Detection of Tuberculosis Using Deep Learning

A major project report submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

Submitted by

Yuyutsu Chinaria (201520)

Shridhar Singh (201526)

Under the guidance & supervision of

Dr. Ekta Gandotra

Department of Computer Science & Engineering and Information Technology Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled **'Detection of Tuberculosis using Deep Learning'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Ekta Gandotra**(Associate Professor, Department of Computer Science & Engineering and InformationTechnology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Yuyutsu Chinaria (201520)

Shridhar Singh (201526)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Ekta Gandotra Associate Professor

Department of Computer Science & Engineering and Information Technology

Date:

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty god for his divine blessing makes it possible for us to complete the project successfully.

I am grateful and wish my profound indebtedness to my supervisor **Dr. Ekta Gandotra**, Department of Computer Science Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of our supervisor in the field of “**Detection of Tuberculosis using Deep learning**” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I am grateful to each and every individual who directly or indirectly helped me in making this project a success.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

LIST OF ABBREVIATIONS

Abbreviation	**Definition**
TB	Tuberculosis
CNN	Convolution neural network
CXR	Chest X Ray
WHO	World Health Organization
GPU	Graphic processing unit
TPU	Tensor Processing Unit
AMBO	Adaptive Monarch Butterfly Optimization
AFCM	Adaptive Fuzzy C-means
DBN	Deep Belief Network
IOT	Internet of Things

ABSTRACT

In this world Tuberculosis is considered as one of the threat to humans and it is considered the fifth major cause of death. Early detection of TB can overcome the spread of the disease. With the advancement of computer aided diagnostic tools the early detection of TB is possible, with a view to provide assistance to healthcare professionals we decided to develop a computer aided tool that will aid and provide a way of early detection of tuberculosis that will curb the fast spread of this life threatening disease.

For development of such computer aided tool a model is required that will be able to detect the identifiers present in the patient’s chest x ray. For the development of model, first a large dataset of CXR is required that should contain both x ray mage of normal and tuberculosis infected one. Afterwards the type of model that will be used to detect the disease is selected. For this cause CNN model is selected, as CNN models are largely used for the purpose of image classification, CXR in our case. Afterwards the model is trained on the obtained dataset. The trained mode is evaluated using metrices such as accuracy, precision and recall. In the testing phase the model resulted in an accuracy of 97.14 % that was sufficient to deploy the model so that it can predict the disease.

For the model to be deployed a platform is required where the model can work in background and provide the result to the patient, for this cause streamlit is used that is a python library that is widely used for creation of web application for machine learning models. The interface that is developed is as minimal as it can be with only a upload tab that not only provide a clean glance but also provide a user friendly experience that can be used by a novice.

During the development of the project challenges such as data collection, low accuracy of model and deployment of model were faced. To begin with , collecting data related to medical field is always a task as there is limited amount of public data available , and the available one’s is either outdated or corrupt, so to overcome this problem a trusted source is required and for that reason Kaggle was chosen as data collection resource. Next comes the low

accuracy of model that was defied by varying the layer depth, hyperparameter tuning that resulted test accuracy of 97.14 %.The last hurdle was to deploy the model on a platform, for that our first approach was to build a website and connect the model in background but that didn’t work, so we explored other platforms where the model can be deployed hassle free, and the search ends at streamlit. Not only streamlit provided a seamless connection for the model but it is easier to setup than conventional methods. Moreover the streamlit platform provides a user friendly interface that enable the user to upload the image of chest x ray and obtain the result.

CHAPTER 1 : INTRODUCTION

1.1. INTRODUCTION

TB remains a persistent and significant public health challenge worldwide, with A total of 1.3 million people died from TB in 2022 (including 167 000 people with HIV), according to the World Health Organization (WHO) [1]. Early and accurate diagnosis is crucial for effective TB management, as delayed detection can lead to increased transmission and poorer treatment outcomes. Traditional diagnostic methods, while valuable, often face limitations in terms of speed, sensitivity, and accessibility.

A new age in medical diagnostics has been created by the development of deep learning, a type of artificial intelligence. Deep learning algorithms have shown impressive performance in a variety of medical imaging applications, especially those that are based on convolutional neural networks, also called CNNs, and neural networks with recurrent . The state of the art techniques have the ability to improve the traditional diagnostic process for TB detection

The report provides a deep learning based model that will be able to detect TB. The proposed model aims to improve the traditional TB detection methods by harnessing the capabilities of the CNN model that we are using in the initial stage of the model development. The CNN based model will extract the important information from the CXR images that will aid both the patient and medical professional.

Using a deep learning approach will be able to train the model on a large dataset that will provide an upper edge for the model to learn complex pattern of TB identifiers, miliary patterns and improve the accuracy of the model further.

1.2. PROBLEM STATEMENT

Worldwide TB is considered a serious health issue and early diagnosis of this disease is important for disease prevention and its diagnosis.For the detection of TB medical imaging technique such as CXR is used but understanding and pointing out patterns in those CXR images visually is laborious and prone to human error. With a view to minimize these life threatening issues we will develop a deep learning based model that will be able to detect TB using CXR images.

The objective is to develop a system in cooperation with the medical professionals' knowledge that will be able to provide a fast, accurate method of TB detection. Thisapproach will reduce the bridge between human intelligence and machine intelligence while improving confidence in incorporating techniques in medical field where there will be less chance of human error and need.

1.3. OBJECTIVE

- To develop a deep learning based model that detects TB.
- Comparing developed models and selecting the best performing model in terms of accuracy.
- Developing a web based application where the chosen model will be deployed at the backend for TB detection.

1.4. SIGNIFICANCE AND MOTIVATION OF THE PROJECT

Early Identification and Management:

Using a deep learning based approach will be able to facilitate the diagnosis procedure by early TB detection .Combining the best of deep learning and medical knowledge will aid in reducing the casualties caused by the disease. TB is a disease where early detection is important due to its contagious nature. Not only early detection reduces the chance of fatality but also provides a sigh of relief for the patient and its family but also limits the disease spread within society.

Accurate Prediction:

Models based on deep learning have an upper hand to any other traditional models in detecting subtle patterns in CXR that are linked to TB as the model is able to betrained on a big dataset. The model has a higher accuracy in identifying minute symptomatic details contrary to manual evaluation. The huge dataset aids the model to expand its learning capabilities whilst incorporating varied TB symptoms. The proposed method provides an assurance of dependability and accuracy in a range of scenarios and reduces the possibility of human error by automation in the process of TBCXR analysis.

Impact On Global Health:

Deep learning has the potential and ability to aid in improving the health sector when used for automated detection of disease. In regions of limited resource, unavailability of medical professionals the proposed model will be able to provide a better and easy to use healthcare accessibility solution.In addition to the statement, the deep learning based approach helps in the diagnosis process by simplifying the diagnostic procedure, reducing the strain from the healthcare professionals, and improving the accuracy in detection.The model will be able to promote better TB detection and diagnosis management

.

1.5. ORGANIZATION OF PROJECT REPORT

Chapter 1: Introduction The introduction chapter covers the groundwork of our project. Chapter 1 covers the problem statement, significance of the project, with the project report organization covering a summary of the chapters that explains the process involved in the project. This introduction provides context for readers by outlining the goal of the project and the main factors influencing its implementation.

Chapter 2: Literature Review Conducts a thorough review of previous studies and publications relevant to your project in the literature review chapter. Give a critical evaluation, examining relevant research and determining how well it fits with ongoing projects.In the chapter we have provided a detailed overview of the research papers that are relevant with the project.Along with the overview we have summarized the research papers highlighting only the aim, resources used, shortcoming that were faced. With a detailed explanation a literary survey table is provided in the chapter that discusses the paper, its journal/conference , tools/techniques used , dataset obtained and their limitations, withdetails of research papers the chapter also discusses the key gaps that were identified in research papers.

Chapter 3: System Development The chapter explores the technical details of the project in this chapter. A description of the project architecture, requirement, and implementation are summarized in this chapter. The prior requirements, working of the model, tools and techniques used are explained in detail. Explanation of the tools used such as Streamlit application for web app development, libraries used such as Matplotlib, Keras, Tensor flow, Kaggle api is provided in the chapter. The architecture of the working model is described with the help of a diagram for better understanding of the reader. Each process involved in the working model such as data collection, preprocessing, classification etc is detailed with supporting screenshots respectively in chapter 3 along with the challenges involved in developing the model like in data collection and privacy concerns , licensing for the model.

Chapter 4: Testing: Chapters 4 cover the techniques, tools and strategies used for the testing phase of the model .In order to develop an accurate and fit to use model in a real world environment the model goes through various testing measures which are described in the testing chapter. The chapter provides a detailed view of the libraries used for the testing phase that include Keras, Sckitlearn, Matplotlib with their working. Along with the libraries used the Test cases and outcome received from the model is detailed in the chapter with supporting results screenshots.

Chapter 5: Result and Evaluation: The chapter provides a detailed view on the results that wegot from the model.The model overall results are given in detail in the chapter with supporting screenshots of the results that we got. The chapter discusses the results obtained from processes such as image segmentation, heatmap , web app with respective screenshots attached.

Chapter 6: Conclusion and Future Scope: The chapter provides a description on the project initial achievements, results, challenges faced during model development and the plan fro the future improvement. Alongside the overview of the model we also added a description on the improvement and iterations needed further for the model. Some suggestions are added in order to further enhance the useability, feasibility and accuracy of the project so that later it can be deployed in real world without any hiccups.

CHAPTER 2: LITERATURE SURVEY

2.1. OVERVIEW OF RELEVANT LITERATURE:

A. S. Becker et al. presents in his survey that in order to aid health care providers in their fight against TB, the study aspires to improve the ability to recognize the disease using chest x-rays [2]. The author deployed new methods with nine different models, including deep learning, data augmentation, and image preprocessing. 3500 TB-positive and 3500 normal chest X-ray photos from public databases were used in the model development. Initially, the researchers used two U-net models to segment X-ray images; followed by the entire X-ray images to classify TB; and in the third, they used segmented lung images to classify TB. When it came to identifying TB from X-ray pictures, the top-performing model,ChexNet, produced results with high levels of accuracy, precision, sensitivity, F1-score, and specificity (96.47%, 96.62%, 96.47%, 96.47%, and 96.51%, respectively).

T. Rahman presents in his survey that photographs of the CXRs were taken of 138 individuals who had both TB and HIV [3]. The researchers then classified the pictures into various pathological patterns, including consolidation, effusion, cavity, interstitial alterations, miliary pattern, and normal. First, problematic areas were located, and then theseareas were classified using Deep Learning software. According to the study, the program performed exceptionally well in localizing diseased areas, particularly when it came to differentiating between intraparenchymal alterations and pleural effusions. As a method of diagnosis for tuberculosis, the study's result indicates that Deep Learning classification of CXR photos shows potential. However, more effort is required to improve performance, especially by creating larger and better-quality datasets.

Oleg Alienin et al. explains the work that evaluates the effect of image augmentation on deep learning performance in order to tackle this difficulty [4]. Important elements as well as general and local features of the TB CXR pictures were successfully highlighted by the image enhancement technique in use. We tested three image improvement algorithms: Contrast Limited Adaptive Histogram Equalization (CLAHE), High-Frequency Emphasis Filtering (HEF), and Unsharp Masking (UM). After that, pre-trained ResNet andEfficientNet models were used for transfer learning with the improved image samples. The study obtained an AUC (Area Under Curve) score of 94.8% and a classification accuracy of 89.92% on a TB picture dataset. Notably, the Shenzhen dataset, which is accessible to the public, was used to obtain all results. This study shows how picture augmentatio might boostthe efficacy of deep learning algorithms for tuberculosis diagnosis.

R. Purkar presents in their survey that the researchers used a transfer learning strategy using pre-trained models (Inception_v3, Xception, ResNet50, VGG19, and VGG16) for identifying TB and normal cases in CXR pictures, and they built a CNN from scratch (ConvNet) for automatic TB identification [5]. The models got satisfactory accuracy for two-class categorization, Slightly less than pre-trained models, the ConvNet model obtained 88.0% precision, 87.0% sensitivity, 87.0% F1-score, 87.0% accuracy, and an AUC of 87.0%. Xception, ResNet50, and VGG16 outscaled the other models with the best results having 90.0% accuracy, 91.0% precision, sensitivity, and F1-score all exceeding 91.0%. The work concluded with the demonstration of a deep CNN-based transfer learning strategy for automatic TB categorization in chest radiographs. With Xception and ResNet50, all models attained an F1-score of more than 87.0% as well as accuracy, precision, and sensitivity.

V. Kumar presents in his survey that Adaptive Fuzzy C-means (AFCM) clustering is first used to preprocess and segment the CXR pictures. The Deep Belief Network (DBN), a DL classifier, is then fed these features [6]. Utilizing an Adaptive Monarch Butterfly Optimization (AMBO) technique, the DBN is optimized and classification accuracy is increased. The objectives of the Deep Belief Network with DBN-AMBO are to maximize weighting parameters, decrease error, and enhance accuracy. Python is the platform used for the entire implementation.

In conclusion, this work offers an improved DL model that uses a methodical approach to segmentation, feature extraction, preprocessing, and optimum classification to identify tuberculosis automatically. By combining the AMBO algorithm with DBN, the model's efficiency and accuracy will be further enhanced, potentially leading to breakthroughs in the use of CXR images for TB diagnosis.

Munadi presents that TB diagnosis in HIV positive patients may be complicated in the areas where HIV and TB coinfection is prevalent and there is an inadequate number of radiologists and atypical X-ray presentations [7]. This was overcome by developing a deep learning algorithm using clinical data and chest X-rays of 677 HIV-positive patients in SouthAfrican hospitals. It sought to ascertain whether this algorithm could be of use to clinicians as an online diagnostic tool. The study revealed that the algorithm led to a moderate, albeit significant, increase in clinician accuracy ($p = 0.002$). Addition of algorithm led to increase in the mean accuracy of clinicians, from 0.60 to 0.65. The stand-alone algorithm however, showed greater accuracy in this case, having accuracy rates of 0.79 on the same unseen test cases. This implies that deep learning support can improve clinician accuracy for TB cases through chest x-rays, in such areas as those highly infected with HIV/TB. Also, the stand-alone algorithm has high accuracy, which makes it an important tool in locations with insufficient radiological skills.

G.Hemalatha presents work that investigates the use of deep learning to the processing of chest X-ray (CXR) images for the purpose of computer-aided diagnosis (CADx) of tuberculosis [7]. The results highlight the usefulness of various data augmentation methods and lung segmentation. When applied to pre-processed datasets produced following lung segmentation, the deep convolutional neural network (CNN) demonstrated its capacity to train efficiently, overcoming obstacles like overfitting. The segmented dataset's lossless data augmentation produced the lowest validation loss without overfitting, and following lossy data augmentation, the accuracy stayed similar to that of the original and other pre-processed datasets. The report concludes by suggesting that better segmentation and data augmentation can lead to better improvements in CADx for small and imbalanced data sets, in addition to using larger datasets and more complicated deep CNNs

2.2. KEY GAPS IN LITERATURE:

The main key gap in our literature surveys is that the dataset in every model was either incomplete or the dataset was imbalanced.

Overfitting, and lower accuracy compared to more complex models. The impact of lossy data augmentation requires further investigation, especially concerning dataset variability.

Limitations such as potential patient overlap between training and test sets. Clinicians lack access to complete clinical data.

The study acknowledged limitations related to GPU and memory requirements. Due to hardware constraints, images were down-sampled to 224×224 pixels before being fed into pre-trained networks, potentially sacrificing some accuracy, especially for subtle findings.

S.NO	Paper Title	Conference	Tools/Technique used	Dataset Used	Results	Limitations
1.	Detection of tuberculosis patterns indigital photographs of chest X-ray images using Deep Learning	Internationa l Union Against Tuberculosi s andLung Disease,201 8	Model used indetection was Unet and Xception	Dataset from 138 cohort patients with HIV and TB Co-infectionwere taken.	Unet Model perfectly distinguish pleural effusions from intraparench ymal is excellently with area under curve roc of 0.82	misclassifi cation were consolidat io ns as cavitations , and miliary patterns as interstitial patterns
2.	Reliable Tuberculosis Detection Using ChestX-Ray WithDeep Learning, Segmentation and Visualization	IEEE,2020	Models used wereU Net, Score,Cam TSNE	Public databases were used to create a database of 3500 TB infected and 3500 normal chest X-ray, NLM dataset,Bel arus Dataset,RS NA CXR Dataset.	ChexNet provided accuracy of 96.47% in non segmented data set while DenseNet provided and accuracy of 98.6% in segmented data.	lack of accessto patient records, prior examinati on s, and lateral views

3.	Chest X-Ray Analysisof Tuberculosis by Deep Learning with Segmentation and Augmentation	IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO),2018	Tools used were CADx,CNN	JSRT dataset with 247 images of cancer, LIDC dataset with ~103 images [8]; Montgomery County (MC) dataset	CADx performed better than VGG16 and ResNet50.	limitations were the small and unbalanced dataset, overfitting,and lower accuracy compared tomore complex models.
4	Deep learning assistance for physician diagnosisof tuberculosis usingchest x-rays in patients with HIV	npj Digital Medicine, 2020	Tool used was CheXaid	GF Jooste Hospital (Novembe r 2011–February 2013), and Khayelitsh a Hospital (March 2013–October 2014).	CHexAid provide Accuracy of95% when trained without clinical covariates	Limitationssuch as potential patient overlap between training andtest sets .Clinicians lack access tocomplete clinical data.

5.	Image Enhancemen tfor Tuberculosis Detection UsingDeep Learning	IEEE Access,2020	TB CXR Image Enhancement Based on Unsharp Masking(UM) and High-Frequency Emphasis Filtering (HEF).Pre-Trained ResNet and EfficientNe tmodels were used	The Shenzhen data set was collected at Shenzhen Hospital, Guangdon g Province	UM model was able to provide 89.92% of classification accuracy	images were down-samp led to 224×224 pixels before being fed into pre-trained networks, potentially sacrificing some accuracy, especially for subtle findings.
6.	Early Diagnosisof Tuberculosi sUsingDeep Learning Approach for IOT Based Healthcare Application s	Computat ional Intelligen ce and Neurosci ence, 2022	Clustering techniques called AFCM, Dl modelDBN-AMBO,Wie ner filtering for noise reduction,	Dataset from Shenzhen China (SC)	DBN-AMB O was usedto classify TB. Evaluation was done on MC and SC datasets, withSC achievinga higher accuracy of 0.992.	

7.	Deep learning-based automatic detectionof tuberculosis disease in chest X-ray images	polish journal of radiology, 2022	CNN-based transfer learning approach using five different pre-trained models, including Inception_v3, Xception, ResNet50, VGG19, and VGG16 was utilized for classifying TB andnormal cases fromCXR images	Two publicly available datasets of postero-anterior chest radiographs, which are from Montgomery County, Maryland, and Shenzhen, China	ConvNet achieved87.0% accuracy,an AUC of 87%.	Limited data leads to low accuracy ofthe proposed model.

Table 2 Literature survey

CHAPTER 3: SYSTEM DEVELOPMENT

3.1. REQUIREMENT AND ANALYSIS:

Developing a model for the detection of TB using deep learning requires a variety of components such as diverse dataset, libraries for model development, hardware such as Graphic processing unit(GPU),Tensor processing unit(TPU)[9]. For the development of the model we initially require an environment that would be suitable for developing a model using python for that we will be using google colab. Afterwards the dataset that would be used for training and testing of the model would be fetched from Kaggle.As for the web application development and model deployment a python based application named streamlitis used for a hassle free use of model for the end user.

3.1.1. Tools Used:

The tools used for deep learning-based tuberculosis detection from chest X-rays include TensorFlow combined with Keras for neural network model construction and training,Google Colab for code execution, and Kaggle for dataset access. Together, these tools expedite model building, training, and data processing in the quest for precise TB identification from medical imaging.

Google collab:

A Python programming environment provided by the web-based platform Google Colab, will be used as the platform for the model development.

Kaggle Api:

The Kaggle API is a command line tool that facilitates easy access to and interaction with Kaggle datasets. It smoothens the process of importing and using datasets from Kaggle for different data science projects. The tool helps in getting the Kaggle datasets easier to access and combine the dataset more efficiently into the developing project.

TensorFlow:

An open-source library that uses TensorFlow and Keras API to create a user friendly environment for the developer. The library aids in building and training the CNN model that specifically made for the image classification task easier [10].

Keras:

Keras is an open-source library that is based on python that provides an interface for easier development, training and deployment of the deep learning model. Keras being an open-source works easily with TensorFlow, Colab, Microsoft cognitive toolkit[11]. The modular and user friendly architecture of keras is ideal for both seasoned and a newcomer towork on.Major networks such as convolutional, recurrent and dense neural networks are supported additionally the smooth integration with major backend frameworks make keras a popular library used worldwide.

Streamlit:

A python based library that is widely used by machine learning developers to make apps with minimal effort. In the project we have used the streamlit app to deploy our model in the backend. By using streamlit it not only saves our front end development time but also bypass the issues faced by primitive ways of deploying models such as long loading time, connection issues. The platform also provides a way to deploy the model on cloud with minimal fees as compared to other cloud service providers.

3.1.2. Training Resources:

For the development of our model apart from GPU(Graphics Processing Unit) a specialized hardware accelerator TPU(Tensor Processing Unit) is used to make the model training more efficient and fast. TPUs are high throughput devices designed specifically for deep learning tasks by Google. GPUs are also effective at speeding up training because they are generally accessible and versatile.

3.2. PROJECT DESIGN AND ARCHITECTURE:

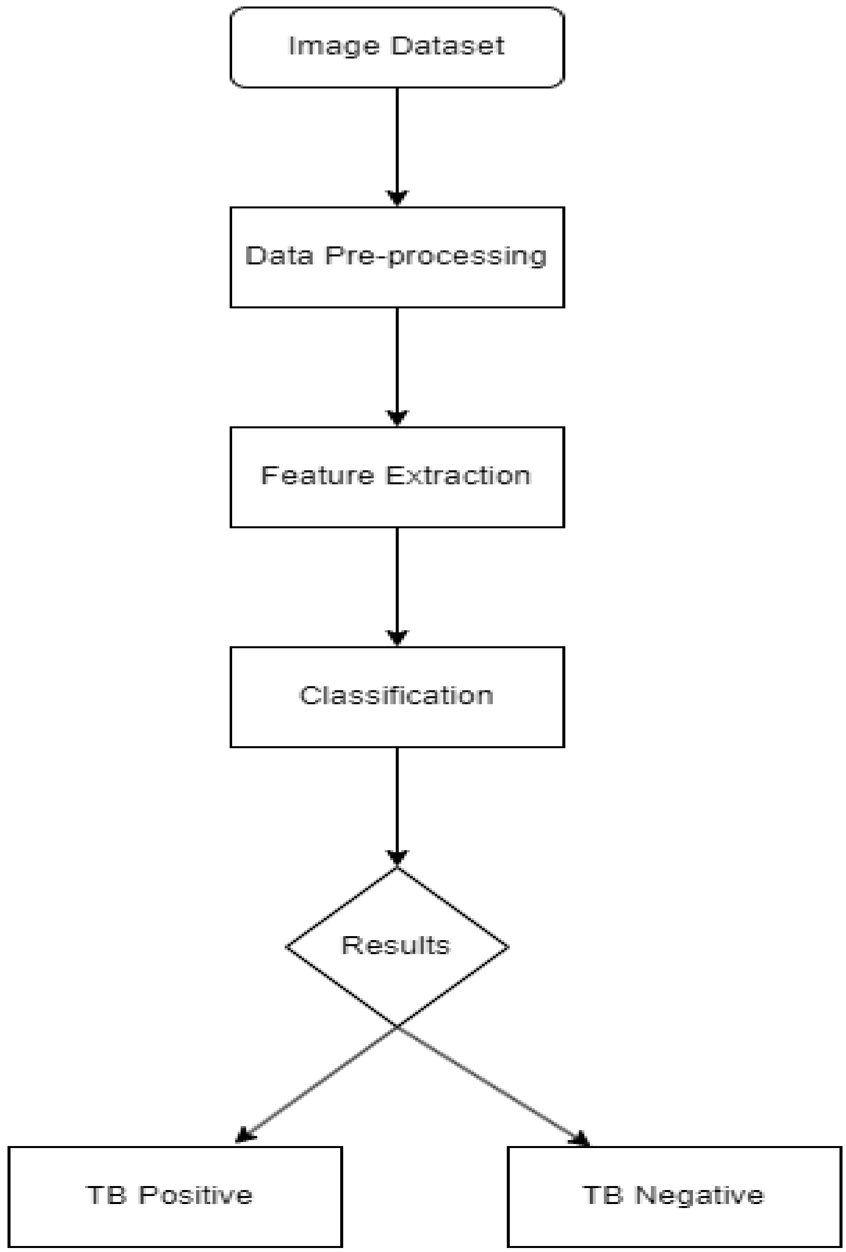

Figure 3.1: Visual representation of working of the model.

Figure 3.1 represents the working of our model that begins with collecting the required dataset followed by preprocessing of the collected data then feature extraction followed by classification and result. A detailed summary of Figure 3.2 is provided below:

3.2.1. Data Collection:

Creating a varied dataset with chest X-ray pictures that have both TB-positive and TB-negative images that were around 7000, in the first step. The dataset is taken from kaggle using the kaggle api which allows data to be directly taken from the kaggle database.

3.2.2. Data Preprocessing:

Preparing images for machine learning requires a number of important steps in the data pretreatment process. As part of this, photos are resized to a standard format to guarantee input dimension uniformity. This preparation improves the dataset's overall quality, which makes machine learning models easier to train and perform at their best[13]. The image size selected is of 500 * 500 with a batch size of 16. For Data augmentation the image data generator is imported from the tensorflow. The image is rescaled to 1/255 and flipped horizontally.

3.2.3. Feature Extraction:

The process of feature extraction involves removing the redundant data that will aid machines in reducing the computational efforts and increasing the learning speed.

3.2.4. Classification:

The steps involve various layers such as convolution layer, pooling layer, flatten layer ,Output layer that will reduce the dimensionality, feature map size and complexity of the CNN model used[14]. CNN model was used for the TB detection as CNN performs better incase of image processing and segmentation. The built in convolution layer reduces the high dimensionality of the image without losing much of the information which makes the CNN a best fit for image processing. Apart from the convolution layer there are other layers also that contribute to the working of the CNN model. Convolution layer provides an feature map that is feeded to pooling layer that reduce the size of the feature map provided by the convolution layer resulting in less computational cost.The next layer is the Flatten layer that reduces the multidimensional array into one dimensional array further reducing the the model.The flatten matrix is fed to the fully connected or the dense layer that will classify the image.The next layer and the last layer is the dense layer or fully connected layer that maps the feature obtained from the previous layer to the final output class. The fully connected layer is simple layer of neurons that receive input from the previous connected layers and using matrix vector multiplication an output is generated.

3.2.5. Results:

The model will provide a result based on the steps involved that will classify whether the patient is TB positive or negative.

3.3. IMPLEMENTATION:

The Model that will detect the TB using CXR images will be implemented using python as programming language on google colab working as the platform[15]. For setting up the environment libraries such as tensorflow, matplot, numpy, kaggle api will be used. CNN will be used initially to build the model for the detection of tuberculosis from deep learning using chest x ray images. The next step is to train the model on the training set of data then evaluate the model on the basis of the training set. The next iteration will be testing the CNNmodel on the x-ray images and evaluating the model on the basis of image classification, accuracy of prediction. The developed model will be deployed on a user friendly interface known as streamlit where user can upload the CXR images and on strreamlit the result of the uploaded CXR images will be delivered.

3.3.1. Importing Libraries:

The python libraries that are necessary for the development of the model will be imported in the first step.

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

Figure 3.2: Importing Libraries

3.3.2. Importing the dataset:

The dataset that we have used in the development of the mode is taken from kaggle using the Kaggle API.

```
! kaggle datasets download -d tawsifurrahman/tuberculosis-tb-chest-xray-dataset
```

```
Downloading tuberculosis-tb-chest-xray-dataset.zip to /content
100% 961M/961M [00:17<00:00, 88.4MB/s]
100% 961M/961M [00:17<00:00, 58.8MB/s]
```

Figure 3.3: Importing dataset

3.3.3. Training and testing of data:

The loaded dataset will be split into training, testing and validation. The ratio which we have selected for the training, testing and validation is 7:2:1 respectively.

```python
train = image_gen.flow_from_directory(
        train_path,
        target_size=(img_height, img_width),
        color_mode='grayscale',
        class_mode='binary',
        batch_size=batch_size
        )

test = test_data_gen.flow_from_directory(
        test_path,
        target_size=(img_height, img_width),
        color_mode='grayscale',
        shuffle=False,
        class_mode='binary',
        batch_size=batch_size
        )

valid = test_data_gen.flow_from_directory(
        valid_path,
        target_size=(img_height, img_width),
        color_mode='grayscale',
        class_mode='binary',
        batch_size=batch_size
        )
```

Figure 3.4: Data splitting into training, testing and validation.

3.3.4. Model Development:

For the detection of TB the model that we have selected to develop in the initial phase in CNN using google colab as the platform.

```
[ ]  cnn = Sequential()
     cnn.add(Conv2D(32, (3, 3), activation="relu", input_shape=(img_width, img_height, 1)))
     cnn.add(MaxPooling2D(pool_size = (2, 2)))
     cnn.add(Conv2D(32, (3, 3), activation="relu", input_shape=(img_width, img_height, 1)))
     cnn.add(MaxPooling2D(pool_size = (2, 2)))
     cnn.add(Conv2D(32, (3, 3), activation="relu", input_shape=(img_width, img_height, 1)))
     cnn.add(MaxPooling2D(pool_size = (2, 2)))
     cnn.add(Conv2D(64, (3, 3), activation="relu", input_shape=(img_width, img_height, 1)))
     cnn.add(MaxPooling2D(pool_size = (2, 2)))
     cnn.add(Conv2D(64, (3, 3), activation="relu", input_shape=(img_width, img_height, 1)))
     cnn.add(MaxPooling2D(pool_size = (2, 2)))
     cnn.add(Flatten())
     cnn.add(Dense(activation = 'relu', units = 128))
     cnn.add(Dense(activation = 'relu', units = 64))
     cnn.add(Dense(activation = 'sigmoid', units = 1))
```

Figure 3.5: Model Development

3.3.5. Model Training:

The model is trained on the imported dataset, 25 epochs are made on the training dataset and below is the attached results that we got.

```
cnn.fit(train,epochs=25, validation_data=valid, class_weight=cw, callbacks=callbacks_list)

Epoch 1/25
307/307 [==============================] - 109s 253ms/step - loss: 0.4736 - accuracy: 0.7789 - val_loss: 0.2
Epoch 2/25
307/307 [==============================] - 75s 244ms/step - loss: 0.2643 - accuracy: 0.8889 - val_loss: 0.15
Epoch 3/25
307/307 [==============================] - 75s 243ms/step - loss: 0.1880 - accuracy: 0.9351 - val_loss: 0.16
Epoch 4/25
307/307 [==============================] - 75s 244ms/step - loss: 0.1377 - accuracy: 0.9444 - val_loss: 0.12
Epoch 5/25
307/307 [==============================] - 75s 244ms/step - loss: 0.1265 - accuracy: 0.9524 - val_loss: 0.09
Epoch 6/25
307/307 [==============================] - 75s 244ms/step - loss: 0.1147 - accuracy: 0.9604 - val_loss: 0.12
Epoch 7/25
307/307 [==============================] - 75s 243ms/step - loss: 0.0736 - accuracy: 0.9766 - val_loss: 0.09
Epoch 8/25
307/307 [==============================] - 74s 241ms/step - loss: 0.0611 - accuracy: 0.9802 - val_loss: 0.05
Epoch 9/25
```

Figure 3.6: Model Training Representation

3.3.6 Model Testing:

```
test_accu = cnn.evaluate(test)
print('The testing accuracy is :',test_accu[1]*100, '%')
```

```
27/27 [==============================] - 5s 172ms/step - loss: 0.0678 - accuracy: 0.9739
The testing accuracy is : 97.38717079162598 %
```

Figure 3.7: Model Testing

3.3.7 Model Evaluation:

The Model will detect and identify tuberculosis cases.

```
test.reset()
x=np.concatenate([test.next()[0] for i in range(test.__len__())])
y=np.concatenate([test.next()[1] for i in range(test.__len__())])
print(x.shape)
print(y.shape)
dic = {0:'NORMAL', 1:'TUBERCULOSIS'}
plt.figure(figsize=(20,20))
for i in range(0+228, 9+228):
  plt.subplot(3, 3, (i-228)+1)
  if preds[i, 0] >= 0.5:
    out = ('{:.2%} probability of being Tuberculosis case'.format(preds[i][0]))


  else:
    out = ('{:.2%} probability of being Normal case'.format(1-preds[i][0]))
  plt.title(out+"\n Actual case : "+ dic.get(y[i]))
  plt.imshow(np.squeeze(x[i]))
  plt.axis('off')
plt.show()
```

```
(421, 500, 500, 1)
(421,)
```

Figure 3.8: Model Evaluation

3.3.8 Stream lit Web application:

```python
import streamlit as st
import numpy as np
from PIL import Image
from tensorflow.keras.models import load_model
from tempfile import NamedTemporaryFile
from tensorflow.keras.preprocessing import image

st.set_option('deprecation.showfileUploaderEncoding', False)

@st.cache(allow_output_mutation=True)
def loading_model():
    fp = r"C:\Users\shrid\Tuberculosis.h5"
    model_loader = load_model(fp)
    return model_loader
```

Figure 3.9: Streamlit webapplication code snippet

3.3.9 Streamlit Interface:

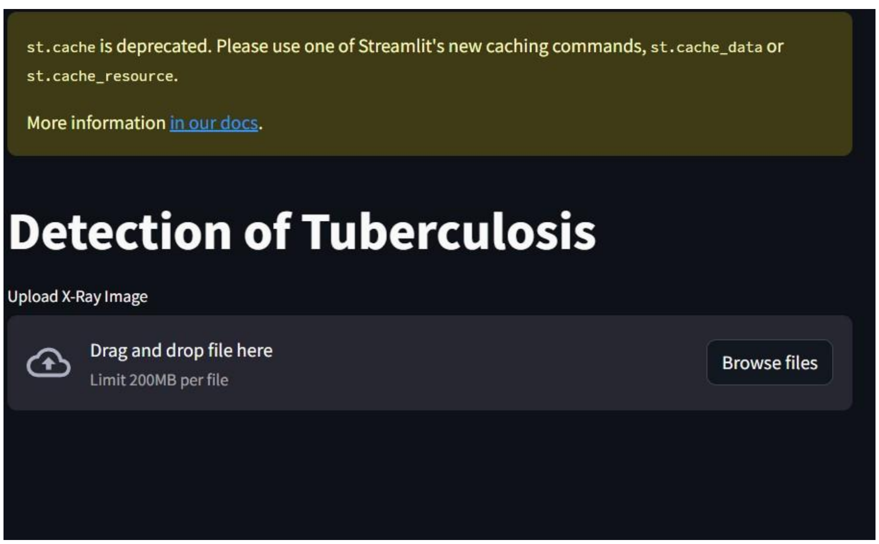

Figure 3.10: Streamlit interface

3.4 CHALLENGES:

Challenges in accessibility of appropriate data, obtaining high model accuracy, developing a user-friendly application of model arise during the development of our tuberculosis detection model. One barrier affecting the robustness of the model is the lack of varied and well-labeled datasets[16]. It is necessary to address infrastructure complexities and user-friendly interfaces in order to deploy the model seamlessly. Continuous refinement ensured the model's accuracy and sensitivity to subtle patterns related to tuberculosis. Overcoming these obstacles is essential to developing a trustworthy and user-friendly instrument for early tuberculosis identification.

3.4.1 Limited and obsolete data:

It was difficult to assemble recently stored data for tuberculosis due to lack of recent and reliable information. However, the datasets that we explored were having old information, noise, and corruption issues that make it difficult to build a resilient training set[17]. This limit may have an effect on how well modern cases are generalized by the machine learning model.

3.4.2 Small Lesions/Identifiers:

The major factor that reduces the accuracy and increases the identification time for the mode are the small hidden lesions and identifiers. Lesion identification could be difficult which is a result of poor lighting conditions , faint damage etc. These issues hurdle the model process by reducing the accuracy due to the reason that small lesions could remain undetected from the model and even sometimes from human invigilation[18]. To tackle this issue what we can do is to take images in good lighting conditions, using good quality x-ray instruments and use algorithms that are immune to such identifiers.

3.4.3 Privacy Concern:

Working with datasets that contain sensitive patient data raises privacy concerns and calls for specialized access rights. It takes a lot of time and typically requires institutional clearance or stringent ethical review committees to obtain these permissions. Moreover it is not an easy task to get hold of datasets that are related to patient information due to privacy concerns and patient approval. These issue primarily affect the development of model training by restricting the access to the data that will be beneficiary for proper model training.

3.4.4 Real World Deployment and Licensing:

For a model that is developed for medical purposes it is crucial to get the model certified from a government testing agency and receive regulatory approval[19]. To get those approval a model must satisfy the strict medical standards that can be achieved by rigorous training of the model on different dataset, tuning the parameters correctly, working alongside the regulatory agency so as to get assurance that the model is able to follow the healthcare guideline and operating procedures. Even if the model gets all the medical certifications the next task is to gain confidence among medical professionals and the patient which is not an easy task to do. Another challenge that the developer may face is regarding the legalities , licencing agreements and third party vendor issue

3.4.5 Creating a user friendly interface:

For the model to present the result it require a interface where the model will be fed with CXR image, to do so a interface is required which not only should be responsive but also user friendly in use. The model will be of no use if the operator is unable to use the interface. To overcome this hurdle streamlit a python based model deployment platform is used which not only provide a platform for the model to be deployed but also offer a user friendly interface.

Chapter 4: TESTING

4.1 TESTING STRATEGY:

Chapter 4 describes the evaluation techniques, tools , libraries that are used to assess the model performance in terms of model performance and accuracy. The testing strategies involve use of testing methods such as unit testing, blackbox testing. Apart from these methods assessment metrices such as accuracy,precision,recall and f1 score are used. In continuation to the methods we have also compared our model with already developed ResUnet.The results obtrained from our model and got from the comparison are detailed below:

4.2 TOOLS USED FOR TESTING:

The tools and libraries that are used in our model testing phase that will be used to improve and optimize our model are described below:

4.2.1 TensorFlow and Keras:

The widely used built-in library provides a detailed framework for developing, training,testing a neural network . These libraries reduce steps for tasks like designing sophisticated architectures and determining model performance

4.2.2 Scikit-Learn:

Scikit-Learn library is made for in depth model evaluation. The library uses a variety of metrics, such as cross-validation , accuracy, precision, and recall. It improves the evaluation process for the user by allowing user to obtain useful information about the working of the model .

4.2.3 Matplotlib:

Python package that is used in our model that provides visual information in form of graphs like ROC curves and confusion matrices[20]. The library allow graphical evaluation through a user understandable interface.

4.2.4 Unittest:

A python library for testing that allow to execute unit test for the model which involve testing individual component within the model which ensure that each component behaves as expected and results a desired output.

4.2.5 Blackbox testing:

The testing strategy involve evaluating the functionality of the system, the procedure highlights only the inputs and outputs of the system rather than its internal working. For the blackbox testing of our system both the platform and model is tested separately.

4.2.6 Comparison of model with ResUnet model.

Apart from the objective to create a model and deploying a TB detection model our other objective is to compare our model with an existing model so that accuracy of the model can be compared and tweaked so as to perform as par or better with the ResUnet model.

CHAPTER 5: RESULT AND EVALUATION

5.1 RESULTS:

The results obtained from the testing of our system are detailed in this chapter with the supporting screenshots and comparison table. During testing the issues tat we faced were rectified that resulted in testing accuracy of 97.14 % onthe dataset obtained from the Kaggle. Moreover the developed model is compared with an existing ResUnet model and changes were made so that the model performs as par with the compared model.The satisfactory testing accuracy provided the confidence that the model will be able to aid the healthcare professionals and it is fit to be deployed on streamlit where the predictions can be made. Below are the attached screenshotsof the results obtained during the development of model and on the deployed platform.

5.1.1 Test cases and outcome:

Testing accuracy:

Accuracy of 97.14 % obtained in the testing phase of our model.

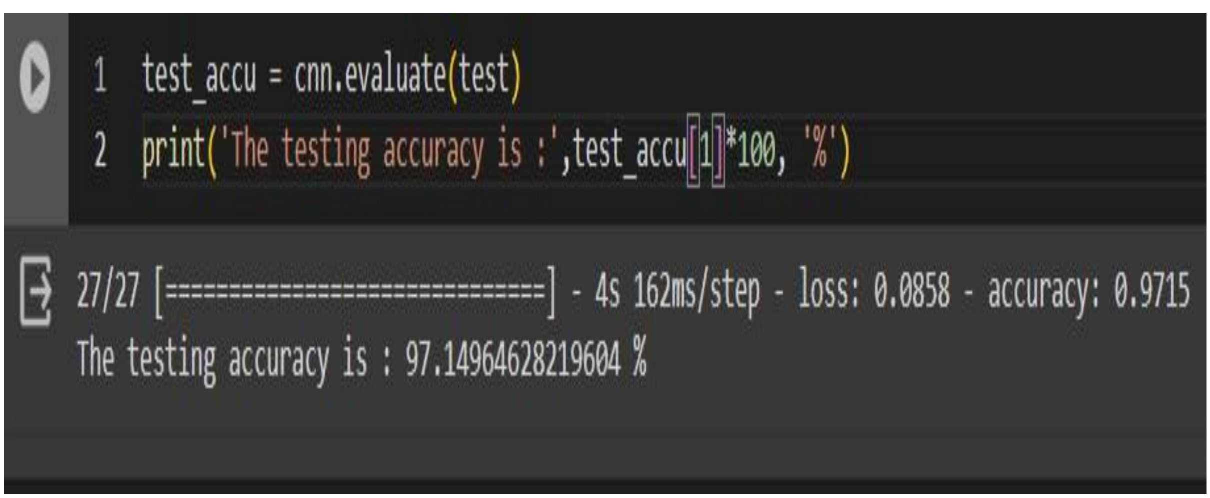

Figure 5.1: Test Accuracy snippet

Unittest result:

The result obtained from unittest when the model is feed with CXR images of both normal and tuberculosis infected one is provided below in form of screen shot.

```
import unittest
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
class TestTuberculosisDetection(unittest.TestCase):
    def setUp(self):
        self.model = load_model('/content/Tuberculosis.h5')
        self.test_normal_image = image.load_img('/content/Normal-185.png', target_size=(150, 150))
        self.test_tb_image = image.load_img('/content/Tuberculosis-9.png', target_size=(150, 150))
    def test_model_accuracy(self):
        normal_image_array = image.img_to_array(self.test_normal_image)
        normal_image_array = np.expand_dims(normal_image_array, axis=0) / 255.0
        tb_image_array = image.img_to_array(self.test_tb_image)
        tb_image_array = np.expand_dims(tb_image_array, axis=0) / 255.0
        normal_pred = self.model.predict(normal_image_array)
        tb_pred = self.model.predict(tb_image_array)
        threshold = 0.5
        normal_class = 1 if normal_pred[0][0] < threshold else 1
        tb_class = 0 if tb_pred[0][0] >= threshold else 0
        self.assertEqual(normal_class, 1)
        self.assertEqual(tb_class, 0)
    def test_prediction_consistency(self):
        normal_image_array = image.img_to_array(self.test_normal_image)
        normal_image_array = np.expand_dims(normal_image_array, axis=0) / 255.0
        first_pred = self.model.predict(normal_image_array)
        second_pred = self.model.predict(normal_image_array)
        self.assertAlmostEqual(first_pred[0][0], second_pred[0][0], delta=0.01)
if __name__ == '__main__':
    unittest.main(argv=[''], exit=False)
```

```
1/1 [==============================] - 0s 126ms/step
1/1 [==============================] - 0s 35ms/step
.1/1 [==============================] - 0s 125ms/step
1/1 [==============================] - 0s 34ms/step
.
----------------------------------------------------------------------
Ran 2 tests in 1.122s

OK
```

Figure 5.2 Unittest result

Blackbox testing:

Checking whether the streamlit platform is working or not, the streamlit run web.py command line is used to check whether the platform is accessible or not. On entering the command line the cmd prompts that the platform is accessible and opens a new tab on browser with streamlit that states the app is working fine.

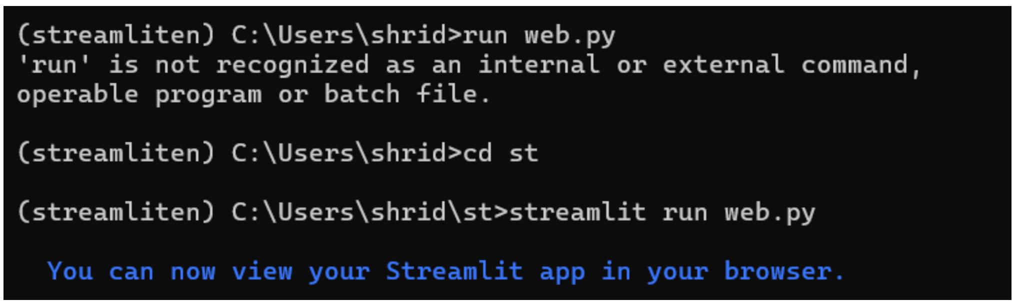

Figure 5.3 Streamlit access prompt

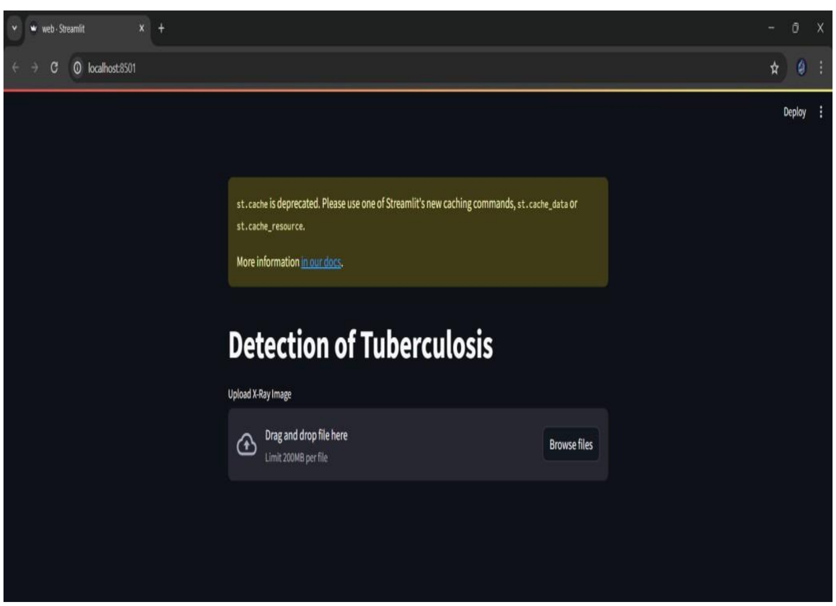

Figure 5.4 Streamlit webpage

Now for the testing accuracy of model running on the streamlit web page two CXR images normal and tuberculosis infected one are uploaded and the result is compared against the uploaded images.

On uploading a normal chest x ray image the model predicts the disease and in this case the model predicted accurately that the image is normal.

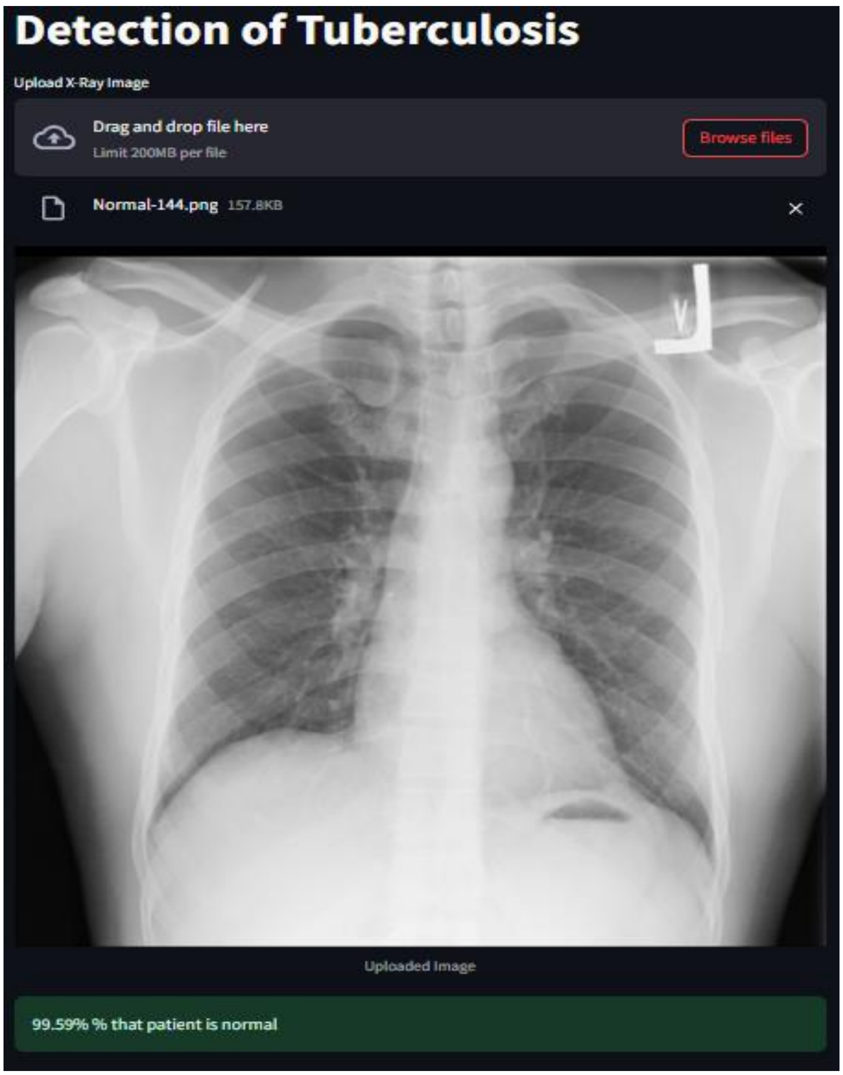

Figure 5.5: Result of normal image

On uploading a tuberculosis infected chest x ray image the model predicts result that the chest x ray image is infected with TB that satisfies the accuracy. Below is the attached screenshot of the result obtained.

Figure 5.6: Result of TB image

Model comparison results:

The comparison of our model and the ResUnet model are done on the basis of matrices such as accuracy graph confusion matrix and classification report.The results that we obtained are provided below with the attached scrrenshots of result and classification report table.

Validation and Accuracy graph comparison:

accuracy obtained from CNN model

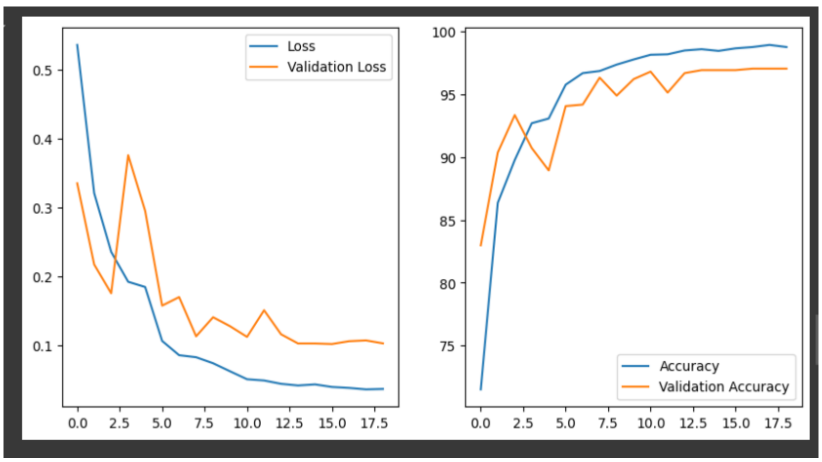

Figure 5.7: Accuracy graph of CNN model

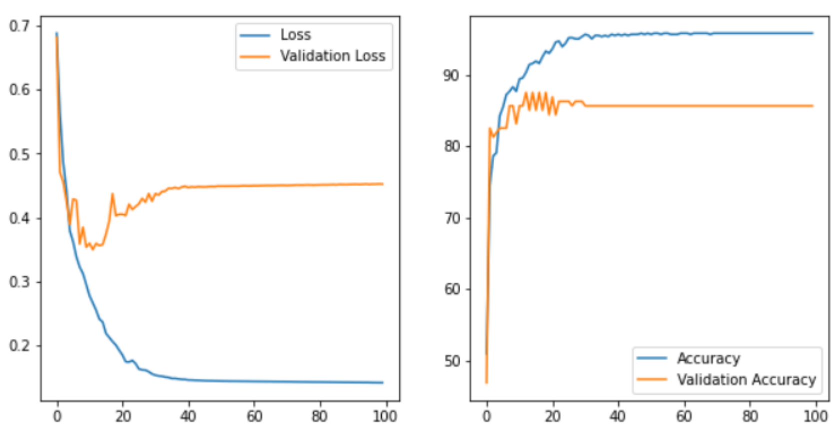

Figure 5.8: Accuracy graph of ResUnet model

Comparison of confusion matrix:

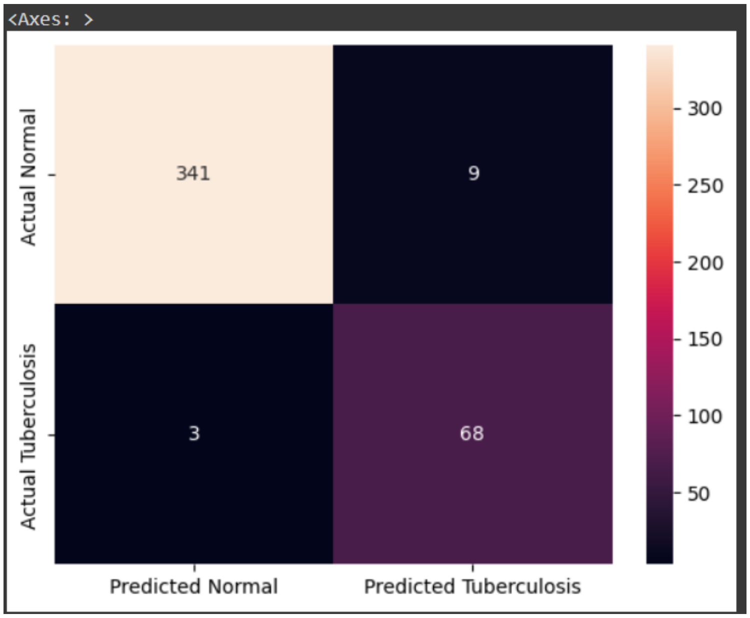

Figure 5.9 Confusion matrix of CNN model.

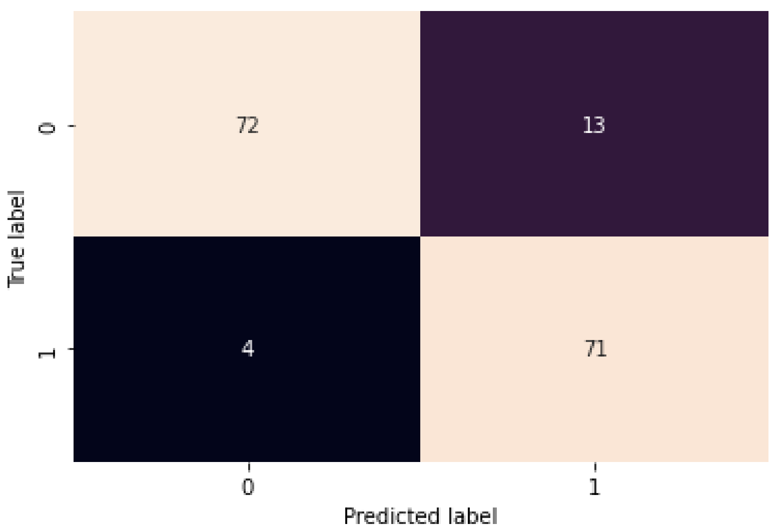

Figure 5.10: Confusion matrix of ResUnet model

Classification report:

The classification report generated from scikit-learn provide statistics for two classes normal and tuberculosis. Precision, recall, F1 score and overall accuracy of the two classes are obtained. Below are the obtained classification reports obtained from the CNN model and the ResUnet model in form of table:

	Precision	recall	F1- score	support
NORMAL	0.99	0.97	0.98	350
TUBERCULOSIS	0.99	0.96	0.92	71
accuracy			0.97	421
Macro avg	0.94	0.97	0.95	421
Weighted avg	0.97	0.97	0.97	421

Table 5.1: Classification report of CNN model

	Precision	recall	F1- score	support
NORMAL	0.93	0.82	0.87	85
TUBERCULOSIS	0.82	0.93	0.87	75
accuracy			0.88	160
Macro avg	0.88	0.88	0.87	160
Weighted avg	0.88	0.88	0.87	160

Table 5.2: Classification report of ResUnet model

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION:

The proposed model that is based on deep learning provides a medium for an early and accurate detection of TB that will aid in limiting the spread of this contagious disease and reducing the death tolls caused by TB. For us to make the idea a reality it was important to consider and tackle issues such as acquiring dataset, ethics and privacy concerns.We had developed the model by keeping all these issues in mind and had tried not to hamper any of these ethics issues. For the dataset we decide to get it from a trustworthy site that is kaggle so that we won't face any privacy concerns in future. to get the model as close to our ambition we tried to make the model not in a rush but give our best so that it works perfectlywell.

6.1.1 Key Findings:

For a model to detect the TB identifiers correctly it was important to train the model on a large dataset that can only be possible using a deep learning approach, by following this method we were able to harness impressive accuracy and reliability from the model developed. For image processing CNN was selected as these neural network works best in extracting features and are able to capture fine structural details that are identifiers for TB. Even though deep learning is best for making a model that would be trained using huge dataset there's always a room for improvement which we will incorporate in the coming months. Apart from the model development hiccups it was important to make the model ethically correct that was achievable by maintaining the privacy concern , not using any unauthorized data. Another add on would be the rapid shifts in disease traits and technological advancements, that include difficulty in generalizing different populations to detect small lesions in the early stage.

6.1.2 Limitations:

Limitations that we face in developing a model using a deep learning approach is the strong point of deep learning that is being able to handle large datasets. The main issue regarding a huge dataset is the data quality and scarcity of appropriately labeled dataset. Small lesions are still hard to identify early that hampers the model sensitivity particularly in early stage of disease. In view of rapid change in disease characteristics,imaging technologies the model and field require constant monitoring and adjustment. The use of medical data appear as the big rock for the model development as use of medical data is strictly adhered to guidelines and not easy to access. The data that is accessible is either obsolete , corrupted or limited. This raises a concern in accuracy in model training and overall model results.

6.1.3 Contributions to the field:

The use of deep learning focuses on making a crucial and remarkable contribution in the medical field.In case of Tb detection from CXR images the ability of the model to process and interpret large data easily aids in improving diagnostic speed and accuracy[21]. As a subsidiary instrument it optimizes the healthcare process, relieving the healthcare professionals from the CXR screening strain. Evenmore the deep learning approach improves the overall screening of CXR images by providing better insights into disease patterns and the ability to adapt in a changing healthcare environment. Being a collaborative field it gives rise to opportunities for collaboration between data scientists , healthcare professionals, and patients that can lead to innovative solutions.

6.2 FUTURE SCOPE:

Our intention is to further improve the accuracy of our proposed model and bridging the gap between human and a machine .To ensure that our model fits all the medical regulations we plan to compare our best fit model with other existing model.Once the model is selected and find fit to use we will incorporate the model with a web based application with main aim of a user friendly interface. We decided to incorporate the model with a web application so that the model can be publicly accessible not only restricted to hospitals only.This method will be more beneficial in tier 3 cities where medical assistance and access is limited.By bridging the gap between a complex technology and common user this user friendly interface intends to make TB detection less time consuming and approachable.By doing so not only we are aiding in accessibility of this technology but also reducing the time in getting the TB results. We see this as a collaboration between modern technology and the human aspect of medical professionals, working in tandem to improve the diagnostic procedure as a whole. Essentially, we want to make the process of TB detection for people as easy as possible by combining cutting-edge technology with humane care. We think we can greatly impact earlydiagnosis and efficient treatment of tuberculosis, improve healthcare outcomes, and positively impact lives through integrating the capabilities of the best deep learning models with user-friendly interfaces while collaborating with healthcare experts.

REFERENCES

[1] https://www.who.int/health-topics/tuberculosis#tab=tab_1.

[2] A. S. Becker, C. Blüthgen, and V. D. Phi van, "Detection of tuberculosis patterns in digital photographs of chest X-ray images using Deep Learning," in International Union Against Tuberculosis and Lung Disease, vol. 22, 2018.

[3] T. Rahman, Amith Khandakar, Rashid Mazhar, and Tahir Hamid . "Tuberculosis Detection Using Chest X-Ray With Deep Learning," Segmentation and Visualization,vol. 8,2018.

[4] S. Stirenko, Yuriy Kochura, Oleg Alienin, and Oleksandr Rokovyi. "Chest X-RayAnalysis of Tuberculosis by Deep Learning with Segmentation and Augmentation,"*2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO), Kyiv, Ukraine*, 2020.

[5] R. Purkar, P., O'Connell, C., Schechter, "CheXaid: deep learning assistance for physician diagnosis of tuberculosis using chest x-rays in patients with HIV",2020.

[6] V. Kumar, G. Simi Margaret, G. Hemalatha, Annapurna Mishra, and H. Shaheen "Early Diagnosis of Tuberculosis Using Deep Learning Approach for IOT Based Healthcare Applications", Volume 2022,Hindawi,2022.

[7] K. Munadi, K. Muchtar, N. Maulina and B. Pradhan, "Image Enhancement forTuberculosis Detection Using Deep Learning," *in IEEE Access*, vol. 8, 2020.

[8] V. Kumar ,S. Margarat, G. Hemalatha, Annapurna Mishra, and H. Shaheen "Early Diagnosis of Tuberculosis Using Deep Learning Approach for IOT Based Healthcare Applications", Volume 2022.

[9] S. Sathitratanacheewin, P Sunanta, K Pongpirul. “Deep learning for automated classification of tuberculosis-related chest X-Ray: dataset distribution shift limits diagnostic performance generalizability.” 6: e04614. *Heliyon* 2020;

[10] C. Silverman. “An appraisal of the contribution of mass radiography in thediscovery of pulmonary tuberculosis.” 60: 466-482.Am Rev Tuberc 2022.

[11] AH. van’t Hoog, HK Meme, KF Laserson, A Janet. Agaya, Benson G. Muchiri, Willi A. Githui, and Lazarus O. Odeny, “Screening strategies for tuberculosis prevalence surveys: the value of chest radiography and symptoms.” 7: e38691. PLoS One 2012.

[12] F. Ahmad Khan, T Pande, B Tessema, Belay Tessema, Rinn Song, Claudia M. Denkinger, “Computer-aided reading of tuberculosis chest radiography: moving the research agenda forward to inform policy.” 50: 1700953 Eur Respir J 2017.

[13] S. Graham, GK Das, RJ Hidvegi, J Kosiuk, , K Al Zahrani,., and D Menzies... “Chest radiograph abnormalities associated with tuberculosis: reproducibility and yield of active cases.” 6: 137-142. Int J Tuberc Lung Dis 2002.

[14] R. Hooda, A. Mittal, and S. Sofat. “Segmentation of lung fields from chest radiographs-a radiomic feature-based approach.” ; 9: 109-117, Biomed Eng Lett 2018.

[15] S. Jaeger, A. Karargyris, S. Candemir, L Folio, J Siegelman, and F Callaghan.. “Automatic tuberculosis screening using chest radiographs.” 33: 233-245, IEEE Trans Med Imaging 2014.

[16] A. Hosny, C. Parmar, J. Quackenbush, LH. Schwartz, HJWL. Aerts. “Artificial intelligence in radiology.” 18: 500-510 Nat Rev Cancer 2018.

[17] T. Rahman, ME. Chowdhury, and A. Khandakar, “Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection using Chest X-ray.” 10: 3233, Appl Sci 2020.

[18] R. Mohammadi, M. Salehi, and H. Ghaffari, “Transfer Learning-Based Automatic Detection of Coronavirus Disease 2019 (COVID-19) from Chest X-ray Images.” 10: 559-568 J Biomed Phys Eng 2020.

[19] G. Choy, O. Khalilzadeh, and M. Michalski “Current Applications and Future Impact of Machine Learning in Radiology.” 288: 318-328, Radiology 2018.

[20] S. Do, KD. Song, JW. Chung. “Basics of Deep Learning: A Radiologist’s Guide to Understanding Published Radiology Articles on Deep Learning.” Korean J 2020.

APPENDIX

USER MANUAL: Use these steps to access the streamlit application:

1. Accessing Streamlit:

To access the streamlit web-application copy the activation path in the command prompt. After activating the path change the directory to streamlit. When the directory is changed use the command "streamlit run web.py" to run the web application.

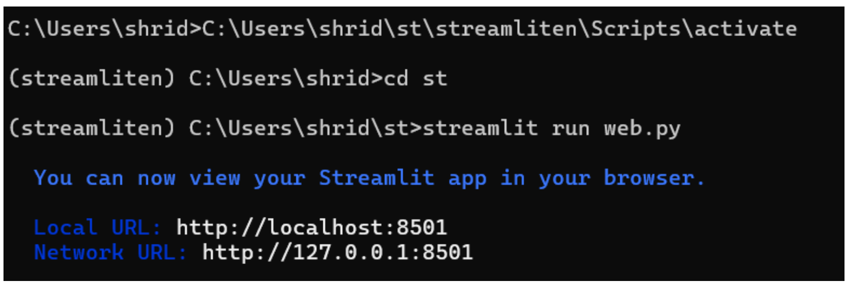

Figure a : Accessing Streamlit

2. Streamlit Interface:

After accessing the web-application an interface will appear in which you have to upload the image of the chest x-ray for further detection.

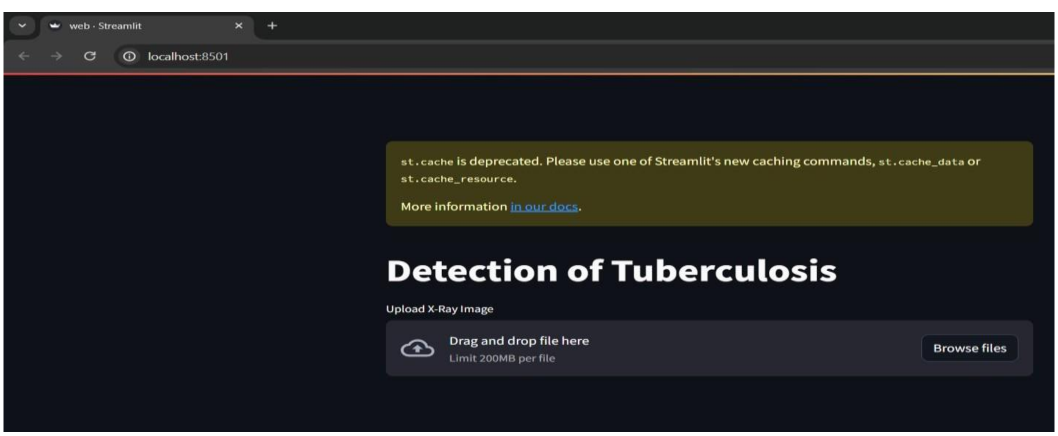

Figure b : Streamlit Interface

3. Uploading the chest x-ray:

To upload the chest x-ray click on the browse file, then upload the image of the chest x-ray.

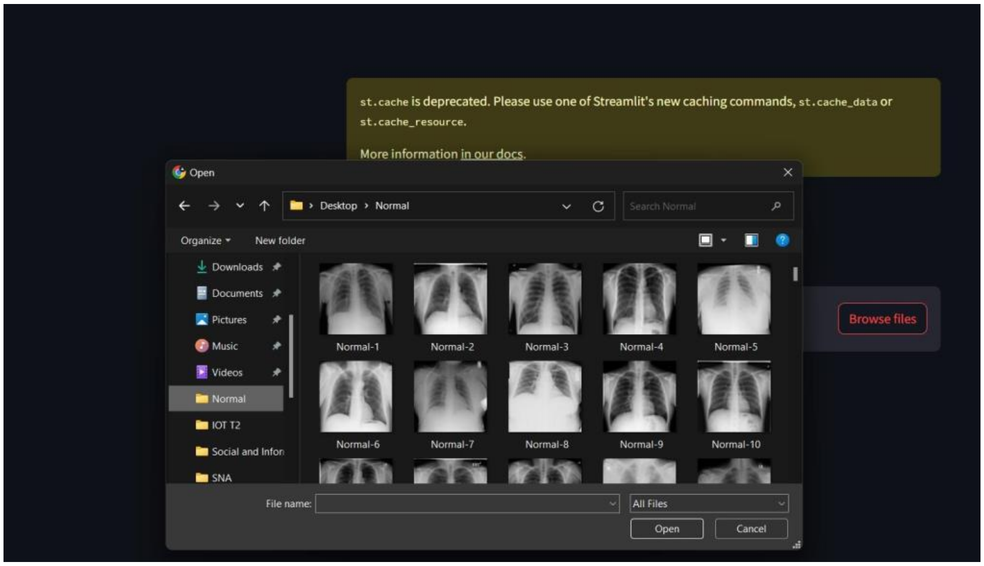

Figure c : Uploading the chest x-ray image

4. Result:

The results of the uploaded chest x ray image will be displayed below the image with detection percentage.

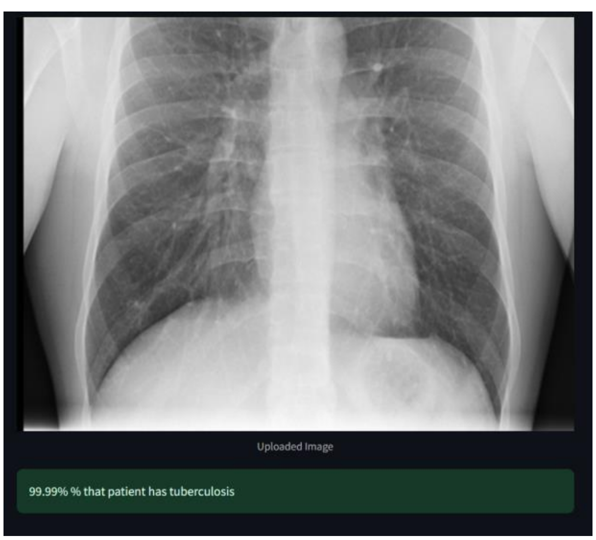

Figure d: Results of the uploaded image.

r

ORIGINALITY REPORT

14%	12%	11%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.polradiol.com Internet Source	1%
2	opus.lib.uts.edu.au Internet Source	1%
3	researchspace.ukzn.ac.za Internet Source	1%
4	www.researchgate.net Internet Source	1%
5	Avala Raji Reddy, Ch Narendar, K Srinu, K Venkateswara Rao, R Venkateswara Reddy, P Sruthi. "COVID 19 Patient Recognition and Prevention Utilizing Machine Learning and CNN Model Techniques", 2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA), 2023 Publication	<1%
6	Submitted to University of North Texas Student Paper	<1%

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ……………………….

Type of Document (Tick): **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report** **Paper**

Name: ___________________________ __**Department:** _________________ **Enrolment No** _________

Contact No. ______________________________**E-mail.** ______________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ________________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

– Total No. of Pages =
– Total No. of Preliminary pages =
– Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……………….(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><th>Copy Received on</th><th>Excluded</th><th>Similarity Index (%)</th><th colspan="2">Generated Plagiarism Report Details
(Title, Abstract & Chapters)</th></tr>
<tr><td></td><td rowspan="4">• All Preliminary Pages
• Bibliography/Images/Quotes
• 14 Words String</td><td rowspan="2"></td><td>Word Counts</td><td></td></tr>
<tr><td>Report Generated on</td><td>Character Counts</td><td></td></tr>
<tr><td rowspan="2"></td><td>Submission ID</td><td>Total Pages Scanned</td><td></td></tr>
<tr><td></td><td>File Size</td><td></td></tr>
</table>

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com