

**RELIABLE TUBERCULOSIS DETECTION FROM CHEST X-RAY  
USING CONVOLUTIONAL NEURAL NETWORK**

by

NAZIFA AFROZ CHOWDHURY

MD SADIQUL ISLAM

MD SAJIDUZZAMAN KHAN



Department of Information and Communication Technology

Faculty of Science and Technology

BANGLADESH UNIVERSITY OF PROFESSIONALS

December 2023

## APPROVAL

The thesis titled "RELIABLE TUBERCULOSIS DETECTION FROM CHEST X-RAY USING CONVOLUTIONAL NEURAL NETWORK" submitted by Nazifa Afroz Chowdhury, Roll No: 2054901027, Md Sadiqul Islam, Roll No: 2054901063 and Md Sajiduzzaman Khan, Roll No: 205490188, Session: 2019-20 is regarded as sufficient to partially meet the requirements for the Bachelor of Science degree in Information and Communication Technology.

---

Dr. Ahmedul Kabir  
Associate Professor  
Institute of Information Technology  
University of Dhaka

## DECLARATION

We declare that this thesis is our authentic creation, and we have composed it in its entirety. I have appropriately recognized and credited all the sources of information that have been utilized in the thesis. The thesis has not been previously submitted, either in full or in part, for any degree or diploma at any institution.

---

Nazifa Afroz Chowdhury

ID: 2054901027

Department of Information and Communication Technology

Faculty of Science and Technology

Bangladesh University of Professionals

20 December, 2023

---

Md Sadiqul Islam

ID: 2054901063

Department of Information and Communication Technology

Faculty of Science and Technology

Bangladesh University of Professionals

20 December, 2023

---

Md Sajiduzzaman Khan

ID: 2054901088

Department of Information and Communication Technology

Faculty of Science and Technology

Bangladesh University of Professionals

20 December, 2023

## ACKNOWLEDGEMENTS

First and foremost, praises and gratitude to the Almighty for his blessing showers during our study work to successfully complete the thesis. We want to give our sincere and honest appreciation for giving us the opportunity to study in this field and provide invaluable guidance during the thesis to our thesis Supervisor, Dr. Ahmedul Kabir, Associate Professor, Institute of Information Technology, University of Dhaka. We were greatly inspired and motivated by his vitality, foresight, integrity, and enthusiasm. His instruction has encompassed the techniques required to carry out the research and ensure the utmost transparency in the thesis. Conducting research under his direction was an honor and a tremendous privilege.

We express our gratitude to the faculty of the Department of Information and Communication Engineering at Bangladesh University of Professionals for imparting information during our studies, enabling us to concentrate on our thesis. We express our utmost gratitude to our parents for their unwavering commitment, supplications, nurturing, and selfless acts of providing us with education and equipping us for the path that lies ahead. We express our gratitude to all our classmates who have provided us with direct or indirect support in completing the research job.

## ABSTRACT

Tuberculosis (TB), also referred to as the "white death" or consumption, is an infectious disease caused by the bacteria *Mycobacterium tuberculosis* (MTB) which mostly targets the lungs but can also impact other organs. Tuberculosis is curable but it can cause death if not treated early. Due to drug-resistant *Mycobacterium* TB strains and other related ailments, this field needs more research. Automated Computer Aided Diagnosis (CAD) systems can enhance and speed up the diagnosis operation. Chest X-ray can indicate lung abnormalities caused due to TB. In this work, we have detected TB reliably from the chest X-ray images using image pre-processing, data augmentation, and deep-learning classification techniques. In this thesis, a Convolutional Neural Network (CNN) has been developed to diagnose TB. The model has been trained, validated, and evaluated on two different data sets. Public data was used for training and validation. We utilised Bangladeshi data for testing that we personally gathered. From the experiment, it is seen that the model has achieved 97.77% accuracy, 95.24% precision, 100% recall and 97.56% f1 score.

**Keywords:** TUBERCULOSIS, CNN, CAD, CXR.

# TABLE OF CONTENTS

<b>APPROVAL</b>	ii
<b>DECLARATION</b>	iii
<b>ACKNOWLEDGEMENTS</b>	iv
<b>ABSTRACT</b>	v
<b>LIST OF FIGURES</b>	viii
<b>LIST OF TABLES</b>	ix
<b>LIST OF ABBREVIATIONS</b>	x
<b>CHAPTER 1. INTRODUCTION</b>	1
1.1 Overview	1
1.2 Background	1
1.3 Motivation of the study	3
1.4 Problem Statement	3
1.5 Objective of the study	4
1.6 Contribution of the study	4
1.7 Outline of the book	5
<b>CHAPTER 2. LITERATURE OVERVIEW</b>	6
2.1 Overview	6
2.2 Existing Datasets	6
2.3 Existing works	7
2.3.1 CNN Based Works	7
2.3.2 Ensemble technique Based Work	9
2.3.3 Transfer Learning Based Works	12
<b>CHAPTER 3. BACKGROUND TOOLS AND TECHNOLOGIES</b>	15

3.1	Overview	15
3.2	Hardware Configuration	15
3.3	Description of Software	15
3.4	Machine Learning Libraries	16
3.4.1	NumPy	16
3.4.2	Pytorch	16
3.4.3	Pandas	17
3.4.4	Matplotlib	17
3.4.5	Scikit-Learn (SK learn)	18
3.4.6	Seaborn	18
3.5	Technologies	19
3.5.1	Convolutional Neural Network	19
3.6	Other Key Concepts	23
3.6.1	Batch normalization	23
3.6.2	Adam Optimizer	24
3.6.3	Activation functions	24
3.6.4	Visualisation	26
<b>CHAPTER 4. SYSTEM ARCHITECTURE AND METHODOLOGY</b>		<b>27</b>
4.1	Overview	27
4.2	Experimental Setup	27
4.3	Dataset Preparation	28
4.4	Proposed model Architecture	31
4.4.1	Feature Extraction	35
4.5	Proposed Model Implementation	37
<b>CHAPTER 5. RESULT AND DISCUSSION</b>		<b>39</b>
5.1	Overview	39
5.2	Evaluation Method	39
5.3	Analysis of Experimental Model Performance	40
5.3.1	Model Evaluation on Test Data	41
5.3.2	Detailed classification report	43
5.3.3	Comparison with Existing Work	45
<b>CHAPTER 6. CONCLUSIONS</b>		<b>48</b>
6.1	Overview	48
6.2	Conclusions	48
6.3	Major Findings and Future Scopes	49
<b>REFERENCES</b>		<b>50</b>

## LIST OF FIGURES

<b>Fig. No.</b>	<b>Title</b>	<b>Page No.</b>
3.1	Convolutional layer	21
3.2	Pooling layer	22
3.3	Fully connected layer	23
3.4	ReLU Activation Function	25
4.1	Data Splitting For Training, Validation And Testing	28
4.2	(a) Steps of pre-processing of Kaggle dataset (b) Steps of pre-processing of Collected dataset	29
4.3	Architecture of the Proposed Model	34
4.4	Diagram of Feature extraction process	35
5.1	Confusion Matrix	41



## LIST OF TABLES

Table No.	Title	Page No.
2.1	Details of the existing datasets.	6
4.1	Description of Kaggle dataset	28
4.2	Description of Collected dataset	29
4.3	Epoch and Time Criteria of Training Model	38
4.4	Optimized value of model parameters	38
5.1	Experimental Model Performance details	42
5.2	Classification report of the proposed model	43
5.3	Comparison of Proposed Model with Existing Works	45

## LIST OF ABBREVIATIONS

**WHO** World Health Organization

**TB** Tuberculosis

**CNN** Convolutional Neural Network

**CXR** Chest X-ray

**CAD** Computer Aided Diagnosis

**LLC** Limited Liability Company

**NIAD** National Institution for Academic Degrees

**SVM** Support Vector Machine

**NLM** Neural Language Models

**RSNA** Radiological Society of North America

**PAD** Peripheral Artery Disease

**CIFAR** Canadian Institute of Advanced Research

**SVHN** Street View House Numbers

**DCNN** Deep Convolutional Neural Network

**LSVR** Liver Segmental Volume Ratio

**ILD** Interstitial Lung Disease

**TSCM** Technical Surveillance Counter Measures

**VGG** Visual Geometry Group

**Numpy** Numerical Python

**ReLU** Rectified Linear Unit

# CHAPTER 1

## INTRODUCTION

### 1.1 Overview

This chapter provides a sequential description of the background information relevant to the proposed work. The discussion of motivation and problem statement provides an overview of the research. The goals of the work are clarified in accordance with the prior discussions. Finally, the paper outlines the contribution of the research in demonstrating the effectiveness of the proposed model in detecting Tuberculosis using CXR images.

### 1.2 Background

Tuberculosis is a deadly transmissible disease that affects people worldwide. It is mostly caused by a type of bacteria named *Mycobacterium tuberculosis*. Tuberculosis can not only infect the lungs, but also impact the bones and spine. Tuberculosis is an air disease that can be transmitted by anyone, anyplace. During inhalation, sneezing, or spitting, tuberculosis bacteria are widely disseminated into the air. According to the WHO, around 10.6 million individuals suffered from TB globally in 2022. This includes 5.8 million males, 3.5 million females, and 1.3 million children. Tuberculosis was responsible for the deaths of 1.3 million people in 2022. Tuberculosis (TB) ranks

as the second most severe transmitted infection worldwide[1]. According to the WHO, over 360,000 individuals in Bangladesh contract tuberculosis annually, resulting in approximately 40,000 deaths each year.

Without timely detection, TB can lead to the widespread spread and formation of cavities filled with dead lung tissue, which raises the risk of severe hemoptysis. The prompt identification and diagnosis of tuberculosis has therefore emerged as a key problem. The diagnosis of tuberculosis relies on the examination of chest X-rays conducted by radiologists. The chest X-ray examination has the ability to not only identify tuberculosis at an early stage, but also to detect minor or concealed lesions. Nevertheless, manual diagnosis is prone to a specific rate of misdiagnosis and is also a time-consuming process. Additionally, it needs radiologists possessing deep expertise in medical imaging. Nevertheless, the lack of adequate healthcare facilities in economically underdeveloped nations that are still in the process of development poses significant challenges to timely disease identification.

Deep learning has demonstrated strong performance in image object detection and classification, and has also found numerous valuable applications in medical imaging. The CNN model, renowned for its popularity, utilizes feature extraction to recognize images. The CNN has made major progress in computer-aided image classification tasks and achieved more significant outcomes compared to traditional machine learning methods [2]. Consequently, numerous researchers are presently adopting deep learning features for detecting abnormalities in the chest region, producing superior outcomes in comparison to standard machine learning methods. The primary benefit of utilizing CNN is its ability to automatically extract distinctive features while taking into account the uniqueness of the data. The pulmonary tuberculosis images often show noise and interference from surrounding tissue. Additionally, there is a limitation in the number of training images, which hinders the ability of the simple CNN model to accurately identify the relevant characteristics.

### 1.3 Motivation of the study

The purpose for researching tuberculosis detection through chest X-rays is to tackle the obstacles presented by this deadly illness, which remains a substantial worldwide health issue. Tuberculosis has a significant global impact, particularly in settings with limited resources and access to advanced diagnostic instruments, where millions are afflicted. Chest X-rays give a potentially fruitful approach owing to their extensive accessibility and economical nature. It is critical to develop precise and effective techniques for identifying tuberculosis-related irregularities in these images in order to facilitate earlier detection, timely initiation of treatment, and containment of the disease's transmission. By harnessing technological advancements like machine learning and CNN, this research endeavors to enhance the accuracy and speed of TB detection through chest X-rays, aiming to significantly impact patient outcomes and public health interventions in the fight against tuberculosis.

### 1.4 Problem Statement

Despite the availability of chest radiography as a widely accessible imaging tool, the accurate and timely identification of tuberculosis-related anomalies remains a critical issue. Existing manual interpretation methods are subjective, time-consuming, and prone to inter-observer variability, impeding swift diagnosis and treatment initiation. The global burden of tuberculosis demands efficient, scalable, and accurate diagnostic solutions, urging the exploration and development of automated, machine-learning-driven approaches that can reliably detect and classify tuberculosis-related patterns in chest X-ray images. Addressing these challenges is pivotal for enhancing diagnostic accuracy, enabling earlier intervention, and curbing the spread of tuberculosis, particularly in areas with limited healthcare resources. As of now, tuberculosis detection model didn't use X-ray from any south asian country as data set.

## 1.5 Objective of the study

- a) To preprocess and augment the unbalanced dataset in order to enhance the model's generalizability and minimize bias by creating a balanced dataset.
- b) To develop a CNN model capable of detecting Tuberculosis.
- c) To investigate data augmentation techniques on the dataset and recommend the most suitable model.
- d) To assess and compare the performances of the implemented models using established metrics such as accuracy, precision, recall, and F1 score.

## 1.6 Contribution of the study

- a) Model implementation using publicly available data as well as data obtained from several hospitals in Bangladesh.
- b) Data Augmentation was utilized for the improvement of precision.
- c) A new CNN architecture was developed for identification of Tuberculosis through chest X-ray.

## 1.7 Outline of the book

The subsequent sections of the book are structured as follows. **Chapter 2** Research and pre-existing datasets are included. A fundamental literature review, along with to the works themselves, can be studied in this section. **Chapter 3**, shows in detail the many different technologies and tools utilized in this chapter. **Chapter 4**, the evaluation of deployed pre-trained models and classifiers is examined, and a novel approach for achieving improved outcomes is introduced. In **Chapter 5**, we conducted an analysis of the experimental results and performed a performance comparison of our suggested approach on both the validation and testing sets. Finally, **Chapter 6**, Concludes the report by providing a succinct analysis of the limitations and potential areas of improvement in the study.



# CHAPTER 2

## LITERATURE OVERVIEW

### 2.1 Overview

This chapter provides a comprehensive review of current datasets and relevant work related to the detection of Tuberculosis using Chest X-Ray.

### 2.2 Existing Datasets

Datasets relating to detection of Tuberculosis using Chest X-Ray can be gathered for research at a number of different websites. The table below displays numerous primary sources for TB datasets.

**Table 2.1:** Details of the existing datasets.

1	Kaggle	<a href="https://www.kaggle.com/datasets">https://www.kaggle.com/datasets</a>
2	NLM dataset	<a href="https://www.ncbi.nlm.nih.gov/datasets">https://www.ncbi.nlm.nih.gov/datasets</a>
3	NIAID TB dataset	<a href="https://www.niaid.nih.gov/research/tb-portals">https://www.niaid.nih.gov/research/tb-portals</a>
4	RSNA dataset	<a href="https://www.rsna.org/rsnai">https://www.rsna.org/rsnai</a>
5	VinDr-CXR dataset	<a href="https://vindr.ai/datasets/cxr">https://vindr.ai/datasets/cxr</a>
6	PadChest dataset	<a href="http://bimcv.cipf.es/bimcv-projects/padchest/">http://bimcv.cipf.es/bimcv-projects/padchest/</a>

## 2.3 Existing works

### 2.3.1 CNN Based Works

F. Pasa et al.[3] (2019) applied hand-crafted algorithms using SVMs and CNNs to automatically diagnose TB from chest X-rays (CXR). Deep neural networks for tuberculosis detection largely categorize real images. These models overfit and are hard to execute in mobile situations due to their many parameters and technology requirements. Optimized for the task at hand, convolutional neural networks exhibit enhanced speed and efficiency compared to preceding models, all the while maintaining accuracy. The graphical representation of the convolutional neural network is understudied. Saliency maps and grad-CAMs were applied by the authors to visualize tuberculosis, and radiology was also explored.

The articles included in this special section on deep learning technology and its applications were suggested by Hayit Greenspan et al.[4] (2016) Deep learning is one of the ten innovative technologies of 2013 and is becoming more prevalent across all data analysis. Deep learning outperforms artificial neural networks and makes more accurate predictions from higher-level data by adding layers. Presently, it stands as the preeminent machine-learning instrument in the domains of general imaging and artificial intelligence. Convolutional neural networks excel at many computer vision applications. Deep CNNs automatically identify mid-level and high-level abstractions from unprocessed data, such as images. Recent studies demonstrate that convolutional neural network (CNN)-derived generic descriptors are effective in natural visual object detection and localization. In a vast array of medical image analysis applications, CNNs and other deep learning methodologies are being rapidly adopted by multinational corporations.

Rahul Hooda et al.[5] (2017) focused on TB, a serious health problem in developing countries. Inadequate treatment and diagnosis errors lead to several patient deaths

annually. Creating a computer-aided diagnosis (CAD) system for TB detection can benefit in early detection and disease control. Many CAD systems currently rely on handwritten features, but there is a growing trend towards deep-learning-based autonomous feature extractors. This research introduces a deep-learning approach for tuberculosis detection, categorizing CXR images as normal or abnormal. The authors utilized a CNN architecture with 7 convolutional and 3 fully linked layers. They compared the performance of three optimizers. The Adam optimizer proved most accurate with an overall accuracy of 94.73% and validation accuracy of 82.09%. All results are based on public domain Montgomery and Shenzhen databases.

The Dense Convolutional Network (DenseNet), proposed by Gao Huang et al.[6] (2017) propagates information from each layer to the subsequent layer in a forward direction. Recent research suggests that convolutional networks with shorter connections between input and output layers have the potential to be deeper, more accurate, and more efficient to train. The convolutional network is unique in that it has  $L(L+1)$  direct connections, which sets it apart from the usual  $L$ -layer networks. Layers utilize the feature maps from previous layers as well as their own feature maps to feed into all subsequent layers. DenseNets address the issue of vanishing gradients, enhance the propagation of features, promote the reuse of information, and reduce the number of parameters. The architecture is evaluated using four challenging object recognition benchmark tasks: CIFAR-10, CIFAR-100, SVHN, and ImageNet. DenseNets provide significantly superior performance compared to the current best methods across most tasks, while requiring less computational resources.

U.K. Lopes et al. [7] (2017) demonstrated the severity of tuberculosis. The majority of the 1.8 million TB deaths in 2015 occurred in underdeveloped nations. Many deaths could have been prevented if the disease had been recognized sooner, but the most advanced testing technologies are too expensive for mass usage. Although frontal thoracic radiograph analysis is a well-liked method for diagnosing tuberculosis,

it has limitations due to the need for individual examination by qualified radiologists. There is a significant study on automating diagnosis using computational approaches on medical images, reducing the requirement for individual picture analysis and overall expenditures. Recent advances in deep learning have led to successful picture classification across various domains, although its use in tuberculosis diagnosis is still limited. This study aims to advance research by proposing three methods for using pre-trained convolutional neural networks as feature extractors to detect disease. This work implements and compares the proposed solutions to the current literature. The results show that pre-trained convolutional networks may effectively extract medical picture features, competing with previous efforts.

Alex Krizhevsky et al.[8] (2012) constructed a deep convolutional neural network to categorize 1.2 million images in the ImageNet LSVRC-2010 challenge. Images with a high resolution were classified into 1000 classes. The top-1 and top-5 error rates of 37.5% and 17.0% were significantly improved above the prior state-of-the-art on test data. The neural network, with 60 million parameters and 650,000 neurons, features five convolutional layers, max-pooling layers, three fully connected layers, and a 1000-way softmax. Using non-saturating neurons and a productive GPU convolution implementation allowed for faster training. Overfitting in fully connected layers was successfully reduced using a new regularization method termed "dropout." A variant of this model won the ILSVRC-2012 competition with a top-5 test error rate of 15.3%, outperforming the second-best entry with 26.2%.

### **2.3.2 Ensemble technique Based Work**

Hanafi Ahmad Hijazi et al.[9] (2014) discussed the global impact of tuberculosis (TB), a highly lethal infectious disease. Mycobacterium tuberculosis causes tuberculosis. Early tuberculosis detection is crucial for reducing illness and mortality. TB diagnosis: chest x-ray and sputum test. Radiologists face challenges in distinguishing

between tuberculosis and lung cancer due to their similar appearance. Identifying the objects of interest is required for machine learning to detect tuberculosis. Detecting early stages may be difficult due to the similarity of small nodules, capillaries, and tissues in the image. This article presents an ensemble deep learning method for TB detection that doesn't need object segmentation. The proposed approach achieved high accuracy (91.0%), sensitivity (89.6%), and specificity (90.7%) in the evaluation.

Hooda Rahul et al.[5] (2017) addressed the infectious disease known as tuberculosis (TB) mostly affects the lungs. Chest radiography is mostly used for its initial screening, as advised by the World Health Organization. Many computer-aided diagnostic (CAD) systems have been developed to aid radiologists in the diagnosis of this disease. However, because it is so difficult to diagnose tuberculosis automatically, these systems are still in the early stages of research. This is because the effects of TB on the CXR vary greatly. This paper presents a deep learning-based TB detection method that achieves remarkably high accuracy. The suggested approach consists of the AlexNet, GoogleNet, and ResNet standard architectures ensembled. This research's key contribution is that it trains these architectures from the ground up and forms an ensemble capable of carrying out TB classification. The combined dataset created by combining publicly accessible standard datasets is used to train and assess the suggested approach. The ensemble outperforms the majority of the current approaches, with an accuracy of 88.24% and an area under the curve of 0.93.

Stefanus Kieu Tao Hwa et al.[6] (2017) discussed Tuberculosis (TB) is a potentially fatal disease if not promptly treated. Ensemble deep learning can assist in the early detection of tuberculosis. The previous study exclusively trained the ensemble classifiers using images that possessed similar attributes. A group or collection of items or individuals Optimal performance is contingent upon the presence of a wide range of mistakes, which can be obtained by the utilization of various classification ap-

proaches or feature sets. This research specifically examines the use of deep learning and contrast-enhanced canny edge detection (CEED-Canny) in the identification of tuberculosis (TB) using x-ray images. The CEED-Canny algorithm was employed to generate edge-enhanced pictures of the lung x-ray. Two categories of features were produced: the first was derived from the enhanced X-ray images, while the second was obtained from the Edge detected images. The suggested modification of characteristics raised the range of mistakes made by the fundamental classifiers and enhanced the detection of TB. The suggested ensemble technique achieved an accuracy of 93.59%, a sensitivity of 92.31%, and a specificity of 94.87%, which is similar to the results obtained in earlier studies.

Priyanka Saha deliberated. [10] (2020) Pulmonary tuberculosis persists as a global epidemic and remains a prominent contributor to mortality, as reported by the World Health Organization (WHO). The global incidence of reported cases is experiencing a significant escalation. A computerized computer-aided diagnostic (CAD) system is necessary for timely identification and mitigation. Deep learning algorithms have become widely used methods for disease detection. Chest X-rays are widely favored for detecting tuberculosis due to their widespread accessibility and comparatively affordable price. The author incorporated an ensemble Convolutional Neural Network (CNN) Dempster-Shafer-based architecture into suggested tuberculosis detection model. The photos are initially upgraded using the CLAHE method to improve the contrast. Subsequently, the training set is expanded to expand the size of the dataset used for training. The experiment yielded a precision rate of 94.21%. The method suggested may accurately identify whether an x-ray image is normal or exhibits signs of tuberculosis. This methodology will facilitate the prompt and precise identification of tuberculosis.

### 2.3.3 Transfer Learning Based Works

Hoo-Chang Shin et al.[11] (2016) explore three significant, although previously understudied features in utilizing deep convolutional neural networks for computer-aided detection challenges. Initially, they examine and assess several convolutional neural network Architecture. The analyzed models consist of a range of 5 thousand to 160 million parameters and exhibit variations in the number of layers. Next, we assess the impact of dataset size and geographical picture context on performance. Ultimately, they investigate the circumstances and rationale for the use of transfer learning from pre-trained ImageNet, namely through the process of fine-tuning. They focus their research on two distinct computer aided detection (CAdE) problems: the identification of lymph nodes in the thoraco-abdominal region, and the categorization of interstitial lung diseases (ILD). They have achieved exceptional performance in detecting mediastinal lymph nodes, with a sensitivity of 85% and just 3 false positives per patient. Additionally, they provide the initial findings of a five-fold cross-validation classification study on predicting ILD categories using axial CT slices. The comprehensive empirical assessment, analysis of the CNN model, and significant insights we have obtained may be applied to develop high-performance CAD systems for various medical imaging applications.

Ray-I Chang et al.[12] (2020) adopt transfer learning approach for Two-stage classification of tuberculosis culture diagnosis. Deep learning is mainly a data-driven approach. Large quantities of data are necessary to learn the rules. It is difficult for deep learning algorithms to maximize the parameters of prediction models when there is a shortage of training data. The lack of annotation and restricted accessibility of medical data pose significant obstacles to the training of supervised neural networks. Transfer learning, which typically employs a pre-trained model as a feature extractor to project the input into another vector space that is less redundant and more informative, or fine-tunes a pre-trained model with a limited quantity of

data, is one solution. Various CADx systems for purposes including interstitial lung disease classification, chest X-ray classification, and tuberculosis acid-fast staining microscopy diagnosis have been developed utilizing transfer learning. These pieces of work illustrate the effectiveness of transfer learning. Due to the limited quantity of data available, they opt for transfer learning as the solution to this problem involving the automated diagnosis of tuberculosis. Furthermore, TSCM is offered as a means to improve the recall of the non-negative class. The the highest recall achieved for the non-negative class is 98%. A high recall value signifies the successful detection of anomalies. Demonstrate that the proposed method can classify the non-negative class with 99% precision and 98% recall.

A. abbas et al.[13] (2020) CNNs facilitate the direct learning of hierarchical and highly representative local image features from data. However, the availability of annotated data continues to be the greatest obstacle in the field, particularly in the domain of medical imaging. Through the transfer of knowledge from generic image recognition tasks to medical image classification, transfer learning may offer a viable and efficient solution. As a result of irregularities in the distribution of the dataset, transfer learning is rarely able to offer a reliable solution. Class decomposition enables the identification of class boundaries within a dataset, which is subsequently more amenable to addressing any anomalies in the data distribution. The paper introduces the Decompose, Transfer, and Compose (DeTraC) approach, an innovative CNN architecture that leverages class decomposition and transfer learning to enhance the performance of medical image classification. Their proposed methodology was verified using three distinct cohorts of digital mammograms, histological images of human colorectal cancer, and chest X-ray images. When class decomposition was implemented using a deep-tuning strategy, DeTraC achieved high accuracies of 99.8% with ResNet on CXR images, 98.5% with VGG16 on digital mammograms, and 99.7% with GoogleNet on CRC images.



M. Ahsan et al.[14] (2019) introduced a ConvNet model that classifies CXR images of patients with tuberculosis using VGG16. They demonstrate that VGG16 can classify the results with comparable accuracy using the raw data without requiring any pre-processing. In an attempt to attain a higher level of accuracy, VGG16 was reapplied to a subset of data subsequent to the augmentation process. The results demonstrated that the application of VGG16 to augmented images improves precision. a combination of the numerous hidden layers containing filters in CNNs, the model achieves an exceptionally high level of accuracy of 80% without augmenting it and 81.25% after augmenting it.

S. Christodoulidis et al.[15] (2016) An advanced approach is introduced to train the suggested network through the transmission of knowledge from a related field, namely general texture classification. The proposed architecture is employed to pre-train networks using six publicly accessible texture databases. Subsequently, the networks undergo fine-tuning using lung tissue data. An ensemble is formed from the resultant CNNs, and the merged knowledge of that ensemble is compressed back into a network with its original architecture. The proposed methodology led to an approximate 2% increase in the overall efficacy of the proposed CNN. The findings illustrate the potential of transfer learning in the domain of medical image analysis, underscore the textural characteristics of the issue, and demonstrate that the training method employed for a network can be equally critical to its architectural design.

## CHAPTER 3

### BACKGROUND TOOLS AND TECHNOLOGIES

#### 3.1 Overview

In the previous chapter, we acquired knowledge about the literature review. In this chapter, we will have a detailed understanding of the tools and techniques.

#### 3.2 Hardware Configuration

The entire work has been carried out using an AMD Ryzen 5 3600 CPU, a 64-bit operating system, 16 GB of RAM, and Windows 10 Pro. Jupyter Notebook is used to implement the model.

#### 3.3 Description of Software

Jupyter Notebook is a dynamic computational platform that enables users to generate and distribute documents that include executable code, mathematical equations, graphic representations, explanatory text, and other features. It enables seamless integration of code execution, statistical analysis, visualizations, and insights in an organized way.

## 3.4 Machine Learning Libraries

A number of widely recognized Python libraries, including Numpy, Matplotlib, Pytorch, Pandas, and Scikit-Learn, are used by the given model.

### 3.4.1 NumPy

NumPy, which stands for Numerical Python, is a fundamental and powerful Python library used for manipulating multidimensional arrays and matrices as well as performing numerical computations. In addition to random number generation, fourier analysis, and mathematical operations, it offers an extensive collection of tools and functions for linear algebra. The array object, known as ndarray, is an essential component that permits the handling of enormous datasets in a manner that is both efficient and quick. It provides a vast array of mathematical functions and high-performance array operations, which facilitate scientific computing, data analysis, and machine learning. As a result, it is a vital foundation for numerical computing in Python.

### 3.4.2 Pytorch

PyTorch is an open source software framework for deep learning that integrates the Torch backend machine learning library with a Python-based high-level API for the development of neural networks. Among other advantages, its adaptability and usability have established it as the leading machine learning framework within the academic and research communities. PyTorch provides extensive support for a diverse range of neural network architectures, comprising both basic linear regression algorithms and complicated convolutional neural networks, as well as generative transformer models that are used in domains such as computer vision and natural language processing (NLP). In addition, data scientists can utilize PyTorch to execute and validate code

segments in real time, eliminating the need to await the complete implementation of the code. As a result, PyTorch is an exceptional environment for rapid prototyping, and debugging is significantly accelerated. The mathematical and programming framework of PyTorch streamlines and simplifies workflows for machine learning while preserving the performance and precision of deep neural networks.

### **3.4.3 Pandas**

Pandas is a Python open-source library designed to facilitate the manipulation of high-performance data. It includes capabilities for data exploration, cleaning, analysis, and manipulation. Pandas is an essential tool for performing a wide range of tasks, including data loading and exploration, feature engineering, data manipulation, and data visualization. Additionally, Pandas can eliminate rows that are irrelevant or contain incorrect values, such as NULL or empty values. This is referred to as "Data Cleaning."

### **3.4.4 Matplotlib**

Matplotlib is a data visualization library for Python that enables users to make top-notch graphs, charts, and plots. In many fields, it is the preferred tool for data visualization due to its extensive set of features that allow users to generate static, interactive, and publishable visualizations. Matplotlib provides users with a lot of ways to change the labels, colours, markers, and styles on line plots, scatter plots, bar charts, histograms, heatmaps, and 3D graphs. Its applications range from visual report or dashboard development to data exploration and analysis, as well as scientific research and data presentation. Matplotlib is an essential tool for Python users of all experience levels when it comes to creating and understanding visual representations of data. It is quite versatile and easy to adapt.

### 3.4.5 Scikit-Learn (SK learn)

Scikit-learn is a Python library for machine learning, data mining, and data analysis. It is open-source and offers efficient tools for performing various tasks. It provides a consistent interface for a diverse set of machine learning methods, making it easy to use and accessible to people with different levels of knowledge. It consists of a wide range of preprocessing, classification, regression, clustering, dimensionality reduction, and model selection techniques. Scikit-learn's user-friendly interface, extensive documentation, and easy functionality make it an excellent choice for implementing and exploring machine learning methods. The applications of scikit-learn span across various domains and sectors, including the development of predictive models, identification of anomalies, recognition of patterns, creation of recommendation systems, processing of natural language, and identification of images, among other uses. It serves as a versatile set of tools to tackle practical difficulties and conduct studies in the fields of data science and machine learning. It simplifies the procedure of developing and deploying machine learning models in various fields.

### 3.4.6 Seaborn

Seaborn is one of the most remarkable Python libraries for graphical statistical plotting and visualization. Seaborn offers a wide variety of color palettes and attractive styles by default to ease the construction of numerous statistical plots in Python. The goal of the Seaborn library is to create a more aesthetically pleasing visualization of the basis of data analysis and perception. It is constructed upon the foundation of the Matplotlib library and offers APIs that are specific to datasets. Seaborn is seamlessly incorporated with the data structures of Panda, enabling users to navigate effortlessly between the diverse visual representations of a specific variable in order to enhance their comprehension of the provided dataset. The Seaborn Collection facilitates the construction of an extensive variety of graphical represen-

tations, including line graphs, scatter diagrams, bar plot lines, bar charts, counting plots, and related plot lines. One of the valuable libraries in projects that integrate intelligent systems with big data to enhance data presentation. Seaborn’s innovative syntax and usability are our greatest appreciations. To produce typical plots, only three functions are necessary.

## 3.5 Technologies

A detailed discussion of the background technologies that were used for developing the proposed model is found in the following section.

### 3.5.1 Convolutional Neural Network

Neural networks are designed in a manner that mirrors the structure of the human brain. Sensory data is used to identify patterns. By employing these numerical representations, the neural network is capable of detecting patterns that are essential to the data. Neural networks demonstrate the capability to execute tasks involving classification and segmentation. By utilising labelled datasets, they categorise and structure unannotated data by detecting similarities discovered during the training phase. The Convolutional Neural Network (CNN) is an architecture of a neural network designed specifically for image input processing. Convolutional neural networks (CNNs) demonstrate superior performance relative to alternative machine learning networks, particularly in the domain of image identification tasks. Convolutional neural networks (CNNs) utilise a hierarchical structure wherein neurons are stratified in order to facilitate the learning of representations and the efficient extraction of features. By utilising this architecture, CNNs are able to achieve remarkable accuracy in duties like image categorization and analysis. Convolutional neural networks are characterised by their exceptional performance when processing picture, speech, or audio

signal inputs, setting them apart from other types of neural networks. Their layers can be classified into three primary categories: 1. Convolutional layer 2. Pooling layer 3. Fully-connected layer

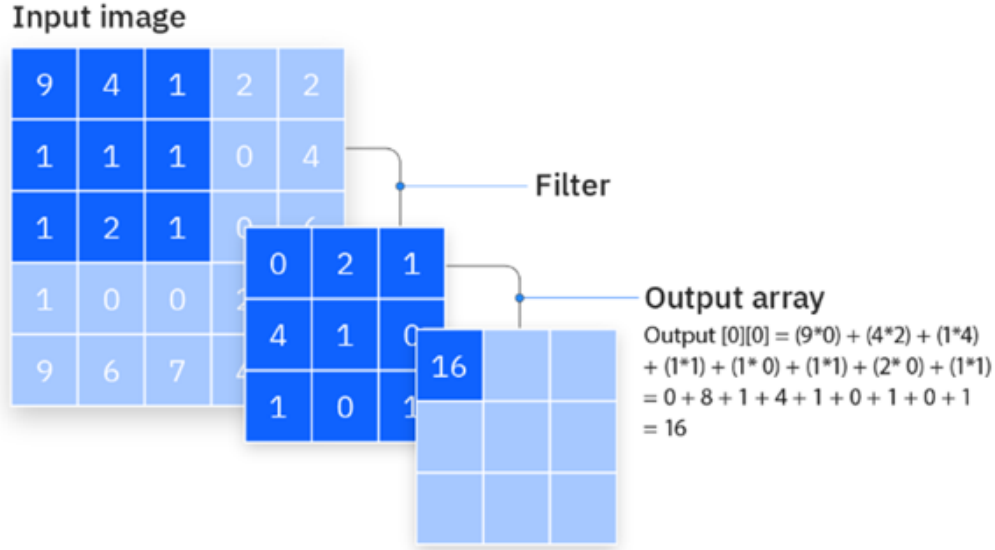
### 3.5.1.1 Convolutional layer

The convolutional layer serves as the initial layer in a convolutional network. As the foundational part of convolutional neural networks (CNNs), it is responsible for the majority of computational processing. The system processes input data, usually in the form of a 3D matrix that represents a colour image. The dimensions of the matrix correspond to the height, width, and depth of the image (RGB channels). In this layer, a feature detector, also known as a filter or kernel, scans the input using convolutions to identify specific characteristics within the image.

The feature detector, typically represented as a 2D array of weights such as a 3x3 matrix, examines the receptive field of the image. The process involves calculating the dot product between the filter and the input pixels, resulting in the creation of an output array referred to as a feature map or activation map. The convolved features are generated by repeatedly applying the filter to the entire image with a defined stride.

Parameter sharing is a key concept in which the weights of filters remain fixed during traversal, while other parameters can be adjusted during training using back propagation and gradient descent. Before training a neural network, three essential hyperparameters determine the size of the output.

A vital component of this type of convolution procedure is weight sharing, in which kernels are pooled in order to represent the entire image. Shared weight enables the generation of certain properties, such as feature extraction motifs, facilitates the pooling operation to generate scalable feature morphologies, and improves model performance through the reduction of variable count.



**Fig. 3.1.** Convolutional layer

$$O = \frac{(W - K + 2P)}{S} + 1 \quad (3.1)$$

Output height/ length is O, input height/ length= W, The filter size= K, The padding= P, and The stride= S.

### 3.5.1.2 Pooling layer

A well-known method of down-sampling, the pooling layer is a technique that decreases the in-plane dimension of the feature maps. This is done in order to reduce the number of learnable constraints that need to be applied and to guarantee a high level of accuracy even when there are slight shift distortions present. Due to the fact that the pooling layers do not contain any learnable parameters, despite the fact that padding, kernel size, and stride are hyper-parameters that are used in pooling operations that are similar to convolution processes.



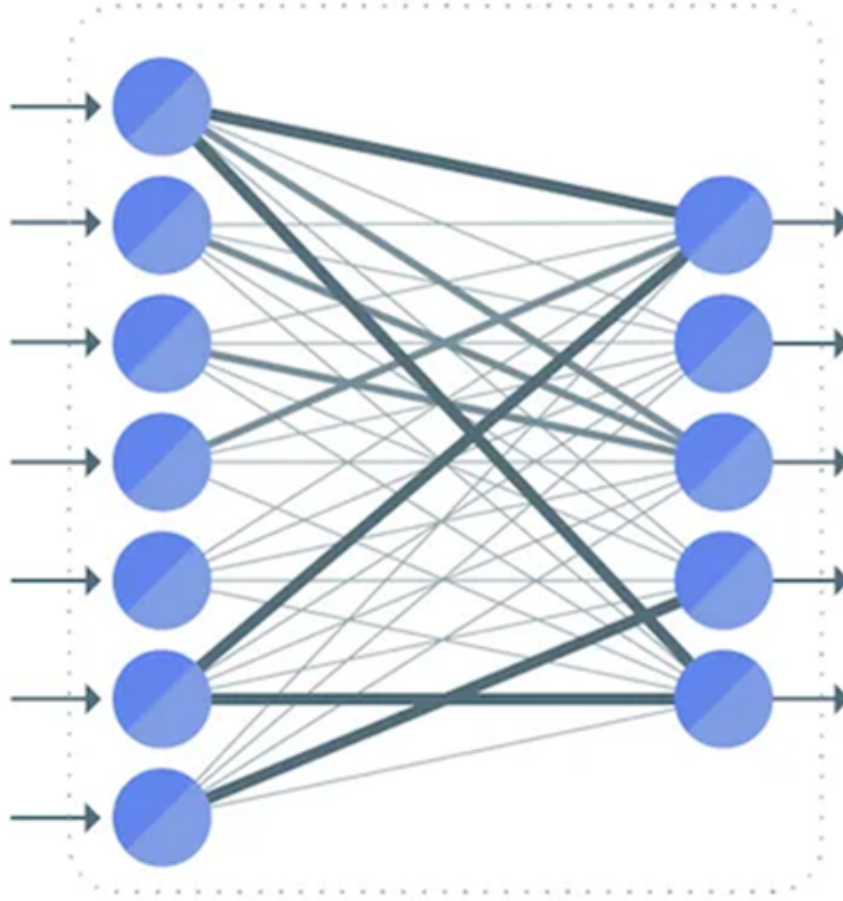


**Fig. 3.2.** Pooling layer

Among the several types of pooling, max pooling and average pooling are two distinct variations that select the highest and average values, respectively. The illustration in the figure depicts max pooling.

### 3.5.1.3 Fully Connected Layer

In Convolutional Neural Networks (CNNs), the output feature maps or the outcome of a pooling layer are frequently converted into a one-dimensional array of values. Afterwards, the resulting representation is fed into one or more fully connected layers, where each neuron is coupled to every neuron in the previous layer. An activation function is utilised to bring nonlinearity into the network after a weighted sum of the input received by each neuron in this layer. By aggregating data from each individual neuron in the previous layer, these layers allow the network to uncover advanced features and connections, which are crucial for identifying intricate patterns in the data. Neural networks often use fully linked layers for classification tasks, which are usually positioned at the end of the network.



**Fig. 3.3.** Fully connected layer

## 3.6 Other Key Concepts

### 3.6.1 Batch normalization

Batch Normalization is a technique that improves deep neural network training speed, stability, and performance. It normalizes layer activations by normalizing their inputs in a mini-batch. This normalization step reduces internal covariate shift, speeding up and improving model learning. Batch Normalization is used to calculate the mean and variance of each mini-batch feature and normalizing it. Subsequently, learnable parameters (gamma and beta) scale and move the normalized features to

give the network flexibility to alter the normalization. Batch Normalization reduces internal covariate shift ,improves gradient flow during backpropagation and speeds up training convergence.

### **3.6.2 Adam Optimizer**

The optimization method that you pick for the deep learning model will determine whether you reach satisfactory results in a few minutes, hours, or days. Through the utilization of the Adam optimization strategy, recent advancements have been made in deep learning for the processing of natural language and machine vision. For the purpose of continually updating the grating weights, Adam is an optimization approach that can be considered nearly new. This is in contrast to the standard stochastic gradient descent.

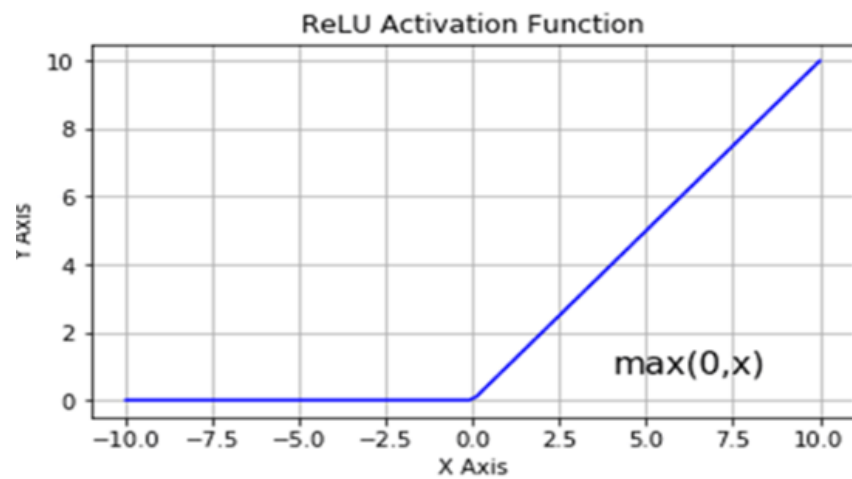
### **3.6.3 Activation functions**

The activation function is a nonlinear transformation that assigns the weighted sum as input and returns the result. Their purpose is to provide a simplified representation of neuronal activity that can be incorporated into deep neural networks. There are numerous additional activation functions that can be applied, such as sigmoid, hyperbolic tangent, logistic, and rectified linear units. It is a type of neural network function involving the activation of neurons. It is utilized to transfer the output of a hidden layer to the input of the subsequent layer.

#### **3.6.3.1 ReLU**

In deep neural networks, the Rectified Linear Unit or shortly known as (ReLU) activation function is a crucial component that introduces nonlinearity by retaining positive values unchanged while outputting zero for negative input values. The ReLU

algorithm calculates the output by summing the values of zero and the input. The computational efficiency, simplicity, and mitigation of the vanishing gradient problem that is commonly associated with conventional activation functions such as sigmoid or tanh are all features of this particular function. Recurrent Log Units (ReLUs) have been extensively implemented in diverse neural network architectures to streamline the convergence process during training, thereby enhancing model performance and learning velocity.



**Fig. 3.4.** ReLU Activation Function

### 3.6.3.2 Data Loader

The PyTorch utility class referred to as the DataLoader facilitates large amounts iteration over a Dataset object. It is optimized for efficiently managing large datasets and can be configured to perform data preprocessing in real time, load data in parallel, and shuffle data for each epoch.

The DataLoader accepts a Dataset object and offers several configuration choices, such as batch size, shuffling , and the quantity of worker processes to facilitate parallel data loading. It is the responsibility of the DataLoader class to batch the data and deliver it to the model in a format that it can consume. By utilizing the DataLoader,

users are able to:

- a) Define a dataset to work with: Determining the source of the data and the appropriate way of accessing it.
- b) Batch the Data : Identify the quantity of training or testing samples to be utilized during a specific iteration.
- c) Data Shuffle : As data is loaded into segments, PyTorch has the capability to manage the shuffling of data for the user. This can prevent accidental bias and increase the dataset's representativeness.
- d) Multiprocessing Support : PyTorch has been optimized to execute multiple processes concurrently, maximizing the utilization of contemporary CPUs and GPUs while reducing the time required for data training and testing. The `DataLoader` class specifies the maximum number of concurrent workers.
- e) Merge datasets : Additional functionality is provided by PyTorch to integrate multiple datasets. Although this may not be a frequent undertaking, its availability is an excellent feature.

### 3.6.4 Visualisation

Data visualisation is an essential component of machine learning as it empowers analysts to understand and derive meaning from data trends, patterns, and connections. By enabling the breakdown and distribution of data patterns and insights to a broader audience, data visualisation is a vital part of machine learning. In numerous CNN models, Matplotlib and Seaborn are utilised to visualise confusion matrices, training and testing metrics (loss, accuracy), and measure the model's quality through a comparison of the two.

# CHAPTER 4

## SYSTEM ARCHITECTURE AND METHODOLOGY

### 4.1 Overview

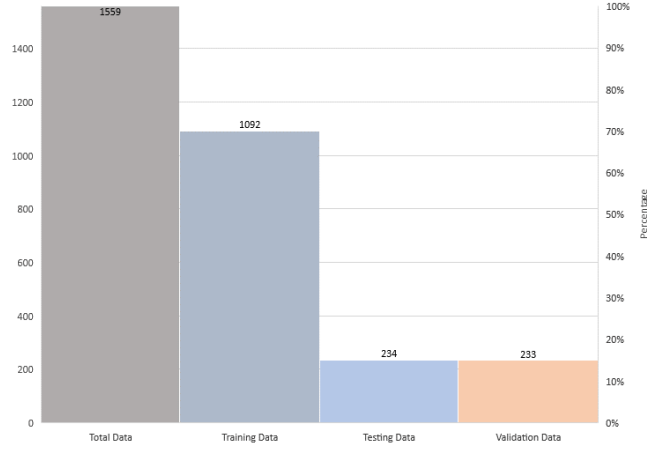
In this chapter, the methodology of the proposed model is discussed. This includes the experimental setup, the preparation of the dataset, the structure of the model, the proposed workflow, as well as the training and validation of the model.

### 4.2 Experimental Setup

Jupyter Notebook and Google Collab, both available via web browsers, served as the platform for running the proposed model. The local Jupyter Notebook machine is equipped with an AMD Ryzen 5 3600 CPU, a 64-bit operating system, 16 GB of RAM, and Windows 10 Pro. Kaggle kernels do not require any extra set up to perform and provide swift training of the model than training the model in a PC (without GPU).

### 4.3 Dataset Preparation

During this experiment, we utilized a variety of datasets for the purposes of training, validation, and testing. Same dataset was used for both the training and the validation processes. For the purpose of testing, a different data set was utilized[16].



**Fig. 4.1.** Data Splitting For Training, Validation And Testing

Combining two different datasets from Kaggle resulted in the creation of the dataset that was used for training and validation purposes. 1325 chest X-ray pictures were utilized for the purposes of training and validation. The testing was carried out using a dataset that we had gathered on our own during the process. The information was gathered from three separate diagnostic centers in Dhaka: the DIGI LAB, the Health Aid Diagnostic Center, and the Ibne Sina Diagnostic Center. We gathered a total of 234 X-ray images.

**Table 4.1:** Description of Kaggle dataset

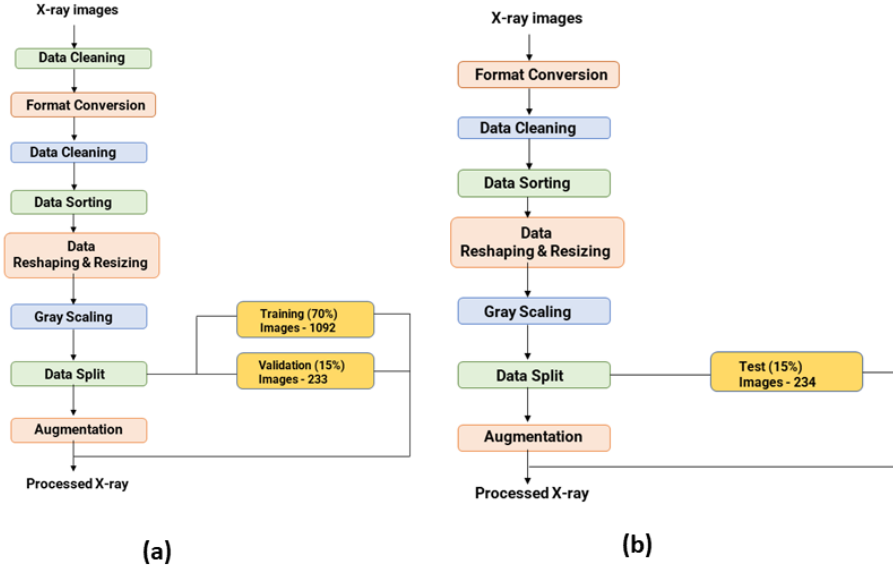
Dataset	No. of TB X-Ray	No. of Normal X-Ray	Total
Kaggle lung x-ray	615	710	1325

**Table 4.2:** Description of Collected dataset

Dataset	No. of TB X-Ray	No. of Normal X-Ray	Total
Ibne Sina, Health aid, DigiLab	89	145	234

The raw photos that were collected were varied in format, noisy, varied in size, and were blended with images of characters that were distinct from one another. In the course of pre-processing, photographs that are noise-free, class-wise assigned, and uniformly resized are eliminated, while unclear images are removed[17].

The flow chart which shows the steps of data pre-processing are given below:



**Fig. 4.2.** (a) Steps of pre-processing of Kaggle dataset (b) Steps of pre-processing of Collected dataset



**Format conversion:** The X-ray that were collected from different laboratories were in different formats such as dcim.,HEIC. etc. The x-ray images were converted to png. So that all x-ray remains in the same format

**Data re-sizing :** The images were resized to 244\*244 pixels.

**Data cleaning:** In order to prevent inaccurate or inconsistent data from impairing the performance of the machine learning model, data cleaning aims to guarantee that the data is accurate, consistent, and error-free. To prevent such an effect, the photos were cleaned.

**Data augmentation :**Data augmentation is a machine learning technique that involves training models on multiple slightly altered copies of existing data in order to reduce overfitting during model training. The augmentation process involved the following actions:

- a) **Horizontal flip=True:** Images are randomly rotated horizontally. This is helpful for tasks when an object's orientation in a picture has no bearing on how it is classified.
- b) **Rotation range=20:**Images within the given range (in degrees) are rotated at random. This aids in the model's invariance to specific rotational degrees.
- c) **Width shift range=0.2:** Moves the pictures horizontally by an arbitrary percentage of their entire width. The model is able to generalise to diverse object positions with more ease because of this.
- d) **Height shift range=0.2:** Positions the photos up or down by an arbitrary percentage of their overall height. This, like width shift, aids the model's generalizability to various object locations in the photos.
- e) **Shear range=0.2:** Applies shear transformations to the photos. This introduces deformation to the forms of objects in the photos, hence adding further diversity to the training data.

- f) **Zoom range=0.2:** Applies random zooming effects to the photos, either by zooming in or out. This is an alternative method to introduce diversity in the size of items.

**Gray Scaling:** A mathematical technique is utilized to combine the red, green, and blue (RGB) values of each pixel in the color image, resulting in a single grayscale value. The initial RGB values of every pixel are substituted with the new grayscale value.

**Shuffling:** Randomizing the dataset enables the network to acquire knowledge from a more inclusive and accurate data subset in every batch. As a result, there is an improvement in overall generalization and performance. It is generally advisable to randomize the dataset prior to training[18].

#### 4.4 Proposed model Architecture

This section focuses on the architecture of the proposed customized CNN model. The utilized architecture is a customized Convolutional Neural Network (CNN) that has demonstrated its viability for image classification tasks, rendering it optimal for the analysis of medical pictures such as X-rays[19].

In this study, a CNN model was employed in a sequential manner. This technique is employed to study several forms of neural networks, in which the model receives a single input as feedback and anticipates a desired result. The Convolutional Neural Network (CNN) model is structured in order, employing the PyTorch framework. The data is transmitted and progresses in a sequential manner, following a top-to-bottom strategy until it reaches the end of the model.

**Convolutional Layers (conv1 to conv5):** Here we have convolutional layers (conv1 to conv5) with increasing numbers of output channels (feature maps). The prevalent form of convolution employed is the 2D convolution layer, commonly ab-

breviated as conv2D. In a conv2D layer, a filter or kernel moves across the 2D input data, conducting an element wise multiplication.

Consequently, it will aggregate the results into a solitary output pixel. The kernel will execute a uniform operation for each position it traverses, converting a 2D array of characteristics into a distinct 2D array of characteristics. Here we have convolutional layers (conv1 to conv5) with increasing numbers of output channels (feature maps).

**Batch normalization:** In the proposed model, each convolutional layer is followed by batch normalization (batch\_norm1 to batch\_norm5), Batch normalization is a technique used to normalize the activations between the layers of a neural network, rather than normalizing the raw input data. The process is conducted using mini-batches rather than the entire data set. Its purpose is to accelerate the training process and enable the utilization of greater learning rates, hence facilitating the learning process. Batch normalization adjusts the values of all pixels in the feature maps of a convolution layer to a different mean and standard deviation[20]. The process typically begins by performing z-score normalization on all pixels. Subsequently, the normalized values are multiplied by an arbitrary parameter alpha (scale) and then added to another arbitrary parameter beta (offset).

**MaxPooling2D((2,2)):** Max-pooling layers (maxpool1 to maxpool5) reduce the spatial dimensions of the feature maps, focusing on the most relevant information. Max pooling is a prominent downsampling method employed in convolutional neural networks (CNNs) to decrease the spatial dimensions of an input volume. It is a technique of non-linear down-sampling that aims to decrease the size and complexity of the representation, as well as to minimise the number of parameters and computations in the network. Max pooling performs spatial resizing on each individual depth slice of the input, operating independently[21]. The fundamental purpose of max pooling is to decrease the quantity of information in an image while preserving the crucial elements required for precise image identification. This method enhances the

ability to recognise characteristics in input data regardless of changes in scale and orientation, and it also helps to prevent overfitting.

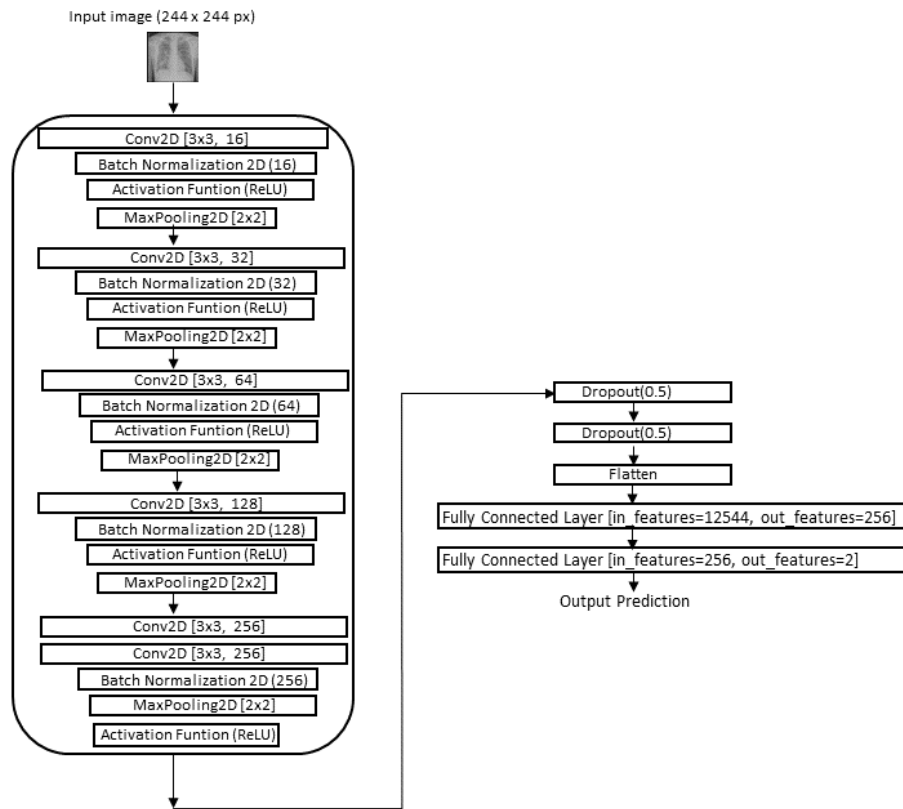
**Dropout(0.2):**Two dropout layers (dropout1 and dropout2 are included with a dropout rate of 0.5. Dropout is a regularisation technique to prevent overfitting. The Dropout layer performs the role of a mask, removing the contributions of certain neurons to the layer that comes after it while preserving the functionality of all of the other neurons. When we apply a Dropout layer to the input vector, we remove some of the features that it contains. On the other hand, when we apply it to a hidden layer, we remove some of the neurons that are within the hidden layer. When it comes to the training of CNNs, dropout layers are extremely important since they prevent the training data from being oversized. If they are not present, the first group of training samples has an excessively enormous influence on the learning process. As a consequence of this, characteristics that are only observed in subsequent samples or batches would not be acquired. Through the process of randomly deactivating a portion of the input units during each training update, the dropout layer is able to perform its role. This suggests that certain neurons in the network are "dropped out" or momentarily ignored during the process of forward propagation, along with the connections that are connected with them, based on a particular likelihood.

**The flatten layer Flatten():**Immediately following the convolutional layers, it is utilized to transform the output tensor from the convolutional layers into a tensor that is only one dimension in size. As part of its job, the flatten layer is in charge of transforming the feature map that it acquired from the max-pooling layer into a format that the dense layers are able to understand.

A feature map is essentially a multi-dimensional array that stores pixel values; the dense layers require a one-dimensional array as input for processing. Feature maps are used to represent an image. For the purpose of representing high-dimensional data, feature maps are utilized. This function is provided by the flatten layer, which

is responsible for transforming the feature maps into a one-dimensional array that may be utilized by the dense layers.

**Fully Connected Layers (fc1 and fc2):** The flattened tensor is then passed through fully connected layers (fc1 and fc2), which further process and combine the learned features for the final classification. A dense layer is a fully connected layer that is utilised in the last stages of the neural network structure in order to alter the dimensionality of the output from the layer that came before it. The relationship between the data values that the model is working with can be better defined with the assistance of dense layers.



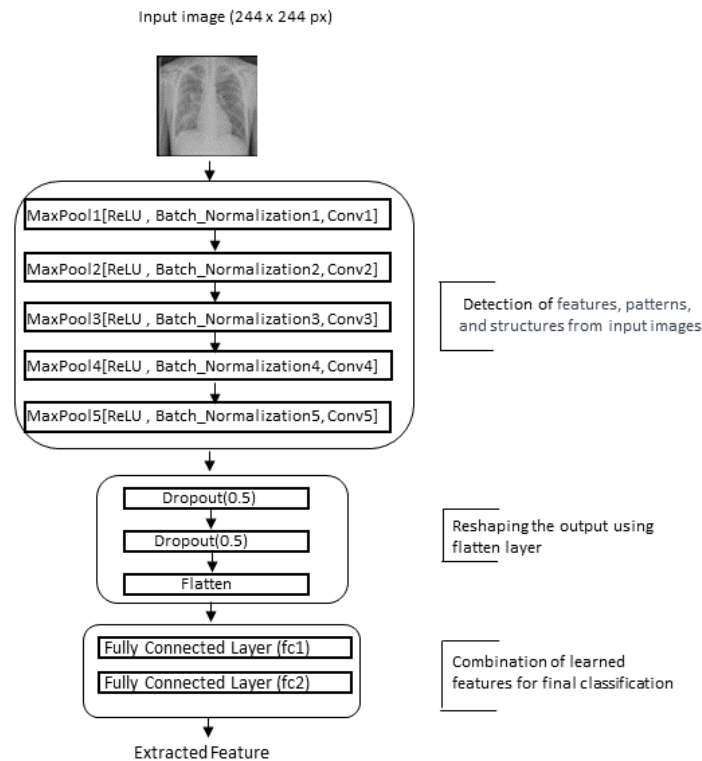
**Fig. 4.3.** Architecture of the Proposed Model

Each and every neuron in the thick layer is coupled to each and every neuron in the layer that comes before it. The neurons that are located in the dense layers of a model are given a result by each neuron that is located in the layer that comes before

them. These neurons of the dense layer are responsible for performing matrix-vector multiplication at that location. Therefore, the dense layer is responsible for doing a matrix-vector multiplication in the background. It is a process in which the row vector of the output from the layers that came before it is equal to the column vector of the dense layer[22].

#### 4.4.1 Feature Extraction

The process of feature extraction is carried out through numerous Convolutional layers, which can extract features from the arrays of input images. CNN has demonstrated superiority over other unsupervised feature extractors in terms of both greater model output rates and the generation of effective features. The selection of the number of convolutional layers and parameters was based on prioritizing the highest classification accuracy and the least calculation time.



**Fig. 4.4.** Diagram of Feature extraction process

Convolutional layers employ filters to extract features from the input images. The output of neurons related to specific regions in the input undergoes a straightforward computational procedure, which involves calculating the dot product between the output of the neurons connected to the local region of the input and their respective weights.

The feature extractor utilizes the Rectified Linear Unit (ReLU) activation filter. The activation function is utilized since all of our pixel values fall within the range of 0 to 1. It produces a linear output for values larger than zero and a zero output for values less than or equal to zero.

In the case of the ‘CustomCNN’ model, feature extraction occurs through a series of convolutional blocks. The feature extraction process is described below:

The process starts with the input layer, where the model receives X-ray images. Each image has two channels corresponding to the black-and-white color information.

The convolutional layers have filters with the following numbers: 16, 32, 64, 128, and 256. . The initial convolutional layer, referred to as ‘conv1, employs a collection of filters, also called kernels, to process the input image. These filters acquire the ability to identify fundamental patterns such as boundaries, textures, or uncomplicated forms[23].

Batch Normalization, implemented as ‘batch\_norm1’ is a technique used to normalize the output of the convolutional layer. This process improves the stability and speed of training. The ‘relu1’ activation, also known as Rectified Linear Unit (ReLU), introduces non-linearity to the model. It applies the ReLU activation function, enabling the network to learn intricate correlations. Max-Pooling, specifically referred to as ‘maxpool1’, decreases the size of the feature maps by selecting the highest-valued elements, retaining the most significant information while also minimizing the computing burden.

The Convolutional Block 2, consisting of the ‘conv2’, ‘batch\_norm2’, ‘relu2’, and

‘maxpool2’ layers, serves the purpose of enhancing the features acquired from the preceding layer. The convolutional filters in the ‘conv2 function now have an enhanced ability to detect intricate patterns and interconnections. In Convolutional Blocks 3 to 5 (‘conv3’ to ‘conv5’), the model adds more convolutional blocks, each consisting of convolutional, batch normalization, ReLU activation, and max-pooling layers. Through each block, the network acquires more complex and organized characteristics. Dropout layers, named ‘dropout1’ and ‘dropout2’, are used in the model. Dropout layers are implemented following the convolutional blocks to enforce regularization. During the training process, a technique called dropout is used to randomly deactivate a portion of the neurons. This helps to prevent overfitting and improves the ability of the model to generalize well to new, unseen data.

Using the ‘nn. Flatten()’ function, the feature maps are transformed into a one-dimensional vector following the completion of the last convolutional block. By performing this action, the data is prepared for processing by layers that are fully connected. Fully Connected levels (fc1 and fc2) the data that has been flattened is then transmitted across two levels that are fully connected. In order to arrive at the final categorization, these layers further combine and process the features that have been learned.

A total of 256 output features from the first fully connected layer (fc1) contribute to the feature combination. Two output features from the last fully connected layer (fc2) are appropriate for binary classification (tuberculosis or non-tuberculosis).

## 4.5 Proposed Model Implementation

The model is implemented using the two datasets for prediction. Two different subsets of the database were used for training and validating the model’s performance. The detailed training criteria of the model is shown in the Table 4.3



**Table 4.3:** Epoch and Time Criteria of Training Model

Parameter	Value
Epoch Stopping criteria	7 epochs
Training time per epoch	32 s
Training Time per iteration	1.02s

The optimized value of the proposed model helps to improve the model's performance. The values are given in the Table 4.4

**Table 4.4:** Optimized value of model parameters

Parameters	Value
Optimizer	Adam
Loss function	Cross Entropy loss
Learning Rate	.001
Metrics	Loss, accuracy
Batch size	32
Epochs	20

## CHAPTER 5

### RESULT AND DISCUSSION

#### 5.1 Overview

This part mostly concerns the evaluation of the experimental outcomes acquired by the proposed approach. This chapter also examines output comparisons between different applied algorithms and other existing models.

#### 5.2 Evaluation Method

The confusion matrix is a specific tabular arrangement often used to visually represent the efficacy of a classification model. The instances of an actual class are represented by each row, while the instances of a predicted class are represented by each column. The Confusion Matrix is a machine learning technique that facilitates the computation of various essential evaluation metrics, such as Accuracy, Precision, Recall, and F1 score. In order to calculate the parameters, it is necessary to have a comprehensive understanding of the fundamental terms associated to the confusion matrix[24].

**True Positive (TP):** Instances where the model correctly predicts the positive class (e.g., tuberculosis) and it is true.

**True Negative (TN):** Instances where the model correctly predicts the negative

class (e.g., non-tuberculosis) and it is false.

**False Positive (FP):** Instances where the model incorrectly predicts the positive class, but it is false (Type I error).

**False Negative (FN):** Instances where the model incorrectly predicts the negative class, but it is true (Type II error).

**Accuracy:** The proportion of correct predictions made by the model over all predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.1)$$

**Precision:** The proportion of true positive predictions out of all positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5.2)$$

**Recall:** The proportion of true positive predictions out of all actual positive instances.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5.3)$$

**F1-score:** The harmonic mean of precision and recall. It provides a balance between precision and recall.

$$\text{F1 - score} = \frac{2TP}{(2TP + FP + FN)} \quad (5.4)$$

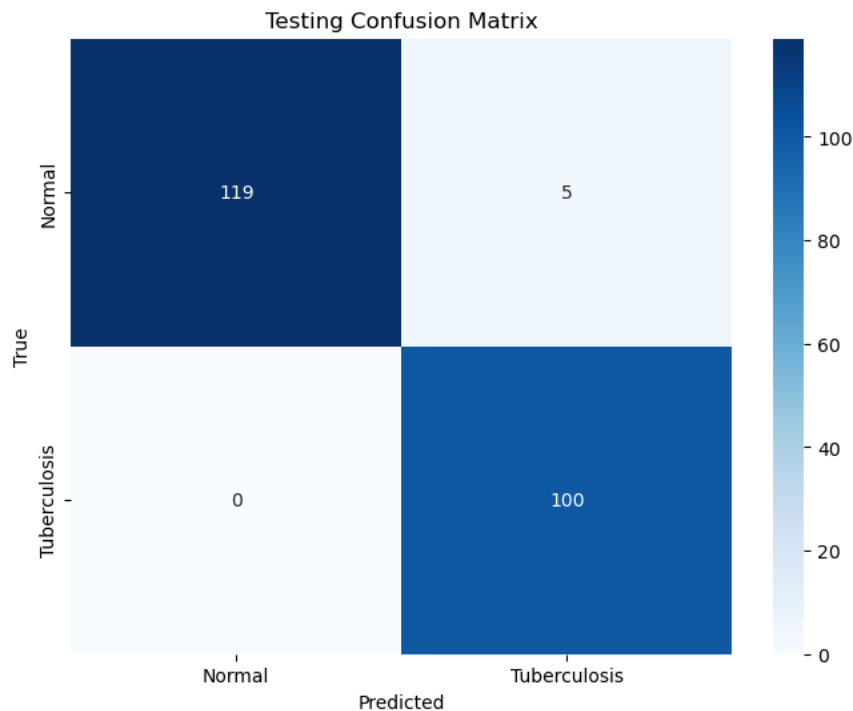
### 5.3 Analysis of Experimental Model Performance

The evaluation of the proposed method comprised the calculation of the accuracy, precision, recall, and F1-score to assess the performance of the model[25]. Due to the imbalanced class distribution in the datasets, it may cause data skewness. As a result,

relying solely on accuracy does not adequately reflect the recognition rate. Therefore, we employed the F1-score, which combines precision and recall in a weighted average, to provide a more precise outcome[26]. We used a confusion matrix to calculate these values. A confusion matrix is a structured table format that enables the visualization of an algorithm's performance, often in the context of supervised learning. In the case of unsupervised learning, it is sometimes referred to as a matching matrix. Each row of the matrix corresponds to the examples in a real class, whereas each column corresponds to the instances in a predicted class.

### 5.3.1 Model Evaluation on Test Data

We have trained and tested the model using two different datasets. The data set that was used for testing was collected from our country. After testing the model with this dataset. We obtained the following confusion matrix:



**Fig. 5.1.** Confusion Matrix

**True Positives (TP): 100** These are instances where the model correctly identi-

fied chest X-rays with tuberculosis. In a medical context, each true positive represents a successful detection of an actual case of tuberculosis.

**True Negatives (TN): 119** These are instances where the model correctly identified chest X-rays without tuberculosis. True negatives represent cases where the model correctly ruled out the presence of the condition.

**False Positives (FP): 5** These are instances where the model incorrectly predicted tuberculosis when it wasn't present. Each false positive represents a case where the model raised a false alarm, potentially causing unnecessary concern for patients.

**False Negatives (FN): 0** There are no instances where the model failed to detect tuberculosis when it was present. A false negative in a medical context could mean a missed opportunity for early intervention.

Using the confusion matrix, we obtained the accuracy, precision, recall, and f1 score. The values are given in the table 5.1:

**Table 5.1:** Experimental Model Performance details

Parameters	value
Accuracy	97.77%
Precision	95.24%
Recall	100%
F1-Score	97.56%

Accuracy is the proportion of correctly classified instances among the total instances. In your case, the model achieved an accuracy of 97.77%, indicating that it correctly predicted the class labels for nearly 98% of the instances in our testing dataset.

Precision is the ratio of true positive predictions to the total predicted positives. In our model, 95.24% of the instances predicted as positive were indeed positive. This indicates a low rate of false positives, which is crucial in medical applications to avoid unnecessary treatments or interventions for non-diseased individuals.

Recall is the ratio of true positive predictions to all instances that are actually

positive. A recall of 100% means that our model successfully identified every instance of tuberculosis in your testing dataset. There were no cases of tuberculosis that the model failed to detect (false negatives). This is a critical metric in medical diagnostics.

The F1-Score is the harmonic mean of precision and recall. It provides a balanced measure, especially when there is an imbalance between the classes. Your F1-Score of 97.56% indicates a strong balance between precision and recall, which is crucial for a model in medical diagnostics.

### 5.3.2 Detailed classification report

The classification report provides a comprehensive analysis of the performance of our model on each class, including the trade-off between precision and recall[27]. Additionally, it provides information on the frequency (support) of each class, which can serve as an indicator of class imbalance or the dataset's size. The classification report is given in the Table 5.2:

**Table 5.2:** Classification report of the proposed model

	Precision	Recall	F1-Score	Support
0	1.00	.96	.98	124
1	0.95	1.00	.98	100
Accuracy	-	-	0.98	224
Macro avg	0.98	0.98	0.98	224
Weighted avg	0.98	0.98	0.98	224

#### **Precision:**

**Precision for class 0:** All instances predicted as class 0 were indeed class 0, showcasing the model's precision in identifying this class.

**Precision for class 1:** 95% of instances predicted as class 1 were correctly classified, indicating a high precision for this class.

#### **Recall (Sensitivity):**

**Recall for class 0:** The model successfully captured 96% of all instances belong-

ing to class 0.

**Recall for class 1:** For class 1, the model identified all instances, with no instances of this class being missed (no false negatives).

**F1-Score:**

**F1-Score for class 0:** The harmonic mean of precision and recall for class 0 is 0.98, indicating a robust balance between precision and recall for this class.

**F1-Score for class 1:** Similarly, class 1 achieved a harmonic mean of 0.98, demonstrating a balanced performance in capturing positive instances while avoiding false positives.

**Support:** Support represents the number of actual occurrences of each class in the dataset.

**Support for class 0:** There are 124 instances of class 0 in the dataset.

**Support for class 1:** There are 100 instances of class 1 in the dataset.

### 5.3.3 Comparison with Existing Work

The suggested model demonstrates superior performance in tuberculosis prediction, surpassing the majority of current models, with a testing accuracy of 98%[28].

The table below presents a selection of existing works for the purpose of comparing them with our own work. The table 5.3 below displays various work models' datasets and associated details.

**Table 5.3:** Comparison of Proposed Model with Existing Works

Author	Year	No. of TB images	Method	Database	Evaluation Matrix
Hrudya et al.[29]	2015	138	Support Vector Machine	MC	Not stated
Jaime al.[30]	2016	392	Statistical Analysis	392 records collected from Cape Town in South Africa	AUC-0.84, Specificity – 49%, and Sensitivity 95%
Rahul et al. [31]	2017	805	CNN	MC and CHN	Accuracy 82.09%
Amuj al.[32]	2017	805	CNN Transfer Learning	MC and CHN	Accuracy>80%
Lopes al.[33]	2017	805	CNN Transfer Learning	MC and CHN	Accuracy 84.7% and AUC-.926
Abbas al.[34]	2018	138	Knowledge transferred via Alexnet	MC	AUC- 99.8
Lucas al.[35]	2018	1052	CNN and two ensembles	JSRT, MC and CHN	Accuracy 88.76%
Ojasvi et al.[36]	2018	805	Transfer learning (ResNet)	MC and CHN	Accuracy 94.89%
Niharika et al.[37]	2019	805	Support Vector Machine	MC and CHN	AUC-0.96 and specificity - 100%
Pasa et al. [38]	2019	1111	Optimized CNN	MC and CHN and Belarus Dataset	AUC-0.811 for MC, 0.9 for CHN nad 0.925 for CHN and 0.925 for combined 100%
Syeda al.[39]	-	805	Transfer Learning	MC and CHN	AUC- 0.85
Mostafa al.[40]	2019	805	Transfer Learning (VGG 16 model)	MC and CHN	Accuracy 80% and 81.25% without and with augmentation
Quang et al.[41]	2019	805	Tuning of DenseNet model	MC and CHN	AUC- 0.94 for CHN and 0.82 for MC
Alfonso et al.[42]	2019	805	3 pre-trained CNNs	MC and CHN	Accuracy 86%
<b>This paper</b>	<b>2023</b>	<b>1559</b>	<b>CNN</b>	<b>Kaggle and Collected</b>	<b>Accuracy 98%</b>

In 2015, Hrudya and others performed work on the Montgomery (MC) database. Hrudya and others constructed this model using the Support Vector Machine (SVM) technique. They utilized only 138 tuberculosis images. Jamie and others developed



a model utilizing statistical analysis in 2016. He gathered a total of 392 tuberculosis images from a database located in Cape Town, South Africa. He attained an AUC of 0.84, a sensitivity of 95%, and a specificity of 49% after implementing this model. In 2017, 805 images of tuberculosis were gathered by Rahul and others from the Montgomery and Shenzhen databases. They utilized a convolutional neural network technique and achieved an accuracy of 82.09%. Amuj, Lopes, and others obtained an accuracy of 84.7% and slightly more than 80%, respectively, when employing the CNN transfer learning technique on the identical set of images from the same database. Similarly, Lopes and others attained an AUC of 0.926 percent. Abbas and others utilized the identical number of datasets and databases as Hrudya, albeit employing the knowledge conveyed via the Alexnet method, which resulted in an AUC of 99.8. Lucas and others utilized 1052 tuberculosis images from the JSRT, MC, and CHN databases in 2018. Utilizing CNN and two additional ensembles, they attained an accuracy of 88.76 percent. Ojasvi and others gathered 805 tuberculosis images from the MC and CHN databases in 2018. Ojasvi and others employed the ResNet transfer learning technique to achieve an accuracy rate of 94.89%. A support vector machine (SVM) was employed by Niharika et al. in 2019 to analyze the identical quantity of tuberculosis images from the same database. An AUC of 0.96 and a specificity of 100% were the results he obtained. Pasa et al. in 2019 utilized 1111 tuberculosis images from the MC, CHN, and Belarus datasets. By employing optimized CNN, they were able to attain AUC-0.811 for MC, 0.9 for CHN, and 0.925 for CHN, for a total of 100%. Syeda, Mostafa, Quang, Alfonso, and others utilized 805 tuberculosis images in 2019. Syeda, Mostafa, Quang, Alfonso, and others obtained these images from the MC and CHN datasets. Mostafa, Syed, and others all employed transfer learning as a technique. In contrast, Quang and Alfonso utilized DenseNet model tuning and three pre-trained CNNs, respectively. This work incorporated a total of 1559 tuberculosis images obtained from Bangladeshi and Kaggle databases into our

paper. The model was constructed using a convolutional neural network and achieved an accuracy of 98%.

The model consistently ranks higher than all other models due to its exceptional accuracy. In terms of performance, studies employing conventional approaches such as Support Vector Machines (SVM) or simplified CNN models (e.g., Rahul et al., 2017; Hrudya et al., 2015) average in nature. While their accuracy may be satisfactory, it falls short of that of advanced deep learning models. Research studies that utilize fundamental statistical analysis or have smaller datasets (e.g., Mostafa et al. in 2019; Jaime et al. in 2016) generally receive lower rankings as a result of their comparatively lower accuracy, area under the curve (AUC), or restricted data usage.

# CHAPTER 6

## CONCLUSIONS

### 6.1 Overview

This chapter provides the concluding observations. It also discusses suggestions for additional research.

### 6.2 Conclusions

The model was implemented by utilising a combination of data that was found to be publicly available as well as data that was gathered from a variety of institutions located in Bangladesh. The methodology of enhancing data in order to get more precise results. An new communication network architecture is now being developed in order to facilitate the identification of tuberculosis through the use of chest X-rays. It is possible that the automated computer-aided diagnosis (CADx) system for tuberculosis identification utilising computed tomography (CXR) pictures could provide assistance to medical professionals and radiologists in the interpretation of images, hence facilitating more accurate diagnoses. The specificity of early tuberculosis diagnosis could potentially be improved with the help of this method. By acting as a second opinion, this automated tuberculosis detection system may also be of use to medical professionals and radiologists in obtaining a definitive diagnosis of the

disease.

### 6.3 Major Findings and Future Scopes

We have developed an advanced model for the detection of tuberculosis using convolutional neural network. The majority of the authors used Keras and TensorFlow for this specific task. Nevertheless, we utilized PyTorch, a widely used Python framework. PyTorch provides researchers with a competitive advantage by offering greater flexibility through its utilization of dynamic computation graphs. This allows researchers to make modifications on the fly and easily manage the method. It is typically more efficient and offers enhanced debugging capabilities. Furthermore, it is noteworthy that no other author has ever previously worked with the dataset from Bangladesh. Regardless of this variety, our model exhibited exceptional performance. This model is highly suitable for detecting tuberculosis and may also be utilized with great efficiency across several other medical fields. We will be analyzing numerous diseases and making minor adjustments to this remarkable model. This approach will enable us to handle several classes, process larger datasets, and compare accuracies across multiple diseases. In the future, we will utilize segmentation in order to enhance accuracy.

# REFERENCES

- [1] A. Kanchar and S. Swaminathan, “Tuberculosis control: Who perspective and guidelines,” *The Indian Journal of Pediatrics*, vol. 86, pp. 703–706, 2019.
- [2] C. Liu, Y. Cao, M. Alcantara, B. Liu, M. Brunette, J. Peinado, and W. Curioso, “Tx-cnn: Detecting tuberculosis in chest x-ray images using convolutional neural network,” in *2017 IEEE international conference on image processing (ICIP)*, pp. 2314–2318, IEEE, 2017.
- [3] F. Pasa, V. Golkov, F. Pfeiffer, D. Cremers, and D. Pfeiffer, “Efficient deep network architectures for fast chest x-ray tuberculosis screening and visualization,” *Scientific reports*, vol. 9, no. 1, p. 6268, 2019.
- [4] H. Greenspan, B. Van Ginneken, and R. M. Summers, “Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique,” *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1153–1159, 2016.
- [5] R. Hooda, S. Sofat, S. Kaur, A. Mittal, and F. Meriaudeau, “Deep-learning: A potential method for tuberculosis detection using chest radiography,” in *2017 IEEE international conference on signal and image processing applications (ICSIPA)*, pp. 497–502, IEEE, 2017.
- [6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected

- convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.
- [7] U. Lopes and J. F. Valiati, “Pre-trained convolutional neural networks as feature extractors for tuberculosis detection,” *Computers in biology and medicine*, vol. 89, pp. 135–143, 2017.
- [8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [9] S. Jaeger, S. Candemir, S. Antani, Y.-X. J. Wáng, P.-X. Lu, and G. Thoma, “Two public chest x-ray datasets for computer-aided screening of pulmonary diseases,” *Quantitative imaging in medicine and surgery*, vol. 4, no. 6, p. 475, 2014.
- [10] P. Chhikara, P. Singh, P. Gupta, and T. Bhatia, “Deep convolutional neural network with transfer learning for detecting pneumonia on chest x-rays,” in *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals: Proceedings of GUCON 2019*, pp. 155–168, Springer, 2020.
- [11] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, “Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning,” *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.
- [12] R.-I. Chang, Y.-H. Chiu, and J.-W. Lin, “Two-stage classification of tuberculosis culture diagnosis using convolutional neural network with transfer learning,” *The Journal of Supercomputing*, vol. 76, pp. 8641–8656, 2020.
- [13] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, “Detrac: Transfer learning of class decomposed medical images in convolutional neural networks,” *IEEE Access*, vol. 8, pp. 74901–74913, 2020.

- [14] M. Ahsan, R. Gomes, and A. Denton, "Application of a convolutional neural network using transfer learning for tuberculosis detection," in *2019 IEEE International Conference on Electro Information Technology (EIT)*, pp. 427–433, IEEE, 2019.
- [15] S. Christodoulidis, M. Anthimopoulos, L. Ebner, A. Christe, and S. Mougiakakou, "Multisource transfer learning with convolutional neural networks for lung pattern analysis," *IEEE journal of biomedical and health informatics*, vol. 21, no. 1, pp. 76–84, 2016.
- [16] Q. H. Nguyen, B. P. Nguyen, S. D. Dao, B. Unnikrishnan, R. Dhingra, S. R. Ravichandran, S. Satpathy, P. N. Raja, and M. C. Chua, "Deep learning models for tuberculosis detection from chest x-ray images," in *2019 26th international conference on telecommunications (ICT)*, pp. 381–385, IEEE, 2019.
- [17] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.
- [18] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE international symposium on circuits and systems*, pp. 253–256, IEEE, 2010.
- [19] B. P. Nguyen, W.-L. Tay, and C.-K. Chui, "Robust biometric recognition from palm depth images for gloved hands," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 6, pp. 799–804, 2015.
- [20] S. Jaeger, A. Karargyris, S. Candemir, L. Folio, J. Siegelman, F. Callaghan, Z. Xue, K. Palaniappan, R. K. Singh, S. Antani, *et al.*, "Automatic tuberculo-

- sis screening using chest radiographs,” *IEEE transactions on medical imaging*, vol. 33, no. 2, pp. 233–245, 2013.
- [21] V. S. Kumar and P. S. Kumar, “Lung nodules detection by computer aided diagnosis (cad) using image processing,” *International Journal of Advance Research in Computer Science and Management Studies*, 2014.
- [22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.
- [23] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, “Score-cam: Score-weighted visual explanations for convolutional neural networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 24–25, 2020.
- [24] A. Hernández, Á. Panizo, and D. Camacho, “An ensemble algorithm based on deep learning for tuberculosis classification,” in *Intelligent Data Engineering and Automated Learning–IDEAL 2019: 20th International Conference, Manchester, UK, November 14–16, 2019, Proceedings, Part I 20*, pp. 145–154, Springer, 2019.
- [25] J. T. Townsend, “Theoretical analysis of an alphabetic confusion matrix,” *Perception & Psychophysics*, vol. 9, pp. 40–50, 1971.
- [26] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18*, pp. 234–241, Springer, 2015.
- [27] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, “Thirty years



- of machine learning: The road to pareto-optimal wireless networks,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1472–1514, 2020.
- [28] T. Rahman, A. Khandakar, M. A. Kadir, K. R. Islam, K. F. Islam, R. Mazhar, T. Hamid, M. T. Islam, S. Kashem, Z. B. Mahbub, *et al.*, “Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization,” *IEEE Access*, vol. 8, pp. 191586–191601, 2020.
- [29] H. Das and A. Nath, “An efficient detection of tuberculosis from chest x-rays,” *International Journal of Advance Research in Computer Science and Management Studies*, vol. 3, no. 5, pp. 149–154, 2015.
- [30] J. Melendez, C. I. Sánchez, R. H. Philipsen, P. Maduskar, R. Dawson, G. Theron, K. Dheda, and B. Van Ginneken, “An automated tuberculosis screening strategy combining x-ray-based computer-aided detection and clinical information,” *Scientific reports*, vol. 6, no. 1, p. 25265, 2016.
- [31] R. Hooda, S. Sofat, S. Kaur, A. Mittal, and F. Meriaudeau, “Deep-learning: A potential method for tuberculosis detection using chest radiography,” in *2017 IEEE international conference on signal and image processing applications (ICSIPA)*, pp. 497–502, IEEE, 2017.
- [32] A. Rohilla, R. Hooda, and A. Mittal, “Tb detection in chest radiograph using deep learning architecture,” *ICETETSM-17*, pp. 136–147, 2017.
- [33] U. Lopes and J. F. Valiati, “Pre-trained convolutional neural networks as feature extractors for tuberculosis detection,” *Computers in biology and medicine*, vol. 89, pp. 135–143, 2017.
- [34] A. Abbas and M. M. Abdelsamea, “Learning transformations for automated classification of manifestation of tuberculosis using convolutional neural network,”

- in *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, pp. 122–126, IEEE, 2018.
- [35] L. G. C. Evalgelista and E. B. Guedes, “Computer-aided tuberculosis detection from chest x-ray images with convolutional neural networks,” in *Anais do XV Encontro Nacional de Inteligência Artificial e Computacional*, pp. 518–527, SBC, 2018.
- [36] O. Yadav, K. Passi, and C. K. Jain, “Using deep learning to classify x-ray images of potential tuberculosis patients,” in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 2368–2375, IEEE, 2018.
- [37] N. Singh and S. Hamde, “Tuberculosis detection using shape and texture features of chest x-rays,” in *Innovations in Electronics and Communication Engineering: Proceedings of the 7th ICIECE 2018*, pp. 43–50, Springer, 2019.
- [38] F. Pasa, V. Golkov, F. Pfeiffer, D. Cremers, and D. Pfeiffer, “Efficient deep network architectures for fast chest x-ray tuberculosis screening and visualization,” *Scientific reports*, vol. 9, no. 1, p. 6268, 2019.
- [39] S. S. Meraj, R. Yaakob, A. Azman, S. Rum, A. Shahrel, A. Nazri, and N. F. Zakaria, “Detection of pulmonary tuberculosis manifestation in chest x-rays using different convolutional neural network (cnn) models,” *Int. J. Eng. Adv. Technol. (IJEAT)*, vol. 9, no. 1, pp. 2270–2275, 2019.
- [40] M. Ahsan, R. Gomes, and A. Denton, “Application of a convolutional neural network using transfer learning for tuberculosis detection,” in *2019 IEEE International Conference on Electro Information Technology (EIT)*, pp. 427–433, IEEE, 2019.
- [41] Q. H. Nguyen, B. P. Nguyen, S. D. Dao, B. Unnikrishnan, R. Dhingra, S. R. Ravichandran, S. Satpathy, P. N. Raja, and M. C. Chua, “Deep learning models

for tuberculosis detection from chest x-ray images,” in *2019 26th international conference on telecommunications (ICT)*, pp. 381–385, IEEE, 2019.

- [42] A. Hernández, Á. Panizo, and D. Camacho, “An ensemble algorithm based on deep learning for tuberculosis classification,” in *Intelligent Data Engineering and Automated Learning-IDEAL 2019: 20th International Conference, Manchester, UK, November 14–16, 2019, Proceedings, Part I 20*, pp. 145–154, Springer, 2019.