SCHOOL OF GRADUATE STUDIES

MULTI-MODAL TUBERCULOSIS DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORK

MSc THESIS

BUSER SHEMSU ALI

DECEMBER, 2023

WOLKITE, ETHIOPIA

Wolkite University

School of Graduate Studies

Multi-modal Tuberculosis Disease Detection Using Convolutional Neural Network

A Thesis Submitted to School of Graduate Studies, in Partial Fulfilment of the Requirements for the Degree of Master of Computer Science and Engineering (Specialization: Computer science)

Buser Shemsu Ali

Major Advisor: Mesfin Abebe (PH.D.)

Co-Advisor: Jemal Ahmed

December, 2023

Wolkite, Ethiopia

APPROVAL SHEET

SCHOOL OF GRADUATE STUDIES

WOLKITE UNIVERSITY

Multi-modal Tuberculosis Disease Detection Using Convolutional Neural Network

Submitted by:

_____	_____	_____
Name of Student	Signature	Date

Approved by:

1. _____	_____	_____
Name of Major Advisor	Signature	Date
2. _____	_____	_____
Name of Co-Advisor	Signature	Date
3. _____	_____	_____
Name of Chairman, DGC	Signature	Date
4. _____	_____	_____
Name of Dean, SGS	Signature	Date

WOLKITE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby certify that we have read and evaluated this Thesis titled "**Multi-Modal Tuberculosis Disease Detection using Deep Learning**" prepared under our guidance by Buser Shemsu Ali. We recommend that the Thesis shall be submitted as fulfilling the requirements for the award of a MSc. degree in Computer Science and Engineering.

_____	_____	_____
Major Advisor	Signature	Date
_____	_____	_____
Co-Advisor	Signature	Date

As members of the Board of Examiners of the Master of Science Thesis open defense examination, we have read and evaluated this Thesis prepared by Buser shemsu Ali and examined the candidate. we hereby certify that, the thesis is accepted for fulfilling the requirements for the award of the degree of Master of Science (M.Sc.) in Computer Science and Engineering.

1. _____	_____	_____
Name of External Examiner	Signature	Date
2._____	_____	_____
Name of Internal Examiner	Signature	Date
3._____	_____	_____
Name of Chairman	Signature	Date

Final approval and acceptance of the Thesis is contingent upon the submission of its final copy to the Council of Postgraduate Program (CPGS) through the candidate's department or school graduate committee (DGC or SGC).

DECLARATION

By my signature below, I declare and affirm that this Thesis is my own work. I have followed all ethical principles of scholarship in the preparation, data collection, data analysis and completion of this thesis. All scholarly matter that is included in the thesis has been given recognition through citation. I affirm that I have cited and referenced all sources used in this document. Every serious effort has been made to avoid any plagiarism in the preparation of this thesis.

This thesis is submitted in partial fulfillment of the requirement for a degree from the School of Graduate Studies at Wolkite University. The thesis is deposited in the Wolkite University Library and is made available to borrowers under the rules of the library. I solemnly declare that this thesis has not been submitted to any other institution anywhere for the award of any academic degree, diploma or certificate.

Brief quotations from this Thesis may be used without special permission provided that accurate and complete acknowledgement of the source is made. Requests for permission for extended quotations from, or reproduction of, this thesis in whole or in part may be granted by the Head of the School or Department or the Dean of the School of Graduate Studies when in his or her judgment the proposed use of the material is in the interest of scholarship. In all other instances, however, permission must be obtained from the author of the thesis.

Name: Signature: _____

Date:

Department:

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **ALLAH** for providing me with the opportunity and the strength to complete this study. I also extend my appreciation to my advisor, **Dr. Mesfin Abebe**, for his invaluable guidance, support, and expertise throughout the research process. Additionally, I am grateful for the assistance and insight provided by my co-advisor, **Mr. Jemal Ahmed**. Their contribution has been instrumental in shaping this study and bringing it to fruition.

I am also thankful to my sister, **Miss. Fedila Shemsu**, for her unwavering support and encouragement throughout my academic journey. Her constant motivation and belief in me have been a source of strength and inspiration. Furthermore, I would like to express my gratitude to my wife, **Miss. Fetiya Behru**, for her unwavering support, patience, and understanding during this period. Her love, care, and encouragement have been a constant source of inspiration and motivation, and I could not have done it without her.

LIST OF ABREVIATIONS AND ACRONYMS

ACC	Accuracy
AFB	Acid-Fast Bacilli
AI	Artificial Intelligence
AIDS	Acquired Immunodeficiency Syndrome
AUC	Area Under the Curve
AUC-ROC	Area Under the Receiver Operating Characteristic Curve
BMI	Body Mass Index
CAD	Computer-Aided Diagnosis
CADe	Computer-Aided Detection
CNN	Convolutional Neural Network
CNN1D	One-Dimensional CNN
CNN2D	Two-Dimensional CNN
COVID-19	Coronavirus Disease 2019
CPU	Central Processing Unit
CSV	Comma-Separated Values
CT	Computed Tomography
CXR	Chest X-ray
DCNN	Deep Convolutional Neural Network
DICOM	Digital Imaging and Communications in Medicine
DL	Deep Learning
DNA	DeoxyriboNucleic Acid
DNN	Deep Neural Network
DSR	Design Science Research
ECG	ElectroCardioGram
EHR	Electronic Health Record
EPTB	Extra-Pulmonary Tuberculosis
FPR	False Positive Rate
ERT	Extremely Randomized Trees
HIV	Human Immunodeficiency Virus
HLP	High-Level Processing
IGRA	Interferon-Gamma Release Assay

JPG (JPEG)	Joint Photographic Expert Group
LiDAR	Light Detection and Ranging
LLP	Low-Level Processing
LTBI	Latent Tuberculosis Infection
ML	Machine Learning
MLP	Medium-Level Processing
MRI	Magnetic Resonance Imaging
mRMR	Minimum Redundancy Maximum Relevance
Mtb	Mycobacterium tuberculosis
MTB-LD	Mycobacterium Tuberculosis Lung Disease
NaNs	Not a Number
NIH	National Institutes of Health
NTM-LD	Non-Tuberculous Mycobacterial Lung Disease
PCR	Polymerase Chain Reaction
PET	Positron Emission Tomography
PNG	Portable Network Graphics
PPD	Purified Protein Derivative
PTB	Pulmonary Tuberculosis
ReLU	Rectified Linear Unit function
RF	Random Forests
RGB	Red Green Blue
RIF	Rifampicin
RNN	Recurrent Neural Network
ROI	Region of Interest
SES	Sensitivity
SMOTE	Synthetic Minority Oversampling Technique
SPC	Specificity
SPECT	Single Photon Emission Computed Tomography
STN	Spatial Transformer Network
TB	Tuberculosis
FPR	True Positive Rate
TST	Tuberculin Skin Test

VGG	Visual Geometry Group-based
WHO	World Health Organization

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHM STEPS

ABSTRACT

Tuberculosis (TB) is a highly infectious disease caused by Mycobacterium tuberculosis that primarily affects the lungs but can also impact other parts of the body. However, accurately diagnosing TB poses a significant challenge, especially in developing countries where access to trained radiologists and expensive imaging methods is limited. Researchers are actively exploring the use of Computer-Aided Diagnosis (CAD) applications as a promising approach to enhance the diagnosis of TB. While many studies focus on analyzing pixel data from X-ray images, relying solely on this modality has limitations in effectively diagnosing the disease. This study proposes a multimodal fusion approach that integrates chest X-ray images and patient symptoms to improve TB detection. The researchers utilized a dataset collected from Wolkite University Specialized Hospital, consisting of 734 cases. Among these, 571 cases were classified as normal, while 163 cases were classified as abnormal. The proposed approach involved various techniques such as data preprocessing, augmentation, segmentation, and data balancing to ensure accurate results. Additionally, hyperparameter tuning was performed to optimize both unimodal and multimodal fusion models. The study employed both individual modality classification and multimodal fusion classification. For multimodal fusion, the researchers extracted features from individual classification models, allowing for a comprehensive analysis that combines the strengths of each modality. The proposed model is rigorously evaluated using 10-fold cross-validation and specific parameters. The results demonstrated that the multimodal fusion technique outperformed individual modalities, with the early fusion model achieving impressive performance metrics. It achieved a remarkable 99% accuracy, 99% sensitivity, and 98% specificity. This comprehensive approach significantly enhances TB detection accuracy, offering potential for real-world clinical applications. By improving diagnostic accuracy, healthcare professionals can devise more effective treatment plans. The use of multimodal fusion not only reduces the workload for radiologists but also ensures a more accurate and timely diagnosis.

Keywords: chest X-ray images; multimodal fusion classification; Mycobacterium tuberculosis; patient symptoms; preprocessing techniques; Tuberculosis; X-ray image segmentation.

CHAPTER ONE

1. Introduction

1.1. Background of the Study

Tuberculosis (TB) is a chronic disease that predominantly targets the lungs, but can also influence the spine and brain (Beguma and P 2020). The disease is transmissible and caused by a rod-shaped bacillus known as Mycobacterium tuberculosis, which has a unique staining property and is sometimes termed as "acid-fast." Mycobacterium bovis and Mycobacterium africanum can also cause the disease, albeit rarely. When an individual with untreated sputum positive pulmonary tuberculosis coughs or sneezes, infectious droplet nuclei (the dried residue of larger respiratory droplets) are released into the air, transmitting the disease to others. Droplets produced by coughing, talking, sneezing, spitting, or singing can contain tubercle bacilli (Cao et al. 2022).

Tuberculosis stands as the foremost infectious cause to global mortality, with an estimated 4,000 deaths reported every day. It holds the 13th among the leading causes of death worldwide and represents the second most fatal infectious ailment, following Coronavirus Disease 2019 (COVID-19) but surpassing Human Immunodeficiency Virus / Acquired Immunodeficiency Syndrome (HIV/AIDS). The likelihood of active tuberculosis increases by 18 times in people with HIV. Individuals with weakened immune systems due to other diseases like HIV, malnutrition, or diabetes, or those who smoke, are also at greater risk of contracting active tuberculosis. Malnourished individuals have a 3% higher chance of falling ill with tuberculosis, and in 2020, 1.9 million new tuberculosis cases were reported worldwide due to malnutrition. In 2020, Tuberculosis claimed the lives of 1.5 million individuals, among whom 214,000 were living with HIV (Anon n.d.).

While tuberculosis can impact individuals across all age groups, its primary focus is on working-age adults. More than 95% of tuberculosis cases and deaths are concentrated in developing nations. Low- and middle-income countries encounter significant challenges in terms of ensuring widespread access to treatment and achieving successful treatment outcomes. Tuberculosis stands as a leading cause of illness and death in Ethiopia, placing among the 30 countries globally with the highest burden of TB (Anon n.d.).

Currently, chest X-rays have emerged as a more appropriate and commonly utilized technique for identifying tuberculosis. The World Health Organization's "End-TB Strategy" highlights the importance of prompt and precise tuberculosis diagnosis in patients and advocates for the use of chest radiography, also known as chest X-ray (CXR). Although CXR is a frequently used method for diagnosing pulmonary tuberculosis, the reading abilities of radiologists in regions with high tuberculosis incidence are insufficient, which can compromise the efficiency of tuberculosis screening and triage (Brady 2017).

Experienced physicians known as radiologists typically review chest radiographs to detect tuberculosis, but this process can be time-consuming (Degnan et al. 2019). Importantly, CXR images of tuberculosis can be misinterpreted as other diseases with similar radiologic patterns (Archer, Levy, and McGregor 1993), resulting in incorrect treatment and exacerbation of the patient's health condition. Radiologists examining chest x-ray images are often affected by subjective differences caused by fatigue from heavy workloads and poor image quality. In low-resource countries, particularly in rural areas, there is also a shortage of trained radiologists. Although Computed Tomography (CT) imaging of the chest can offer considerably more diagnostic insights compared to a chest radiograph, its adoption is less common among low-income populations due to its high cost (Traub et al. 2007). Consequently, more individuals would benefit from automatic and reliable tuberculosis detection from chest radiographs.

Computer-aided diagnosis (CAD) systems have a vital role to play in the widespread screening of pulmonary tuberculosis by examining chest X-ray images and medical records. The timely identification of tuberculosis from X-ray images and health records is critical for radiologists as it enhances the likelihood of successful treatment. Convolutional Neural Networks (CNNs), a type of Neural Network, have recently shown substantial progress in computer-aided classification tasks, surpassing conventional machine learning methods and producing more encouraging outcomes (Lakhani and Sundaram 2017).

The main contribution of this research is the introduction of a highly effective fusion model that combines X-ray images and patient symptom data extracted from their medical records. This fusion model incorporates preprocessing techniques, lung object detection, and parameter tuning to enhance the accuracy and reliability of tuberculosis detection. By integrating both X-ray images and relevant patient symptom data, this fusion model aims to provide a comprehensive and robust approach to tuberculosis diagnosis.

1.2. Statement of the Problems

Tuberculosis (TB) is a severe infectious disease caused by Mycobacterium tuberculosis (Mtb). According to the World Health Organization's (WHO) 2021 report, TB has emerged as one of the most severe infectious diseases affecting humanity, with approximately 10 million new cases and 1.6 million deaths recorded. The diverse outcomes of Mtb infection, ranging from latent tuberculosis infection (LTBI) to TB reactivation, poses significant challenges in diagnosing and treating the disease. Despite efforts to develop efficient and non-invasive diagnostic methods, there is a need to further investigate and address these challenges. The impact of TB on global health and the limitations in current diagnostic approaches highlight the importance of finding innovative solutions to improve TB diagnosis and management (Gough et al. 2022).

The increasing use of radiological imaging exams in the digital era has placed a growing workload on radiologists, leading to fatigue, burnout, and an increased likelihood of errors. To cope with this demand, an average radiologist may need to interpret an image every 3-4 seconds over the course of an 8-hour workday (Dunnmon et al. 2019). Additionally, there is a shortage of healthcare experts who can analyze the vast amounts of healthcare data that is being generated. Early computer-aided diagnosis (CAD) applications have been developed for TB analysis using chest X-rays, without effectively integrate additional clinical variables. Relying solely on imaging features is insufficient to differentiate alternative diagnoses, and achieving an accurate diagnosis necessitates the integration of clinical and laboratory data.

Therefore, this study aims to address these critical gaps in TB diagnosis and management by investigating innovative solutions. By integrating advanced techniques, such as lung object detection and multimodal fusion, into the diagnostic process, we aim to enhance the accuracy and efficiency of TB diagnosis. This research seeks to bridge the current gap between radiological imaging and clinical information, ultimately contributing to improved TB diagnosis and patient care. Furthermore, by integrating radiological imaging findings with clinical information, healthcare experts will be able to promptly assess and treat TB cases, leading to the development of personalized treatment plans and improved disease management.

1.3. Research Questions

The study is designed to address the existing problems by formulating the following questions mentioned on *sub section 1.2*.

- ***Q1***. Can the accuracy of tuberculosis detection be improved by integrating the technique of lung object detection during the diagnostic process?
- ***Q2***. Which approach, individual modality classification models or fusion techniques, enhances the accuracy of the tuberculosis classification model?
- ***Q3***. If the performance of multimodal fusion is improved, what is the most effective multimodal fusion technique to enhance the accuracy of tuberculosis diagnosis?

1.4. Objective of the Study

1.4.1. General Objective

The general objective of the study is to develop a Deep Learning-based multi-modal model for diagnosing TB patients using both chest X-ray images and their medical history.

1.4.2. Specific Objectives

The study aims to:

- Preprocess the collected dataset
- Build separate classification models for symptom data and radiology image data
- Build a multi-modal fusion classification model
- Evaluate models performance

1.5. Significance of the research

The research on multimodal tuberculosis disease detection using convolutional neural networks is crucial to various stakeholders. Firstly, it is significant for tuberculosis patients in Ethiopia, where the number of cases is high. This research aims to develop a reliable Computer-Aided Diagnosis (CAD) system to address this issue. By utilizing a multi-modal fusion technique, the proposed system can aid physicians in diagnosing pulmonary tuberculosis with higher accuracy and in less time. This will improve patient outcomes by enabling early treatment. Secondly, the CAD system can ease the workload of physicians by enhancing diagnosis precision and reducing subjectivity. It will serve as a valuable tool for physicians, enabling accurate and timely diagnoses and more effective treatment plans.

Involvement of radiologists and technicians in data collection is crucial for obtaining high-quality chest x-ray images. Their contributions improve the overall quality of tuberculosis diagnosis. Additionally, researchers and academia can benefit from this study as it provides insights for creating automated systems to identify pulmonary tuberculosis. The research findings and methodology offer guidance for future studies in tuberculosis diagnosis and treatment. Lastly, the healthcare system in Ethiopia lacks a reliable CAD system for tuberculosis diagnosis. This research aims to bridge this gap and strengthen the healthcare system by improving accuracy and efficiency in diagnosis.

1.6. Scope and limitations of the research

The main objective of this study is to determine the prevalent cases of pulmonary tuberculosis by employing automatic multi-modal fusion deep learning models using chest X-ray images and clinical symptoms of patients. However, it excludes other illnesses like pneumonia, lung cancer, and COVID-19, as well as other types of diagnostic tests such as sputum, skin, and CT scans.

1.6.1. Scope of the Study

The study is limited to the classification of pulmonary tuberculosis disease, utilizing multi-modal fusion deep learning models. The models analyze chest X-ray images and clinical patient symptoms. A new dataset was created by gathering daily confirmed TB cases at Wolkite University Specialized Hospital. Finally, the model's performance is evaluated using a variety of assessment techniques.

1.6.2. Limitation of the Study

During our study, we encountered difficulties in finding an appropriate public dataset for multimodal tuberculosis classification. To overcome this, we gathered heterogenous data from Wolkite University Specialized Hospital. As we continued our research, we faced numerous challenges in implementing our multimodal tuberculosis detection models, one of which was the computing power required for training the model. Additionally, we had to carefully consider the resources necessary for deploying the models, including integrating it into healthcare systems. Throughout this process, we prioritized ethical and legal considerations to protect patient privacy and maintain ethical standards.

1.7. Application of the Study

The study focuses on creating a system that combines heterogeneous data to detect tuberculosis disease. By utilizing artificial intelligence and machine learning methods, the system aims to minimize human errors and enhance the accuracy of disease diagnosis, ultimately improving clinical outcomes. The system has the capacity to significantly benefit the healthcare industry, especially in regions with a scarcity of experienced radiologists or experts in CXR imaging. By detecting TB disease early, individuals and communities can benefit from timely treatment and prevent its spread. The suggested study can also act as a basis for forthcoming research endeavors and improvements in the realm of disease detection through the utilization of multimodal fusion techniques. In general, this study holds significant promise for improving healthcare outcomes and reducing the impact of tuberculosis in affected communities. The discoveries from this study can be implemented in digitalized healthcare systems, facilitating the monitoring of data over time and offering valuable insights for more effective disease management and prevention strategies.

1.8. Organization of the Thesis

The overall workflow of the thesis is depicted in *Figure 1.1*. The study is structured into five chapters, with each chapter dedicated to exploring a crucial aspect of the investigation:

Chapter One: This chapter presents an introduction to the research by offering the necessary information to comprehend the topic and providing a compelling rationale for the research problem.

Chapter Two: This chapter incorporates a comprehensive examination of relevant background information, which provides a theoretical foundation for the study. The literature review follows a systematic approach to data processing, which involves conducting a literature search and screening, extracting and analyzing data, and finally, composing the literature review.

Chapter Three: This chapter offers a comprehensive explanation of the research methodology, starting from collecting the dataset and ending with evaluating the model's performance using suitable performance metrics.

Chapter Four: This chapter outlines the study's findings, offers a detail description of the proposed approach, and tackles the challenges raised in Chapter One. It contains a comprehensive analysis and interpretation of the results, emphasizing their importance.

Chapter Five: It focuses on providing a detailed explanation of how the proposed solution will be implemented. To give an overview of the implementation's source code, numerous code samples are included. Furthermore, it wraps up the study by providing an overview of the key findings, deriving a summary from the outcomes, and offering suggestions for future research.

Figure 1.1 Workflow of Thesis Organization

CHAPTER TWO

2. Literature Review and Related Works

2.1. Introduction

During regular clinical examinations, patients may examine digital data in many formats, including pathological, radiological, and camera images, as well as laboratory test results and clinical data. This diverse set of data can provide different perspectives on the same patient, which can aid in making clinical decisions such as disease diagnosis and prognosis. However, these decisions can be subjective and qualitative, and may vary significantly between different people. To address this, deep learning methods have been developed for multi-modal learning in medical applications, which is made possible by the rapid progress in artificial intelligence technologies (Cui et al. 2023).

This chapter is structured into thirteen subsections, starting with an introduction, followed by a detailed discussion on TB and its microbiology, pathophysiology, and anatomical pathology. The section on TB also covers its screening mechanisms and the challenges associated with its diagnosis. The subsequent section provides an overview of deep learning models and evaluation methods, with a specific focus on CNNs, overfitting, regularization techniques, and strategies to avoid the black box features of CNNs. The chapter also discusses data splitting techniques and evaluation methods used for assessing the performance of deep learning models. The subsequent sections focus on medical data processing, including image processing and symptom processing. The chapter also covers the concept of heterogeneous data fusion, discussing different types of data fusion and their applications in deep learning. Additionally, the chapter offers a summary of the main findings and contributions of the literature review, along with an overview of related works.

2.2. Tuberculosis

TB is an airborne bacterial infection caused by Mycobacterium tuberculosis. It is the most prevalent infectious disease worldwide and causes more fatalities than HIV/AIDS. While TB primarily affects the lungs, it can also attack other parts of the body (Ma et al. 2018). The WHO has identified TB as one of the most contagious and widespread diseases that can lead to death. The transmission of TB occurs through various means, including coughing, spitting, sneezing, and contaminated water. Pulmonary tuberculosis happens when the

bacteria inject the lungs, whereas extrapulmonary tuberculosis occurs when the bacteria affect other organs like the spine and bones. Fever, weight loss, cough, and night sweats are among the main clinical symptoms associated with tuberculosis. Early diagnosis is crucial as these symptoms can be life-threatening if left untreated.

2.2.1. Pulmonary TB

Pulmonary tuberculosis (PTB) is a bacterial infection primarily impacting the lungs, although it can extend its effects to other areas of the body. PTB symptoms includes chest pain, a persistent cough lasting more than three weeks, coughing up blood, fever, night sweats, and weight loss. Children and individuals with weakened immune systems are more susceptible to PTB (Vinnarasi and Saravanabavan 2017). There are two types of PTB (Loddenkemper, Lipman, and Zumla 2016): primary PTB, which affects people who have never had TB before, and post primary PTB, which affects people who have previously been infected with TB. Primary PTB is commonly observed in children and individuals in high TB prevalence countries, while post primary PTB usually affects adults and can occur many years after exposure to infectious TB.

2.2.2. Extra Pulmonary Tuberculosis

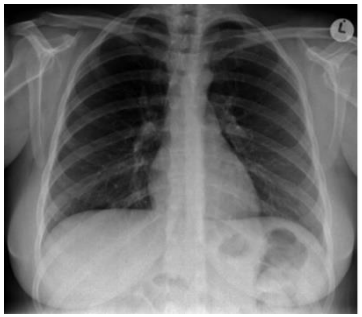
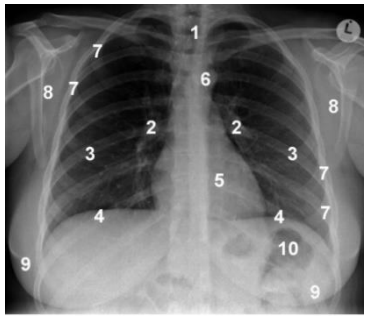
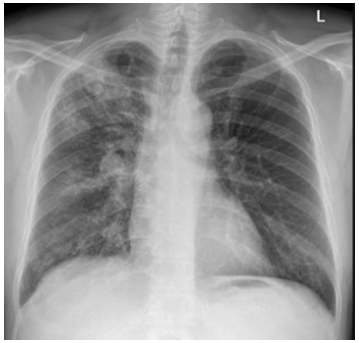
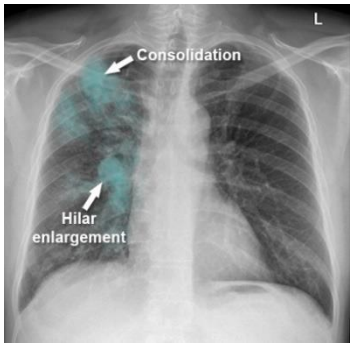
Extra-pulmonary tuberculosis is a type of tuberculosis that is identified through either bacterial or clinical evidence, and it impacts organs in the body apart from the lungs. These organs are pleura, lymph nodes, abdomen, genitourinary tract, skin, bones, joints, and the lining of the brain (Djannah et al. 2022). It is commonly believed that the majority of EPTB cases occur due to the reactivation of a latent infection that was acquired during a primary infection that may have occurred several years ago (Peirse and Houston 2017).

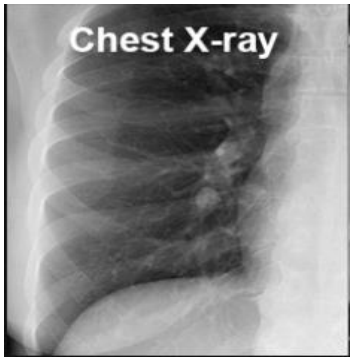
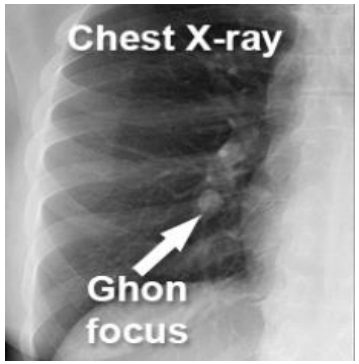
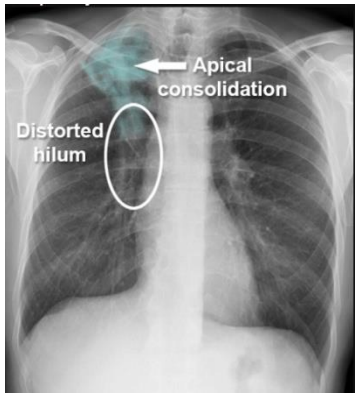
2.3. Anatomical Pathology of TB

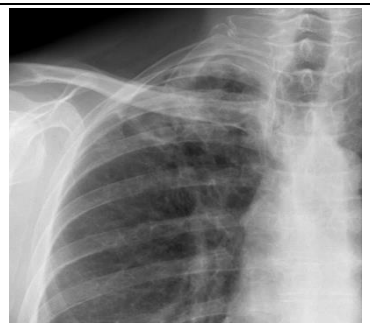
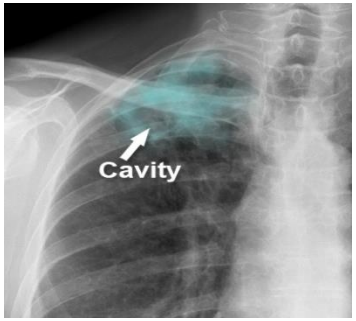
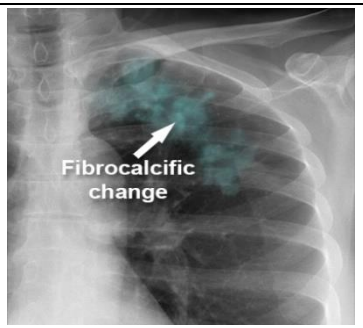
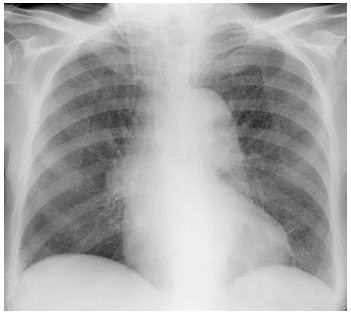
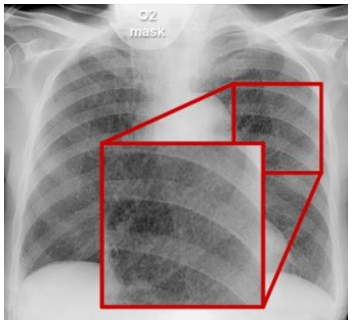
2.3.1. Chest X-ray anatomy

A chest X-ray can easily display numerous chest structures, while some significant structures, such as the pleura, are only noticeable when they are not normal, and a few structures like the phrenic nerve cannot be seen at all. *Table 2.1* below shows the presentation of CXR anatomy, as well as various stages of TB, including possible TB abnormalities in the lungs. (Lloyd n.d.).

Table 2.1 CXR-anatomy and lung abnormalities

Manifestation Stage	***Anatomy***	***Findings***	***Description***
Normal chest X-ray			▪ 1 - Trachea ▪ 2 - Hilum (hila) ▪ 3 - Lungs ▪ 4 - Diaphragm ▪ 5 - Heart ▪ 6 - Aortic knuckle ▪ 7 - Ribs ▪ 8 - Scapulae ▪ 9 - Breasts ▪ 10 - Bowel gas
Primary TB			▪ There are no radiological features which are in themselves diagnostic of primary mycobacterium tuberculosis infection (TB) but a chest X-ray may provide some clues to the diagnosis ▪ This image shows consolidation of the upper zone with ipsilateral hilar enlargement due to lymphadenopathy

Manifestation Stage	***Anatomy***	***Findings***	***Description***
Healed primary TB			▪ Following an immune response to primary infection, granuloma forms which calcifies over time – this is known as a 'Ghon focus' – TB has gone! ▪ A Ghon focus is a rounded, well-defined focus of calcific density (as dense as bone) usually located in the periphery of the lung, shows a large, rounded calcified focus near the right hilum
Post-primary TB			▪ Post-primary TB (secondary TB or reactivation TB) is more common in immunocompromised individuals ▪ The upper lobes are more commonly affected ▪ Consolidation often extends to the hilum ▪ The hilar structures may be distorted due to volume loss of the upper lobe

Manifestation Stage	***Anatomy***	***Findings***	***Description***
Post-primary TB – Lung cavity			▪ (Same patient as image above – 4 months later) ▪ Cavities are a common finding in mycobacterial infection
Healed post-primary TB			▪ Following an immune response to post-primary infection, the affected area often becomes scarred (fibrotic) and calcified ▪ The combined fibrosis and calcification can be described as ‘fibro-calcific change’
Miliary TB			▪ Miliary TB is due to disseminated spread of mycobacterial infection ▪ It can occur either at the time of primary infection or on disease reactivation – prognosis is poor ▪ Very fine nodules are typically seen scattered throughout the lungs

2.4. Tuberculosis Screening Mechanisms

TB screening is a crucial procedure used to identify individuals who may be infected with TB disease. TB can be contagious and can cause severe health issues if not treated promptly. There are several approaches to TB screening, including chest X-rays, which quickly image the lungs to detect any abnormalities, chest X-rays have been a crucial diagnostic tool in the past, particularly for pulmonary TB (World Health Organization 2016); Tuberculin Skin Tests (TSTs) are a procedure where a small amount of Purified Protein Derivative (PPD) is injected into the skin of the forearm. After 48-72 hours, the area is examined to check for any response or reaction (Pai and Behr 2016); Interferon-Gamma Release Assays (IGRAs) are tests that evaluate the release of interferon-gamma in response to specific TB antigens. These tests are more accurate than Tuberculin Skin Tests (TSTs) (Nienhaus, Schablon, and Diel 2008); and molecular diagnostics, which use nucleic acid amplification methods to identify the existence of TB DNA in medical samples and are more sensitive than chest X-rays, able to detect TB in the initial phases of infection (Boehme et al. 2010).

2.5. Diagnosis of TB

Mycobacterium tuberculosis is a bacterium that causes tuberculosis, a communicable disease that mostly affects the lungs. To diagnose TB, a combination of clinical, radiological, and laboratory tests are necessary (Anon 2020). The first step involves performing a comprehensive medical history and physical examination to identify factors that may elevate the likelihood of contracting tuberculosis, such as a past history of TB exposure, recent travel to regions with a widespread occurrence of TB, or having HIV infection (Menzies, Pai, and Comstock n.d.) (Sulis et al. 2014). Radiological exams, such as chest X-rays, play a significant role in identifying pulmonary TB, as they can reveal characteristic lung changes like nodules, consolidation, and cavitation. However, chest X-rays may appear normal in as many as 10% of cases, particularly during the early stages of the disease (van Zyl-Smit et al. 2009). Laboratory tests for TB include sputum microscopy, culture, and molecular tests such as polymerase chain reaction (PCR). Sputum microscopy involves staining sputum samples with specialized dyes and analyzing them under a microscope to identify acid-fast bacilli (AFB), which are unique to M. tuberculosis (Denkinger et al. 2014) (Cobelens et al. 2012). Culture involves cultivating the bacteria from a sputum sample on specific media, which can take a few weeks (Lawn and Nicol 2011). Molecular tests such as

PCR can identify M. tuberculosis DNA in sputum samples within hours, enabling quick diagnosis (Breen 2004).

2.6. Challenges of Tuberculosis

Tuberculosis is a significant public health issue on a global scale, especially in developing countries. It is a primary cause of mortality worldwide, with approximately 10 million new cases and 1.4 million deaths reported in 2019 (Anon 2020). Factors such as poverty, overcrowding, malnutrition, and weak healthcare systems contribute to its prevalence in these regions (Lönnroth and Raviglione 2016). A significant hurdle in TB control is accurate diagnosis, with sputum microscopy, the most common method, having limited sensitivity and specificity, especially in HIV-positive individuals and children (Anon 2023). The accessibility of recent innovations such as the Cepheid Xpert MTB/RIF assay is constrained in resource-limited areas due to its high cost and the requirement for sophisticated laboratory equipment, despite its ability to provide quicker tuberculosis detection and assess rifampicin resistance. Furthermore, certain TB forms, like extra-pulmonary TB, cannot be identified through sputum-based tests, adding to the diagnostic challenge (Zumla et al. 2013).

2.7. Machine Learning and Deep Learning

2.7.1. Machine Learning

Machine Learning (ML) is a well-established concept in computer science that is constantly advancing and developing. It involves using machines to replicate human behavior and capacity, and is considered a technology that can generate meaning from data and answer queries. ML merges principles from statistics and computer science, empowering computers or machines to execute particular tasks. In contrast to humans, ML has the capability to swiftly acquire knowledge and adjust its performance accordingly. It represents a subset of AI that employs various approaches, models, and algorithms to glean insights and patterns from data, facilitating the extraction of knowledge and enabling the development of AI applications. Machine learning comprises a collection of algorithms that leverage historical datasets for training, enabling the generation of predictions for future predictions. The increasing advancement of technology and the internet is generating an enormous amount of data, which provides the machine with a vast amount of information to learn from and evaluate (El-Yaniv and Souroujon 2003).

2.7.2. Deep Learning

Deep learning, emerging in 2006, revolutionized machine learning by using multi-layered structures and layer-wise training to improve data representation. Inspired by the learning mechanism of the human brain, this approach integrates feature extraction and categorical regression into a unified model, eliminating the necessity for artificial feature engineering. Deep learning encompasses generative deep architectures (e.g., Deep Boltzmann Machine), discriminative deep architectures (e.g., Convolutional Neural Networks), and hybrid deep architectures (e.g., Recurrent Neural Networks). It excels at extracting contextual and global features, making it ideal for solving complex problems. Recent research has yielded adaptable and customizable models tailored to specific needs (Omar, Nasim, and Izharuddin 2021).

2.7.3. Convolutional Neural Network

A Convolutional Neural Network (CNN) is particularly suitable for data that has a grid-like structure, such as images and videos, as well as for sequential data types like time series or one-dimensional categorical data. For two-dimensional data, like images, CNNs use convolutional filters to recognize patterns, like edges and textures, by sliding over the input and performing dot products with filter weights. For one-dimensional categorical data, like text or DNA sequences, CNNs also use convolutional filters on a one-hot encoded representation of the input sequence. The structure of a CNN includes convolutional layers, pooling layers, and fully connected layers (Kiranyaz et al. 2021).

2.7.3.1. CNN-1D VS CNN-2D

The computational complexity varies notably between 1D and 2D convolutions. Convoluting an N x N image with a K x K kernel demands substantial computation, whereas the equivalent dimensions in a 1D convolution do not entail the same level of computational intensity. This suggests that a 1D CNN has a much lower computational complexity than a 2D CNN with the same design, network, and hyper parameters. Most 1D CNN applications use compact setups with 1-2 hidden CNN layers and networks that have 10 K parameters. On the other hand, 2D CNN applications generally use deep architectures with more than 1 M (sometimes exceeding 10 M) parameters. Training deep 2D CNNs often requires specialized hardware such as Cloud computing or GPU farms. In contrast, compact 1D CNNs with few hidden layers (2 or less) and neurons (50 or fewer) can be trained on a typical

computer's CPU relatively quickly. Because they have cheap processing needs, compact 1D CNNs are highly relevant for real-time and low-cost applications, especially on mobile or handheld devices. In situations where there is little labeled data and large signal fluctuations from various sources (such as patient ECG, civil, mechanical, or aerospace structures, high-power circuitry, power engines or motors, etc.), compact 1D CNNs have shown better performance (Kiranyaz et al. 2021).

2.7.4. Under fitting and Overfitting

The aim of a learning algorithm is to acquire knowledge about a concept or function, also called a model that characterizes the training data and can work well on new, independent data. To accomplish this, the algorithm should avoid both under fitting and overfitting. Overfitting occurs when the algorithm excessively adapts to patterns in the training data, resulting in high accuracy during training but suboptimal performance during testing and validation. conversely, underfitting occurs when the algorithm is not sufficiently responsive and overly simplistic, resulting in an inability to accurately model the training data. As a result, its performance is expected to be poor when applied to the testing dataset. Achieving an ideal balance between underfitting and overfitting is important for successful machine learning engineering. Poor performance of machine learning algorithms is usually caused by inadequate fitting (McCoubrey et al. 2021). *Figure 2.5*, illustrates under fitting vs overfitting.

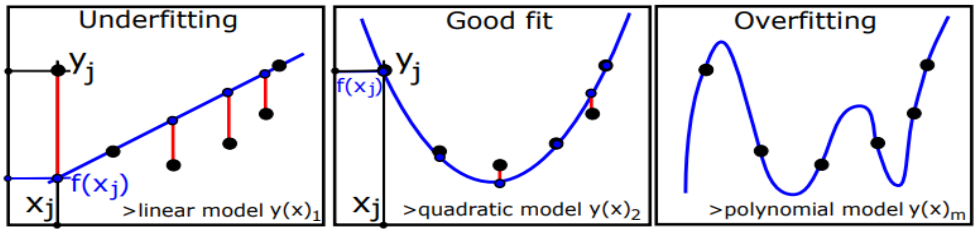


Figure 2.1 Under fitting vs. Overfitting (Shao et al. 2017)

2.7.4.1. Regularization Techniques

In machine learning, regularization (Murugan and Durairaj 2017) is employed as a method to counteract overfitting in models. It operates by giving a penalty term into the loss function, discouraging the model from assigning excessive significance to any specific feature. Various regularization techniques exist, including L1 and L2 regularization and dropout. L1 regularization imposes a penalty that is directly relative to the absolute value of the weights present in the model. This motivates the model to select only the most significant features

and reduce the weight of the remaining features to zero. In contrast, L2 regularization imposes a penalty proportional to the squared value of the weights, which encourages the model to allocate the importance of the features more evenly. Dropout is a technique that randomly drops out some neurons during training, which prevents the model from becoming too reliant on any one particular set of neurons and encourages it to learn more robust features.

2.7.5. Data Augmentation

Data augmentation is a method employed in machine learning and computer vision to increase a dataset by generating new data instances from existing ones. It proves valuable when data is scarce or when a model is susceptible to overfitting (Shorten and Khoshgoftaar 2019). This method involves applying various modifications, such as cropping, rotating, scaling, flipping, and adjusting attributes like hue and brightness, to diversify the data (Ait Nasser and Akhloufi 2023a). These augmentations create additional samples with slight variations. For imbalanced datasets, techniques like SMOTE (Synthetic Minority Over-sampling Technique) can be utilized to generate synthetic samples for the minority class by interpolating attributes from nearest neighbors, rebalancing the dataset and preventing overfitting (Elreedy, Atiya, and Kamalov 2023a), prevents loss of information from the majority class and impact the model's ability to learn patterns and generalize effectively.

2.7.6. Performance Metrics

Assessing how well a deep learning model can accurately classify input data is called evaluation. Binary classification is a frequent type of issue in deep learning where the model is taught to classify input data into one of two categories. There exist various ways to evaluate binary classification problems in deep learning (Tohka and van Gils 2021). Some of the commonly used performance measures for deep learning algorithms in healthcare are:

- **Accuracy**: refers to the ratio of correct predictions made by a model to the overall number of predictions made (Kim 2016). It is frequently utilized in healthcare contexts, such as medical image analysis and disease diagnosis. The formula for accuracy is:

$$Accuracy = \frac{True\ Positive+True\ Negative}{True\ Positive+False\ Positve+True\ Negative+False\ Negative} X\ 100 \qquad 2.1$$

Where:

True Positive: the number of individuals with the condition who are accurately identified as positive by the test.

False Negative: the number of individuals with the condition who are mistakenly identified as negative by the test.

True Negative: the number of individuals without the condition who are accurately identified as negative by the test.

False Positive: the number of individuals without the condition who are mistakenly identified as positive by the test.

- **Sensitivity (Recall**) **and Specificity**: Sensitivity evaluates the ratio of true positives (i.e., positive cases correctly identified) to all actual positive cases, whereas specificity evaluates the ratio of true negatives (i.e., negative cases correctly identified) to all actual negative cases (Saito and Rehmsmeier 2015). These measures are particularly valuable in healthcare contexts where false positives or false negatives have significant implications, such as TB screening.

$$Sensitivity\ (Recall)\ = \frac{True\ Positive}{True\ Positive + False\ Negative} X\ 100 \qquad 2.2$$

$$Specificity\ = \frac{True\ Negative}{True\ Negative + False\ Positive} X\ 100 \qquad 2.3$$

- **Precision:** Precision is a metric that assesses the ratio of true positives to all positive predictions made by the model, whereas recall assesses the ratio of true positives to all actual positive cases (Davis and Goadrich 2006).

$$Precision\ = \frac{True\ Positive}{True\ Positive + False\ Positive} X\ 100 \qquad 2.4$$

- **Area under the Receiver Operating Characteristic Curve (AUC-ROC):** is a metric that gauges the model's overall ability to differentiate between positive and negative cases, irrespective of the chosen threshold (Fawcett 2006). AUC-ROC is frequently employed in healthcare settings where striking the balance between sensitivity and specificity is crucial, such as forecasting the risk of cardiovascular disease.
- **F1-score:** is a metric that is the weighted harmonic mean of precision and recall, with both measures given equal importance (Powers 2020). This score is valuable in healthcare scenarios where both false positives and false negatives carry equal significance, such as predicting patient outcomes.

The F1-score is calculated as:

$$F1 - score = 2 * \frac{precision * recall}{precision + recall} \quad 2.5$$

Where:

Precision: the ratio of correctly identified positive predictions to the total number of positive predictions

Recall: the ratio of accurately predicted positive instances to the total number of actual positive cases

2.8. Sampling Methods for Dataset Splitting

Data splitting is the process of dividing a dataset into smaller groups or subsets, commonly referred to as a training set and a testing set. This division allows for the evaluation of a deep learning model's performance on new, unseen data. By doing so, it helps to prevent overfitting, where the model becomes too specialized in the training data, and ensures that the model can generalize well to new, unseen data. Random sampling, like an 80-20 split, is common but may yield biased splits if data isn't evenly distributed (Joseph 2022). Advanced techniques like Stratified Sampling, which maintains class ratios, K-fold Cross-Validation, where data is split into k subsets for precise performance estimation, and Leave-One-Out Cross-Validation, using each sample for testing, offer more robust alternatives to traditional data splitting methods (Kuhn and Johnson 2013).

2.9. Medical Data Processing

2.9.1. Image Processing

Digital image processing involves manipulating images with a digital computer, using mathematical and statistical techniques to extract meaningful information or modify them for specific purposes. Images are composed of pixels, each having a location and value representing gray levels or colors. A digital image is finite and discrete in terms of its spatial coordinates, making it suitable for processing (Shawal, Shoyab, and Begum 2014). There are three main types of digital images: Binary Images, with only two values (typically black and white); Grayscale Images, with varying intensity levels; and True Color Images, which use three values to define red, green, and blue components. Grayscale and True Color images find applications in medical imaging, digital photography, and computer vision (Kumar and Verma 2010).

Image processing is broadly categorized into three types. Low-level processing (LLP) involves basic pixel-level operations like noise reduction and contrast enhancement, applied to raw image data. Medium-level processing (MLP) identifies image features such as texture and object boundaries, using advanced operations like segmentation and pattern recognition. High-level processing (HLP) utilizes techniques like artificial intelligence and machine learning for complex tasks such as object recognition and image classification, commonly used in medical image analysis and autonomous vehicles (Dastres and Soori n.d.). In medical image processing, various imaging modalities like MRI, CT, X-Rays, Ultrasound, PET, and SPECT are employed. Key techniques include preprocessing, enhancement, segmentation, feature extraction, and classification, catering to physician requirements and disease type (Chinmayi, Agilandeeswari, and Prabukumar 2018). These methods are illustrated in *Figure 1*.

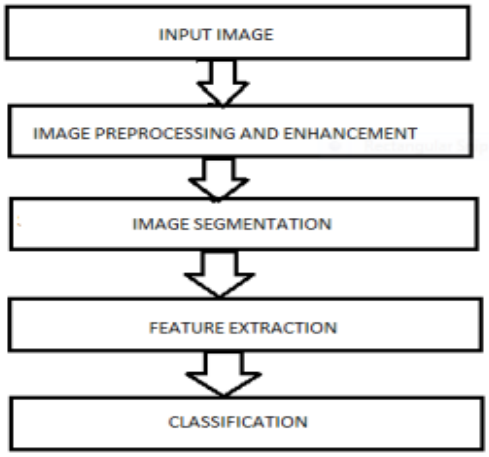


Figure 2.2 Steps in image processing (Antony 2017)

2.9.1.1. Preprocessing Techniques

The procedure of improving the quality of X-ray images by converting them into a more useful and informative form is called preprocessing. DICOM (Digital Imaging and Communications in Medicine) is the format in which most CXR images are produced and it contains a large number of metadata, making it difficult for non-radiology experts to understand. In computer vision, DICOM images are typically stored in PNG or JPG formats using specialized algorithms that compress the images without sacrificing critical information. Preprocessing involves two primary steps: first, de-identifying patient information to protect their privacy, and second, converting DICOM images to PNG, JPEG, or other formats. Radiological images typically have large dimensions of 3000 × 2000 pixels,

which can be computationally intensive to work with in their original size. Thus, it is necessary to resize the images while preserving their essential information. Many datasets already contain resized images, such as the Indiana dataset with CXR images resized to 512 × 512 pixels and the Chest X-ray dataset with images resized to 1024 × 1024 pixels (Ait Nasser and Akhloufi 2023b).

2.9.1.2. Segmentation Techniques

Medical image segmentation methods are essential for isolating specific structures or regions of interest within medical images. Common techniques include thresholding, region growing Active Contours, and Pre-trained Deep Learning Models. Thresholding sets a value to distinguish objects from the background based on pixel intensity (Chakraborty et al. 2021). Region growing groups connected pixels based on predefined criteria (Dabass, Vashisth, and Vig 2018). Active Contours use prior knowledge and grayscale/edge data to achieve stable segmentation by minimizing an energy function (Chen et al. 2023). Pre-trained Deep Learning Models, like VGG16, have enhanced image segmentation, with semantic segmentation classifying pixels and instance segmentation outlining objects of interest (Malhotra et al. 2022). Traditional CNN models like UNet and SegNet have improved through pre-trained DL models, serving as encoders and decoders for feature extraction and segmentation (Kadry et al. 2022). For more details on the VGG-UNet approach, refer to (Rajinikanth, Kadry, and Nam 2021).

2.9.1.3. Feature Extraction Techniques

Deep learning has revolutionized medical image analysis, offering accurate automated techniques for feature extraction, especially in tasks like x-ray diagnosis. Convolutional Neural Networks (CNNs), autoencoders, and Transfer Learning have surpassed traditional methods (Litjens et al. 2017). CNNs use convolutional and pooling layers to extract spatial features like edges, textures, and shapes from x-ray images. Autoencoders, typically used for image reconstruction, can extract relevant features from x-ray images during training for further analysis. Transfer Learning involves fine-tuning pre-trained models, such as CNNs, on a target dataset of x-ray images to achieve precise disease classification (Wang et al. 2017). These deep learning techniques have highly enhance the accuracy and effectiveness of medical image analysis, including detection, segmentation, and classification (Srinivas, Roy, and Mohan 2016).

2.9.1.4. Classification Techniques

The technique of using deep neural networks to classify x-ray images, known as deep learning based x-ray image classification, is commonly used in medical imaging (Rajpurkar et al. 2017). The goal is to automatically detect and diagnose a variety of medical conditions from x-ray images, including pneumonia, tuberculosis, and COVID-19. To achieve this, various methods are employed, including CNNs, transfer learning, and ensemble learning. CNNs are a type of neural network that are specifically designed to process and extract features from images (Krizhevsky, Sutskever, and Hinton 2017). Transfer learning involves using pre-trained models as a starting point for a new classification task, which helps the model learn from new data more efficiently and accurately (Shin et al. 2016). Finally, ensemble learning involves combining multiple models to enhance the classification performance (Wang et al. 2021).

2.9.2. Symptom Processing

In healthcare, categorical data pertains to information that can be sorted into distinct groups or categories that are mutually exclusive and collectively exhaustive. This means that each observation can only belong to one category, and all possible categories are covered (Imrey and Koch 2005). Patient characteristics, such as age group, ethnicity, gender, and medical history (Preisser and Koch 1997), are examples of categorical data in healthcare. Extracted from medical history, patient symptom data is another example of categorical data that can be analyzed and modeled to identify relationships and patterns between symptoms and various health outcomes. To handle categorical data, preprocessing and classification techniques are commonly used.

2.9.2.1. Preprocessing

Preparing categorical data for analysis is a critical step involving the transformation of categorical variables into numerical values suitable for statistical models, known as preprocessing. This may involve data cleaning to ensure accuracy and consistency. Common preprocessing techniques for categorical data (Reilly et al. 2022) include handling missing data by replacing them with mode or median values, encoding categorical data into numerical format using methods like Label Encoding or One-Hot Encoding. Label Encoding assigns unique numerical values to categories, potentially introducing unintended ordinal relationships, while One-Hot Encoding creates binary columns for each category, avoiding

ordinal associations. Additionally, handling outliers in categorical data involves merging low-frequency categories into an "Other" category to reduce their impact and enhance data manageability for analysis.

2.9.2.2. Classification

Categorical data classification refers to the task of setting a categorical label to each instance in a dataset based on its features. Deep learning algorithms, particularly deep neural networks (DNNs), have shown great promise in healthcare categorical data classification. DNNs have the capability to automatically learn sophisticated patterns from raw data. This ability often results in enhanced performance when compared to conventional machine learning algorithms. CNNs and RNNs are two popular types of deep learning algorithms used in healthcare categorical data classification (Rajkomar et al. 2018).

2.10. Heterogeneous Data Fusion

In the modern medical field, integrating information from diverse sources is essential for providing effective care. This includes using pixel data from imaging, structured data from lab results, unstructured data from narratives, and sometimes audio or observational data. In medical image interpretation, having clinical context is vital for diagnostic decisions. Research consistently demonstrates that image interpretation without clinical and lab data leads to lower performance and reduced usefulness (Leslie, Jones, and Goddard 2000), (Cohen 2007). In a survey of radiologists, 87% emphasized the significant impact of clinical information on their interpretations (Boonn and Langlotz 2009). Notably, clinical context is vital not just in radiology but also in other specialties like pathology, ophthalmology, and dermatology (Comfere et al. 2014), (Comfere et al. 2015).

2.10.1. Types of Data Fusion

Data fusion is the process of merging information from various sources, achieved through techniques like machine learning, deep learning, or basic arithmetic operations. It occurs at different stages, categorized into three levels: early fusion, joint fusion, and late fusion (Amal et al. 2022). Early fusion combines input modalities into a single feature vector before training a machine learning model, with two types: Type I merging original features and Type II merging extracted features. Joint fusion, or intermediate fusion, combines learned feature representations from neural network layers with other modality features, back-

propagating loss for improved representations. Late fusion, or decision-level fusion, employs predictions from separate models, making a final decision using aggregation functions like averaging or majority voting. The choice of aggregation function depends on empirical evaluation and the application's requirements (Huang et al. 2020). The details of the three fusion techniques are shown below in *Figure 2.3*.

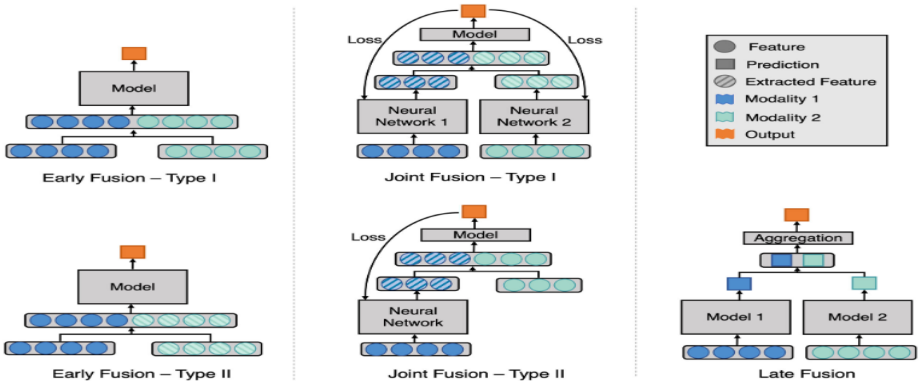


Figure 2.3 Model architecture for different fusion strategies (Huang et al. 2020)

2.10.2. Deep Learning uses for Multi-modal Fusion

Deep learning has the potential to revolutionize healthcare, particularly in medical imaging. present radiology models focus solely on pixel-based data, neglecting vital clinical information from electronic health records (EHR). Integrating this contextual data into deep learning models can significantly enhance diagnostic accuracy and support informed clinical decisions, ultimately improving patient outcomes (Huang et al. 2020). While convolutional neural networks (CNNs) have excelled at image recognition in medicine, they solely rely on pixel data, limiting their clinical utility. In various fields, multimodal deep learning models that combine pixel data with other clinical information have demonstrated success. Now, these models are considered appropriate for medical imaging applications, where the integration of comprehensive data is necessary (Hinton 2018, Person et al. 2019).

2.11. Related Works

This section discusses a range of previous research studies that have been conducted on detecting lung tuberculosis through an automated system using various methods. Several researchers have proposed different techniques for Computer Aided Diagnosis to identify lung tuberculosis. In the previous years, numerous research Investigations have been carried out in the medical domain, focusing particularly on the development of techniques for processing medical images. These studies have utilized DL. Deep learning models are

popular in medical imaging systems as they can dynamically extract features or use pre-trained networks. This section introduces and examines the research gap and the findings with a specific emphasis on the area of interest.

According to JIMMY et al. (Anon 2022), tuberculosis remains a major health threat worldwide, and the field of computer-aided diagnosis has shown considerable interest in identifying TB using chest X-ray images. Although Promising outcomes have been demonstrated by deep learning techniques. in this domain, most of them only consider images as input sources, ignoring the potential information contained in non-imaging data such as patient demographics or medical record history. Therefore, in their study, the authors proposed a multimodal ensemble model that uses both chest X-ray images and patient demographics to enhance the accuracy of TB detection. Their proposed model used EfficientNet for image classification and XGBoost for demographic variables classification, followed by a weighted ensemble model to combine the individually predicted results. A collection of 754 images was utilized by the authors (552 normal and 202 TB) collected from the daily radiology examination routines at Murni Teguh Memorial Hospital between 2014 and 2021, along with demographic variables such as age, sex, and Body Mass Index (BMI). The evaluation results showed that their approach is better than the previous novel multimodal model by increasing the area under curve (AUC) value by 0.0075 (0.0213 vs. 0.0138, respectively) and achieved high accuracy in TB detection. This study highlights the importance of using a multimodal approach that incorporates both imaging and non-imaging data to improve the accuracy of disease detection in clinical practice. Although the proposed multimodal ensemble model in the study by JIMMY et al. (Anon 2022) achieved higher accuracy and outperformed the previous novel multimodal model, the study only used a limited number of demographic variables, such as age, sex, and BMI, and did not consider other potentially relevant clinical information that could improve the accuracy of TB detection. Furthermore, it is worth noting that BMI is a population-level measure of adiposity and does not necessarily reflect individual-level body composition. For example, individuals with the same BMI may have different body fat percentages or muscle mass, which could have different implications for TB risk and disease severity.

The study (D'Souza et al. 2022) proposes a new framework for multimodal fusion of clinical, genomic, and imaging data in tuberculosis using multiplexed graphs and a new graph neural network. The proposed method outperforms existing fusion methods for multi-outcome prediction on a large TB dataset. The study uses the Tuberculosis Data Exploration Portal

and data from 3051 patients with five classes of treatment outcomes. The proposed framework uses domain-specific autoencoders to convert each modality into a compact feature space and a graph neural network to propagate information across the multiplexed graph. They use AU-ROC as the evaluation metric and report significant differences between their Multiplex GNN and baseline AU-ROC using a DeLong test. The results show that their framework outperforms common multimodal fusion baselines and provides improved performance over single modality outcome classifiers. The main drawback of this study is that it lacks a discussion on their proposed framework. Additionally, the paper does not mention the computational requirements and scalability of their proposed method, which could be important for its practical use in real-world scenarios.

Minwoo Park et al. (Park et al. 2023) studied Distinguishing nontuberculous mycobacterial lung disease and Mycobacterium tuberculosis lung disease on X-ray images using deep transfer learning. The clinical features of Non tuberculous mycobacterial lung disease (NTM-LD) and Mycobacterium tuberculosis lung disease (MTB-LD) are quite similar, which often leads to misdiagnosis and incorrect treatment. In order to address this issue, the author employed deep learning technology to differentiate between the two diseases using chest X-ray images. The study included 1082 patients with MTB-LD and 260 patients with NTM-LD, and the chest X-ray images were pre-processed using lung segmentation and image augmentation before training a deep learning model. Three different CNN models, Densenet 201, ResNet 50, and Efficientnet B4, were used to develop the model with transfer learning using ImageNet, and the models were then evaluated using various metrics including precision, recall, F1 score, accuracy, and AUROC. The EfficientNet B4 model outperformed the other models in all of the evaluation metrics.

In their study, Minwoo Park et al. (Park et al. 2023) employed the cropping method for lung segmentation. However, this method has some limitations. One major drawback of using cropping is the potential loss of important contextual information surrounding the lungs, such as the diaphragm, pleura, or other structures. Additionally, cropping may result in variable input sizes, making it difficult to train deep learning models effectively. This variability may lead to difficulties in generalizing to new images if different images are cropped to different sizes. Furthermore, manual cropping or a high degree of variability in the size and position of the lungs in the images may introduce artifacts or distortions that can negatively impact the accuracy of the segmentation.

Ekin Yagis et al. (Yagis et al. 2022) developed Ensemble Deep Learning Architectures for Automated Diagnosis of Pulmonary Tuberculosis using Chest X-ray. Employing an automated and affordable approach can assist in conducting screening assessments and detecting illnesses early in underprivileged countries. The researchers presented a deep ensemble learning framework that combines two deep learning-based techniques using multisource data for the automated diagnosis of TB. They tested the integrated model framework on three datasets, including two publicly available and one private dataset. By combining supervised prediction and unsupervised representation, the ensemble method accurately classified TB using chest radiography with an area under the receiver operating characteristic (AUROC) of up to 0.98, outperforming other tested classifiers and achieving state-of-the-art performance. While this investigation yielded good results, it was noted that the quality of the images used in lung disease classification can be improved through the use of histogram processing and automated central cropping, although these techniques may have some limitations. **1)** ***Over-correction:*** Pre-processing steps can sometimes over-correct the image, resulting in the image becoming overly bright or dark. For example, histogram processing may cause some areas of the image to become too bright or too dark, which can make it difficult to see important details. **2)** ***Dependence on Image Quality:*** The effectiveness of pre-processing techniques can be influenced by the quality of the input images. In some cases, poor quality images may not benefit from pre-processing steps or may even be negatively impacted by them.

Sana Sahar et al. (Guia et al. 2023) novel approach was introduced to detect tuberculosis in chest X-ray images by utilizing a deep learning model. As the author notes, TB is highly infectious disease caused by the bacterium Mycobacterium tuberculosis. The bacteria responsible for TB can lie dormant in the body for extended periods, and can become active when the immune system is weakened. The proposed model utilizes two pre-trained models (vgg16 and vgg19) and a block attention module to obtain spatial data. The model was trained and tested on a dataset comprising 7000 images from four publicly available datasets and was evaluated using binary classification. The authors report that the model achieved high accuracy, precision, recall, and f1-score when tested on four different datasets. The model's performance was evaluated using a test set, and the authors obtained excellent scores (0.9992) for all metrics (accuracy, precision, recall, f1-score). While the authors of the study reported achieving high scores for evaluation metrics such as accuracy, precision, recall, and f1-score, it may be argued that these metrics may not be sufficient for evaluating the

performance of a deep learning model for a healthcare binary classification problem such as TB detection. For instance, the model's ability to accurately detect true positive cases (i.e., patients with TB) while minimizing false positives (i.e., patients who do not have TB but are incorrectly classified as having the disease) is crucial for healthcare applications.

In their study, (Melendez et al. 2016) propose an automated tuberculosis screening strategy that combines X-ray-based CAD with clinical information to enhance TB detection accuracy. They evaluate this strategy on a database of 392 patient records from suspected TB subjects in Cape Town, South Africa, including CAD scores from chest X-rays and 12 clinical features. The proposed framework involves two key components: feature ranking and classification by multiple learner fusion. The feature ranking component utilizes the minimum redundancy maximum relevance (mRMR) algorithm to assess the relevance of each feature, considering both its individual performance and its relation to other features. The classification by multiple learner fusion involves the combination of random forests (RF) and extremely randomized trees (ERT), two state-of-the-art classification techniques. The fusion mechanism incorporates a second ERT to optimize the combination of outputs from RF and ERT. The evaluation results show that the combined approach outperforms individual strategies relying only on CAD scores or clinical information. It achieves a higher specificity (49%) and NPV (98%) at 95% sensitivity compared to using CAD scores or clinical information alone. The authors emphasize the potential of combining CAD and clinical information as a promising tool for TB screening, especially in resource-limited settings. The study highlights the importance of leveraging imaging and non-imaging data for effective TB detection. The proposed approach improves TB screening accuracy by considering a combination of chest X-ray images and clinical information.

The study briefly mentions the use of RF and ERT for classification but lacks a detailed explanation of these techniques. Additionally, the reliance on traditional machine learning algorithms like random forests and extremely randomized trees may limit the model's ability to capture complex patterns and relationships in the data. These algorithms are based on predefined rules and assumptions, which may not be flexible enough to handle the complexities of tuberculosis detection. Furthermore, the optimization of machine learning parameters, such as the number and maximum depth of trees, may introduce a risk of model overfitting. Therefore, a comprehensive understanding of the chosen machine-learning algorithms and machine learning optimization is essential for the robustness and generalizability of the proposed model.

Table 2.2 Summary of related works

Citation	***Author & Year***	***Title***	***Performance***	***Dataset***	***Gap***
(Park et al. 2023)	Minwoo Park et al. (2023)	Distinguishing nontuberculous mycobacterial lung disease and Mycobacterium tuberculosis lung disease on X-ray images using deep transfer learning	Precision, Recall, F1 score, AUROC and Accuracy are 0.96, 0.92, 0.94, 0.88 and 0.92 respectively	raw data	difficulties in generalizing to new images
(Guia et al. 2023)	Sana Sahar et al (2023)	Tuberculosis Detection Using Chest X-Ray Image Classification by Deep Learning	accuracy, precision, recall and f1-score of 99%	Montgomery and Shenzhen datasets	Needs further evaluation metrics
(Anon 2022)	JIMMY et al. (2022)	Detection of pulmonary tuberculosis from chest X-Ray images using multimodal ensemble method	Increasing the area under curve (AUC) value by 0.0075	Murni Teguh Memorial Hospital	This study is for specific area. Therefore, it is population dependent
(D'Souza et al. 2022)	D'Souza et al. (2022)	Fusing Modalities by Multiplexed Graph Neural Networks for Outcome Prediction in Tuberculosis	AU-ROC with $p < 0.01$	raw data	Lack of detail discussion on their proposed framework
(Yagis et al. 2022)	Ekin Yagis et al. (2022)	Ensemble Deep Learning Architectures for Automated Diagnosis of Pulmonary Tuberculosis using Chest X-ray	AU-ROC 98%	Montgomery, Shenzhen and Songklanagarind Hospital datasets	Pre-processing Issue
(Melendez et al., 2016	Melendez et al., (2016)	An automated tuberculosis screening strategy combining X-ray-based computer-aided detection and clinical information	specificity (49%) and NPV (98%) at 95% sensitivity	Cape Town, South Africa	Limited modeling capability and risk of model overfitting

In this subsection, we thoroughly examined six articles that are closely related to our proposed title. We categorized these articles based on their nature and the intended focus of the studies they presented. By doing so, we were able to identify key open issues that need to be addressed, and we have summarized these issues in *Table 2.2*. It is important to note that our study incorporates a local raw dataset about a total of 734 instances, which allows us to tackle the specific challenges highlighted in the articles we reviewed. By leveraging this dataset, we aim to provide valuable insights and potential solutions to the unresolved problems identified in the existing literature.

This study presents an innovative method for tuberculosis detection that has the potential to assist radiologists and physicians, enhancing their diagnostic capabilities and contributing to a more accessible and cost-effective solution for identifying TB cases. Unlike traditional TB detection methods that rely on predefined lung models or masks constructed by experts, this new model operates independently without such constraints. The proposed model aims to enhance tuberculosis detection by employing a multimodal approach, seamlessly integrating x-ray images and corresponding symptom records. By harnessing this technique, the model has the potential to improve the accuracy and efficiency of tuberculosis screening, ultimately resulting in earlier diagnoses and better patient outcomes. In essence, this new method holds the power to advance the scene of tuberculosis detection and diagnosis, marking a pivotal advancement in the field.

2.12. Conclusion

The chapter emphasizes the importance of utilizing deep learning techniques for multi-modal learning in medical applications, particularly in TB diagnosis. TB is a serious global public health issue that can affect various parts of the body and its early detection is crucial for effective treatment. The chapter offers the summary of the different diagnostic tools used for TB screening and explains how deep learning techniques can mimic the human brain's ability to learn. The chapter also discusses the use of CNNs for handling data in a grid-like format, such as medical images, and describes how regularization techniques can be used to prevent overfitting in machine learning. Additionally, the chapter covers different evaluation metrics for binary classification problems and data splitting techniques for model validation. The section further explores previous research studies that have attempted to detect lung tuberculosis through automated systems, deep learning methods. The section highlights research gaps and findings, focusing on the medical data processing field.

CHAPTER THREE

3. Materials and Methods

3.1. Introduction

This chapter offers a comprehensive explanation of the complete research methodology undertaken to address the questions addressed in *chapter 1*. The methodology comprises a series of comprehensive procedures that serve as a roadmap for the entire research process, from problem formulation to evaluation. It directly represents the steps involved in achieving the proposed solution. This proposed model is founded on the current knowledge in the research area, and aims to introduce a fresh and creative approach to address existing issues.

The proposed system requires specific hardware and software to achieve its objectives. For software requirements, the system utilizes various development tools, with Python being the programming language used to implement the complete solution. Additionally, a toolbox of machine learning libraries such as TensorFlow, Keras, Pandas, NumPy, Matplotlib, and Opencv is utilized for the overall implementation, evaluation, and interpretation of the system. As for hardware requirements, the system is developed on a desktop computer with specific specifications. The computer model is DESKTOP-N30UPRO, with an Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz, 16.0 GB of RAM, a 2TB HDD, and a x64-based processor and x64-bit Operating System (Windows 10). These hardware and software components are necessary to conduct the entire experimental procedures of the proposed system.

The methodology utilized in this research to address the research question and objective is outlined in this section. The process begins with a comprehensive literature review (*sub section 3.2*) is conducted to identify relevant studies and establish the foundation for the research. This is followed by defining the problem and objective of the study (*sub section 3.3*), which clarifies the specific focus and goals of the research. The data collection process (*sub section 3.4*) is then described, including an explanation of the datasets required for the proposed solution. The proposed system (*sub section 3.5*) is presented, outlining the conceptual framework and approach taken in the research. To support the implementation of the proposed system, the proposed system requirements (*sub section 3.6*) are detailed. Next, the dataset preparation (*sub section 3.7*) is discussed, which involves several preprocessing steps (*sub section 3.8*). To optimize the performance of the proposed system,

hyperparameter optimization (*sub section 3.9*) is conducted to fine-tune the parameters. The subsequent subsections focus on the classification process (*sub section 3.10*). To evaluate the performance of the proposed system, a performance evaluation (*sub section 3.11*) is conducted. Finally, the section concludes with a conclusion (*sub section 3.12*) summarizing the key findings and implications of the research.

3.2. Literature Review

In order to determine the relevant entities involved, goals, observed occurrences, assessment methods, diagnostic criteria, ways of presenting information, root causes, mechanisms, and underlying reasons, it is necessary to extract information systematically from previous studies. As a result, an extensive review of literature was conducted on both image processing and symptom processing to cover all aspects of the system. In order to gain a more comprehensive understanding of the topic, various sources were analyzed, including relevant journals, books, case studies, previous research works, and guidelines.

To conduct the literature review, the snowball approach was used for searching and analyzing articles, beginning with a broad research area and progressively focusing on specific articles within the field. The search process was guided by three key-phrases, namely "preprocessing, classification, and multi-modal fusion for screening TB in CXR images and symptom feature. In order to collect suitable resources for the study on TB detection using chest x-ray image analysis and related symptoms, a customized search query was employed to explore various reference databases. The retrieved references were then manually assessed based on pre-determined inclusion and exclusion criteria. The study utilized PubMed, IEE Xplore, Google Scholar, ScienceDirect, Springer, and Elsevier reference databases to explore relevant literature regarding the identification or diagnosis of tuberculosis using chest X-ray images along with the related symptoms.

3.3. Defining Problem and Objective of the Study

The proposed study aims to address the research questions mentioned in *subsection 1.4* by investigating the integration of lung object detection technique during the diagnostic process to enhance the precision of tuberculosis detection. Additionally, the study aims to compare the effectiveness of individual modality classification models and fusion techniques in enhancing the accuracy of the tuberculosis classification model. Furthermore, if the

performance of multimodal fusion is improved, the study seeks to identify the most effective multimodal fusion technique to enhance the accuracy of tuberculosis diagnosis.

The goal of this research is to examine and evaluate different multi-modal fusion techniques to optimize the accuracy measure for tuberculosis screening using x-ray images and patient symptom records. The study also aims to identify and analyze various performance metrics that can be used to measure the model's performance. By achieving these objectives, the study intends to contribute to the advancement of tuberculosis diagnosis techniques and provide valuable insights for improving the accuracy of tuberculosis detection.

3.4. Data Collection

To gather a complete and thorough dataset that would allow for an analysis of the relationship between tuberculosis chest x-ray images and patient symptoms, we conducted a data collection at Wolkite University specialized hospital. The collection process spanned a year, and ultimately, we obtained 734 records of chest x-ray images and corresponding patient symptom data. This dataset will serve as the foundation for our analysis, which will utilize various statistical techniques to identify the pattern of the disease by merging chest x-ray images with the associated patient symptoms. By examining this data, we hope to achieve better results for tuberculosis diagnosis, ultimately leading to improved patient outcomes.

In our research, we engaged the expertise of three domain experts: a TB clinical doctor, a radiologist, and a patient record manager. The TB clinical doctor provided us with a general overview of the disease and treatment steps, giving us detailed explanations about the clinical features of tuberculosis disease. The radiologist doctor and two radiology technicians participated in the data collection process and guided us through the general steps of the x-ray image process. Their expertise ensured that the collected chest x-ray images were of high quality and relevant to our study. This collaboration was essential in obtaining accurate and reliable chest x-ray images for analysis. Additionally, the patient record manager played a vital role in helping us organize and manage the patient symptom data. Their expertise ensured the accuracy and completeness of the patient symptom data, which is crucial for our analysis. By involving these domain experts, we were able to incorporate their knowledge and insights into our study, enhancing the validity and relevance of our findings. Their contributions have been instrumental in shaping our research on multimodal tuberculosis disease detection using convolutional neural networks.

3.5. Proposed System

In this study, a proposed model combines symptom and image data. The two types of data are processed separately and then merged using the Information Fusion technique to make the final classification. The model involves different stages, such as data preparation, pre-processing of symptom and image data, image augmentation, image segmentation, balancing datasets, parameter tuning and model training. The symptom and image models are used for extracting the features and followed by training the multimodal fusion. The efficiency of the system being proposed is examined through a 10-fold cross-validation process. and various evaluation metrics such as Accuracy, Sensitivity, and Specificity. The details of the proposed model are discussed in *Figure 3.2* below.

3.6. Dataset Preparation

The dataset of proposed model was gathered from Wolkite University Specialized Hospital. The dataset comprised a total of 734 data points, which included both symptoms and CXR images. Out of these, 163 were abnormal cases, while the remaining 571 were normal cases. Initially, the images were stored in the DICOM format, which is not commonly used for image processing. Therefore, to facilitate the analysis of this heterogeneous data, the X-ray images were transformed into the Portable Network Graphics (PNG) format, which is more widely recognized and utilized for image processing. Additionally, the symptoms were encoded to enable their analysis in a more standardized format. The process of preparing the symptoms was described in detail in *Algorithm 3.1*, which outlined the steps involved in encoding the symptoms. On the other hand, *Algorithm 3.2* described the process of converting the CXR images from DICOM to PNG format in detail.

3.7. Preprocessing

3.7.1. Symptom Data Preprocessing

The main goal of symptom data preprocessing in the proposed system is to transform categorical data into numerical features that can be used in machine learning models. This process is commonly referred to as data encoding. In medical datasets, symptoms are often denoted as categorical variables, such as the presence or absence of a specific symptom. One common technique for encoding categorical data is one-hot encoding (Reilly et al. 2022), which generates a binary feature for each category. In this study, one-hot encoding was applied. However, the resulting matrix may include missing values represented by NaNs

(Not a Number), and thus it was necessary to replace these values before conducting any further analysis. A typical method to replace missing values is to use the mean value of the corresponding feature (Reilly et al. 2022). The mean value was calculated first by disregarding NaNs and then used to replace them. To ensure uniformity in the data type, all values, including NaNs, were converted to the float32 data type. Finally, the resulting matrix was transformed back to a numpy array, and the target variable was extracted for subsequent analysis.

Algorithm 3.1 Symptom data preprocessing

Input: *Categorical data in the form of a pandas DataFrame 'df'*

Output: *Numeric feature matrix 'X' and target variable 'y'*

Select categorical columns by index;

Extract categorical data from the DataFrame using the selected columns;

Apply one-hot encoding to convert categorical data into numeric features;

Convert sparse matrix to dense matrix and then to pandas DataFrame;

Replace missing values with the mean value;

Convert all values to float32, including NaNs;

Compute the mean of non-NaN values;

Replace NaNs with the mean value;

Convert back to numpy array;

Extract the target variable (i.e., the label) from the DataFrame;

3.7.2. Symptom Data Balancing

Machine learning datasets frequently have imbalanced class distribution, where one class has significantly more instances than the others. This may result in the model exhibiting bias towards the majority class and showing suboptimal performance on the minority class. SMOTE is a technique used to address the class imbalance problem (Elreedy, Atiya, and Kamalov 2023b). In this study, SMOTE technique is applied to balance the class distribution of the target variable ('label') in the dataset. The SMOTE algorithm is applied to the feature matrix X and the target variable y using the fit_resample() function from the imblearn library.

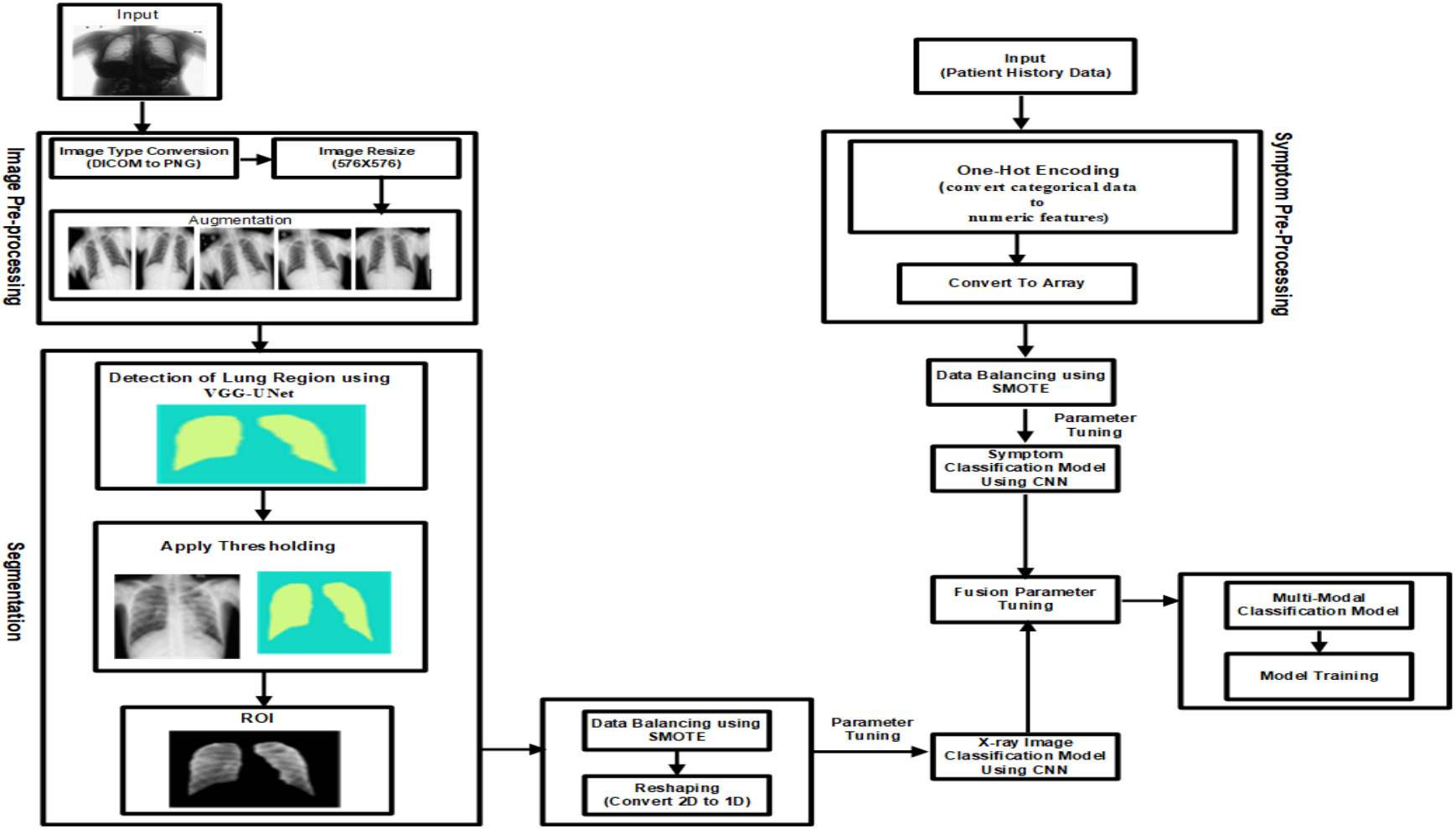


Figure 3.2 A block diagram for the proposed model

3.7.3. CXR Image Preprocessing

3.7.3.1. DICOM Image Conversion to PNG

The proposed model uses CXR images from the Wolkite University Hospital as its training and testing dataset. These images are in DICOM format, which needs to be converted to a PNG using specialized algorithms (Ait Nasser and Akhloufi 2023b). Additionally, the images have varying resolutions, even within the same dataset. To address this, image normalization is required. The normalization process aims to achieve consistent image resolution across all images. In this study, the images were resized to an average resolution of 576 x 576 pixels to ensure that they are all of the same size and to facilitate easier processing by the machine learning model.

Algorithm 3.2 DICOM Image Conversion to PNG

Input - *DICOM files or binary data in DICOM format*

Output - *DICOM to PNG image Conversion using the Python libraries*

Set the path to the directory "dir_path" containing the DICOM files;

Set the image size to 1024 X 1024;

For *each DICOM image file in the directory "dir_path":*

Check if the file has the extension ".dcm";

Load the DICOM image using the pydicom library;

Convert the pixel array to a 16-bit unsigned integer;

Resize the image using the cv2 library;

Convert the pixel array to an 8-bit unsigned integer;

Check if the image has PhotometricInterpretation equal to "MONOCHROME1". If so, subtract the pixel array from 255;

Save the image as a PNG in the PNG directory using the cv2 library;

End

3.7.3.2. CXR Image Augmentation

The goal of image augmentation is to increase the size and variety of a given dataset by applying different transformations to the images. These transformed images are then used to train or test machine learning models, resulting in improved performance and generalization. Several techniques can be utilized for image augmentation, such as position-based techniques which include cropping, rotating, scaling, flipping, padding, and elastic

deformations(Ait Nasser and Akhloufi 2023b). These techniques play a crucial role in creating variations of the original images and reducing overfitting by providing the model with a wider range of diverse examples to learn from.

However, it is important to note that not all augmentation techniques were included in this study due to their computational expense when applied to the proposed model. In this study, the ImageDataGenerator was employed to apply augmentation techniques to the images. The ImageDataGenerator library offers a diverse range of image augmentation techniques, including rotation, flipping, shifting, and zooming.

3.7.3.3. Detection of Lung Region

The study focused on evaluating the effectiveness of a pre-trained model that underwent the augmentation process to generate image masks. To accomplish this, the VGG-UNet (Kadry et al. 2022) method was utilized, which employs a pre-trained segmentation model. This model is accessed from (Chollet [2016] 2023) and saved the model as 'model.h5' that combines the VGG16 architecture with a UNet decoder. The model weights were loaded, and a list of input images in PNG format was also loaded using glob. The model was then used to generate a segmentation mask for each image in the list. These masks were saved to a specific directory, and their filenames were added to a list. Once all images were processed, the study checked if there were at least two images in the input images list. If there were, the study randomly selected four images from the input images list and displayed each image along with its corresponding mask. This procedure aids in assessing the accuracy of the generated masks, thereby ensuring the performance of the model. The algorithm used for the segmentation tasks is presented below.

Algorithm 3.3 Detection of Lung Region

Input: *A directory of input images in PNG format, a pre-trained segmentation model*

Output: *Segmentation masks for the input images*

Load the pre-trained segmentation model using load_model function;

Create an empty list to store mask filenames;

For *each input image in the directory "input_images":*

Read the image file using cv2.imread function;

Predict the segmentation mask for the input image using the model's predict_segmentation function;

Generate the mask filename by appending the basename of the input image to the mask directory path using os.path.join function;

Save the predicted mask to the mask filename using the out_fname parameter of the predict_segmentation function;

Add the mask filename to the list of mask filenames;

Check if there are at least 2 images in the input directory using an if statement;

If there are at least 2 images, select 4 random images from the input directory using the random.sample function;

For *each randomly selected input image in "random_images":*

Read the image file using cv2.imread function;

Find the corresponding mask filename by appending the basename of the input image to the mask directory path using os.path.join function;

Display the original and masked images using matplotlib's subplots function and imshow function;

Set the title of the first subplot to "Original Image: " followed by the basename of the input image;

Set the title of the second subplot to "Mask of " followed by the basename of the input image;

Show the subplots using plt.show function;

End

End

3.7.3.4. Segmentation

The study examined thresholding to distinguish an object from its background by setting a threshold value on the intensity values of the image (Chakraborty et al. 2021). The 'threshold_slow' function is defined to apply a binary threshold to the input images. The function loops through each pixel of the input image and compares it to the threshold value T. If the pixel value is less than or equal to T, it sets the corresponding pixel value in the output image to zero, otherwise, it retains the original pixel value. A binary mask is then applied to remove unwanted dots. The input images are read from a directory using the 'glob' module.

For each input image, the model predicts a segmentation mask using the model predict segmentation function and saves the mask to a corresponding directory. The mask filenames are stored in a list, then selects 4 random images and their corresponding masks, and displays them side by side using Matplotlib. Finally, the code applies the 'threshold_slow' function to

the mask and input image of each input image-mask pair. The resulting thresholded images are saved to a directory named '_roi_folder'. The algorithm used for the segmentation tasks is presented below.

Algorithm 3.4 Segmentation

Input: *A directory of input images and their corresponding masks in PNG format, a threshold value for image binarization*

Output: *Processed images with regions of interest (ROIs) extracted*

Get the list of input images and masks using the glob.glob function;

For *each input image and corresponding mask:*

Read the input image and mask files using cv2.imread function;

Convert the image and mask to grayscale using cv2.cvtColor function;

Apply the threshold_slow function to the grayscale image with the given threshold value and the grayscale mask;

Check if the ROI directory exists; if not, create it using os.makedirs function;

Generate the ROI filename by appending the basename of the input image to the ROI directory path using os.path.join function;

Save the thresholded image to the ROI filename using cv2.imwrite function;

End

3.7.3.5. CXR Image Data Balancing

The SMOTE algorithm is a technique used to balance imbalanced datasets. In imbalanced datasets, there may be very few examples of one class known as the minority class compared to the other classes known as the majority class. This might result in inadequate performance of machine learning models since they likely be biased towards the majority class [1]. In this study, SMOTE is implemented for X-RAY image using fit_resample method of the SMOTE object to oversample minority class examples.

3.8. Hyperparameter Optimization

The selection of hyper-parameters in Convolutional Neural Networks (CNN) can vary among researchers, leading to different optimization goals. Some researchers focus on specific hyper-parameters of individual layers while keeping the network architecture fixed. On the other hand, others consider a broader range of hyper-parameters, including the number of layers, learning rate, dropout rate, and more (Mohakud and Dash 2022). Grid search is a systematic approach to hyperparameter optimization where each combination of

hyperparameters is defined and assessed on the model. It is a trial-and-error technique (Saad et al. 2022).

Determining the best or optimum set of hyperparameters for training a CNN relies on various elements such as the specific CNN model, the dataset being used, the size of the dataset, and the specific problem being addressed. Deep learning practitioners strive to discover these optimal values using various approaches. One approach is to use automatic software tools like Optuna, which can perform hyperparameter optimization. Another approach is to use a trial-and-error method, where different combinations of hyperparameters are tested and evaluated (Monshi et al. 2021).

By considering these factors and utilizing tools and methods for hyperparameter optimization, researchers can boost the performance and effectiveness of CNN models in various applications. In this study, we employ the grid search algorithm to systematically explore and select the optimal hyperparameters for our CNN model. In addition to hyperparameter tuning, the choice of specific components within the model architecture also plays a pivotal role in achieving superior performance. In this context, we adopt the Adam optimizer (Kim and Choi 2021), the Rectified Linear Unit (ReLU) activation function (Reshi et al. 2021), and the binary cross-entropy loss function (Reshi et al. 2021). These components collectively contribute to the model's efficiency and accuracy. The details of the tuned parameters using Grid Search algorithm is presented on the next chapter.

3.9. Classification

The purpose of this thesis is to explore the use of multimodal fusion for detecting tuberculosis (TB) in patients. Specifically, the study aims to combine information from patient symptoms and chest X-ray images to classify cases as either infected or non-infected with TB. To accomplish this task, we employ a supervised learning algorithm known as CNN as a classifier. The CNN is used to identify TB manifestations in both the patient symptoms and CXR images and for the subsequent multimodal fusion. Additional details will be described in the following sub sections.

3.9.1. Symptom Classification

In this study, symptoms were extracted from the patient history records and were categorized into different features. As the symptoms were categorical, the researchers employed CNN as a classifier for the data. Specifically, they utilized a One-Dimensional CNN (CNN1D)

algorithm, which is a type of supervised learning algorithm commonly used for classification purposes. The objective of symptom classification was to compare the classification results with multimodal fusion. To achieve this, the CNN1D algorithm was trained on symptom data extracted from patient history records, enabling the algorithm to analyze and classify the data based on specific patterns and features found in symptoms.

3.9.2. Image Classification

In this study, CXR images were gathered based on the patient's identification number of symptoms, and CNN was employed as a classifier for the two-dimensional data. Specifically, a Two-Dimensional CNN (CNN2D) algorithm was used, which is a supervised learning algorithm commonly used for classification tasks. The main goal of image classification was to compare the classification results with multimodal fusion. To achieve this, the CNN2D algorithm was trained on CXR data collected from radiology records, enabling the algorithm to analyze and classify the data based on specific patterns and features found in CXR images.

3.9.3. Multimodal Fusion

As researchers, our primary goal was to develop a reliable and effective approach to diagnose tuberculosis by combining two important sources of data - patient symptoms and CXR images. To achieve this, we employed a multimodal fusion algorithm that integrates information from both sources. We explored three different types of fusion algorithms - early, joint, and late fusion - each with its own unique approach to integrating the symptom and CXR data. In order to accurately classify the data, we utilized CNN as our primary classifier. To extract the necessary features from the data, we relied on classification models for both symptom and image classification. These models were able to determine patterns and characteristics within the data that were critical to the successful implementation of our fusion algorithms. The description of the three models are the following:

Early Fusion Model: Early Fusion involves combining multiple input modalities into a single feature vector before feeding it into a machine learning model for training (Huang et al. 2020). In this study, the early fusion model is done by concatenating the outputs from both modalities as they are and feeds them through some dense layers to perform the classification. Before concatenation, each modality is processed independently by a classification model that extracts features from the respective input. The output from each

model is a set of features that represent the input. In this model, the two sets of features are concatenated and passed through some dense layers.

Joint Fusion Model: The joint fusion model, as described in reference (Huang et al. 2020), carries out separate processing of each modality before integrating them. In this particular study, the Joint Fusion approach was executed as follows: Each input modality underwent feature extraction via a classification model. Once the features were extracted, each modality was subjected to a set of dense layers, after which they were combined using concatenation. The combined features were then processed through additional dense layers to facilitate classification.

Late Fusion Model: The late fusion model also processes each modality independently before combining them (Huang et al. 2020). Each modality is passed through a classification model that extracts features from the respective input. After feature extraction, each modality is processed by some dense layers before being concatenated. The resulting concatenated features are then passed through additional dense layers for classification.

3.10. Performance Evaluation

The proposed solution's overall performance is evaluated using three performance metrics, which are Accuracy (ACC), Sensitivity (SES) and Specificity (SPC). *Equation 2.7* for Accuracy, *Equation 2.8* covers the mathematical equation for Sensitivity and *Equation 2.9* for Specificity.

3.11. Conclusion

In conclusion, a thorough summary of the research methodology used in the study was presented. It describes the literature review process, data collection, and preprocessing techniques, as well as the proposed system and its requirements. The chapter also discusses the classification approach used, including symptom classification, image classification, and multimodal fusion. Lastly, the chapter outlines the performance evaluation metrics used to assess the proposed solution's overall performance. By following this methodology, the study aims to optimize the accuracy measure for tuberculosis diagnosis from chest x-ray images and patient symptom records, ultimately leading to improved patient outcomes.

CHAPTER FOUR

4. Result and Discussion

4.1. Introduction

In this chapter, we delve into the discoveries of our study and engage in a meaningful discussion about our findings. However, before we can do that, there are crucial steps we must undertake. Firstly, we must prepare and organize our data to make it suitable for analysis. Subsequently, we embark on a preprocessing journey, fine-tuning both symptom and x-ray image data for optimal analysis. Once this groundwork is complete, we transition to the classification phase, where we extract essential features from our symptom and x-ray image models. We construct both individual and combined classification models, integrating information from symptoms and images. Afterward, we assess the performance of these models to gauge their effectiveness. Finally, we offer our research findings and engage in a comprehensive discussion to provide valuable insights.

4.2. Dataset Preparation

In this study, first we load symptom data from a CSV file and generates an exploration of the data to see the details of the symptom data. The images are stored in a DICOM format. This DICOM images are typically grayscale, which means they only have shades of gray instead of colors, and have a high pixel depth of 16 bits. This high pixel depth allows for a lot of detail and precision in the images, however, depending on the type of medical imaging equipment used to acquire the image, DICOM can also support other data types and color images. The *figure 4.1*, shows an example of a grayscale DICOM image with a high pixel depth. The implementation code is shown in *Appendix A*.

4.3. Preprocessing

4.3.1. Symptom Data Preprocessing

The first step in preprocessing the symptom data involves selecting specific symptom columns. Next, one-hot encoding is applied to convert categorical data to numeric features. Missing values are replaced with the mean value and all values, including NaNs, are converted to float32. The code computes the mean of non-NaN values and replaces NaNs with the mean value. The target variable, i.e., the label, is then extracted from the data. Once

these steps are completed, the processed data is suitable for use as input to machine learning algorithms. The implementation of this preprocessing procedure is shown in *Appendix B*.

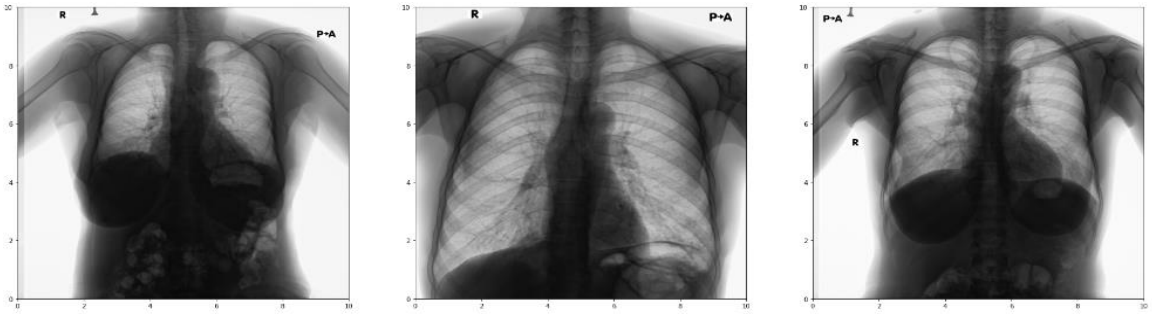


Figure 4.1 Visualization of DICOM image

4.3.2. Symptom Data Balancing

Symptom data are loaded from a CSV file, preprocesses it, and applies the SMOTE to balance the class distribution. The SMOTE algorithm creates synthetic samples of the minority class to increase its representation and improve the classification performance of the model (Elreedy et al. 2023b). The target variable (i.e., the label) is extracted from the data, and SMOTE is applied to the feature matrix (i.e., X) and the target variable (i.e., y). The resampled data and target variable are combined into a Data Frame, and the class distribution is visualized before and after SMOTE.

4.3.3. CXR Image Preprocessing

4.3.3.1. DICOM Image Conversion to PNG

Our process involves converting DICOM medical image files into PNG format. We begin by defining the directory path containing the DICOM files and specifying the desired output image size in our code. Next, we iterate through each file in the directory, and we use the pydicom library to read the DICOM file. We extract the pixel array from the DICOM file, and convert it into a 16-bit unsigned integer. We then resize the image to the desired size using the OpenCV library. After resizing, we convert the pixel array into an 8-bit unsigned integer. If the DICOM image is in "MONOCHROME1" format, we invert the pixel array to create a white-on-black image. Lastly, we save the PNG image to a separate directory using the cv2.imwrite function. *Appendix C* provides a detailed implementation of this process.

4.3.3.2. CXR Image Augmentation

This implementation shows how to use Python libraries to automatically generate new images from a set of PNG files. The TensorFlow Keras library is used to perform image augmentation, which consists transforming the original images using various techniques., such as rotating, shifting, zooming, and flipping, as shown in *Figure 4.2*. The number of augmented images to generate per input image is specified, and the resulting images are saved to a designated directory. The code provided in *Appendix D* demonstrates how to implement this process using OpenCV to load the images and define the augmentation parameters, as well as how to save the resulting images to the designated directory.

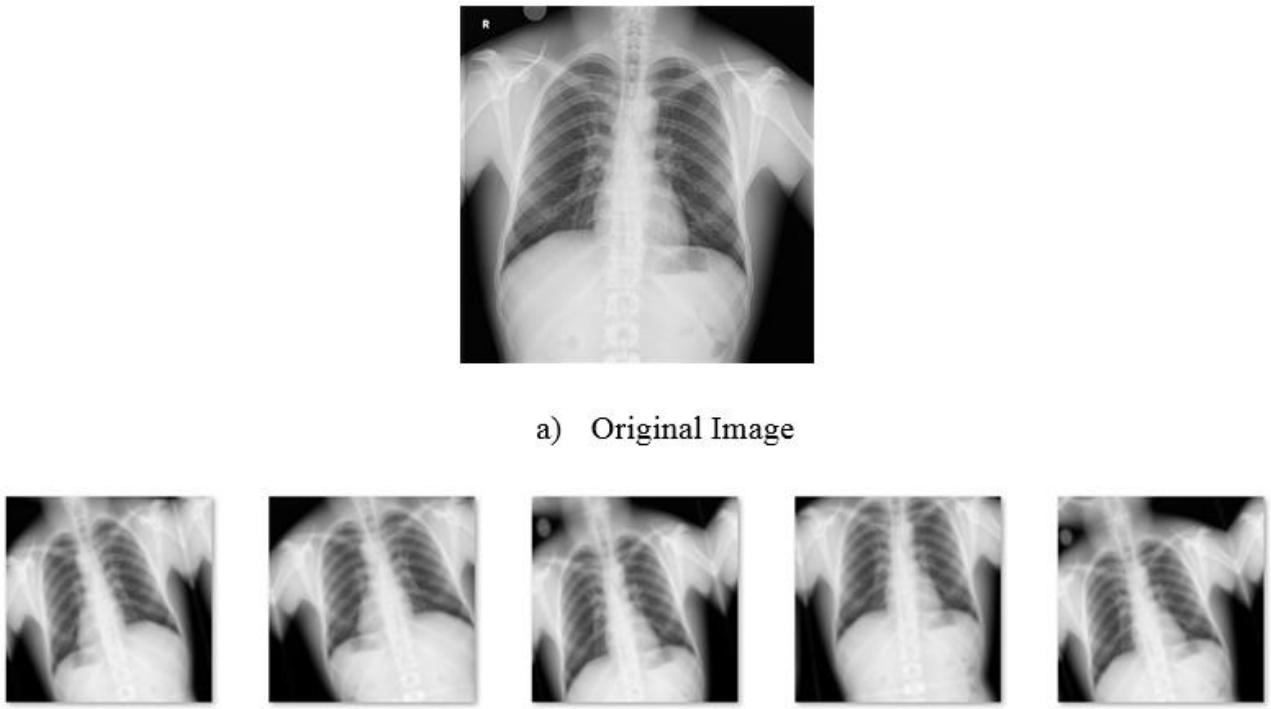


Figure 4.2 CXR Image Augmentation

4.3.3.3. Detection of Lung Region

Image segmentation is done using a pre-trained model. It takes a set of input images, generates a segmentation mask for each image using the pre-trained model, and saves the masks in a separate directory. The filenames of the masks are stored in a list for later use and then selects four random images from the list to display the original and masked images side by side using matplotlib. The purpose of this is to visualize the effectiveness of the image

segmentation algorithm by comparing the original images with their corresponding masks as shown in *Figure 4.3*. This can help to evaluate how well the algorithm is able to distinguish different objects or regions within the image. The implementation of this code is provided in *Appendix E*.

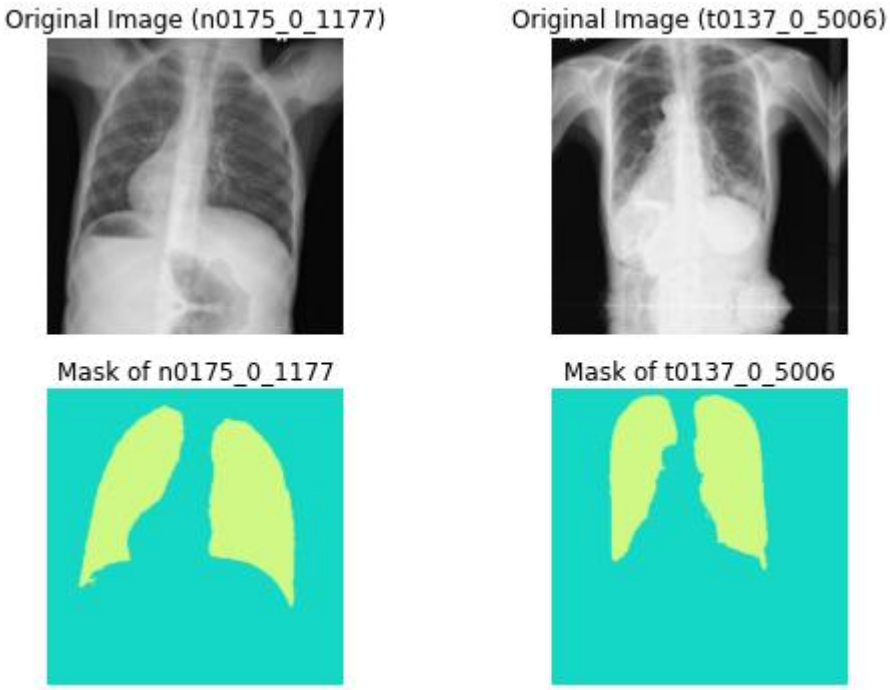


Figure 4.3 Detection of lung region

4.3.3.4. Segmentation

We perform an important task of extracting the region of interest (ROI) from XRAY images using thresholding image segmentation technique (Chakraborty et al. 2021). The code uses a thresholding algorithm to convert the grayscale XRAY images into binary images, which helps to isolate the ROIs. The OpenCV library functions are used to load and manipulate the images. The threshold_slow function is the core of the algorithm, which applies the threshold value to each pixel of the image 'img1' to convert it into a binary image. It then applies a binary mask to the image 'img2' to remove unwanted areas, and finally returns the thresholded image. This process is repeated for each pair of images and masks loaded from the specified directories using the glob function. The result of this process is shown in *Figure 4.4*, also the implementation for this process is shown in *Appendix F*.

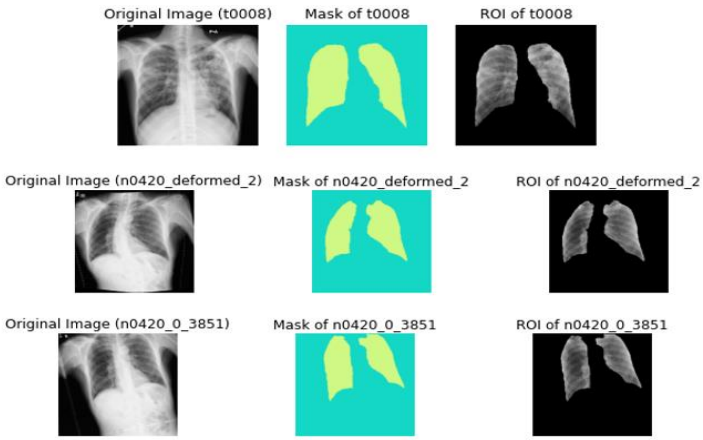


Figure 4.4 CXR Image Segmentation

4.3.3.5. CXR Image Data Balancing

The SMOTE algorithm is used for oversampling the minority class to balance the data (Elreedy et al. 2023b). The data is initially loaded from two directories and then converted to a pandas Data Frame. After preprocessing the images, the labels are mapped to numeric values and the SMOTE algorithm is applied to the data to balance the classes.

4.4. Hyperparameter Optimization

We conducted an extensive hyperparameter optimization process using a grid search algorithm to fine-tune our CNN model for training. This optimization aimed to maximize accuracy on independent datasets. Here's a breakdown of the key hyperparameters we considered and how we explored them:

Learning Rate (Lr): We started with a base learning rate of 0.001, which is commonly used in the Adam optimizer. To find the best learning rate around this base value, we selected one learning rate greater than the base (0.009) and one learning rates less than the base (0.0005). The learning rate values we considered were: Lr = {0.0005, 0.001, 0.009}.

Epochs: The number of training epochs, indicates how many times the entire dataset is processed during the training stage, was an important parameter. We explored different values for the number of epochs, including 10, 15, and 20.

Batch Size (Bs): We experimented with different batch sizes to optimize training efficiency. Our batch size values formed a set starting from 16 and increasing by 16 for each subsequent value. The batch size values we considered were: Bs = {16, 32, 64}.

Kernel Size (Ks): Kernel size determines the size of the convolutional kernels used in the network. We explored three different kernel sizes. The kernel size values we considered were: Ks = {3, 5, 7}.

Dropout Rate (Dr): Dropout serves as a regularization technique to mitigate overfitting by randomly deactivating a subset of input units, setting them to zero during the training process. We considered different dropout rates in our grid search. The dropout rate values we explored were: Dr = {0.5, 0.6, 0.7}.

Kernel Filters: While the number of kernel filters isn't explicitly mentioned in the provided text, it's an important hyperparameter to consider. The number of kernel filters (filters per layer) can impact the network's capacity to learn features. The actual values for the number of filters were likely part of the grid search process.

After the initial hyperparameter tuning, we further ensured the robustness of our model by employing a 10-fold cross-validation technique. In this technique, our dataset was divided into 10 subsets or "folds." We trained and evaluated the model 10 times, each time using a different fold as the validation set while using the remaining nine folds for training. This helped us assess the model's performance across various data splits and reduce the risk of overfitting.

4.5. Classification

4.5.1. Symptom Classification

In order to effectively analyze and classify the categorical symptoms, a Convolutional Neural Network (CNN) was employed as a classifier. The symptoms were grouped into distinct features, and the 1D CNN algorithm (CNN1D) was utilized for classification purposes. By training the CNN1D algorithm on the symptom data, the researchers aimed to enable the model to identify specific patterns and features within the symptoms, thereby achieving accurate classification. During the training process of the symptom classification model, various hyperparameters were tuned to optimize its performance. *Table 4.1* shows the parameters and their respective values that were explored using a grid search:

Table 4.4.1 Parameters Tuning of Symptom Classification

Parameters	***Ranges***	***Optimal Parameters***
Learning Rate	0.0005,0.001, 0.009	0.001
Batch Size	16, 32, 64	16
Epochs	10, 15, 20	20
Dropout Rate	0.5, 0.6, 0.7	0.7
Kernel Size	3, 5, 7	3
Layer Units	[8, 16, 32], [16, 32, 64]	[16, 32, 64]

In the pursuit of optimizing the Symptom Classification model, a meticulous exploration of key hyperparameters was undertaken. The values for Learning Rate were tested at 0.0005, 0.001, and 0.009, with 0.001 emerging as the most favorable. For Batch Size, the options of 16, 32, and 64 were considered, highlighting 16 as the optimal choice. The Epochs parameter was evaluated across 10, 15 and 20, with 20 demonstrating superior performance. Among the dropout rate 0.5, 0.6 and 0.7, 0.7 exhibited the highest efficacy. Kernel Size, gauged at 3, 5, and 7, found its peak at 3, indicating a more comprehensive receptive field. Lastly, layer unit's options of [8, 16, 32] and [16, 32, 64] were weighed, with [16, 32, 64] being deemed the most advantageous for the task at hand. This intricate selection of hyperparameters coalesces to enhance the Symptom Classification model's accuracy and its capacity to decipher intricate patterns within the symptom data. The architecture of the symptom model, including details about the different layers and their parameters are shown in *Table 4.2*. Also, the implementation to train this model architecture is presented in *Appendix F.*

Table 4.4.2 Symptom Data Classification Model Architecture

Layer	***Parameters***
Input	shape=(symptom_train_resampled.shape[1], 1)
Hidden-1	Conv1D with filters=16, kernel_size=3, activation='relu', kernel_regularizer=regularizers. l1_l2 (l1=0.003, l2=0.003), MaxPooling2D with pool_size =2 and Dropout (0.7)

Layer	*Parameters*
Hidden-2	Conv1D with filters=32, kernel_size=3, activation='relu', kernel_regularizer=regularizers.l1_l2(l1=0.002, l2=0.002), MaxPooling2D with pool_size =2 and Flatten
Hidden-3	Dense with units=64, activation='relu', kernel_regularizer=regularizers.l1_l2(l1=0.002, l2=0.002),
Output	units=1, activation='sigmoid', optimizer='adam', learning_rate=0.001, loss='binary_crossentropy' and metrics=['accuracy']

4.5.2. Image Classification

In order to achieve precise categorization of Chest X-ray single modality data and analyze the result, we incorporate CNN as a classifier. To enhance the accuracy of chest X-ray image classification, the researchers implemented a grid search for hyperparameter tuning. This technique allows for the selection of the best combination of hyperparameters for the CNN2D algorithm, which serves as the classifier in this classification task. The grid search involves systematically testing different values of hyperparameters to find the optimal combination that yields the highest performance. The parameters that were tuned in this study are shown in *Table 4.3*:

Table 4.4.3 Parameters Tuning of X-ray Image Classification

Parameters	***Ranges***	***Optimal Parameters After preprocessing***
Learning Rate	0.0005,0.001, 0.009	0.0005
Batch Size	16, 32, 64	16
Epochs	10, 15, 20	10
Dropout Rate	0.5, 0.6, 0.7	0.5
Kernel Size	3, 5, 7	3
Layer Units	[8, 16, 32], [16, 32, 64]	[8, 16, 32]

By performing the grid search with the above parameter grid, the researchers aimed to identify the most effective hyperparameters for the CNN2D algorithm. This process helps to

optimize the model's performance and enhance its ability to accurately identify distinct patterns and features in the X-ray images. The researchers likely tested various combinations of hyperparameter values within the specified range for each hyperparameter. By systematically exploring different hyperparameter combinations, researchers can identify the best-performing configuration that leads to accurate and reliable classification results. As shown on the table the researchers found that using a learning rate of 0. 0005, a batch size of 16, and training for 10 epochs resulted in the highest performance. Additionally, using the Dropout Rate of 0.5, a kernel size of (3,3) and Layer Units of [8, 16, 32] yielded the best results for the CNN2D algorithm in classifying chest X-ray images. The model architecture for this classification model is shown in *Table 4.4*. The implementation details for this model training process can be found in *Appendix G*.

Table 4.4.4 X-RAY Image Classification Model Architecture

Layer	***Parameters***
Hidden-1	Conv2D with filters=8, kernel_size=(3, 3), padding='same', activation='relu', input_shape=(224, 224, 3) and MaxPooling2D with pool_size=(2, 2) and Dropout(0.5)
Hidden-2	Conv2D with filters=16, kernel_size=(3, 3), padding='same', activation='relu', kernel_regularizer=regularizers.l2(0.001) and MaxPooling2D with pool_size=(2, 2) and Flatten
Hidden-3	Dense (32, activation='relu', kernel_regularizer=regularizers.l2(0.001))
Output	Dense with units=1, activation='sigmoid', optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'] and learning_rate= 0.0005

4.5.3. Multimodal Fusion

To classify combined data into two distinct categories, normal and abnormal, we utilize three different multimodal classification models: early fusion, joint fusion, and late fusion. All of these models utilize Convolutional Neural Networks (CNNs) to process the data. For each multimodal fusion technique, we performed hyperparameter tuning using a k-fold cross-validation approach. The number of folds for the cross-validation was set to 10, ensuring robust evaluation of the models' performance. The parameters that were tuned for each fusion techniques are presented below Table:

Table 4.4.5 Parameters Tuning of Multi-Modal Fusion Classification

Parameters	***Ranges***	***Optimal for Early Fusion***	***Optimal for Joint Fusion***	***Optimal for Late Fusion***
Learning Rate	0.0005,0.001, 0.009	0.001	0.001	0.009
Epochs	10, 15, 20	10	10	10
Batch Size	16, 32, 64	64	64	64
Layer Units	[8, 16, 32], [16, 32, 64]	[16, 32, 64]	[8, 16, 32],	No Layer Unit

The optimal parameters for the multimodal fusion classification were determined through a grid search as indicated in the above table. The researchers identified that a learning rate of 0.001 consistently produced the best results for both early and joint fusion techniques, and 0.009 was the optimal learning rate for late fusion. The number of epochs was set to 10 for all fusion techniques. Additionally, a batch size of 64 was found to be optimal across all fusion approaches. These finely tuned hyperparameters are expected to contribute to improved accuracy and robustness in the classification of combined data into two distinct categories: normal and abnormal, utilizing the multimodal fusion models. The layer unit's information adds another layer of detail to the parameter optimization process. For the Early Fusion and Joint Fusion techniques, the choice of layer units was from the range [16, 32, 64], indicating that these values were considered for the models. The architecture details for these three types of multimodal fusion can be found in *Table 4.6*, *Table 4.7, Table 4.8* respectively, providing a comprehensive overview.

Table 4.4.6 Early Fusion Model Architecture

Layer	***Parameters***
Input	Symptom and Image input
Symptom Output	Symptom Model (Symptom Input)
Image Output	Image Model (Image Input)
Hidden-1	Concatenate ([Symptom Output, Image Output])
Hidden-2	Dense (16, activation='relu', kernel_regularizer=regularizers.l2(0.08))
Hidden-3	Dense (32, activation='relu', kernel_regularizer=regularizers.l2(0.07))
Hidden-4	Dense (64, activation='relu', kernel_regularizer=regularizers.l2(0.07))
Output	Dense (1, activation='sigmoid'), learning_rate=0.001, optimizer='adam', loss='binary_crossentropy', metrics=['accuracy']

The above table provides an overview of a fusion-based model architecture that combines symptom data and X-ray image data for a specific task. The model follows an early fusion approach, where the information from the two modalities is fused at an early stage. This fusion-based model architecture combines the information from symptom data and X-ray image data to make predictions for the given task, using a combination of 1D and 2D convolutional layers, dense layers, and fusion techniques.

Table 4.4.7 Joint Fusion Model Architecture

Layer	***Parameters***
Input	Symptom and Image Input
Symptom Output	Symptom Model (Symptom Input)
	Dense (1, activation='sigmoid')
Image Output	Image Model (Image Input)
	Dense (1, activation='sigmoid')
Hidden-1	Concatenate ([Symptom Output, Image Output])
Hidden-2	Dense (8, activation='relu') and BatchNormalization (Hidden-1)
Hidden-3	Dense (16, activation='relu') and BatchNormalization (Hidden-2)
Hidden-4	Dense (32, activation='relu') and BatchNormalization (Hidden-3)
Output	Dense (1, activation='sigmoid'), learning_rate=0.001, optimizer='adam', loss = 'binary_crossentropy', metrics=['accuracy']

The table describes a joint fusion model architecture that combines symptom data and X-ray image data for a fusion task. The joint fusion approach aims to merge the information from both modalities at a later step in the model. This joint fusion model architecture combines the information from symptom data and X-ray image data at a later stage in the model, using 1D and 2D convolutional layers, dense layers, batch normalization, and fusion techniques.

Table 4.4.8 Late Fusion Model Architecture

Layer	***Parameters***
Input	Symptom and Image Input
Symptom Output	Symptom Model (Symptom Input)
Image Output	Image Model (Image Input)
Hidden-1	Concatenate ([Symptom Output, Image Output])
Output	Dense (1, activation='sigmoid'), learning_rate=0.009, optimizer='adam', loss='binary_crossentropy', metrics=['accuracy']

The table describes a late fusion model architecture that combines symptom data and X-ray image data for a specific task. In contrast to the joint fusion approach, the late fusion approach combines the information from both modalities at the output layer. This late fusion model architecture combines the information from symptom data and X-ray image data at

the output layer, using 1D and 2D convolutional layers, dense layers, and fusion techniques. The concatenated feature vector is passed through a dense layer with sigmoid activation for binary classification.

4.6. Performance Evaluation

We use the evaluation method called k-fold cross-validation on a binary classification model. The data is split into 10 folds, where each part is used as a validation set once while the other parts are used for training. This process repeats 10 times, so every data point gets used for both training and validation. Three important metrics are calculated: accuracy, sensitivity, and specificity.

4.7. Result of the Study

In this study, we introduced an approach for tuberculosis disease classification that utilizes a multimodal fusion of x-ray images and patient symptoms. Our implementation of the proposed model yielded significant and promising results throughout the various stages of the study. We gathered a comprehensive set of local datasets from Wolkite University Specialized Hospital, encompassing 734 x-ray images alongside their corresponding clinical patient symptoms. Out of these datasets, 571 were classified as normal, while 163 were classified as abnormal cases.

To enhance the diversity of the dataset, we employed x-ray image augmentation. This involved applying cropping, rotating, scaling, and flipping to the images, resulting in an augmented dataset comprising 3426 normal cases and 978 abnormal cases. Furthermore, the augmented datasets underwent additional preprocessing steps, including masking and segmentation, prior to the classification process.

To examine the performance of each modality and the fusion of modalities, we conducted a comprehensive analysis and comparison. The results of this evaluation before and after segmentation are summarized in *Table 4.9*, *Table 4.10*, *Table 4.11 and Table 4.12*. Our study's findings clearly demonstrate the Efficiency of our suggested multimodal fusion approach for tuberculosis disease classification. These results hold promising potential for real-world clinical applications, showcasing the value of integrating both x-ray images and patient symptoms in improving diagnostic accuracy.

Table 4.9 Training and Testing Accuracy before Segmentation

Models	***Training Accuracy (%)***	***Testing Accuracy (%)***
Symptom	0.9230	0.9239
X-RAY Image	0.8304	0.7775
Early Fusion	0.6949	0.6953
Joint Fusion	0.5251	0.5307
Late Fusion	0.6960	0.6691

The above table presents the training and testing accuracy of different models before segmentation. The models under consideration are Symptom, X-RAY Image, Early Fusion, Joint Fusion, and Late Fusion. Based on the presented results, the Symptom model stands out as the most promising option, demonstrating superior performance in both training and testing accuracy. The X-RAY Image and Fusion models, while showing varying levels of accuracy, may benefit from further optimization or feature engineering to enhance their predictive capabilities.

Table 4.10 Training and Testing Accuracy after Segmentation

Models	***Training Accuracy (%)***	***Testing Accuracy (%)***
X-RAY Image	0.9268	0.8430
Early Fusion	0.9766	0.9723
Joint Fusion	0.7584	0.7475
Late Fusion	0.8922	0.8586

The above table presents the training and testing accuracy of different models after segmentation. After segmentation, all models show improvements in both training and testing accuracy. The Early Fusion model stands out as the top performer, achieving remarkably high accuracy levels. The X-RAY Image and Late Fusion models also demonstrate significant improvements, while the Joint Fusion model, though enhanced, may still require further refinement for optimal performance. Overall, the segmentation process appears to have positively influenced the models' predictive capabilities.

Table 4.11 Performance Metrics Result of the Proposed Models before segmentation

Models	***Accuracy (%)***	***Sensitivity (%)***	***Specificity (%)***
Symptom	0.8345	0.8107	0.8581
X-RAY Image	0.9407	0.9496	0.9322
Early Fusion	0.6771	0.8063	0.5768
Joint Fusion	0.5009	0.6000	0.4000
Late Fusion	0.6743	0.5972	0.7526

The given table presents the performance metrics of different models before segmentation. These models are evaluated based on their accuracy, sensitivity, and specificity for a particular task. The symptom model and X-ray image model achieved high accuracy, sensitivity, and specificity, indicating their effectiveness in the task. However, the early fusion and joint fusion models performed relatively poorly, with lower accuracy, sensitivity, and specificity values. The late fusion model also exhibited lower accuracy compared to the symptom and X-ray image models but showed better sensitivity and specificity than the early and joint fusion models. These results suggest that the symptom and X-ray image models individually perform well, while the fusion models (both early and joint) have lower performance before segmentation. The late fusion model shows some improvement compared to other fusion models but is still not as effective as the individual symptom and X-ray image models.

Table 4.12 Performance Metrics Result of the Proposed Models after segmentation

Models	***Accuracy (%)***	***Sensitivity (%)***	***Specificity (%)***
X-RAY Image	0.9487	0.9294	0.9679
Early Fusion	0.9915	0.9942	0.9889
Joint Fusion	0.9889	1.0000	0.7095
Late Fusion	0.8070	0.8950	0.8511

Compared to the previous findings, the X-ray Image Model has shown considerable improvement in all aspects, achieving an accuracy of 94.87%, sensitivity of 92.94%, and specificity of 96.79%. The Early Fusion Model stands out as the top performer, with an accuracy of 99.15%, sensitivity of 99.42%, and specificity of 98.89%. The Joint Fusion Model has also made positive strides, with an accuracy of 98.89%, sensitivity 100.00%, and a specificity of 70.95%. However, the Late Fusion Model has experienced a drop in accuracy, now at 80.70%, while maintaining moderate sensitivity and specificity values at 89.50% and 85.11% respectively. In summary, these updated results highlight the superior performance of the Early Fusion Model, while the X-ray Image Model and Joint Fusion Model also show significant improvements, although the latter sacrifices specificity. On the other hand, the Late Fusion Model falls behind in terms of accuracy compared to previous results.

The visual representation of the Receiver Operating Characteristic (ROC) curve results for the overall proposed models before and after segmentation are presented below in *Figure*

4.5 : and Figure 4.6. The ROC curve, or Receiver Operating Characteristic curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The plot illustrates the relationship between the true positive rate (TPR) and the false positive rate (FPR) at different threshold settings. The TPR, also known as sensitivity, represents the proportion of actual positives that are correctly identified as such. On the other hand, the FPR, or 1-specificity, shows the ratio of true negatives that were wrongly classified as positives.

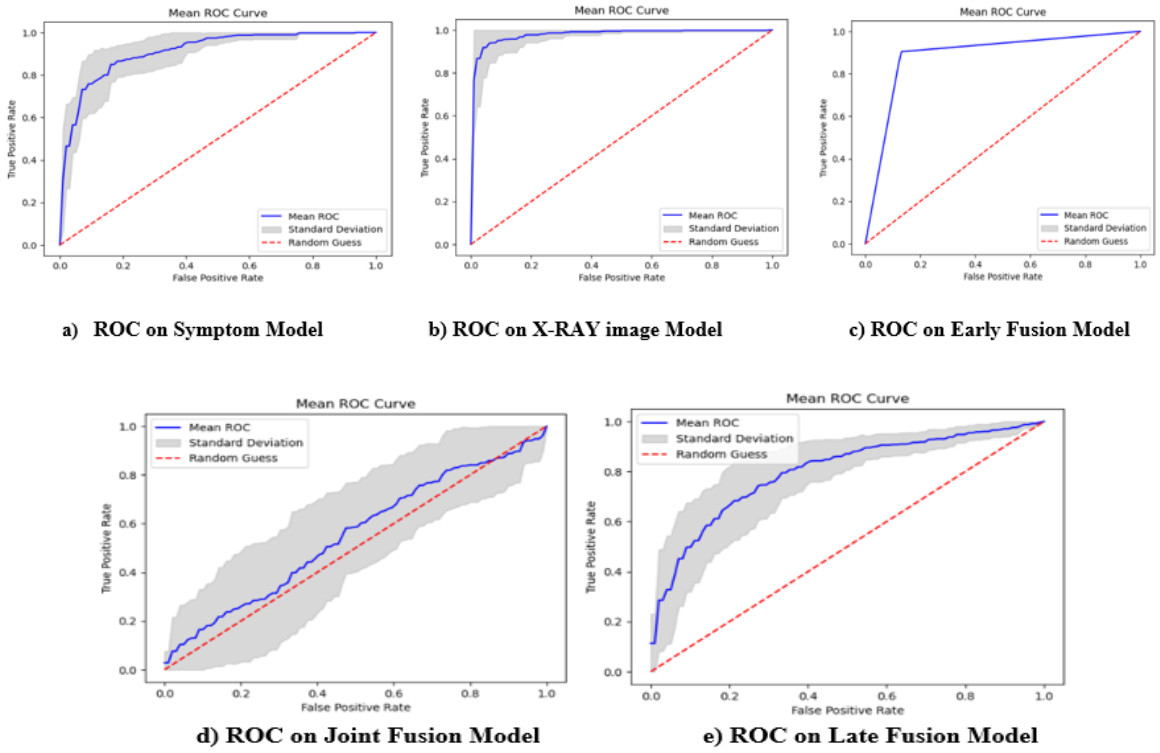


Figure 4.5 ROC Results of Proposed Models before Segmentation

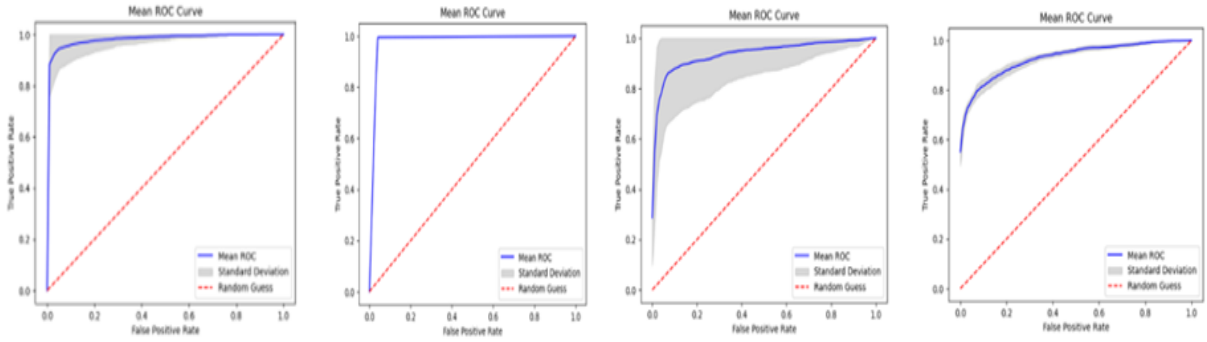
The results shown in above Figure (diagrams a, b, c, d, e) are based on a 10-fold cross-validation. This process involves training the model on each training fold and evaluating it on the corresponding test fold. For each fold, the TPR and FPR are calculated at various thresholds, and the ROC curve is plotted. The mean ROC curve is then plotted by averaging the TPR values at each FPR across all 10 folds. The shaded area in the diagrams represents the standard deviation of the TPR values.

Based on the results provided:

- Symptom has a mean TPR of 0.8345, a maximum TPR of 0.8107, and a standard deviation of 0.8581.

- X-RAY Image has a mean TPR of 0.9407, a maximum TPR of 0.9496, and a standard deviation of 0.9322.
- Early Fusion has a mean TPR of 0.6771, a maximum TPR of 0.8063, and a standard deviation of 0.5768.
- Joint Fusion has a mean TPR of 0.5009, a maximum TPR of 0.6000, and a standard deviation of 0.4000.
- Late Fusion has a mean TPR of 0.6743, a maximum TPR of 0.5972, and a standard deviation of 0.7526.

From these results, it appears that the X-RAY Image model shows the best performance, as it has the highest mean TPR and maximum TPR, indicating a high proportion of actual positives correctly identified. The standard deviation is also relatively low, suggesting a consistent performance across different thresholds.



a) ROC on X-RAY Model b) ROC on Early Fusion Model c) ROC on Joint Fusion Model d) ROC on Late Fusion Model

Figure 4.6 ROC Results of Proposed Models after Segmentation

As shown in the above Figure (diagrams a, b, c, d), the TPR and FPR are calculated at various thresholds, and the ROC curve is plotted. The mean ROC curve is then plotted by averaging the TPR values at each FPR across all 10 folds. The shaded area in the diagrams represents the standard deviation of the TPR values.

Based on the results provided:

- X-RAY Image has a mean TPR of 0.9487, a maximum TPR of 0.9294, and a standard deviation of 0.9679.
- Early Fusion has a mean TPR of 0.9915, a maximum TPR of 0.9942, and a standard deviation of 0.9889.

- Joint Fusion has a mean TPR of 0.9889, a maximum TPR of 1.0000, and a standard deviation of 0.7095.
- Late Fusion has a mean TPR of 0.8070, a maximum TPR of 0.8950, and a standard deviation of 0.8511.

From these results, it appears that the Early Fusion model shows the best performance, as it has the highest mean TPR and maximum TPR, indicating a high proportion of actual positives correctly identified. The standard deviation is also relatively low, suggesting a consistent performance across different thresholds.

4.8. Discussion and Interpretation

In this thesis, we present a new approach for TB classification, which involves the fusion of chest CXR images with their corresponding clinical symptoms. To further explore and expand upon this concept, the thesis addresses three primary research questions. These research questions serve as the foundation for investigating various aspects of the proposed multimodal TB classification model, enabling a comprehensive analysis and evaluation of its effectiveness.

Based on the findings provided in the tables above, the answers of my research questions are explained below:

Q1. Can the accuracy of tuberculosis detection be improved by integrating the technique of lung object detection during the diagnostic process?

The study presented a new and innovative method for classifying tuberculosis disease by combining x-ray images and patient symptoms. To ensure a diverse dataset, the researchers utilized pre-processing techniques called lung object detection. They then assessed the performance of different models before and after identifying the lung region. The findings demonstrated that the X-ray Image Model, Early Fusion Model, and Joint Fusion Model all saw improved accuracy, sensitivity, and specificity following segmentation. The X-ray Image Model achieved an accuracy of 94.87%, with a sensitivity of 92.94% and specificity of 96.79%. The Early Fusion Model showed significant enhancement, with an accuracy of 99.15%, sensitivity of 99.42%, and specificity of 98.89%. The Joint Fusion Model also displayed improved accuracy, sensitivity, and specificity, achieving an accuracy of 98.89%, sensitivity of 100.00%, and specificity of 70.95%. In conclusion, the study highlights the effectiveness of employing segmentation, or lung object detection, as a preprocessing

technique to improve the accuracy of tuberculosis detection. When combined with a multimodal fusion approach utilizing x-ray images and patient symptoms, this method holds promise for improving the diagnosis of TB.

Q2. Which approach, individual modality classification models or fusion techniques, enhances the accuracy of the tuberculosis classification model?

Before segmentation, the individual symptom model and x-ray image model achieved high accuracy, sensitivity, and specificity. However, the early fusion and joint fusion models performed relatively poorly, with lower accuracy, sensitivity, and specificity values. The late fusion model showed some improvement compared to the fusion models but was still not as effective as the individual symptom and x-ray image models.

After segmentation, the x-ray image model, early fusion model, and joint fusion model demonstrated improved accuracy, sensitivity, and specificity. The x-ray image model achieved an accuracy of 94.87%, the early fusion model achieved an accuracy of 99.15%, and the joint fusion model achieved an accuracy of 98.89%. The late fusion model showed a decrease in accuracy compared to the previous results.

The study's findings indicate that the multimodal fusion approach, specifically the early fusion and joint fusion models, enhances the accuracy of the tuberculosis classification model after segmentation. However, it is worth noting that the late fusion model did not perform as well as the individual symptom and x-ray image models. These results suggest that integrating both x-ray images and patient symptoms can improve the diagnostic accuracy of tuberculosis classification.

Q3. If the performance of multimodal fusion is improved, what is the most effective multimodal fusion technique to enhance the accuracy of tuberculosis diagnosis using a combination of x-ray images and patient symptoms?

The study built a new method for tuberculosis disease classification that employs a multimodal fusion of X-ray images and patient symptoms. The proposed model showed promising results throughout the various stages of the study. The performance of different models before and after segmentation was evaluated and compared.

Before Segmentation:

- The symptom model and X-ray image model achieved high accuracy, sensitivity, and specificity, indicating their effectiveness in the task.
- The early fusion and joint fusion models performed relatively poorly, with lower accuracy, sensitivity, and specificity values.
- The late fusion model showed some improvement compared to the fusion models but was still not as effective as the individual symptom and X-ray image models.

After Segmentation:

- The X-ray Image Model showed improved accuracy, sensitivity, and specificity, achieving an accuracy of 94.87%, sensitivity of 92.94%, and specificity of 96.79%.
- The Early Fusion Model showed significant improvement, with a high accuracy of 99.15%, sensitivity of 99.42%, and specificity of 98.89%.
- The Joint Fusion Model also exhibited improved accuracy, sensitivity, and specificity, achieving an accuracy of 98.89%, sensitivity of 100.00%, and specificity of 70.95%.
- The Late Fusion Model demonstrated a decrease in accuracy compared to the previous results, with an accuracy of 80.70%, sensitivity of 89.50%, and specificity of 85.11%.

Based on the results, the most effective multimodal fusion technique to enhance the accuracy of tuberculosis diagnosis using a combination of X-ray images and patient symptoms is the Early Fusion Model. It showed the highest accuracy, sensitivity, and specificity after segmentation.

CHAPTER FIVE

5. Conclusions and Recommendations

5.1. Conclusion

Tuberculosis is a persistent and communicable illness caused by Mycobacterium tuberculosis. It primarily affects the lungs but can also impact the spine and brain. The disease spreads when an untreated individual with TB coughs or sneezes, releasing infectious droplet nuclei containing tubercle bacilli. TB is a major global cause of mortality, particularly among individuals with weakened immune systems or other health conditions. Developing countries bear the highest burden of TB cases and deaths. Accurate diagnosis relies on chest radiography, but interpretation skills of radiologists in high-prevalence areas can be inadequate, leading to misdiagnosis and delayed treatment. Automated TB detection from chest radiographs and clinical health record could greatly assist low-resource settings with limited access to trained radiologists and expensive CT imaging.

This research introduces a model designed to help in the detection of tuberculosis through the combination of chest X-ray images and patient symptoms. Through a comprehensive an overview of related works, several unresolved issues have been discovered, which call for further attention and investigation by researchers in the field. By addressing these gaps, we can enhance the accuracy and effectiveness of TB detection and aid to the improvement of healthcare outcomes in tuberculosis management.

This research presents a new method for tuberculosis diagnosis by fusing chest X-ray images and clinical patient symptoms through multimodal fusion. The study applies various preprocessing techniques, such as symptom preprocessing using one-hot encoding and data balancing with SMOTE. Additionally, for the X-ray images, preprocessing involves DICOM to PNG conversion and augmentation techniques like cropping, rotating, scaling, and flipping, resulting in a dataset of 571 normal and 163 abnormal cases.

Furthermore, the study incorporates lung region detection using a pre-trained VGG-UNet model and thresholding to extract the region of interest. Data balancing with SMOTE is also applied to the X-ray images. The study then implements individual modality classification and multimodal fusion classification, including early, joint, and multimodal fusion. The proposed model undergoes evaluation using 10-fold cross-validation.

The proposed system was assessed using performance evaluation metrics, including accuracy, sensitivity, and specificity. The findings revealed that the multimodal fusion technique yielded better results compared to individual modalities. Furthermore, implementing the region of interest enhanced the performance.

In summary, the proposed multimodal fusion approach, especially the Early Fusion model after segmentation, showcased promising potential for tuberculosis diagnosis. The study's findings underscore the importance of integrating both chest X-ray images and patient symptoms to enhance diagnostic accuracy. These results contribute to addressing the challenges of TB diagnosis in resource-limited settings, where access to trained radiologists is limited. Further research and optimization efforts can potentially refine the proposed model for broader clinical applications, ultimately contributing to improved healthcare outcomes in tuberculosis management.

5.2. Recommendations

5.2.1. Recommendation for Hospitals

The integration and utilization of IT infrastructure within the healthcare industry have the potential to bring significant improvements in information exchange, time management, and the effective allocation of human resources. Through the implementation of computer technology, hospitals can streamline the handling of urgent cases, resulting in more efficient and timely interventions. In light of this, the researcher strongly recommends that health sectors consider incorporating a multimodal tuberculosis disease detection system as an invaluable diagnostic and treatment support tool for TB. The application of machine learning techniques, particularly deep learning, within the healthcare field has shown promising outcomes in various studies. By harnessing the capabilities of CNNs (Convolutional Neural Networks) and IT infrastructure, healthcare providers can make better-informed decisions and deliver appropriate care to their patients. These techniques can aid healthcare professionals in the diagnostic process, ultimately leading to improved patient outcomes.

5.2.2. Future Work

The models proposed in this thesis serve as valuable inspiration for future research endeavors and offer valuable insights into areas that can enhance the advancement of future CAD

systems for screening TB abnormalities. The following areas are highly recommended for further investigation and development:

1) **Explainable AI Techniques:** The incorporation of explainable AI techniques in the multimodal tuberculosis diagnosis model using X-ray images and clinical patient symptoms can greatly enhance the interpretability of the model's predictions. By employing these techniques, the model can provide clinicians with transparent and understandable explanations for its diagnostic decisions. This means that instead of solely relying on the model's output, clinicians will have access to insights into the reasoning behind a particular diagnosis.
2) **Radiologists' Analysis Integration using Natural Language Processing (NLP):** Expand the current scope by integrating radiologists' analyses through the incorporation of Natural Language Processing (NLP) techniques. Develop an innovative framework that extracts and interprets textual descriptions provided by radiologists alongside the multimodal data. By fusing the quantitative insights from the convolutional neural network with the qualitative observations from radiologists, the model's diagnostic capabilities can be further enhanced. This approach will provide a holistic understanding of disease patterns, ensuring a more comprehensive and accurate diagnostic outcome.
3) **Expanding the multimodal approach:** The study utilized chest X-ray images and clinical patient symptoms for TB diagnosis. However, other modalities such as CT scans can also be included in the multimodal fusion approach to increase the accuracy of diagnosis.
4) **Extending the study to other diseases:** The proposed approach can be applied to other diseases that impact the respiratory system or various body parts, like pneumonia and lung cancer. This can be achieved by combining chest X-ray images and clinical patient symptoms through multimodal fusion. This could be used to figure out of these diseases and improve the diagnosis and treatment outcomes.

REFERENCES

Ait Nasser, Adnane, and Moulay A. Akhloufi. 2023a. “A Review of Recent Advances in Deep Learning Models for Chest Disease Detection Using Radiography.” Diagnostics 13(1):159. doi: 10.3390/diagnostics13010159.

Ait Nasser, Adnane, and Moulay A. Akhloufi. 2023b. “A Review of Recent Advances in Deep Learning Models for Chest Disease Detection Using Radiography.” Diagnostics 13(1):159. doi: 10.3390/diagnostics13010159.

Anon. 2020. Global Tuberculosis Report 2020. Geneva: World Health Organization.

Anon. 2022. “Detection of Pulmonary Tuberculosis from Chest X-Ray Images Using Multimodal Ensemble Method.” Communications in Mathematical Biology and Neuroscience. doi: 10.28919/cmbn/7776.

Anon. 2023. “Tuberculosis (TB).” Retrieved April 9, 2023 (https://www.who.int/news-room/fact-sheets/detail/tuberculosis).

Anon. n.d. “Global Tuberculosis Report 2021.” Retrieved April 9, 2023 (https://www.who.int/teams/global-tuberculosis-programme/tb-reports/global-tuberculosis-report-2021).

Antony, Betsy. 2017. “Lung Tuberculosis Detection Using X-Ray Images.” 12(24).

Archer, Chantai, Adrian R. Levy, and Maurice McGregor. 1993. “Value of Routine Preoperative Chest X-Rays: A Meta-Analysis.” Canadian Journal of Anaesthesia 40(11):1022–27. doi: 10.1007/BF03009471.

Beguma, Sayyada Hajera, and Vidyullatha P. 2020. “A Contemporary Study of Deep Learning in Diagnosis and Prediction of Diseases.” SSRN Electronic Journal. doi: 10.2139/ssrn.3734301.

Boehme, Catharina C., Pamela Nabeta, Doris Hillemann, Mark P. Nicol, Shubhada Shenai, Fiorella Krapp, Jenny Allen, Rasim Tahirli, Robert Blakemore, Roxana Rustomjee, Ana Milovic, Martin Jones, Sean M. O’Brien, David H. Persing, Sabine Ruesch-Gerdes, Eduardo Gotuzzo, Camilla Rodrigues, David Alland, and Mark D. Perkins. 2010. “Rapid Molecular Detection of Tuberculosis and Rifampin Resistance.” New England Journal of Medicine 363(11):1005–15. doi: 10.1056/NEJMoa0907847.

Brady, Adrian P. 2017. “Error and Discrepancy in Radiology: Inevitable or Avoidable?” Insights into Imaging 8(1):171–82. doi: 10.1007/s13244-016-0534-1.

Breen, R. A. M. 2004. “Paradoxical Reactions during Tuberculosis Treatment in Patients with and without HIV Co-Infection.” Thorax 59(8):704–7. doi: 10.1136/thx.2003.019224.

Cao, Yan, Xinjing Wang, Ping Liu, Yue Su, Haotian Yu, and Jingli Du. 2022. “Vitamin D and the Risk of Latent Tuberculosis Infection: A Systematic Review and Meta-Analysis.” BMC Pulmonary Medicine 22(1):39. doi: 10.1186/s12890-022-01830-5.

Chakraborty, Sanjoy, Apu Kumar Saha, Sukanta Nama, and Sudhan Debnath. 2021. “COVID-19 X-Ray Image Segmentation by Modified Whale Optimization Algorithm with Population Reduction.” Computers in Biology and Medicine 139:104984. doi: 10.1016/j.compbiomed.2021.104984.

Chen, Yiyang, Pengqiang Ge, Guina Wang, Guirong Weng, and Hongtian Chen. 2023. “An Overview of Intelligent Image Segmentation Using Active Contour Models.” Intelligence & Robotics 3(1):23–55. doi: 10.20517/ir.2023.02.

Chinmayi, P., L. Agilandeeswari, and M. Prabukumar. 2018. “Survey of Image Processing Techniques in Medical Image Analysis: Challenges and Methodologies.” Pp. 460–71 in Proceedings of the Eighth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2016). Vol. 614, Advances in Intelligent Systems and Computing, edited by A. Abraham, A. K. Cherukuri, A. M. Madureira, and A. K. Muda. Cham: Springer International Publishing.

Chollet, François. [2016] 2023. “Trained Image Classification Models for Keras.”

Cobelens, Frank, Susan van den Hof, Madhukar Pai, S. Bertel Squire, Andrew Ramsay, and Michael E. Kimerling. 2012. “Which New Diagnostics for Tuberculosis, and When?” The Journal of Infectious Diseases 205(suppl_2):S191–98. doi: 10.1093/infdis/jis188.

Cui, Can, Haichun Yang, Yaohong Wang, Shilin Zhao, Zuhayr Asad, Lori A. Coburn, Keith T. Wilson, Bennett Landman, and Yuankai Huo. 2023. “Deep Multi-Modal Fusion of Image and Non-Image Data in Disease Diagnosis and Prognosis: A Review.” Progress in Biomedical Engineering. doi: 10.1088/2516-1091/acc2fe.

Dabass, Manju, Sharda Vashisth, and Rekha Vig. 2018. “Effectiveness of Region Growing Based Segmentation Technique for Various Medical Images - A Study.” Pp. 234–59 in Data Science and Analytics. Vol. 799, Communications in Computer and Information Science, edited by B. Panda, S. Sharma, and N. R. Roy. Singapore: Springer Singapore.

Dastres, Roza, and Mohsen Soori. n.d. “Advanced Image Processing Systems.”

Davis, Jesse, and Mark Goadrich. 2006. “The Relationship between Precision-Recall and ROC Curves.” Pp. 233–40 in Proceedings of the 23rd international conference on Machine learning - ICML ’06. Pittsburgh, Pennsylvania: ACM Press.

Degnan, Andrew J., Emily H. Ghobadi, Peter Hardy, Elizabeth Krupinski, Elena P. Scali, Lindsay Stratchko, Adam Ulano, Eric Walker, Ashish P. Wasnik, and William F. Auffermann. 2019. “Perceptual and Interpretive Error in Diagnostic Radiology—Causes and Potential Solutions.” Academic Radiology 26(6):833–45. doi: 10.1016/j.acra.2018.11.006.

Denkinger, C. M., S. G. Schumacher, C. C. Boehme, N. Dendukuri, M. Pai, and K. R. Steingart. 2014. “Xpert MTB/RIF Assay for the Diagnosis of Extrapulmonary Tuberculosis: A Systematic Review and Meta-Analysis.” European Respiratory Journal 44(2):435–46. doi: 10.1183/09031936.00007814.

Djannah, Fathul, Muhammad Nasrum Massi, Mochammad Hatta, Agussalim Bukhari, and Idyatul Hasanah. 2022. “Profile and Histopathology Features of Top Three Cases of Extra Pulmonary Tuberculosis (EPTB) in West Nusa Tenggara: A Retrospective Cross-Sectional Study.” Annals of Medicine & Surgery 75. doi: 10.1016/j.amsu.2022.103318.

D’Souza, Niharika S., Hongzhi Wang, Andrea Giovannini, Antonio Foncubierta-Rodriguez, Kristen L. Beck, Orest Boyko, and Tanveer Syeda-Mahmood. 2022. “Fusing Modalities by Multiplexed Graph Neural Networks for Outcome Prediction in Tuberculosis.” Pp. 287–97 in Vol. 13437.

Dunnmon, Jared A., Darvin Yi, Curtis P. Langlotz, Christopher Ré, Daniel L. Rubin, and Matthew P. Lungren. 2019. “Assessment of Convolutional Neural Networks for Automated Classification of Chest Radiographs.” Radiology 290(2):537–44. doi: 10.1148/radiol.2018181422.

Elreedy, Dina, Amir F. Atiya, and Firuz Kamalov. 2023a. “A Theoretical Distribution Analysis of Synthetic Minority Oversampling Technique (SMOTE) for Imbalanced Learning.” Machine Learning. doi: 10.1007/s10994-022-06296-4.

Elreedy, Dina, Amir F. Atiya, and Firuz Kamalov. 2023b. “A Theoretical Distribution Analysis of Synthetic Minority Oversampling Technique (SMOTE) for Imbalanced Learning.” Machine Learning. doi: 10.1007/s10994-022-06296-4.

El-Yaniv, Ran, and Oren Souroujon. 2003. “Iterative Double Clustering for Unsupervised and Semi-Supervised Learning.” Pp. 121–32 in Machine Learning: ECML 2001.

Vol. 2167, Lecture Notes in Computer Science, edited by L. De Raedt and P. Flach. Berlin, Heidelberg: Springer Berlin Heidelberg.

Fawcett, Tom. 2006. “An Introduction to ROC Analysis.” Pattern Recognition Letters 27(8):861–74. doi: 10.1016/j.patrec.2005.10.010.

Gough, Maya, Dhiraj K. Singh, Chivonne Moodley, Tianhua Niu, Nadia A. Golden, Deepak Kaushal, and Smriti Mehra. 2022. “Peripheral Blood Markers Correlate with the Progression of Active Tuberculosis Relative to Latent Control of Mycobacterium Tuberculosis Infection in Macaques.” Pathogens 11(5):544. doi: 10.3390/pathogens11050544.

Guia, Sana Sahar, Abdelkader Laouid, Mostefa Kara, and Mohammad Hammoudeh. 2023. Tuberculosis Detection Using Chest X-Ray Image Classification by Deep Learning. preprint. In Review. doi: 10.21203/rs.3.rs-2509525/v2.

Huang, Shih-Cheng, Anuj Pareek, Saeed Seyyedi, Imon Banerjee, and Matthew P. Lungren. 2020. “Fusion of Medical Imaging and Electronic Health Records Using Deep Learning: A Systematic Review and Implementation Guidelines.” Npj Digital Medicine 3(1):136. doi: 10.1038/s41746-020-00341-z.

Imrey, Peter B., and Gary G. Koch. 2005. “Categorical Data Analysis.” Categorical Data Analysis.

Joseph, V. Roshan. 2022. “Optimal Ratio for Data Splitting.” Statistical Analysis and Data Mining: The ASA Data Science Journal 15(4):531–38. doi: 10.1002/sam.11583.

Kadry, Seifedine, Gautam Srivastava, Venkatesan Rajinikanth, Seungmin Rho, and Yongsung Kim. 2022. “Tuberculosis Detection in Chest Radiographs Using Spotted Hyena Algorithm Optimized Deep and Handcrafted Features” edited by A. Loddo. Computational Intelligence and Neuroscience 2022:1–12. doi: 10.1155/2022/9263379.

Kim, Kwang Gi. 2016. “Book Review: Deep Learning.” Healthcare Informatics Research 22(4):351. doi: 10.4258/hir.2016.22.4.351.

Kim, Kyung-Soo, and Yong-Suk Choi. 2021. “HyAdamC: A New Adam-Based Hybrid Optimization Algorithm for Convolution Neural Networks.” Sensors 21(12):4054. doi: 10.3390/s21124054.

Kiranyaz, Serkan, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. 2021. “1D Convolutional Neural Networks and Applications: A Survey.” Mechanical Systems and Signal Processing 151:107398. doi: 10.1016/j.ymssp.2020.107398.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2017. “ImageNet Classification with Deep Convolutional Neural Networks.” Communications of the ACM 60(6):84–90. doi: 10.1145/3065386.

Kuhn, Max, and Kjell Johnson. 2013. Applied Predictive Modeling. New York, NY: Springer New York.

Kumar, Tarun, and Karun Verma. 2010. “A Theory Based on Conversion of RGB Image to Gray Image.” International Journal of Computer Applications 7(2):5–12. doi: 10.5120/1140-1493.

Lakhani, Paras, and Baskaran Sundaram. 2017. “Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks.” Radiology 284(2):574–82. doi: 10.1148/radiol.2017162326.

Lawn, Stephen D., and Mark P. Nicol. 2011. “Xpert ® MTB/RIF Assay: Development, Evaluation and Implementation of a New Rapid Molecular Diagnostic for Tuberculosis and Rifampicin Resistance.” Future Microbiology 6(9):1067–82. doi: 10.2217/fmb.11.84.

Lloyd, Dr Graham. n.d. “Radiology Masterclass.” Retrieved February 4, 2023 (https://www.radiologymasterclass.co.uk/).

Loddenkemper, Robert, Marc Lipman, and Alimuddin Zumla. 2016. “Clinical Aspects of Adult Tuberculosis.” Cold Spring Harbor Perspectives in Medicine 6(1):a017848. doi: 10.1101/cshperspect.a017848.

Lönnroth, Knut, and Mario Raviglione. 2016. “The WHO’s New End TB Strategy in the Post-2015 Era of the Sustainable Development Goals.” Transactions of The Royal Society of Tropical Medicine and Hygiene 110(3):148–50. doi: 10.1093/trstmh/trv108.

Ma, Y., C. R. Horsburgh, L. F. White, and H. E. Jenkins. 2018. “Quantifying TB Transmission: A Systematic Review of Reproduction Number and Serial Interval Estimates for Tuberculosis.” Epidemiology and Infection 146(12):1478–94. doi: 10.1017/S0950268818001760.

Malhotra, Priyanka, Sheifali Gupta, Deepika Koundal, Atef Zaguia, and Wegayehu Enbeyle. 2022. “Deep Neural Networks for Medical Image Segmentation” edited by C. Chakraborty. Journal of Healthcare Engineering 2022:1–15. doi: 10.1155/2022/9580991.

McCoubrey, Laura E., Moe Elbadawi, Mine Orlu, Simon Gaisford, and Abdul W. Basit. 2021. “Harnessing Machine Learning for Development of Microbiome Therapeutics.” Gut Microbes 13(1):1872323. doi: 10.1080/19490976.2021.1872323.

Melendez, Jaime, Clara I. Sánchez, Rick H. H. M. Philipsen, Pragnya Maduskar, Rodney Dawson, Grant Theron, Keertan Dheda, and Bram Van Ginneken. 2016. “An Automated Tuberculosis Screening Strategy Combining X-Ray-Based Computer-Aided Detection and Clinical Information.” Scientific Reports 6(1):25265. doi: 10.1038/srep25265.

Menzies, Dick, Madhukar Pai, and George Comstock. n.d. “Meta-Analysis: New Tests for the Diagnosis of Latent Tuberculosis Infection: Areas of Uncertainty and Recommendations for Research.”

Mohakud, Rasmiranjan, and Rajashree Dash. 2022. “Designing a Grey Wolf Optimization Based Hyper-Parameter Optimized Convolutional Neural Network Classifier for Skin Cancer Detection.” Journal of King Saud University - Computer and Information Sciences 34(8):6280–91. doi: 10.1016/j.jksuci.2021.05.012.

Monshi, Maram Mahmoud A., Josiah Poon, Vera Chung, and Fahad Mahmoud Monshi. 2021. “CovidXrayNet: Optimizing Data Augmentation and CNN Hyperparameters for Improved COVID-19 Detection from CXR.” Computers in Biology and Medicine 133:104375. doi: 10.1016/j.compbiomed.2021.104375.

Murugan, Pushparaja, and Shanmugasundaram Durairaj. 2017. “Regularization and Optimization Strategies in Deep Convolutional Neural Network.”

Nienhaus, Albert, Anja Schablon, and Roland Diel. 2008. “Interferon-Gamma Release Assay for the Diagnosis of Latent TB Infection – Analysis of Discordant Results, When Compared to the Tuberculin Skin Test.” PLoS ONE 3(7).

Pai, Madhukar, and Marcel Behr. 2016. “Latent Mycobacterium Tuberculosis Infection and Interferon-Gamma Release Assays.”

Park, Minwoo, Youjin Lee, Sangil Kim, Young-Jin Kim, Shin Young Kim, Yeongsic Kim, and Hyun-Min Kim. 2023. “Distinguishing Nontuberculous Mycobacterial Lung Disease and Mycobacterium Tuberculosis Lung Disease on X-Ray Images Using Deep Transfer Learning.” BMC Infectious Diseases 23(1):32. doi: 10.1186/s12879-023-07996-5.

Peirse, Mary, and Angela Houston. 2017. “Extrapulmonary Tuberculosis.” Medicine 45(12):747–52. doi: 10.1016/j.mpmed.2017.09.008.

Powers, David MW. 2020. “Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation.” arXiv Preprint arXiv:2010.16061.

Preisser, John S., and Gary G. Koch. 1997. “CATEGORICAL DATA ANALYSIS IN PUBLIC HEALTH.” Annual Review of Public Health 18(1):51–82. doi: 10.1146/annurev.publhealth.18.1.51.

Rajinikanth, Venkatesan, Seifedine Kadry, and Yunyoung Nam. 2021. “Convolutional-Neural-Network Assisted Segmentation and SVM Classification of Brain Tumor in Clinical MRI Slices.” Information Technology and Control 50(2):342–56. doi: 10.5755/j01.itc.50.2.28087.

Rajkomar, Alvin, Eyal Oren, Kai Chen, Andrew M. Dai, Nissan Hajaj, Michaela Hardt, Peter J. Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, Patrik Sundberg, Hector Yee, Kun Zhang, Yi Zhang, Gerardo Flores, Gavin E. Duggan, Jamie Irvine, Quoc Le, Kurt Litsch, Alexander Mossin, Justin Tansuwan, De Wang, James Wexler, Jimbo Wilson, Dana Ludwig, Samuel L. Volchenboum, Katherine Chou, Michael Pearson, Srinivasan Madabushi, Nigam H. Shah, Atul J. Butte, Michael D. Howell, Claire Cui, Greg S. Corrado, and Jeffrey Dean. 2018. “Scalable and Accurate Deep Learning with Electronic Health Records.” Npj Digital Medicine 1(1):18. doi: 10.1038/s41746-018-0029-1.

Rajpurkar, Pranav, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng. 2017. “CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning.”

Reilly, Denis, Mark Taylor, Paul Fergus, Carl Chalmers, and Steven Thompson. 2022. “The Categorical Data Conundrum: Heuristics for Classification Problems—A Case Study on Domestic Fire Injuries.” IEEE Access 10:70113–25. doi: 10.1109/ACCESS.2022.3187287.

Reshi, Aijaz Ahmad, Furqan Rustam, Arif Mehmood, Abdulaziz Alhossan, Ziyad Alrabiah, Ajaz Ahmad, Hessa Alsuwailem, and Gyu Sang Choi. 2021. “An Efficient CNN Model for COVID-19 Disease Detection Based on X-Ray Image Classification” edited by Á. Madureira Bueno. Complexity 2021:1–12. doi: 10.1155/2021/6621607.

Saad, Mohamed H., Sherief Hashima, Wessam Sayed, Ehab H. El-Shazly, Ahmed H. Madian, and Mostafa M. Fouda. 2022. “Early Diagnosis of COVID-19 Images Using Optimal CNN Hyperparameters.” Diagnostics 13(1):76. doi: 10.3390/diagnostics13010076.

Saito, Takaya, and Marc Rehmsmeier. 2015. “The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets” edited by G. Brock. PLOS ONE 10(3):e0118432. doi: 10.1371/journal.pone.0118432.

Shawal, Shammi, Muhammad Shoyab, and Suraiya Begum. 2014. “Fundamentals of Digital Image Processing and Basic Concept of Classification.” International Journal of Chemical and Process Engineering Research 1(6):98–108. doi: 10.18488/journal.65/2014.1.6/65.6.98.108.

Shin, Hoo-Chang, Holger R. Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M. Summers. 2016. “Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning.” IEEE Transactions on Medical Imaging 35(5):1285–98. doi: 10.1109/TMI.2016.2528162.

Shorten, Connor, and Taghi M. Khoshgoftaar. 2019. “A Survey on Image Data Augmentation for Deep Learning.” Journal of Big Data 6(1):60. doi: 10.1186/s40537-019-0197-0.

Sulis, G., A. Roggi, A. Matteelli, and MC Raviglione. 2014. “Tuberculosis: Epidemiology and Control. Mediterr J Hematol Infect Dis [Internet]. 2014; 6 (1): E2014070.”

Tohka, Jussi, and Mark van Gils. 2021. “Evaluation of Machine Learning Algorithms for Health and Wellness Applications: A Tutorial.” Computers in Biology and Medicine 132:104324. doi: 10.1016/j.compbiomed.2021.104324.

Traub, Matthias, Mark Stevenson, Suzanne McEvoy, Greg Briggs, Sing Kai Lo, Steven Leibman, and Tony Joseph. 2007. “The Use of Chest Computed Tomography versus Chest X-Ray in Patients with Major Blunt Trauma.” Injury 38(1):43–47. doi: 10.1016/j.injury.2006.07.006.

Vinnarasi, J. Vimala, and V. Saravanabavan. 2017. “Tuberculosis Types and Its Characteristics in Dindigul District-A Geomedical Study Using GIS.” International Journal of Geomatics and Geosciences 7(3):262–74.

Wang, Shuai, Bo Kang, Jinlu Ma, Xianjun Zeng, Mingming Xiao, Jia Guo, Mengjiao Cai, Jingyi Yang, Yaodong Li, Xiangfei Meng, and Bo Xu. 2021. “A Deep Learning Algorithm Using CT Images to Screen for Corona Virus Disease (COVID-19).” European Radiology 31(8):6096–6104. doi: 10.1007/s00330-021-07715-1.

World Health Organization. 2016. Chest Radiography in Tuberculosis Detection: Summary of Current WHO Recommendations and Guidance on Programmatic Approaches. Geneva: World Health Organization.

Yagis, Ekin, Nichapat Pinpo, Janadhip Jacutprakart, Vahid Abolghasemi, Jarutas Andritsch, Sitthichok Chaichulee, Yashin Dicente Cid, Alba Garcia Seco de Herrera, and Thammasin Ingviya. 2022. Ensemble Deep Learning Architectures for Automated Diagnosis of Pulmonary Tuberculosis Using Chest X-Ray. preprint. doi: 10.36227/techrxiv.21543309.

Zumla, Alimuddin, Andrew George, Virendra Sharma, Nick Herbert, and Baroness Masham of Ilton. 2013. "WHO's 2013 Global Report on Tuberculosis: Successes, Threats, and Opportunities." The Lancet 382(9907):1765–67. doi: 10.1016/S0140-6736(13)62078-4.

van Zyl-Smit, Richard N., Madhukar Pai, Kwaku Peprah, Richard Meldau, Jackie Kieck, June Juritz, Motasim Badri, Alimuddin Zumla, Leonardo A. Sechi, Eric D. Bateman, and Keertan Dheda. 2009. "Within-Subject Variability and Boosting of T-Cell Interferon-γ Responses after Tuberculin Skin Testing." American Journal of Respiratory and Critical Care Medicine 180(1):49–58. doi: 10.1164/rccm.200811-1704OC.

APPENDIXES

Appendix A DICOM Image Visualization (Python Code)

```
# Set the directory path where the DICOM files for images are located.
dir_path = "C:/Users/Buser at
wku/Desktop/environment/implementation/preprocess_file_path/__dicom_folder"
# Get a list of all DICOM files in the directory
dicom_files = [os.path.join(dir_path, f) for f in os.listdir(dir_path) if f.endswith(".dcm")]
# Select three random DICOM files from the list
random_files = random.sample(dicom_files, 3)
# Set the desired image size in inches
img_width = 10
img_height = 10
# Create a figure with 3 subplots arranged horizontally
fig, axs = plt.subplots(1, 3, figsize=(3*img_width, img_height))
# Loop through the random files and display the images horizontally
for i, file in enumerate(random_files):
    # Load the DICOM image
    ds = pydicom.dcmread(file)
    # Convert the pixel array to a 16-bit unsigned integer
    pixel_array = ds.pixel_array.astype(np.uint16)
    # Display the image in the i-th subplot
    axs[i].imshow(pixel_array, cmap=plt.cm.gray, extent=[0, img_width, 0, img_height])
# Show the figure
plt.show()
```

Appendix B Symptom Data Preprocessing

```
# Retrieve the data from the CSV file
df = pd.read_csv('tbsymptom.csv')
symptom_cols = [4,5,6,7,8,9,10,11,12,13,14,15,16,17,18]
df['text'] = df.iloc[:, symptom_cols].apply(lambda x: ' '.join([str(i) for i in x]), axis=1)
# Select the categorical columns by index
cat_cols_index = symptom_cols
X_cat = df.iloc[:, cat_cols_index]
# Apply one-hot encoding to convert categorical data into numeric features
encoder = OneHotEncoder()
X = encoder.fit_transform(X_cat)
# Convert sparse matrix to dense matrix and then to pandas DataFrame
symptom_features = pd.DataFrame(X.toarray())
# Replace missing values with the mean value
symptom_features = symptom_features.replace('.', np.nan)
# Convert all values to float32, including NaNs
symptom_features = symptom_features.astype(np.float32)
# Compute the mean of non-NaN values
symptom_mean = np.nanmean(symptom_features)
# Replace NaNs with the mean value
symptom_features = symptom_features.fillna(symptom_mean)
# Convert back to numpy array
X = symptom_features.values
# Extract the target variable (i.e., the label) from the data
y = df['label']
```

Appendix C Convert DICOM Image to PNG

```
# Set the path to the directory where the DICOM image files are located
dir_path = "C:/Users/Buser at wku/Desktop/environment/implementation/preprocess_file_path/__dicom_folder"
# Set the desired image size
new_size = (576, 576)
# Loop through all DICOM files in the directory
for filename in tqdm(os.listdir(dir_path)):
    if filename.endswith(".dcm"):
        # Load the DICOM image
        ds = pydicom.dcmread(os.path.join(dir_path, filename))
        # Convert the pixel array to a 16-bit unsigned integer
        pixel_array = ds.pixel_array.astype(np.uint16)
        # Resize the image
        pixel_array = cv2.resize(pixel_array, new_size)
        # Convert the pixel array to 8-bit unsigned integer
        pixel_array = pixel_array.astype(np.uint8)
        # Display the image
        if ds.PhotometricInterpretation == "MONOCHROME1":
            pixel_array = 255 - pixel_array
        # Save the image as PNG in the PNG directory
        cv2.imwrite(os.path.join("C:/Users/Buser at wku/Desktop/environment/implementation/preprocess_file_path/__png_folder", os.path.splitext(filename)[0] + ".png"), pixel_array)
```

Appendix D CXR Image Augmentation

```
# Set the path to the directory where the PNG image files are located
dir_path = "C:/Users/Buser at wku/Desktop/environment/implementation/augmentation_1/__png_folder"
# Set the number of augmented images to generate per input image
num_augmented_images = 5
# Set the augmentation parameters
datagen = ImageDataGenerator(
    rotation_range=20, # rotation range in degrees
    width_shift_range=0.1, # horizontal shift range as a fraction of image width
    height_shift_range=0.1, # vertical shift range as a fraction of image height
    zoom_range=0.1, # zoom range as a fraction of original image size
    horizontal_flip=True, # randomly flip images horizontally
    vertical_flip=False, # randomly flip images vertically
    fill_mode='reflect', # fill mode for newly created pixels
)
# Loop through all PNG files in the directory
for filename in os.listdir(dir_path):
    if filename.endswith(".png"):
        # Load the PNG image
        img = cv2.imread(os.path.join(dir_path, filename),    cv2.IMREAD_GRAYSCALE)
        # Reshape the image to have four dimensions
        img = np.expand_dims(img, axis=-1) # Add an extra dimension for the channel
        img = np.expand_dims(img, axis=0)  # Add an extra dimension for the batch
        # Generate augmented images
        i = 0
        for batch in datagen.flow(img, batch_size=1, save_to_dir='C:/Users/Buser at wku/Desktop/environment/implementation/augmentation_1/__augmented_folder', save_prefix=filename[:-4], save_format='png'):
            i += 1
            if i >= num_augmented_images:
                break
```

Appendix E Detection of lung region

```
# Get a list of input images
input_images = glob.glob("__augmented_folder/*.png")
# Create a list to store the mask filenames
mask_filenames = []
# loop over each input image and generate a segmentation mask
for input_filename in tqdm(input_images):
    # Load the image
    img = cv2.imread(input_filename)
    # Predict segmentation mask and save to the mask's directory
    mask_filename = os.path.join("_mask_folder", os.path.basename(input_filename))
    out = model.predict_segmentation(
        inp=img,
        out_fname=mask_filename
    )
    # Add the mask filename to the list
    mask_filenames.append(mask_filename)
# check that there are at least 2 images in the input_images list
if len(input_images) < 2:
    print("Error: There are fewer than 2 images in the input_images list")
else:
    # Select 2 random images
    random_images = random.sample(input_images, 4)
    # loop over the random images
    for input_filename in random_images:
        # Load the image
        img = cv2.imread(input_filename)
        # Find the corresponding mask filename
        mask_filename = os.path.join("_mask_folder", os.path.basename(input_filename))
        # Display the original and masked images
        fig, axs = plt.subplots(1, 2, figsize=(10, 10))
        original_filename = os.path.basename(input_filename)
        axs[0].imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
        axs[0].set_title("Original Image: "+ original_filename)
```

```
axs[0].axis("off")
axs[1].imshow(cv2.cvtColor(cv2.imread(mask_filename), cv2.COLOR_BGR2RGB))
axs[1].set_title("Mask of "+ original_filename)
axs[1].axis("off")
plt.show()
```

Appendix F CXR Image Segmentation

```
def threshold_slow(T, img1, img2):
    # Grab the image dimensions
    h = img1.shape[0]
    w = img1.shape[1]
    # loop over the image, pixel by pixel
    for y in range(0, h):
        for x in range(0, w):
            # threshold the pixel
            org=img2[y, x]
            img2[y, x] = 0 if img1[y, x] <= T else org
    # Apply binary mask to remove unwanted dots
    mask = cv2.threshold(img1, T, 255, cv2.THRESH_BINARY)[1]
    img2 = cv2.bitwise_and(img2, img2, mask=mask)
    # Return the thresholded image
    return img2
# Get a list of input images and masks
input_images = glob.glob("__augmented_folder/*.png")
masks = glob.glob("_mask_folder/*.png")
# loop over each input image and mask
for i in range(len(input_images)):
    # Load the image and mask
    input_filename = input_images[i]
    mask_filename = masks[i]
    img = cv2.imread(input_filename)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    mask = cv2.imread(mask_filename)
    mask = cv2.cvtColor(mask, cv2.COLOR_BGR2GRAY)
    # Apply the threshold_slow function
    thresholded_img = threshold_slow(160, mask, img)
    # Create the ROI directory if it doesn't exist
    if not os.path.exists("_roi_folder"):
        os.makedirs("_roi_folder")
    # Construct the ROI filename from the input filename
```

```
input_basename = os.path.basename(input_filename)
result_filename = os.path.join("_roi_folder", input_basename)
# Save the result to the ROI directory
cv2.imwrite(result_filename, thresholded_img)
```

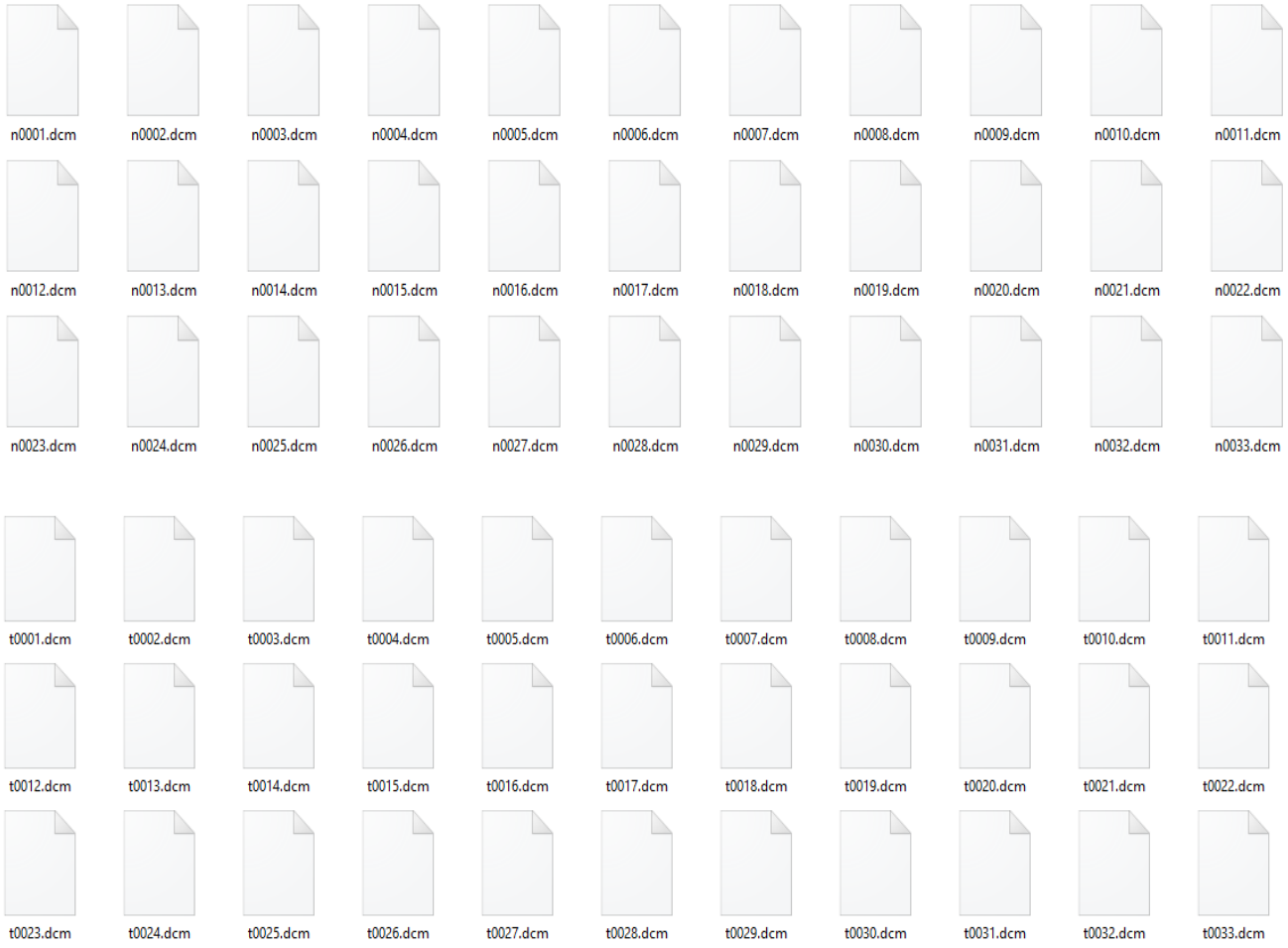
Appendix F Symptom Classification

```
# Define the number of folds for k-fold cross-validation
num_folds = 10
# Create a stratified k-fold object
kf = StratifiedKFold(n_splits=num_folds, shuffle=True)
# Create an empty list to store the accuracy for each fold
training_accs = []
testing_accs = []
# Loop through each fold
for fold, (train_idx, test_idx) in enumerate(kf.split(symp_train_data_arr_reshaped,
symp_train_labels_arr_reshaped)):
    print(f'Fold {fold+1}/{num_folds}')
# Split the data into training and testing sets for this fold
    X_train_fold, y_train_fold = symp_train_data_arr_reshaped[train_idx],
symp_train_labels_arr_reshaped[train_idx]
    X_test_fold, y_test_fold = symp_train_data_arr_reshaped[test_idx],
symp_train_labels_arr_reshaped[test_idx]
    # Train the model on this fold and store the accuracy
    training_acc, testing_acc = train_symptom_model(X_train_fold, y_train_fold,
X_test_fold, y_test_fold, create_symptom_model())
    training_accs.append(training_acc)
    testing_accs.append(testing_acc)
    # Print the training and testing accuracy for this fold
    print(f"Fold {fold+1}: Training Accuracy = {training_acc:.4f}, Testing Accuracy =
{testing_acc:.4f}")
   print("----------------------------------------------------------------------------------------------------
")
# Calculate the average training and testing accuracy across all folds
avg_training_acc = sum(training_accs) / len(training_accs)
avg_testing_acc = sum(testing_accs) / len(testing_accs)
# Print the average training and testing accuracy
print(f"Average Training Accuracy = {avg_training_acc:.4f}, Average Testing Accuracy =
{avg_testing_acc:.4f}")
```

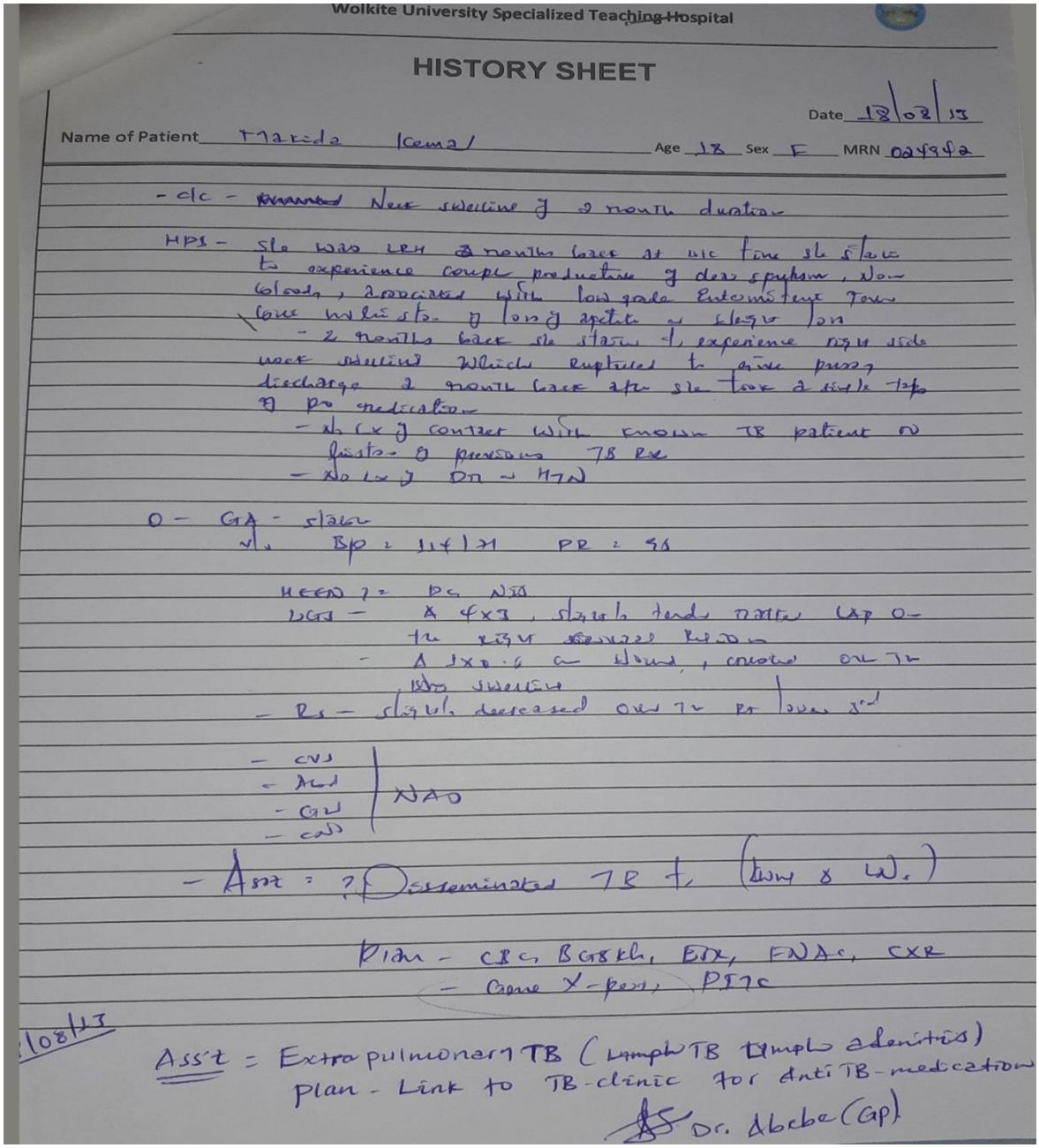
Appendix G Image Classification

```
# Define the number of folds for k-fold cross-validation
num_folds = 10
# Create a stratified k-fold object
kf = StratifiedKFold(n_splits=num_folds, shuffle=True)
# Create an empty list to store the accuracy for each fold
training_accs = []
testing_accs = []
# Loop through each fold
for img_fold, (img_train_idx, img_test_idx) in
enumerate(kf.split(img_train_data_arr_reshaped, img_train_labels_arr_reshaped)):
    print(f'Fold {img_fold+1}/{num_folds}')
    # Split the data into training and testing sets for this fold
    X_train_fold, y_train_fold = img_train_data_arr_reshaped[img_train_idx],
img_train_labels_arr_reshaped[img_train_idx]
    X_test_fold, y_test_fold = img_train_data_arr_reshaped[img_test_idx],
img_train_labels_arr_reshaped[img_test_idx]
    # Train the model on this fold and store the accuracy
    training_acc, testing_acc = train_image_model(X_train_fold, y_train_fold, X_test_fold,
y_test_fold, create_image_model())
    training_accs.append(training_acc)
    testing_accs.append(testing_acc)
    # Print the training and testing accuracy for this fold
    print(f"Fold {img_fold+1}: Training Accuracy = {training_acc:.4f}, Testing Accuracy =
{testing_acc:.4f}")
    print("----------------------------------------------------------------------------------------------------
")
# Calculate the average training and testing accuracy across all folds
avg_training_acc = sum(training_accs) / len(training_accs)
avg_testing_acc = sum(testing_accs) / len(testing_accs)
# Print the average training and testing accuracy
print(f"Average Training Accuracy = {avg_training_acc:.4f}, Average Testing Accuracy =
{avg_testing_acc:.4f}")
```

Appendix H DICOM X-ray Image File Sample



Appendix I TB Patient History Sample



Wolkite University Specialized Teaching Hospital

HISTORY SHEET

Date 18/08/13

Name of Patient Tigeste (Cema) Age 18 Sex F MRN 024942

- c/c - Neck swelling of 2 month duration

HPI - She was LRH 2 months back at this time she starts to experience cough productive of dark sputum, Non bloody, associated with low grade Intermittent fever, Loss of weight & loss of appetite & sleep loss
- 2 months back she starts to experience right side neck swelling which ruptured to give pussy discharge 2 month back after she took 2 weeks Tx of Po medication
- No Hx of contact with known TB patient or history of previous TB Rx
- No Hx of DM & HTN

O - GA - stable
VS Bp = 110/70 PR = 96

HEENT = Bs WNL
Neck - A 4x3, slightly tender matted LAP on the right supraclavicular region
- A 1x0.6 cm round, circular ulcer with swelling
- Rs - slightly decreased over Rt lower zone

- CVS
- Abd
- GU
- CNS

WNL

- Asst : ? Disseminated TB + (LNs & Lu.)

Plan - CBC, Blood Ch, BIX, FNAC, CXR
- Gene X-pert, PTTC

10/8/13

Asst = Extra pulmonary TB (Lymph TB Lymph adenitis)
Plan - Link to TB-clinic for anti TB-medication

Dr. Abebe (GP)