train_data.append(img)
    train_labels.append(label)

```

```

# Tuberculosis cases
for img in tqdm(Tuberculosis_cases):
    img = cv2.imread(str(img))
    img = cv2.resize(img, (28,28))
    if img.shape[2] ==1:
        img = np.dstack([img, img, img])
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img=np.array(img)
    img = img/255
    label = 'Tuberculosis'
    train_data.append(img)
    train_labels.append(label)

# Convert the list into numpy arrays

train_data1 = np.array(train_data)
train_labels1 = np.array(train_labels)

print("Total number of validation examples: ", train_data1.shape)
print("Total number of labels:", train_labels1.shape)
from imblearn.over_sampling import SMOTE
smt = SMOTE()
train_rows=len(train_data1)
train_data1 = train_data1.reshape(train_rows,-1)
train_data2, train_labels2 = smt.fit_resample(train_data1, train_labels1)

cases_count1 = train_labels2['label'].value_counts()

```

```

print(cases_count1)

# Plot the results
plt.figure(figsize=(10,8))
sns.barplot(x=cases_count1.index, y= cases_count1.values)
plt.title('Number of cases', fontsize=14)
plt.xlabel('Case type', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.xticks(range(len(cases_count.index)), ['Normal(0)', 'Tuberculosis(1)'])
plt.show()

train_data2.shape
train_labels2.shape
train_labels2

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(train_data2, train_labels2,
test_size=0.13, random_state=42)

from tensorflow import keras
from tensorflow.keras import layers
data_augmentation = keras.Sequential(
[
    layers.RandomFlip("horizontal",
                        input_shape=(28,
                                     28,
                                     3)),

```

```

        layers.RandomRotation(0.1),
        layers.RandomZoom(0.1),

    ]
)
import tensorflow as tf
from tensorflow.keras import layers, models

model = models.Sequential([
    data_augmentation,
    layers.Conv2D(28, (3, 3), activation='relu', input_shape=(28, 28, 3)) ,
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu')

])

model.summary()
model.add(layers.Flatten())
model.add(layers.Dense(640, activation='tanh'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(564, activation='tanh'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(64, activation='tanh'))
model.add(layers.Dense(64, activation='sigmoid'))
model.add(layers.Dense(2))

```

```

model.summary()

test_data = []
image=r'C:\Users\kuchi\Downloads\tb-
project\TB_Chest_Radiography_Database\Tuberculosis\Tuberculosis-101.png'
img = cv2.imread(str(image))
img = cv2.resize(img, (28,28))
if img.shape[2] ==1:
    img = np.dstack([img, img, img])
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img=np.array(img)
img = img/255
test_data.append(img)

# Convert the list into numpy arrays

test_data1 = np.array(test_data)

# Importing Image module from PIL package
from PIL import Image
import PIL

```

REFERENCES

- [1] D. R. Nayak, D. Das, R. Dash, S. Majhi, and B. Majhi, “Deep extreme learning machine with leaky rectified linear unit for multiclass classification of pathological brain images,” *Multimedia Tools and Applications*, vol. 79, no. 21-22, pp. 15381–15396, 2021 .
- [2] E. Paul, P. Gowsalya, N. Devadarshini, M. P. Indhumathi, and M. Iniyadharshini, “Plant leaf perception using convolutional neural network,” *International Journal of Psychosocial Rehabilitation*, vol. 24, no. 5, pp. 5753–5762, 2020.
- [3] H. Shouno, and S. Kido, “Deep convolution neural network with 2-stage transfer learning for medical image classification,” *Brain Neural Networks*, vol. 24, no. 1, pp. 3–12, 2021.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2020.
- [5] M. S. Faridi, M. A. Zia, Z. Javed, I. Mumtaz, and S. Ali, “A comparative analysis using different machine learning: an efficient approach for measuring accuracy of face recognition,” *International Journal of Machine Learning and Computing*, vol. 11, no. 2, pp. 115–120, 2021.
- [6] M. Sarıgul, B. M Ozyildirim, and M Avci, “Differential convolutional neural network,” *Neural Networks: 9e Official Journal of the International Neural Network Society*, vol. 116, no. 9, pp. 279–287, 2019.
- [7] M. van Cleeff, L. E. K. Ndugga, H. Meme, J. Odhiambo, and P. Klatser, “*e role and performance of chest X-ray for the diagnosis of tuberculosis: a cost-effectiveness analysis in Nairobi, Kenya,” *BMC Infectious Diseases*, vol. 5, no. 1, p. 111, 2019.
- [8] P. Sadaphal, J. Rao, G. W. Comstock, and M. F. Beg, “Image processing techniques for identifying mycobacterium tuberculosis in ziehl-neelsen stains”, vol. 12, no. 5, , pp. 579–582, 2020.
- [9] R. Ramya and P. S. Babu, “Automatic tuberculosis screening using canny Edge detection method,” in *Proceedings of the 2nd International Conference on Electronics and Communication Systems* , pp. 282–285 2019.

- [10] S. K. Sharma and A. Mohan, “Tuberculosis: from an incurable scourge to a curable disease journey over a millennium,” *Indian Journal of Medical Research*, vol. 137, no. 9, pp. 455–93, 2020.