WEB BASED APPLICATION TO DETECT TUBERCULOSIS USING CXR

By:

GC M. Shehryar

GC Alamgir Hasni

GC Mudassir Ahmed

GC Zeeshan Saqib

Supervised by:

Asst Prof. Dr. Nauman Ali

A thesis submitted to the faculty of Computer Software Engineering Department, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan in partial fulfillment of the requirements of the degree Bachelors in (Software) Engineering.

In the name of ALLAH, the Most benevolent, the Most Courteous

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

"WEB BASED APPLICATION TO DETECT TUBERCULOSIS USING CXR"

is carried out by

GC M. Shehryar

GC Alamgir Hasni

GC Mudasir Ahmed

GC Zeeshan Saqib

under my supervision and that in my judgment, it is fully ample, in scope and excellence, for the degree of Bachelor of Software Engineering at Military College of Signals, National University of Sciences and Technology (NUST), Islamabad.

Supervisor Name: **Asst Prof Dr. Nauman Ali**

Supervisor Signature:_________________

Date: _______________

Place: ______________

DECLARATION OF ORIGINALITY

We hereby declare that no portion of the work presented in this Thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

ACKNOWLEDGEMENTS

Allah Subhan'Wa'Tala is the sole guidance in all domains.

Our parents, colleagues, and most of all our supervisor, Asst Prof. Dr Nauman Ali

Without

his guidance this wouldn't be possible.

And all the group members, who through all adversities worked steadfastly.

PLAGIARISM CERTIFICATE (Turnitin Report)

This thesis has______________similarity index. Turnitin report endorsed by Supervisor is attached.

____

GC M. Shehryar
325046

____

GC Alamgir Hasni
325024

____

GC Mudasir Ahmed

325033

____

GC Zeeshan Saqib
325017

Signature of Supervisor

ABSTRACT

Tuberculosis is a very infectious respiratory disease and is currently the leading cause of mortality worldwide, ranking higher than both malaria and HIV/AIDS. As a result, it is vital to promptly diagnose TB to limit its transmission, enhance preventative measures, and reduce the mortality rate associated with the disease. Various procedures and tools have been employed to diagnose TB early, practically all of which needed a visit to the doctor and were not available to the public. This work presents an automated and accurate approach for diagnosing TB that may be used by the general population and does not require special imaging equipment or conditions. An application will be developed for the detection of TB using CXRs and deep learning techniques. The application will use a convolutional neural network (CNN) to classify CXRs as normal or indicative of TB. The CNN will be trained on dataset of annotated CXRs to learn the relevant features for TB detection.

Table of Contents

LIST OF FIGURES

Chapter 1

INTRODUCTION

Deep learning is a type of artificial neural network (ANN) that employs mathematical approaches to create a structure that resembles the human brain. On the other hand, technological advancements have resulted in algorithms for optimizing regular neural networks so that the number of neural layers may be counted. Neural networks have grown from multiple layers to hundreds of layers with thousands of neurons in each layer, which could not have been achieved until recently. A deep learning network is a specific form of neural network. For Tuberculosis diagnosis, several strategies involving various types of ANNs and other machine learning techniques have been introduced. Li et al., for example, proposed a new data synthesis methodology that combines individual's CXRs photos with significantly processed data. The research team used a convolutional neural network (CNN) to outperform traditional detection and tracking methods. Furthermore, the system was taught using simple criteria by people.

1.1 Overview

Tuberculosis (TB) is a highly infectious disease caused by the bacterium Mycobacterium tuberculosis. It primarily affects the lungs, but can also infect other parts of the body such as the kidneys, bones, and brain. TB remains a significant public health challenge globally, with an estimated 10 million cases and 1.4 million deaths reported in 2019. Early diagnosis and treatment are critical to prevent transmission and reduce morbidity and mortality.

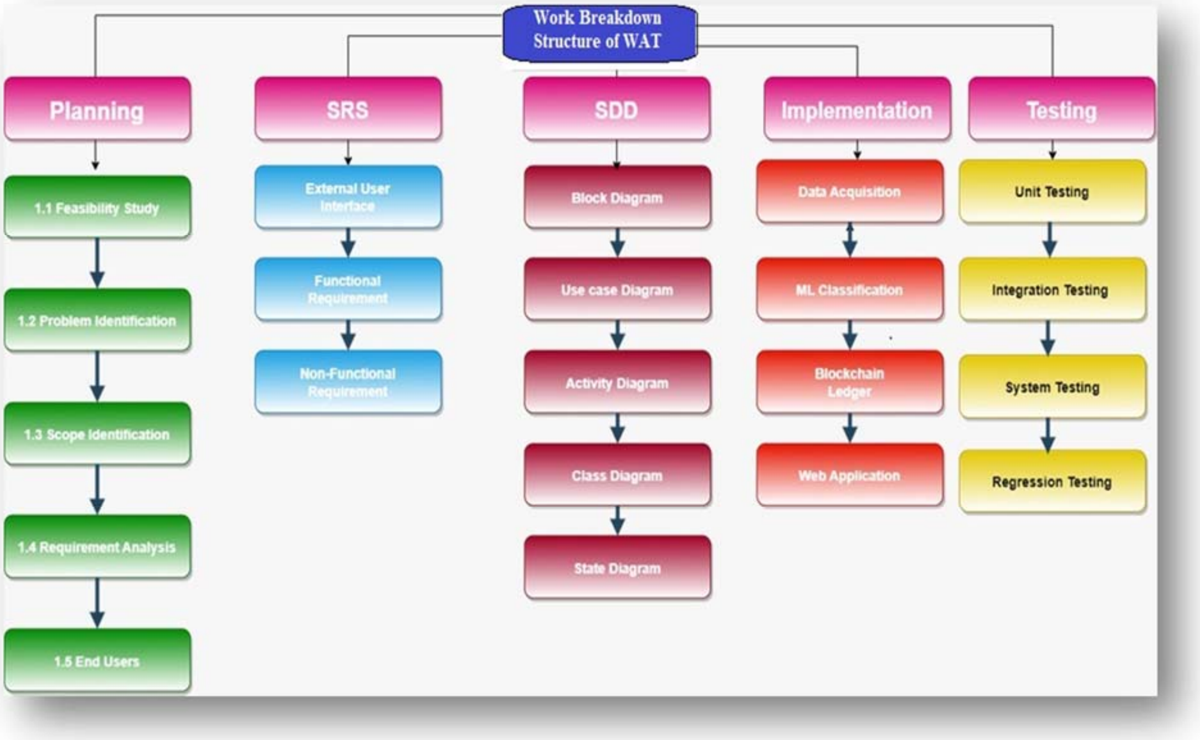

Figure 1: Work Breakdown Structure

1.2 Novelty and Challenges

Tuberculosis (TB) remains a major public health challenge globally, with an estimated 10 million cases and 1.4 million deaths reported in 2019. Early diagnosis and treatment are critical to prevent transmission and reduce morbidity and mortality. However, TB diagnosis can be challenging, especially in resource-limited settings where access to trained radiologists and laboratory facilities is limited. Chest X-rays (CXRs) are commonly used for TB screening and diagnosis, but interpreting CXRs can be subjective and time-consuming. There is a need for accurate and efficient TB detection methods to improve diagnosis and treatment outcomes.

1.3 Proposed Solution

Machine learning techniques, particularly deep learning, have shown promise in improving TB detection accuracy and efficiency. However, the performance of these methods may be affected by various factors, such as the quality of the CXRs, the size and diversity of the training dataset, and the presence of other comorbidities that may affect the CXR interpretation. Therefore, the ultimate goal is to develop an accurate and efficient TB detection method using CXRs and deep learning techniques that can overcome the limitations of current methods and improve TB diagnosis and treatment outcomes.

1.4 Framework

The study focuses on image processing concepts combined with deep learning methods. The project is organized into various modules, each of which is intertwined with the next. The following is a list of modules:

- Datasets and annotations
- Dataset training and processing
- Output extraction
- Decision based upon Output
- Integration
- GUI presentation

1.4.1 Datasets and annotations:

The preparation of datasets is an important aspect of the process. The collection includes photos of Normal Chest X-Rays and Affected Chest X-Rays with Tuberculosis.

1.4.2 Dataset training and processing:

The prepared dataset is used as input to train object detection models using deep learning.

1.4.3 CNN algorithm:

The acronym CNN stands for "Convolutional Neural Network." This is a method of detecting and distinguishing various objects in an image (in real-time). Object detection in CNN is approached as a classification problem, with class probabilities for the detected images provided. In our project, we use the CNN (Convolutional Neural Network) technique to train the dataset, which is then used to develop the object detection model.

1.4.4 Output Extraction:

The output extraction would involve determining whether the input CXR image is indicative of TB or not. The deep learning model would output a binary classification, indicating whether the CXR is normal or abnormal.

1.4.5 Decision based upon Outputs:

The outputs retrieved from Chest X-Rays is employed in decision-making

1.4.6 Integration:

The many modules are then combined into a single standalone entity. For a compact solution, this stand-alone entity is required.

1.4.7 GUI presentation:

The visual demonstration of the project is done through the aid of GUI (graphical user interface).

1.4.8 Python-Django Framework:

Django is a Python framework for developing online apps and graphical user interfaces, which is used in this project.

1.4.9 General Objectives:

"To develop cutting-edge software based on Deep Learning (DL) and Internet Protocol (IP) techniques that will serve as a smart administrative tool for reducing the Tuberculosis Cases."

1.4.10 Academic Objectives:

- Development of a smart and intelligent Machine System
- To implement Deep Learning techniques and simulate the results.
- To decrease disease by working in a team
- To design a project that contributes to the General hospitals.

1.5 Scope

The scope for improvement in TB detection using deep learning techniques and CXRs is significant, and several areas of research and development can be pursued to enhance the accuracy and efficiency of TB detection. One area of improvement is to train the deep learning models using even larger datasets of CXRs, including diverse populations and a broad range of TB disease stages. This will enable the model to learn from a wider variety of CXR images and improve its ability to detect TB accurately. Another critical area for improvement is to have access to individuals with possible TB cases, as this will enable the model to learn from real-world scenarios and improve its ability to detect TB in a clinical setting. In addition, testing the model under various environmental conditions, such as low light or poor image quality, can help to assess its robustness and performance in challenging conditions. To make the software user-friendly and easy-to-use for the end-user, the design of the graphical user interface (GUI) is essential. The GUI should be designed to be intuitive and straightforward, with clear instructions and visualizations to assist the user in understanding the TB detection results. Moreover, the GUI should have the flexibility to accommodate different users with varying levels of technical expertise and different language capabilities. The development of the TB detection software should also consider the ethical and legal implications of using deep learning models for medical

diagnostics. The model should comply with data privacy regulations and maintain confidentiality of patient data. Additionally, the model should be transparent in its decision-making process and provide clear explanations for its classification decisions. The democratization of such medical diagnostic systems is critical to improving access to healthcare for underserved populations. The software should be designed to be scalable, cost-effective, and accessible to healthcare providers globally. This can be achieved through the development of cloud-based solutions or open-source software, which can be deployed in resource-limited settings without requiring significant infrastructure or hardware investment. The scope for improvement in TB detection using deep learning techniques and CXRs is vast, with several areas of research and development to enhance the accuracy, efficiency, and accessibility of TB diagnosis. The development of user-friendly and transparent software is critical to democratizing healthcare and improving the outcomes for TB patients worldwide.

1.6 Deliverables

Software Requirement Specification:

The purpose of this document is to present a detailed description of our Web-based Application for detecting Tuberculosis through CXRs. It will explain the purpose and features of the system, the interfaces of the system, what the systemwill do, its entire process, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the users.

Software Architecture Document:

In this document, the overall architecture of the system is discussed, including the introduction of various components and subsystems. It is mainly supported by system Architecture diagram which shows an insider's perspective of the system by describing thehigh-level software components that perform the major functions to make the system operational.

Software Design Document:

The design document captures all our functional requirements and shows how they interactwith each other conceptually. The low-level design also shows as to how we have been implementing how we are going to implement all of these requirements.

Implementation Code Document:

The implementation code document provides details about the pseudo code for the application and project prototype.

Software Testing Document:

This document has testing modules in which there are certain test cases which depicts th**e** correctness and accuracy of the project.

Final Project Report

This is the thesis report which compiles all the previous and current working for the project.Thesis report provides the whole summary for the project and also give details about each and every aspect of the project starting from the introduction of the project, literature review, requirements leading to design discussions then testing and lastly future work and conclusion.

.

1.7 Structure of Thesis

In summary, the thesis breakdown is as follows:

Chapter 2: Literature Review discusses the popular existing SE Schemes. Their comparisons and architectural models already existing relevant to the project's domain.

Chapter 3: Proposed Scheme presents the risks to our system and the scheme derived to tackle. It also presents the system specifications and working algorithm of our project.

Chapter 4: The methods, approaches, tools, techniques, algorithms, or other aspects of the solution

are sufficiently discussed with sufficient details and supporting figures. It also discusses our target market, Operating environment and some constraints, assumptions and dependencies.

Chapter 5: System Design presents system architectural design and other relevant UML diagrams to explain the design of the system.

Chapter 6: System Implementation discusses how the system is actually implemented. It includes pseudo code that presents the working of APIs, Demo outputs of APIs and User interface of the application and dashboard.

Chapter 7: A cognitive evaluation of the system using the test results. System testing is performed through a viable and relatable testing strategy to cover all the use cases of application.

Chapter 8: Future Direction disuses the possible directions available to grow the specifications.

Chapter 2

LITERATURE REVIEW

Between 2000 and 2018, statistics indicate that timely diagnosis and treatment of tuberculosis have the potential to save up to 58 million lives. Therefore, prompt detection of TB is crucial in reducing its transmission, enhancing preventive measures, and decreasing mortality rates associated with TB.

- Background
- Existing solutions and their drawbacks
- Research Papers

2.1 Background

Tuberculosis (TB) is a contagious infection caused by the bacterium Mycobacterium tuberculosis. It primarily affects the lungs but can also spread to other parts of the body. TB has been a significant public health concern for centuries and is one of the top 10 causes of death worldwide TB is transmitted through the air when an infected person coughs, sneezes, or speaks, and another person inhales the bacteria. TB is more prevalent in low and middle-income countries, where factors such as poor living conditions, malnutrition, and limited access to healthcare contribute to the spread of the disease. The development of antibiotics such as streptomycin and rifampin in the mid-20th century led to effective treatments for TB. However, the emergence of drug-resistant strains of TB, particularly multidrug-resistant TB (MDR-TB) and extensively drug-resistant TB (XDR-TB), has presented a new challenge in the fight against TB. The World Health Organization (WHO) estimates that approximately 10 million people developed TB in 2019, and 1.4 million died from the disease.

Efforts to combat TB include early detection, treatment with a combination of antibiotics, and public health measures such as vaccination and improved living conditions. Advances in technology, including molecular diagnostics and artificial intelligence, have the potential to improve TB diagnosis and treatment.

2.2 Tuberculosis in Pakistan

Tuberculosis (TB) is a significant public health issue in Pakistan, with a high burden of the disease and a considerable number of TB-related deaths every year. According to the World Health Organization (WHO), Pakistan ranks fifth among high TB burden countries, with an estimated 510,000 new cases of TB in 2020 alone.

The reasons for the high TB burden in Pakistan are multifactorial and include poor living conditions, limited access to healthcare, poverty, malnutrition, and a high prevalence of risk factors such as HIV and smoking. The situation is further complicated by the emergence of drug-resistant TB and the lack of adequate diagnostic facilities and treatment options in many parts of the country. The Pakistan government has recognized TB as a major public health concern and has launched several initiatives to combat the disease. The National TB Control Program (NTP) was established in 1995 to coordinate TB control efforts at the national level, and since then, the program has made significant progress in expanding access to TB diagnosis and treatment. However, challenges remain in the effective management of TB in Pakistan. There is a need for increased awareness and education about TB among the general population and healthcare providers, as well as the development and implementation of innovative approaches to TB diagnosis and treatment. The use of deep learning techniques and CXRs for TB detection has shown promise in improving the accuracy and efficiency of TB diagnosis, and such technologies can be particularly useful in resource-limited settings where traditional diagnostic methods are not readily available remains a significant public health issue in Pakistan, and there is a need for continued efforts to combat the disease. The use of innovative technologies such as deep learning and CXRs can play an essential role in improving TB diagnosis and treatment outcomes in the country. However, these technologies must be integrated into a comprehensive approach to TB control that addresses the social, economic, and environmental determinants of the disease.

2.3 Improving TB Diagnosis with Deep Learning Interpretation Techniques:

Tuberculosis (TB) interpretation in deep learning involves the use of computer algorithms to analyze chest X-rays (CXRs) and identify patterns associated with TB infection. Deep learning is a subset of machine learning that utilizes artificial neural networks to extract features from large datasets and develop predictive models. In TB interpretation, deep learning models are trained using a large dataset of CXRs, including both normal and TB-infected images. The model learns to recognize patterns and features associated with TB infection, and when presented with a new CXR, it predicts whether the image is normal or shows signs of TB infection. The use of deep learning for TB interpretation has shown promise in improving the accuracy and efficiency of TB diagnosis. One advantage of this approach is that it can analyze large datasets of CXRs quickly and accurately, which is especially useful in resource-limited settings where traditional diagnostic methods may not be readily available.

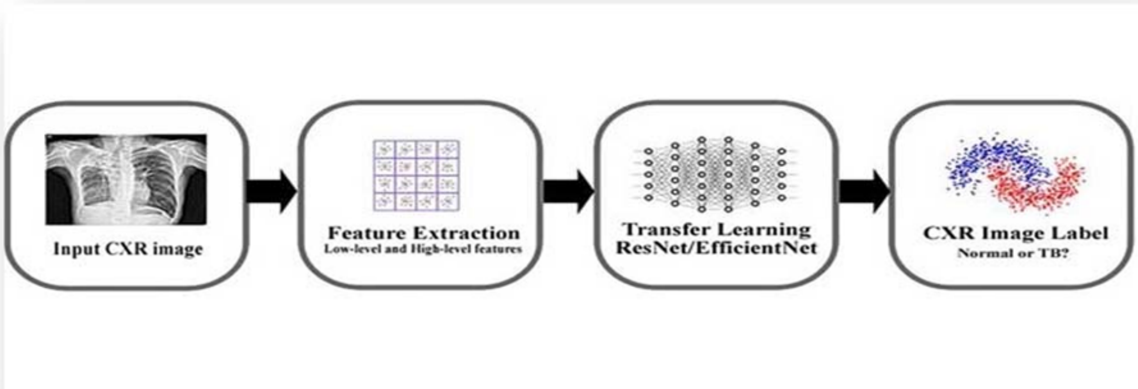

Figure 2: TB Diagnosis with Deep Learning Interpretation

2.4 Limitations and Areas of Improvement

Limitations:

- **Limited availability of high-quality CXR datasets:** Deep learning models require large and diverse datasets to train accurately. However, the availability of high-quality CXR datasets for TB diagnosis is limited, particularly in resource-limited settings.
- **Variability in CXR interpretation:** CXR interpretation is subjective and can vary depending on the experience and expertise of the radiologist. This variability can affect the accuracy and consistency of deep learning models trained on CXR data.
- **Comorbidities affecting CXR interpretation:** TB often co-occurs with other respiratory diseases, such as pneumonia and lung cancer, which can affect CXR interpretation and potentially lead to misdiagnosis.
- **Limited access to diagnostic facilities:** In many parts of the world, including Pakistan, there is a lack of access to diagnostic facilities for TB. Deep learning interpretation techniques may not be accessible in these areas due to the need for specialized equipment and expertise.

Improvement:

- **Developing larger and more diverse CXR datasets**: Efforts can be made to collect and curate larger and more diverse CXR datasets for TB diagnosis, particularly in resource-limited settings.
- **Incorporating expert opinions and guidelines:** Expert opinions and guidelines can be incorporated into deep learning models to help reduce variability in CXR interpretation and improve the accuracy and consistency of TB diagnosis.
- **Incorporating information from other diagnostic modalities:** Deep learning models can be trained to incorporate information from other diagnostic modalities, such as sputum microscopy and molecular testing, to improve TB diagnosis accuracy.
- **Developing portable and affordable diagnostic tools:** Efforts can be made to develop portable and affordable diagnostic tools that use deep learning interpretation techniques for TB diagnosis, making them more accessible in resource-limited settings.

2.5 Existing solutions and their drawbacks

Sputum microscopy: Sputum microscopy is a widely used method for TB diagnosis, particularly in resource-limited settings. It involves examining sputum samples under a microscope to detect the presence of TB bacteria. However, sputum microscopy has limitations in terms of sensitivity, as it may not detect low levels of bacteria, leading to false-negative results. Additionally, it requires skilled laboratory technicians and may take several days to obtain results.

Chest X-rays (CXR): CXR is another commonly used tool for TB diagnosis. Radiologists analyze the images for the presence of abnormalities indicative of TB, such as opacities or cavities. However, CXR interpretation can be subjective and dependent on the expertise of the radiologist. It also requires access to trained radiologists and may not be readily available in resource-limited settings.

Tuberculin skin test (TST): TST involves injecting a small amount of purified protein derivative (PPD) into the skin and assessing the immune response. However, TST has limitations, including false-positive results due to cross-reactivity with other mycobacteria and false-negative results in individuals with weakened immune systems.

GeneXpert MTB/RIF: GeneXpert is a molecular diagnostic test that detects the presence of TB bacteria and assesses their resistance to the antibiotic rifampicin. It provides rapid results and has improved sensitivity compared to microscopy. However, GeneXpert requires expensive equipment, a controlled laboratory setting, and may not be widely available in resource-limited settings.

Limited sensitivity: Many existing solutions, such as microscopy and TST, have limitations in terms of sensitivity, leading to missed or delayed diagnoses.

Subjectivity and variability: CXR interpretation and sputum microscopy results can vary depending on the expertise of the healthcare provider, leading to inconsistencies in diagnosis.

Infrastructure and resource requirements: Some existing solutions, like GeneXpert, require specialized equipment and laboratory infrastructure, making them inaccessible in resource-limited settings.

Lack of point-of-care availability: Existing solutions often require samples to be sent to laboratories for testing, causing delays in diagnosis and treatment initiation.

Chapter 3

PROBLEM DEFINITION

The problem definition of tuberculosis detection using deep learning is to develop an accurate and efficient method for diagnosing TB using chest X-rays (CXRs) and deep learning techniques. The current methods for TB diagnosis, such as sputum microscopy and culture, have limitations in terms of accuracy and turnaround time. CXRs have the potential to serve as a rapid and non-invasive screening tool for TB, but their interpretation is subjective and can vary depending on the experience and expertise of the radiologist.

Deep learning interpretation techniques can potentially overcome these limitations by providing an automated and objective method for CXR interpretation. However, there are challenges in developing accurate and robust deep learning models for TB detection, such as the limited availability of high-quality CXR datasets and the presence of comorbidities that may affect CXR interpretation.

3.1 Working Elements and Techniques

The following components and techniques will feature the application towards its goal:

3.1.1 Machine Learning Model

Our machine learning model will use algorithms to meet our application's needs. Our model will be trained with compatible data to allow us the use of deep learning via TensorFlow library to detect any errors.

3.1.2 Object Detection API

The CXRs are preprocessed to enhance their quality and reduce noise, as discussed earlier. Then, the preprocessed images are fed into the CNN model, which is implemented using TensorFlow. The CNN model consists of multiple layers, including convolutional, pooling, and fully connected layers, which learn to extract relevant features from the input CXRs. These features are then used to classify the CXRs as either TB-positive or TB-negative.

3.1.3 Database Engine

It is a basic software component that is used by database management systems to search, read, and retrieve data from a database. This component is most frequently used with database server and interchangeably in our project.

3.1.4 Database Server

A database server is a computer program that provides database services to other computer programs or terminals as defined by the client–server model. Most of the database servers work on the basis of query language.

3.1.5 User Interface

The goal of user interface design is to make it easy, efficient, and enjoyable for the users while working with the application. In this project, we have used HTML, CSS.

3.1.6 Deep Learning

Deep leaning is a machine learning approach and it depends on neural networks that produce a very good accuracy. These deep neural networks require huge dataset to learn and find a pattern between data. We use TensorFlow library to make detection of Tuberculosis.

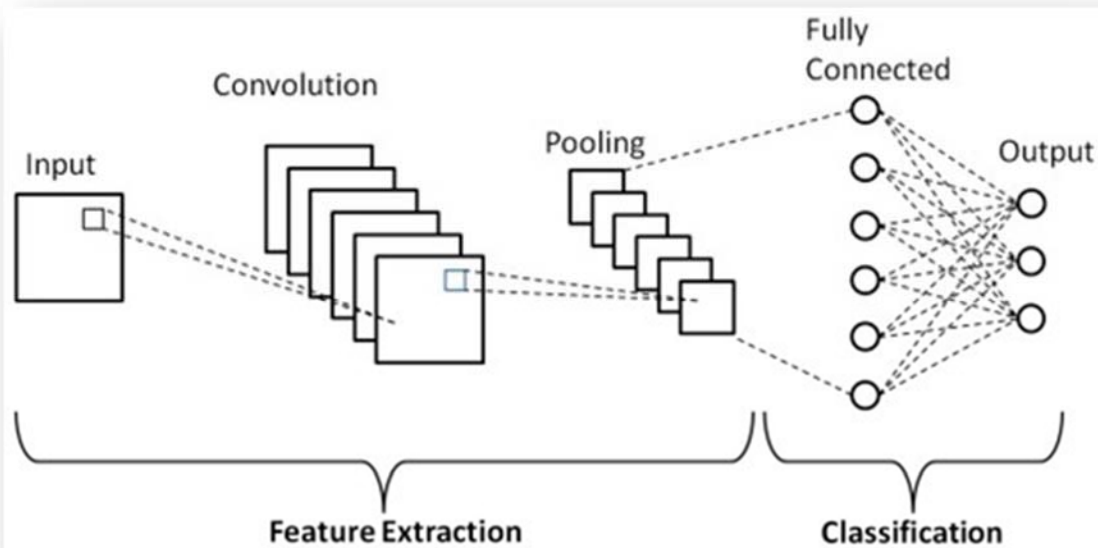

Figure 3: Convolution Neural Networks

3.1.7 Programming Languages

A programming language is designed to provide the medium of communication between user instructions and a machine. Therefore, programming language is to control the behavior of a machine, express and implement algorithms, and to create applications. Following programming languages are utilized in our project for different aspects:

- The algorithms of machine learning will execute using android programming language.

- The interaction between user interface and the system will also be controlled by android programming language.
- HTML,CSS, JavaScript
- Using Python in model.

Chapter 4

METHODOLOGY

4.1 Project Steps

i. Study and analysis the project idea from all aspects.
ii. Search and choose the best algorithm that applies our project idea.
iii. Modeling and simulating our system.
iv. Solve problems and errors in the system.
v. Deploy and implement the system in real world.
vi. Test the accuracy of the app and machine-learning model 7. Evolution the system in the future.

4.2 Working Principle

The study focuses on image processing concepts combined with deep learning methods. The project is organized into various modules, each of which is intertwined with the next. The following is a list of modules:

4.2.1 Data Acquisition:

CXR images can be acquired from various sources, such as hospitals or clinics, and must be of high quality and resolution to ensure accurate diagnosis. Once the CXR image is acquired it will be used as input to the deep learning model for TB detection

4.2.2 Preprocessing:

The acquired CXR images may need to be preprocessed to improve the quality and enhance the features relevant to TB detection. This may involve operations such as resizing the images to a consistent resolution, adjusting the brightness and contrast, and applying noise reduction techniques to remove any artifacts or distortions. The goal of preprocessing is to improve the accuracy of the subsequent TB detection step using deep learning techniques.

4.2.3 Feature Extraction:

The preprocessed CXR image is used as input to the deep learning model, which extracts relevant features from the image. This process involves analyzing the image to identify regions of interest, such as lesions or infiltrates, and extracting meaningful features from these regions. Convolutional neural networks (CNNs) are commonly used for this task, as they are effective at learning spatial patterns in images. The CNN model can be trained on a large dataset of CXR images to learn the most discriminative features for TB detection. Once the features are extracted, they are passed on to the next step for classification as normal or TB.

4.2.4 Output Presentation:

In the proposed TB detection system, the output presentation involves classifying the preprocessed CXR image as normal or TB. The deep learning model, which has been trained on a large dataset of CXR images, extracts relevant features from the input image and passes them through a classification algorithm. The output of the algorithm is a prediction of whether the input image represents a normal CXR or a CXR with TB. The system can then present this output to the end-user, such as displaying the output on a computer screen. The output presentation is designed to be easy-to-understand and user-friendly, enabling medical professionals to make informed decisions about patient care based on the system's results.

4.2.5 Feedback and Refinement:

feedback and refinement are crucial components of the working principle. The system may generate false positives or false negatives, which can be problematic in a medical setting. To improve the accuracy of the system, the output can be fed back into the system for further refinement. This may involve adjusting the parameters of the deep learning model, increasing the size of the training dataset, or incorporating additional features into the model. The feedback loop can help the system learn from its mistakes and improve over time, leading to more accurate and reliable TB detection results. Additionally, the system can be refined by incorporating feedback from medical professionals and patients, who can provide insights and suggestions for improvement. This iterative approach to feedback and refinement is a key aspect of the proposed TB detection system, as it enables ongoing improvement and adaptation to the needs of the medical community.

Chapter 5

DETAILED DESIGN AND ARCHITECTURE

This section will provide a design detail of our application including high level system design and UML diagrams depicting the system processes. Our Application will follow client server architecture model. Where user is the client side and application will be the server as per the system model.

5.1 Architectural Design

The proposed application will take chest X-ray images as input and use deep learning techniques to analyze them for the presence of tuberculosis. The model will be trained on a large dataset of CXRs and will use advanced algorithms to accurately identify the presence of TB.

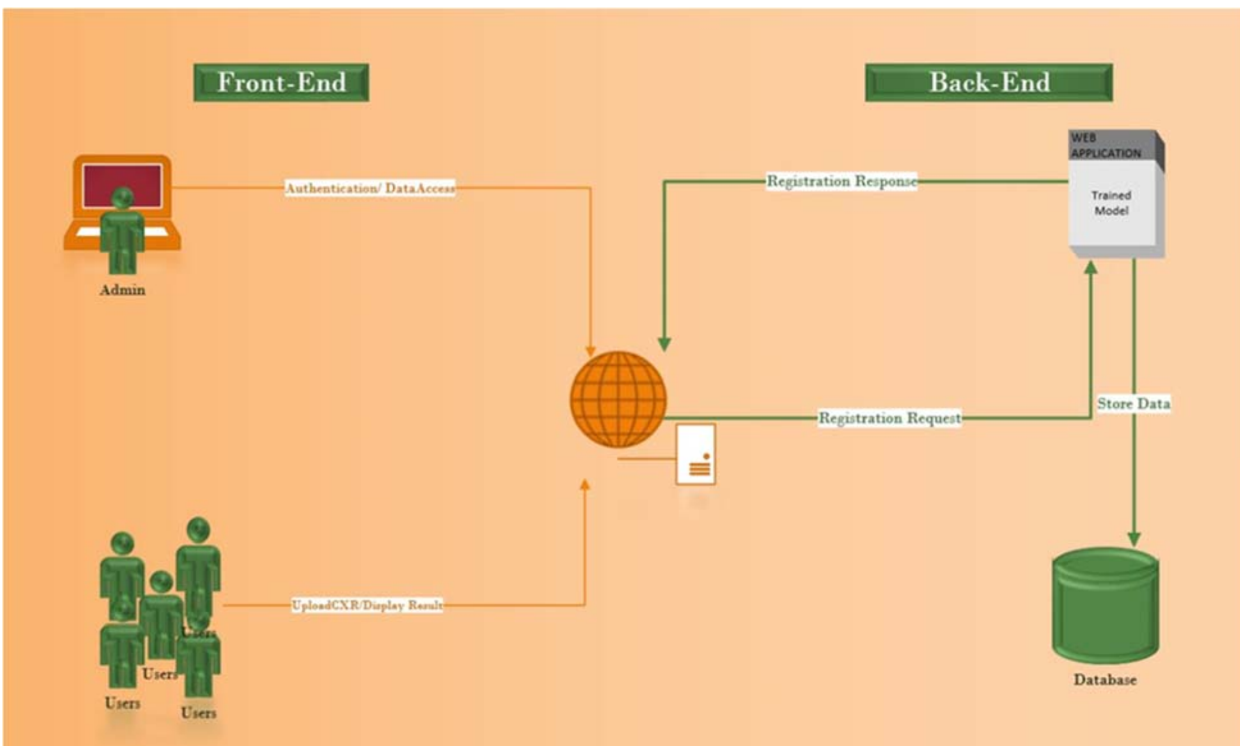

Figure 4: Architecture Diagram

5.2 Component Design

This section will take a closer look at each component of the Application in a more systematic way through a component diagram. The component diagram shows three major components; the client machine, the API server and cloud server which shows several subcomponents.

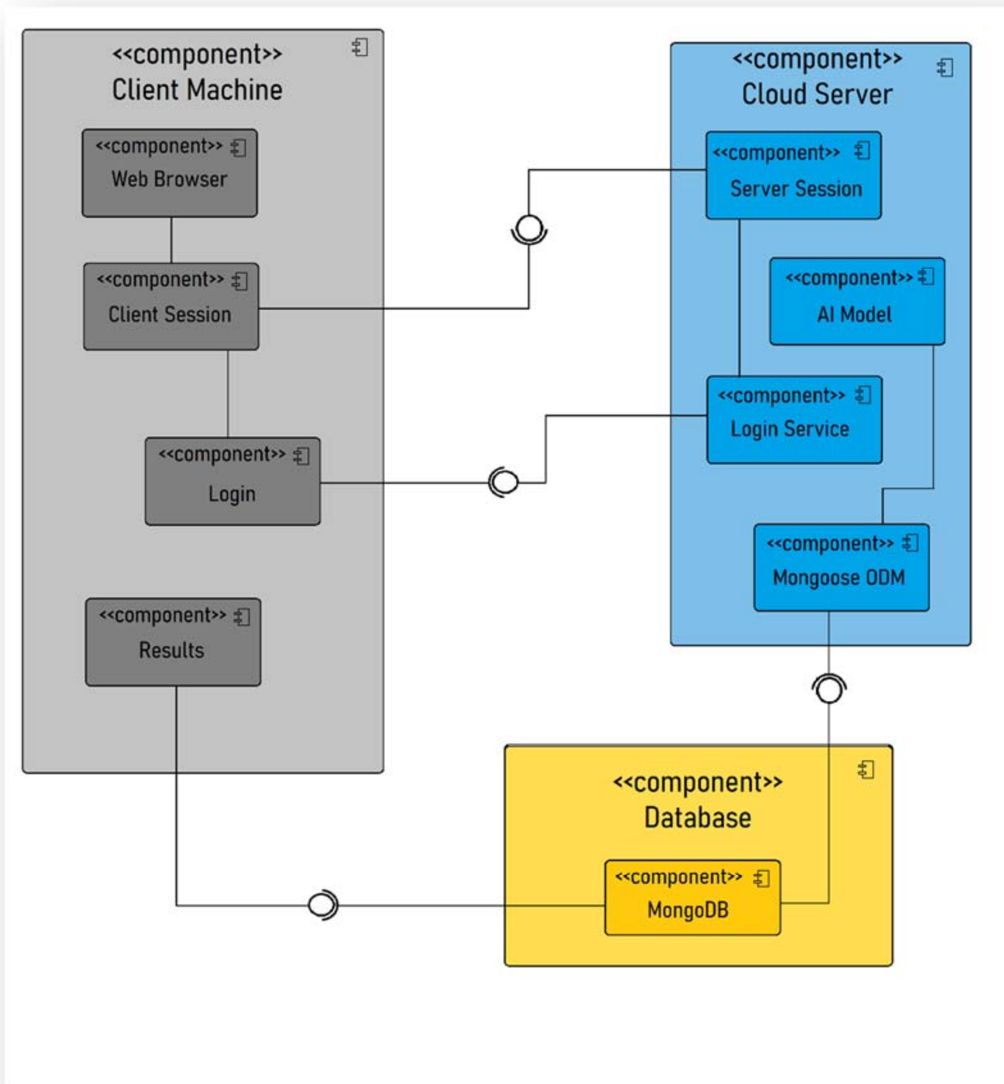

Figure 5: Component Diagram

5.3 Decomposition Description

The decomposition of the subsystems shown in the architectural design is explained as per the module and process decomposition:

5.3.1 Module Decomposition – Class Diagram

Class diagram below shows the Module Decomposition of our application.

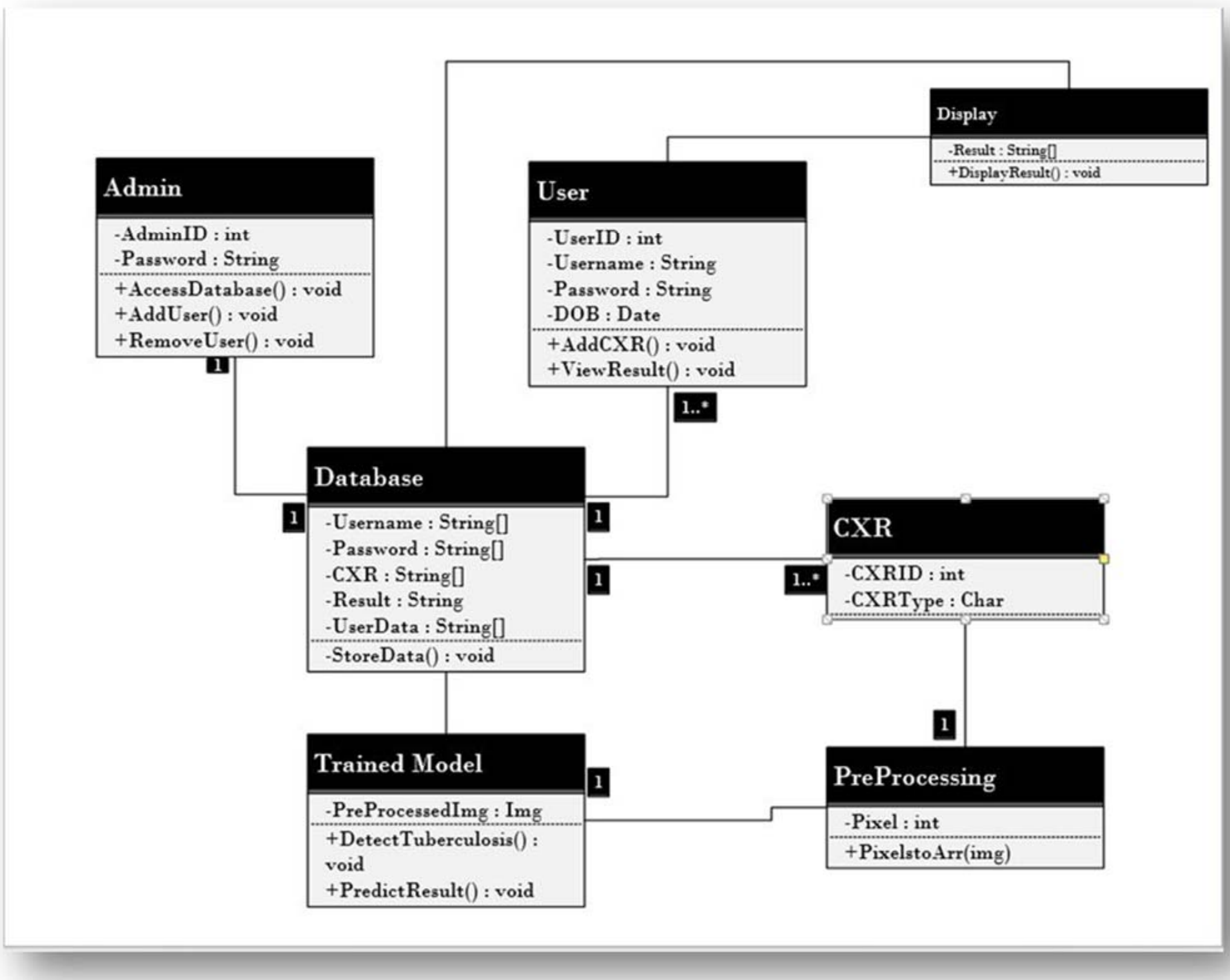

Figure 6: Class Diagram

5.3.2 Process Decomposition – Use Case Diagram

The process decomposition is explained through use case, sequence and activity diagrams which decompose the system into well-defined and cohesive processes. The use cases and the subsequent use case narratives explain the set of actions that a user undertakes.

The following figure explains the use cases and subsequent narratives as per the actors performing actions while interacting with the application.

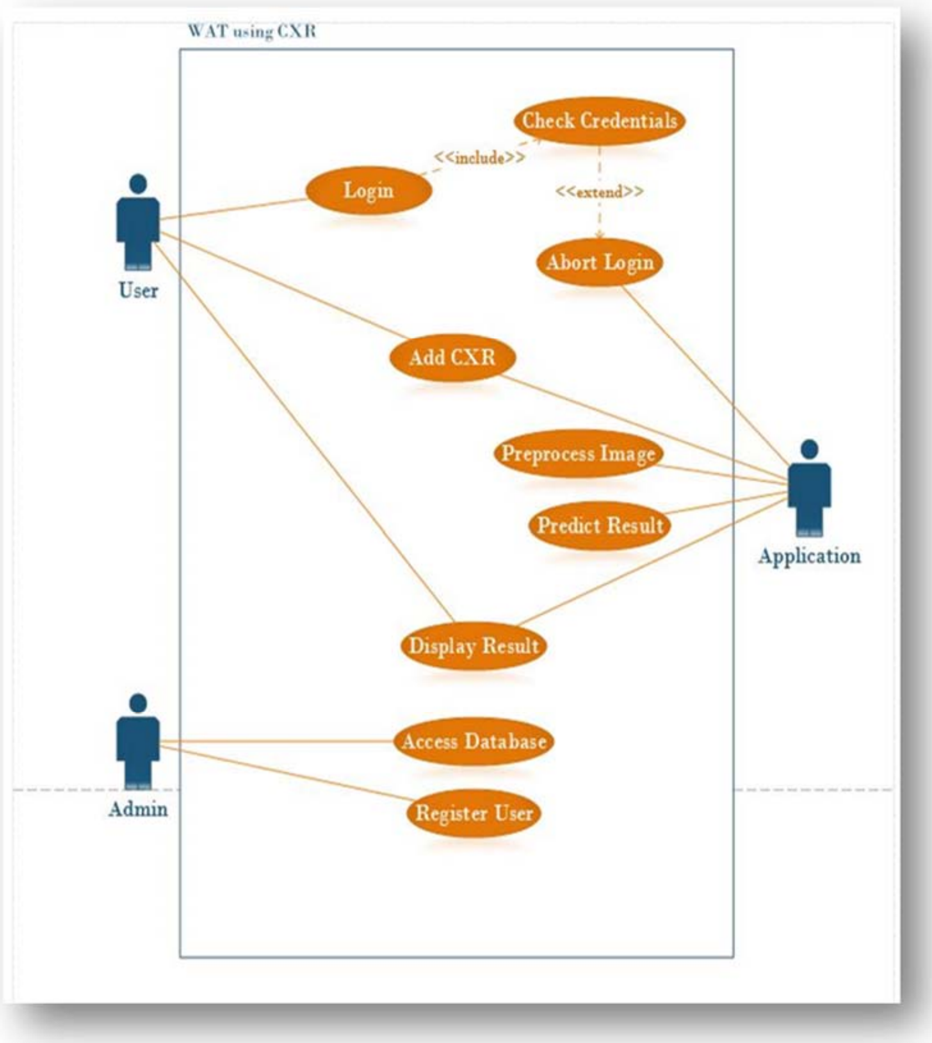

Figure 7: Use Case Diagram

5.4 Sequence Diagram:

The sequence diagram below shows the simplified and tentative illustration of the component interactions that are involved in system implementation.

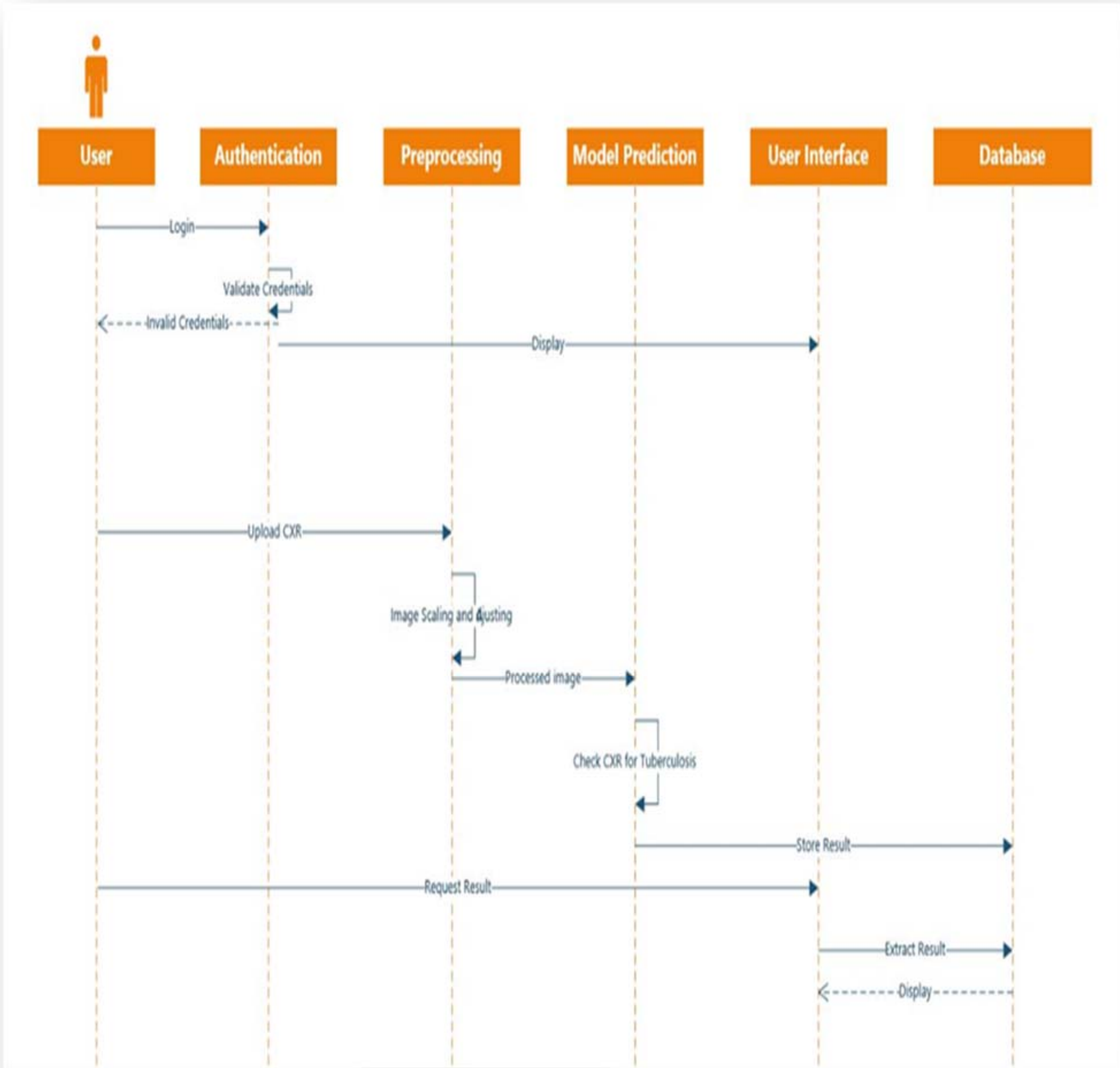

Figure 8: Sequence Diagram

5.5 Activity Diagram:

The below activity diagrams provide visual representation of the flow of activities and actions in the application. The systematic purpose of this activity diagram is to help in understanding the high-level behavior and interaction of the system and its components.

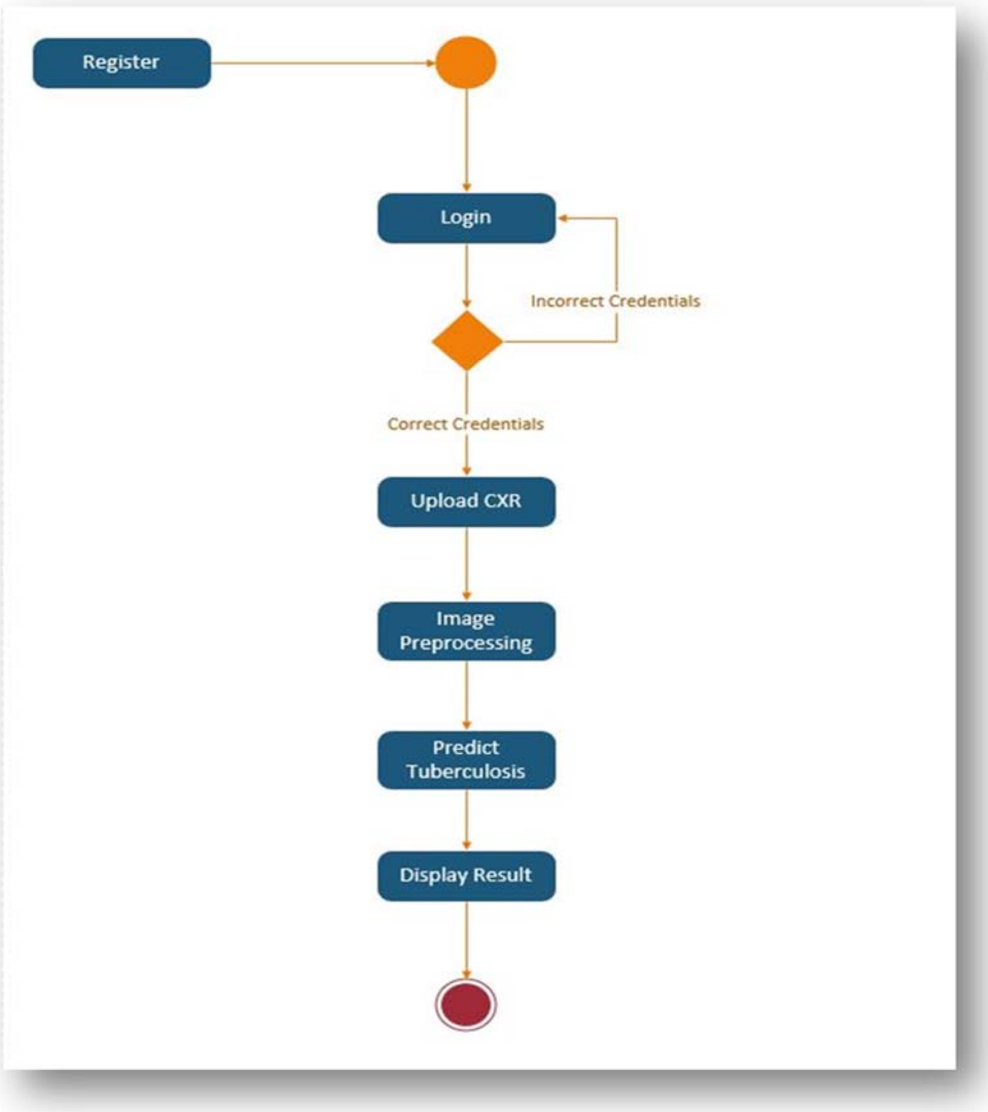

Figure 9: Activity Diagram

5.6 Interface:

In the application, the user will firstly be presented with the login screen where the user will enter their verified credentials. After authentication, the user will be taken to the main upload screen where they will be able to upload their X-Ray image data for prediction in the model. Lastly the user is taken to the final screen where the results are displayed to the user and they are given choices as to what to do next.

Sign-In Screen:

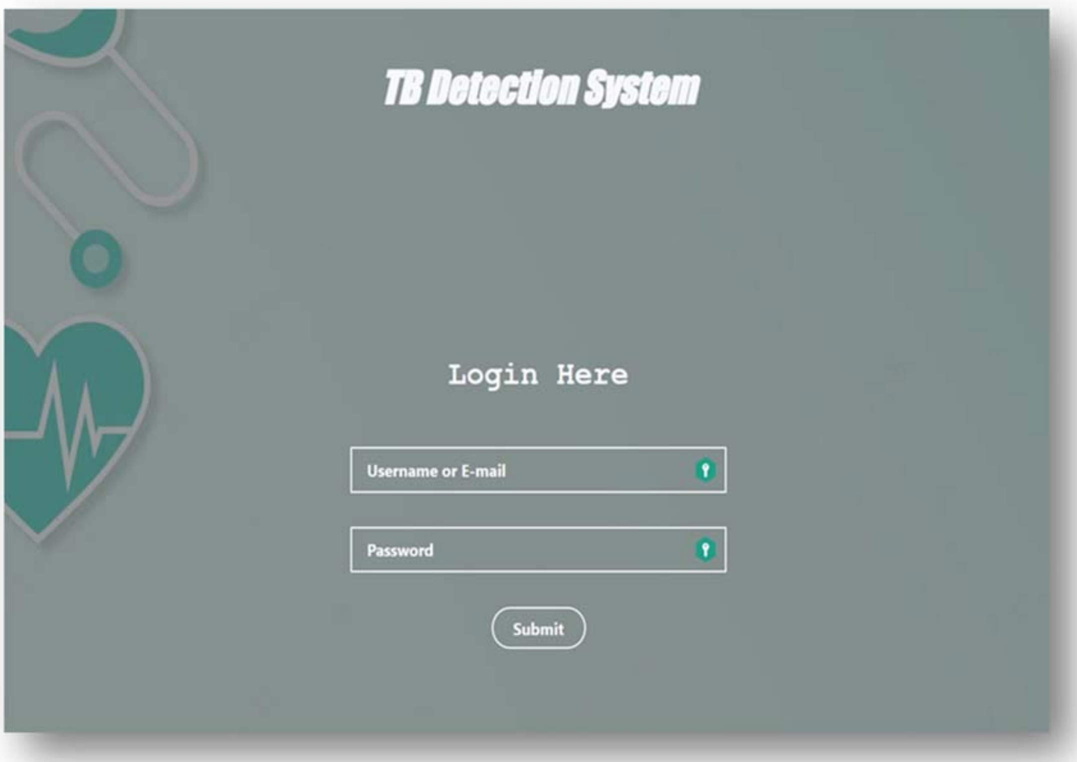

Figure 10: Login Page

Upload Screen

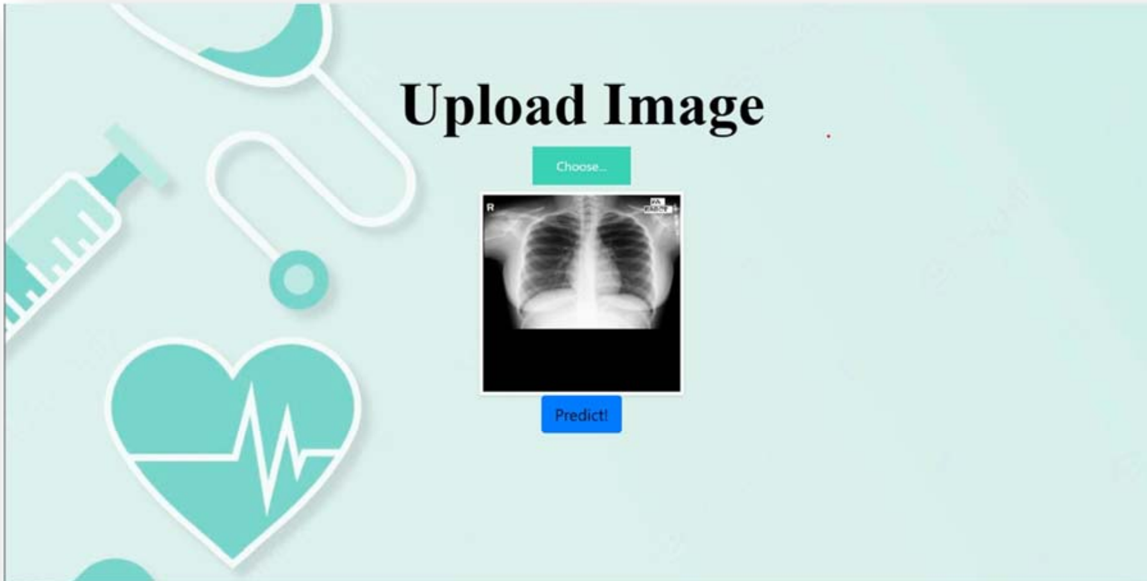

Figure 11: Main Page

Results Screen

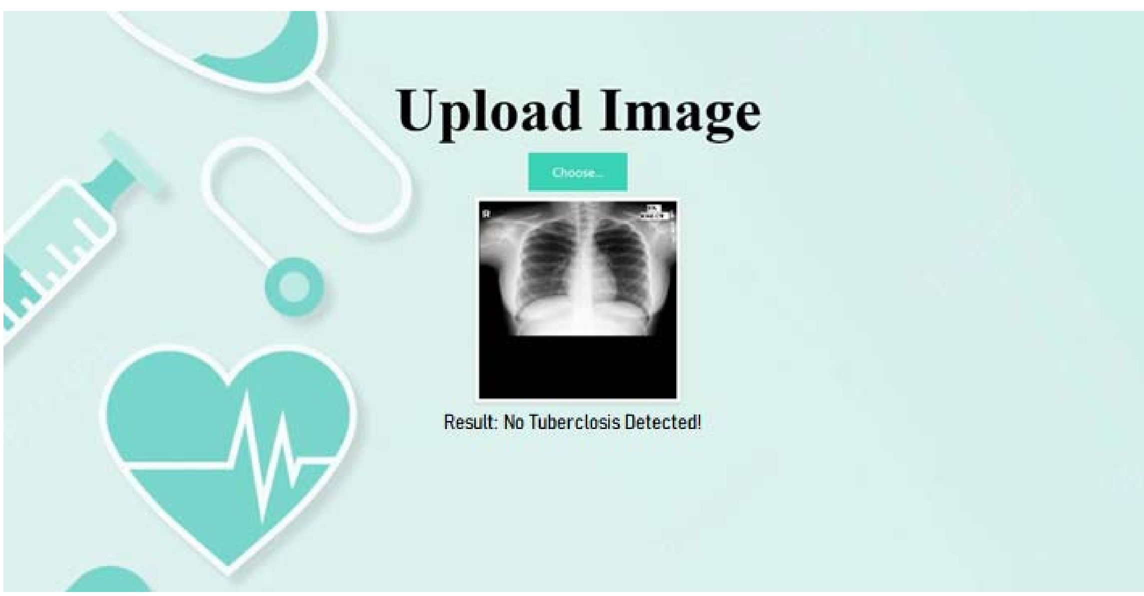

Figure 12: Result Page

Chapter 6

Code Analysis and Evaluation

Code:

Importing Essential Libraries:

```
from numpy.random import seed
seed(1)
from tensorflow import random
random.set_seed(1)

import pandas as pd
import numpy as np

import tensorflow

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Conv2D, MaxPooling2D, Flatten
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint
from tensorflow.keras.metrics import binary_accuracy

import os
import cv2

import imageio
import skimage
import skimage.io
import skimage.transform

from sklearn.utils import shuffle
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
import itertools
import shutil
import matplotlib.pyplot as plt

%matplotlib inline
```

Creating Model Architecture:

```python
# Source: https://www.kaggle.com/fmarazzi/baseline-keras-cnn-roc-fast-5min-0-8253-lb

kernel_size = (3,3)
pool_size= (2,2)
first_filters = 32
second_filters = 64
third_filters = 128

dropout_conv = 0.3
dropout_dense = 0.3


model = Sequential()
model.add(Conv2D(first_filters, kernel_size, activation = 'relu',
                 input_shape = (IMAGE_HEIGHT, IMAGE_WIDTH, 3)))
model.add(Conv2D(first_filters, kernel_size, activation = 'relu'))
model.add(Conv2D(first_filters, kernel_size, activation = 'relu'))
model.add(MaxPooling2D(pool_size = pool_size))
model.add(Dropout(dropout_conv))

model.add(Conv2D(second_filters, kernel_size, activation ='relu'))
model.add(Conv2D(second_filters, kernel_size, activation ='relu'))
model.add(Conv2D(second_filters, kernel_size, activation ='relu'))
model.add(MaxPooling2D(pool_size = pool_size))
model.add(Dropout(dropout_conv))

model.add(Conv2D(third_filters, kernel_size, activation ='relu'))
model.add(Conv2D(third_filters, kernel_size, activation ='relu'))
model.add(Conv2D(third_filters, kernel_size, activation ='relu'))
model.add(MaxPooling2D(pool_size = pool_size))
model.add(Dropout(dropout_conv))

model.add(Flatten())
model.add(Dense(256, activation = "relu"))
model.add(Dropout(dropout_dense))
model.add(Dense(2, activation = "softmax"))
```

Training The Model:

```
model.compile(Adam(lr=0.0001), loss='binary_crossentropy',
              metrics=['accuracy'])
```

```
filepath = "model2.h5"
checkpoint = ModelCheckpoint(filepath, monitor='val_acc', verbose=1,
                             save_best_only=True, mode='max')

reduce_lr = ReduceLROnPlateau(monitor='val_acc', factor=0.5, patience=2,
                                   verbose=1, mode='max', min_lr=0.00001)


callbacks_list = [checkpoint, reduce_lr]

history = model.fit_generator(train_gen, steps_per_epoch=train_steps,
                            validation_data=val_gen,
                            validation_steps=val_steps,
                            epochs=100, verbose=1,
                           callbacks=callbacks_list)
```

Loading and Testing:

```
from keras.models import load_model
import keras.utils as image
from keras.applications.vgg16 import preprocess_input
import numpy as np
model = load_model('/content/drive/MyDrive/saved/model2.h5')
img = image.load_img('/content/drive/MyDrive/mono/Tuberclosis/Tuberculosis-10.png', target_size=(96, 96))
x = image.img_to_array(img)
img_test = np.expand_dims(x, axis=0)
classes = model.predict(img_test)
print (classes)
```

Prediction Code:

```
myapp > views.py > ...
from imp import load_module
from django.http import HttpResponse
from django.shortcuts import render
from tensorflow.keras.models import load_model

import sys
import os
import glob
import re
import numpy as np
import tensorflow as tf
from werkzeug.utils import secure_filename

from tensorflow.keras.applications.resnet50 import preprocess_input
from tensorflow.keras.preprocessing import image


from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession
from .models import Users_data

import cv2
import numpy as np
import os


from django.http import HttpResponse
from django.shortcuts import render, redirect
from .forms import *

```

```python
def index(request):
    return render(request, 'ANALYSIS.html')


def Mainpage(request):
    msg = ""
    return render(request, 'index.html',{'msg':msg})

def OurProject(request):
    userName = request.POST['username']
    Password = request.POST['userpassword']
    print(userName)
    print(Password)
    all_data_from_table = Users_data.objects.all()
    for row in all_data_from_table:
        # all_questions[row.chat_msg] = row.Answer

        if userName==row.Name and Password ==row.Password:
            return render(request, 'Mainscreen.html')

    msg = "Wrong user name or password."
    return render(request, 'index.html',{'msg':msg})
```

```python
def model_predict2(img_path, model):
    #try:
        img = image.load_img(img_path, target_size=(96, 96))

        # Preprocessing the image
        x = image.img_to_array(img)

        x = np.expand_dims(x, axis=0)
        preds = model.predict(x)
        #preds=np.argmax(preds, axis=1)
        if preds[0,0]>preds[0,1]:

            preds="No Tubercolosis Detected!! "


        else:
            preds="Tubercolosis Detected!!"
        return preds
```

```python
def predict_output(request):
    form = RiceForm()
    if request.method == 'POST':
        print(request.FILES['file'].name)
        form = RiceForm(request.POST, request.FILES)
        if form.is_valid():
            user_pr = form.save(commit=False)
            user_pr.file = request.FILES['file']
            print(user_pr.file.url)
            # f = request.post.files['file']

            basepath = os.path.dirname(__file__)
            print(basepath)
            user_pr.save()
            print("photo saved at ",(user_pr.file.url))

            MODEL_PATH = os.path.join(basepath,'static\\model2.h5')
            print(MODEL_PATH)
            model = load_module(MODEL_PATH)
            filename = request.FILES['file'].name
            file_path = os.getcwd()
            file_path_full = os.path.join(file_path,'media',filename)
            if "skin" in filename:
                # Make prediction
                preds = model_predict(file_path_full, model)
                result=preds
                return HttpResponse(result)
            else:
                result = model_predict2(file_path_full, model)
                return HttpResponse(result)

        else:
            return HttpResponse('Invalid')
```

```python
urlpatterns = [

    path('', views.index),
    path('admin/', admin.site.urls),
    path('home', views.Mainpage,name="home"),
    path('mainPage/', views.OurProject,name="mainPage"),
    path('mainPage/mainPage/', views.OurProject,name="mainPage"),
    path('mainPage/predict_it', views.predict_output,name="predict_it"),
    path('mainPage/mainPage/predict_it', views.predict_output,name="predict_it"),
    path('predict', views.predict_output,name="predict"),
    # path('', views.index),/predict
    path('mainPage/cam_pic', views.Live_cam,name="live_cam"),
]

if settings.DEBUG:
        urlpatterns += static(settings.MEDIA_URL,
                              document_root=settings.MEDIA_ROOT)
```

Chapter 7

RESULTS AND DISCUSSION

7.1 Results:

The end result of the project is a web-based application that uses deep learning techniques to accurately detect the presence of tuberculosis in chest X-ray images. The application takes a chest X-ray image as input and processes it using a trained deep learning model. The output of the application is a binary classification result indicating whether the image shows signs of TB or not.

This application has the potential to improve the accuracy and efficiency of TB detection, particularly in resource-limited settings where trained radiologists may not be readily available. It can also serve as a valuable tool for TB screening programs and help in early detection and treatment of the disease. The user-friendly interface of the application makes it accessible to healthcare professionals and even non-experts, contributing to the democratization of medical diagnostic systems.

The successful implementation of the project demonstrates the feasibility of using deep learning techniques for TB detection and sets the foundation for further research in this area. The limitations and areas for improvement identified during the project also provide directions for future work to enhance the accuracy and robustness of the application.

7.2 Discussion:

The proposed project of developing a web-based application for tuberculosis detection using chest X-rays and deep learning techniques has several potential benefits. It can significantly improve the accuracy and efficiency of TB diagnosis, especially in resource-limited settings where there is a shortage of trained medical professionals. The use of deep learning algorithms can help to overcome the limitations of current methods, such as the subjectivity and variability of human interpretation, and improve the overall quality of TB screening and diagnosis.

However, there are also several challenges and limitations to consider. One major limitation is the quality and diversity of the training dataset. The accuracy and generalizability of the deep learning model depend heavily on the size and quality of the dataset used for training. Obtaining a large and diverse dataset of CXRs with confirmed TB cases can be difficult, especially in low-resource settings where TB is more prevalent.

Another challenge is the potential presence of other comorbidities or abnormalities that may affect the interpretation of CXRs. For example, individuals with HIV, pneumonia, or lung cancer may have CXR images that resemble those of TB, which could lead to false positive or false negative results. Therefore, it will be important to carefully validate the model's performance using a diverse set of CXRs with various comorbidities.

Furthermore, the use of deep learning models in medical diagnostics raises ethical and regulatory concerns related to privacy, data security, and potential biases in the algorithm. It will be important to address these concerns and ensure that the application is developed and deployed in an ethical and responsible manner.

The proposed project has the potential to greatly improve TB detection and diagnosis, it also faces several challenges and limitations that need to be carefully considered and addressed. With careful planning, validation, and ethical considerations, this project could have a significant impact on global TB control efforts.

Chapter 8

CONCLUSIONS AND FUTURE WORK

The results of this thesis demonstrate the potential of using machine learning techniques, such as CNNs, for feature extraction in improving the accuracy of our applications. The successful implementation of the application using Python highlights the potential for the use of technology to improve TB Diagnosis using Chest X-Rays.

However, there are still several areas where future work can be done to improve the application. One potential area for improvement is to increase the size of the dataset used for training and testing the application. A larger dataset may result in improved accuracy and robustness of the application.
Furthermore, it may be beneficial to explore the use of other machine learning techniques and algorithms for feature extraction and classification of Normalities and abnormalities. This can lead to further improvements in the accuracy of the application.

In conclusion the development of a web-based application for TB detection using CXRs and deep learning techniques is a significant step towards improving TB diagnosis and treatment outcomes, and has the potential to make a positive impact on the lives of millions of people affected by this disease.

8.1 Future Work

There are several potential areas of future work that can be explored to improve the Sign Language Recognition (SLR) application developed in this thesis. These include:

Integration with other diagnostic modalities: While CXRs are widely used for TB diagnosis, they may not always provide a definitive diagnosis. Therefore, integrating the proposed system with other diagnostic modalities such as sputum analysis or molecular testing could improve the overall accuracy and reliability of TB diagnosis.

Validation in diverse populations: The proposed system was trained and tested using CXR images from a specific population. To validate its performance in diverse populations, it should be tested using CXRs from other regions and populations with varying disease prevalence and clinical characteristics.

Incorporation of clinical data: CXRs alone may not always provide enough information for an accurate TB diagnosis. Incorporating clinical data such as symptoms, medical history, and demographic information could further improve the accuracy of the proposed system.

Deployment in resource-limited settings: TB is particularly prevalent in resource-limited settings where access to diagnostic tools and trained personnel is limited. Therefore, deploying the proposed system in these settings could improve TB diagnosis and treatment outcomes.

Exploration of explainable AI techniques: While deep learning has shown promise in TB diagnosis, its "black box" nature limits its interpretability and raises ethical concerns. Exploring explainable AI techniques could help increase the transparency and trustworthiness of the proposed system.

REFERENCES

[1] Bykov, K., Hedström, A., Nakajima, S., & Höhne, M. M. C. (2022, June). NoiseGrad—Enhancing Explanations by Introducing Stochasticity to Model Weights. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 36, No. 6, pp. 6132-6140).

[2] Chen, L., Chen, J., Hajimirsadeghi, H., & Mori, G. (2020). Adapting grad-cam for embedding networks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 2794-2803).

[3] Rahman, T., Khandakar, A., Kadir, M. A., Islam, K. R., Islam, K. F., Mazhar, R., ... & Chowdhury, M. E. (2020). Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. IEEE Access, 8, 191586-191601.

[4] Hwang, E. J., Park, S., Jin, K. N., Kim, J. I., Choi, S. Y., Lee, J. H., ... & Deep Learning-Based Automatic Detection Algorithm Development and Evaluation Group Kim Dong Hyeon Woo Woo Choi Choi Hwang In Pyung Song Yong Sub Lim Lim Kim Kim Wi Jae Yeon Oh Su Suk Kang Mi-Jin. (2019). Development and validation of a deep learning–based automatic detection algorithm for active pulmonary tuberculosis on chest radiographs. Clinical Infectious Diseases, 69(5), 739-747.

[5] Rajpurkar, P., O'Connell, C., Schechter, A., Asnani, N., Li, J., Kiani, A., ... & Lungren, M. P. (2020). CheXaid: deep learning assistance for physician diagnosis of tuberculosis using chest x-rays in patients with HIV. NPJ digital medicine, 3(1), 1-8.

[6] Heo, S. J., Kim, Y., Yun, S., Lim, S. S., Kim, J., Nam, C. M., ... & Yoon, J. H. (2019). Deep learning algorithms with demographic information help to detect tuberculosis in chest radiographs in annual workers' health examination data. International journal of environmental research and public health, 16(2), 250.

[7] Munadi, K., Muchtar, K., Maulina, N., & Pradhan, B. (2020). Image enhancement for tuberculosis detection using deep learning. IEEE Access, 8, 217897-217907.

[8] Oloko-Oba, M., & Viriri, S. (2020, October). Pre-trained convolutional neural network for the diagnosis of tuberculosis. In International Symposium on Visual Computing (pp. 558-569). Springer, Cham.

[9] Rahman, T., Khandakar, A., Kadir, M. A., Islam, K. R., Islam, K. F., Mazhar, R., ... & Chowdhury, M. E. (2020). Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. IEEE Access, 8, 191586-191601.

TB Detection System

ORIGINALITY REPORT

14%	9%	4%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	2%
2	seecs.nust.edu.pk Internet Source	1%
3	Submitted to Engineers Australia Student Paper	1%
4	nurseinnursing.com Internet Source	1%
5	Submitted to London School of Science & Technology Student Paper	<1%
6	Submitted to Liberty University Student Paper	<1%
7	lib.iium.edu.my Internet Source	<1%
8	Submitted to The University of the West of Scotland Student Paper	<1%