DEVELOPMENT OF A CLINICAL DIAGNOSIS AND DECISION SUPPORT SYSTEM FOR CHEST RADIOGRAPHY USING CNN

BIPIN THAPA MAGAR

M.Sc. ENGINEERING THESIS

DEPARTMENT OF BIOMEDICAL ENGINEERING

MILITARY INSTITUTE OF SCIENCE AND TECHNOLOGY

DHAKA, BANGLADESH

MARCH 2023

DEVELOPMENT OF A CLINICAL DIAGNOSIS AND DECISION SUPPORT SYSTEM FOR CHEST RADIOGRAPHY USING CNN

BIPIN THAPA MAGAR (SN. 0421260012)

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Biomedical Engineering

DEPARTMENT OF BIOMEDICAL ENGINEERING
MILITARY INSTITUTE OF SCIENCE AND TECHNOLOGY
DHAKA, BANGLADESH

MARCH 2023

DEVELOPMENT OF A CLINICAL DIAGNOSIS AND DECISION SUPPORT SYSTEM FOR CHEST RADIOGRAPHY USING CNN

M. Sc. Engineering Thesis

By

BIPIN THAPA MAGAR
(SN: 0421260012)

Approved as to style and content by the Board of Examination on 27 March 2023

1. Dr. Md. Asadur Rahman
Assistant Professor
Department of Biomedical Engineering
Military Institute of Science and Technology (MIST)

Supervisor
(Chairman)

2. Col Syed Mahfuzur Rahman
Senior Instructor and the Head of the Department
Department of Biomedical Engineering
Military Institute of Science and Technology (MIST)

Member
(Ex-Officio)

3. Lt Col Hussain Md. Abu Nyeem, PhD
Instructor Class A
Department of EECE
Military Institute of Science and Technology (MIST)

Member
(Internal)

4. Dr Mohiuddin Ahmad
Professor
Department of Electrical and Electronic Engineering
Khulna University of Engineering & Technology (KUET)

Ahmad.

Member
(External)

Department of Biomedical Engineering, MIST, Dhaka

DEVELOPMENT OF A CLINICAL DIAGNOSIS AND DECISION SUPPORT SYSTEM FOR CHEST RADIOGRAPHY USING CNN

DECLARATION

I hereby declare that the study reported in this thesis entitled above is my own original work and has not been submitted anywhere for any degree or other purposes. Further, I declare that the intellectual content and the results of this thesis are the outcomes of my work. Also, all the sources and assistance received in preparing this thesis work have been properly acknowledged and/or cited in the reference section.

Bipin Thapa Magar

Department of Biomedical Engineering, MIST, Dhaka

ABSTRACT

Development of a Clinical Diagnosis and Decision Support System for Chest Radiography Using CNN

Chest X-ray (CXR) image is a widely used diagnostic tool for various chest diseases. Human interpretation of CXR images, however, has never been always effective. The diagnostic level of the radiologists along with other factors like cognitive ability, experience, fatigue, and other human-dependent factors may impair the diagnostic procedure with missed information, misinterpretation, and requiring more time and cost. Computer-aided analysis of CXR images has already demonstrated its potential over manual or human screening to facilitate rapid, correct, and low-cost diagnosis of chest diseases. Existing computer-aided systems are still not suitable for real-time applications due to limited findings, limited generalizability across wide datasets, and not being computationally and economically affordable as well. Therefore, an efficient Convolution Neural Network (CNN) based computer-aided decision support system, the **CXRNet**, was developed for the automatic detection of abnormalities from CXR images in a real-time clinical scenario. The proposed CXRNet model is a 16-layered CNN architecture with 5 output classes: Cardiomegaly, COVID, Normal, Pneumonia, and Tuberculosis. This architecture is trained with frontal CXR images obtained from various sources to improve the generalization of the model across multiple datasets. Upon testing the model on three different data distribution conditions (70% training and 30% testing, 80% training and 20% testing, and 90% training and 10% testing), it achieved a state-of-the-art performance with an average accuracy of **95.7%**, a precision of **95.3%**, a recall of **95.3%**, and an f1-score of **95.3%** for the multiclass classification task. The proposed CXRNet also demonstrates excellent performance on binary classification tasks with an average accuracy of over **98%** for each disease condition. The results obtained from this work outperform several other custom-designed CNN architectures as well as pre-trained models-based architectures like ResNet, VGG, DenseNet, Xception, Inception, etc. Furthermore, with proper testing, validation, and debugging of the model in clinical practice, it can be successfully deployed as a decision support system for radiologists.

Keywords: *Chest X-ray image, Convolution Neural Network, Clinical Diagnosis, Decision Support System, Multiclass Classification, Binary Classification*

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor, **Md. Asadur Rahman,** Ph.D., Assistant Professor, Department of Biomedical Engineering, MIST, for his effective guidance and support throughout this research work whenever necessary.

My sincere gratitude goes to **Colonel Syed Mahfuzur Rahman**, the respected Head of the Department, for his persistent guidance and financial allocation that made this research work possible. I also extend my profound appreciation to the Department of Biomedical Engineering, MIST, for their co-operations and for facilitating me with the materials and laboratory resources to carry out my research work.

I am also indebted to all the individuals who directly or indirectly helped and supported me with their technical and editorial feedback in carrying out my research work. I want to acknowledge **Capt Kumar Shrestha,** my fellow companion, for his continuous support and cooperation in the journey of completing my M.Sc. Engineering together.

Finally, I would like to acknowledge the continuous love and support from my loving family and friends throughout the journey and am grateful to everyone who has made this M.Sc. thesis a reality.

Author

TABLE OF CONTENTS

ABSTRACT v
ACKNOWLEDGMENTS vi
TABLE OF CONTENTS vii
LIST OF FIGURES ix
LIST OF TABLES xi
LIST OF ABBREVIATIONS xii
CHAPTER 1: INTRODUCTION 1
1.1 Background 1
1.2 Problem Statement 2
1.3 Objectives 4
1.4 Contribution of this Proposed Research 4
1.5 Thesis Organization 5
CHAPTER 2: LITERATURE REVIEW 6
2.1 Introduction 6
2.2 Present State of DL Models 6
2.3 DL models for CXR Image Analysis 8
2.3.1 Pneumonia Detection 8
2.3.2 COVID-19 Detection 10
2.3.3 Tuberculosis Detection 13
2.3.4 Multiple Abnormalities Detection 14
2.4 Discussion 16
2.5 Chapter Summary 19
CHAPTER 3: A NEW CXRNet MODEL 20
3.1 Introduction 20
3.2 Dataset 20
3.3 Data Pre-processing and Augmentation 24
3.3.1 Rescaling 24
3.3.2 Resizing 25
3.3.3 Augmentation 25
3.4 CXRNet Architecture 27
3.4.1 Input Layer 31
3.4.2 Convolution Layer 31
3.4.3 Pooling Layer 33
3.4.4 Dropout Layer 34
3.4.5 Flatten Layer 34
3.4.6 Dense Layer 35
3.4.7 Activation Function 36
3.5 Model Training 38

3.5.1 Forward Propagation ... 39
3.5.2 Backpropagation ... 39
3.6 Adam Optimization Algorithm ... 40
3.7 Improving and Optimization of CNN Network ... 42
3.7.1 Bias/Variance Problem ... 42
3.7.2 Input Shape ... 44
3.7.3 Model Checkpoint ... 45
3.7.4 Early Stopping ... 45
3.7.5 Hyperparameters Tuning ... 45
CHAPTER 4: RESULTS AND DISCUSSION ... 48
4.1 Introduction ... 48
4.2 Results ... 48
4.2.1 Input Pre-processing and Augmentation ... 48
4.2.2 Model Training ... 50
4.3 Qualitative Evaluation ... 57
4.4 Comparison with Related Works ... 64
4.5 CXRNet as a Clinical Diagnosis and Decision Support System ... 68
4.6 Discussion ... 70
CHAPTER 5: CONCLUSION AND FUTURE WORKS ... 73
5.1 Outcomes ... 73
5.2 Limitations ... 74
5.3 Future Work ... 74
References ... 76
List of publications ... 83

LIST OF FIGURES

Figure 2.1: The evolution of DL in the field of medical imaging 6

Figure 3.1: The general flow diagram of the working process of this research work 22

Figure 3.2: Chest Radiographs samples; (a) Cardiomegaly, (b) COVID, (c) Pneumonia, (d) Normal, and (e) Tuberculosis 23

Figure 3.3: The original distribution of images in each category collected from various sources 28

Figure 3.4: The general block diagram of the proposed CXRNet model 29

Figure 3.5: The detailed architecture of the proposed CXRNet model 30

Figure 3.6: Max pooling operation illustration 33

Figure 3.7 The flattening operation of two-dimensional input vector 35

Figure 3.8: An example of the fully connected layer in a deep neural network 36

Figure 3.9: The line plot of ReLU for negative and positive inputs 37

Figure 3.10: The line plot of the sigmoid function for negative and positive inputs 37

Figure 3.11: Working procedure to address bias/variance problem while designing proposed CXRNet architecture 44

Figure 4.1: The samples of images before and after image pre-processing and augmentation tasks, (a) before, and (b) after 49

Figure 4.2: Training accuracy vs validation accuracy curve of proposed CXRNet model for **70% - 30%** data split condition 51

Figure 4.3: Training loss vs validation loss curve of proposed CXRNet model for **70% - 30%** data split condition 52

Figure 4.4: Training accuracy vs validation accuracy curve of proposed CXRNet model for **80% - 20%** data split condition 52

Figure 4.5 Training loss vs validation loss curve of proposed CXRNet model for **80% - 20%** data split condition 53

Figure 4.6 Training accuracy vs validation accuracy curve of proposed CXRNet model for **90% - 10%** data split condition 53

Figure 4.7 Training loss vs validation loss curve of proposed CXRNet model for **90% - 10%** data split condition 54

Figure 4.8: Graph of learning rate values during model training, (a) 70% - 30% split, (b) 80% - 20% split, and (c) 90% - 10% split 55

Figure 4.9: Model performance without data augmentation techniques, (a) accuracy graph, and (b) loss graph 56

Figure 4.10: Model performance without dropout layers, (a) accuracy graph, and (b) loss graph 56

Figure 4.11: Training performance of the model for different image shapes, (a) accuracy graph, and (b) loss graph 57

Figure 4.12: The resulting confusion matrix of the CXRNet on 70% - 30% split case 59

Figure 4.13: The resulting confusion matrix of the CXRNet on 80% - 20% split case 60

Figure 4.14: The resulting confusion matrix of the CXRNet on 90% - 10% split case 61

Figure 4.15: The resulting confusion matrix of the CXRNet for the binary classification task, (a) Cardiomegaly vs Normal, (b) COVID vs Normal, (c) Pneumonia vs Normal, and (d) Tuberculosis vs Normal 63

Figure 4.16: Predicted images from the proposed CXRNet model with its corresponding heat-map activated image from the last convolutional layer: (a) Cardiomegaly original image, (b) Heat-map activated Cardiomegaly image, (c) COVID original image, (d) Heat-map activated COVID image, (e) Normal original image, (f) Heat-map activated Normal image, (g) Pneumonia original image, (h) Heat-map activated Pneumonia image, (i) Tuberculosis original image, and (j) Heat-map activated Tuberculosis image 68

Figure 4.17: An illustration of the graphical abstract of this research work in clinical diagnosis and decision support system 71

LIST OF TABLES

Table 2.1: Performance comparison of different DL techniques for several abnormalities detection on CXR images 17

Table 3.1: The detailed composition of the dataset collected from various sources 21

Table 3.2: Data augmentation techniques 26

Table 3.3: Data distribution in each category after oversampling the minority classes using data augmentation strategies (Balanced dataset) 27

Table 3.4: Summary of the proposed CXRNet architecture with output shape and parameters for each layer 31

Table 3.5: The hyper-parameters of the proposed CXRNet model 38

Table 4.1: The distribution of the dataset for various training-testing splits 50

Table 4.2: The summary of the results from CXRNet training for different dataset splits 54

Table 4.3: Experimental performance metrics results of CXRNet on 70% - 30% split case 58

Table 4.4: Experimental performance metrics results of CXRNet on 80% - 20% split case 59

Table 4.5: Experimental performance metrics results of CXRNet on 90% - 10% split case 60

Table 4.6: The summary of evaluation metrics of the CXRNet model 61

Table 4.7: Experimental performance metrics results of CXRNet on the binary classification task 62

Table 4.8: Comparison of the proposed CXRNet model with other related research works 65

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANNs	Artificial Neural Networks
BCET	Balance Contrast Enhancement Technique
CAD	Computer Aided Diagnosis
CDS	Clinical Decision Support
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolution Neural Network
CT	Computed Tomography
CXR	Chest X-Ray
DL	Deep Learning
DBN	Deep Belief Network
DNN	Deep Neural Network
FFNN	Feed Forward Neural Network
GPU	Graphics Processing Unit
Grad-CAM	Gradient-weighted Class Activation Mapping
GRU	Gated Recurrent Unit
HE	Histogram Equalization
KNN	K-Nearest Neighbor
LDA	Linear discriminant analysis
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multi-Layer Perceptron
MTANN	Massive Training Artificial Neural Network
NFNets	Normalization-Free Networks
NLP	Natural Language Processing

NN	Neural Network
PA	Posterior Anterior
QDA	Quadratic Discriminant Analysis
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SVM	Support Vector Machine
YOLO	You Only Look Once

CHAPTER 1
INTRODUCTION

1.1 Background

Chest radiography (chest X-ray or CXR) is one of the most routinely used medical imaging and diagnostic modalities. Over 2 billion chest radiograph procedures are performed each year [1]. On the other hand, chest radiography is immensely common and popular in that it is available even in under-resourced small-scaled hospitals to advanced large-scale hospitals around the world. This worldwide popularity and acceptance of chest radiography as a diagnostic tool are due to its non-invasive nature, operational speed, lower cost, lower infrastructure setup, ease of acquisition, accessibility, portability, and relatively lower radiation dose [2-4]. It has an important clinical value in the diagnosis/screening of various abnormalities in the cardiothoracic region such as pneumonia, tuberculosis, interstitial lung diseases, early lung cancer, etc. [5-6]. Meanwhile, in present medical practice, the interpretation of image information from CXRs is performed by the radiologists and the diagnostic expertise of the radiologist has a significant impact on the accuracy of the results [7].

In the last decades, the landscape of healthcare services is continuing to shift. In radiology including other diagnostic imaging-related domains, the use of Clinical Decision Support (CDS) system tools has emerged greatly to deliver high-quality and affordable care [8]. Today, the growth of daily increase in radiological examinations far exceeds the growth of radiologists who are responsible to interpret these studies [9]. In this context, the role of the CDS system in the field of radiology has been evolving showing the great potential to assist in the process of decision-making by radiologists in clinically relevant real-world medical practice [10]. With the Artificial Intelligence (AI) boom gaining momentum, particularly with the development of many deep learning techniques, it has led to achieving the automatic diagnosis of diseases as part of the CDS system thereby minimizing the lengthy screening period and considerably enhancing the efficiency of diagnosis [11].

More recently, developments in the deep learning field, especially CNNs, and the availability of a large volume of data sets showed great success in the automatic diagnosis of diseases in chest radiographs [12]. A CNN is an excellent deep learning algorithm for image classification because of its strong ability for feature extraction and classification even from scratch images. It is based upon the functioning of sophisticated neural networks in the human brain and after being trained on a huge data set, identifies salient clinical features in images [13-14]. Previously, the need for vast amounts of data was a barrier to the effective training of deep learning models. However, in recent years, the availability of large publicly accessible datasets and the increased numbers of high-end computing power led to an increase in several research works that dealt with disease detection in CXRs using CNN models [15-16]. All of these works clarified that CNN systems can outperform humans in various cases of disease detection and help to make decisions on CXR interpretation as part of the CDS system. Taking into consideration of progress in Deep Learning (DL) and CNN, this research work is based on building a CNN architecture from scratch that detects pulmonary abnormalities from CXR images and efficiently classifies them into one of five different classes on a wide range of datasets. Besides, the model performance is observed and validated with real-time datasets collected from various hospitals and imaging centers in an attempt to implement an efficient decision support system for radiologists in medical practice.

1.2 Problem Statement

Despite the wide use of CXR images in clinical diagnosis, the correct interpretation of the information in the CXR images is always a major challenge even for experienced radiologists and other physicians [17]. The major challenges are summarized below:

(i) A chest radiograph is a 2-dimensional representation of a 3-dimensional anatomical structure. As the X-rays pass through the body, these are absorbed by multiple anatomical structures producing different pixel values in the respective radiograph. Besides, different anatomic components might overlap in a single 2-dimensional image, and various pathological and physiological alterations can appear identical or single pathology may display various features.

(ii) In practice, radiologists are primarily responsible for interpreting information from CXR images. However, not all radiologists are equally qualified or experienced, and hence, different radiologists can make an inconsistent diagnosis on the same CXR image. This indicates that the results are influenced by cognitive ability, experience, fatigue, and other human-dependent factors as well [18].

Screening of CXR images is thus highly susceptible to misinterpretation or missed information. For example, over 22% of all errors in diagnostic radiology are made during chest radiograph interpretation [19]. Besides, the adequate interpretation of the CXR findings demands a radiologist's expertise, time, and energy. However, the number of qualified radiologists has remained limited to properly address the highly increasing number of examinations that further widen the scope of misinterpretation [20]. Thus, addressing the issue of misinterpretation or missed information by overcoming the inherent limitations of human perception and workload has been one of the key motivating factors of this research work.

In recent years, several research works have been proposed on different AI-based models to improve the efficiency and accuracy of diagnosis. However, real-time adoption of these models remains limited as follows:

(i) Most of these works dealt with two or three class abnormalities of CXRs classification problems, however, real-world applications may require multiple-class disease detection from CXRs.

(ii) They have limited generalizability across datasets being trained and tested with a few datasets from a single source. Real-world applications demand efficient performance across a wide dataset.

(iii) Existing works mostly employed pre-trained models such as LeNet, AlexNet, Inception, VGGNet, etc, which are huge with millions of trainable parameters requiring much computational power and time. Additionally, the use of pre-trained models requires fine-tuning the CXR images to perform better.

(iv) These research works lack rigorous comparative assessment against standard data sets for real-time applications.

Thus, addressing the above challenges by developing a custom-designed CNN model and then training and testing the model over multiple source datasets for efficient performance has become the other motivating factor in carrying out this research work.

1.3 Objectives

This research aims to design a more efficient CNN-based model for classifying input CXR image into one of five classes including 4 major abnormalities findings that is common in CXR readings. The specific objectives of this research are:

(i) To develop a CNN-based model to classify lung diseases from chest radiographs.

(ii) To model the system by optimizing the hyper-parameters of the CNN algorithm.

(iii) To validate the performance of the proposed model from actual medical data collected from hospitals to employ it as a decision support system for radiologists.

1.4 Contribution of this Proposed Research

The primary technical contributions reported in this thesis work are noted below:

(i) **Development of an efficient CNN architecture from scratch**: This research work developed the **CXRNet**, an efficient CNN architecture from scratch, to detect abnormalities from CXR images and effectively classify them into one of five different classes. Also, since the CXRNet has been developed from scratch, it is smaller in size with fewer trainable parameters compared to various available pre-trained models thus overcoming the limitations of research works using such pre-trained models.

Also, the study demonstrates the tuning of hyperparameters and the use of various optimization algorithms to achieve better model performance.

(ii) **Development of a generalized dataset**: This work developed a generalized dataset collected from various publicly accessible sources originally from different hospitals and research centers, thus enhancing the generalizability of the purposed model across data sets. In this work, a total of 13,619 Posterior Anterior (PA) CXR images are collected corresponding to five different categories namely, Cardiomegaly, COVID-19, Pneumonia, normal, and Tuberculosis, thus, addressing the issues of limited findings in most of the relevant research works.

(iii) **Development of a robust image processing algorithm**: An effective image processing algorithm is developed that incorporates the input CXR pre-processing techniques including data augmentation and also demonstrates the impact of image pre-processing on the model performance.

(iv) **A comprehensive review of various deep-learning techniques**: The study presents a comprehensive review of various DL techniques in the detection of various abnormalities present in chest radiographs. After careful analysis, their performance is observed, and the limitations have also been investigated.

1.5 Thesis Organization

This thesis is divided into five different chapters. In addition to Chapter 1, the thesis is organized as follows:

Chapter 2 presents a review of related work providing a thorough review of various DL techniques in the detection of various abnormalities present in chest radiographs investigating various limitations as well.

Chapter 3 covers the research methodology of the proposed model. It discusses the various materials and methods used in designing the CNN architecture from scratch including relevant formulations and algorithms.

Chapter 4 presents a thorough analysis and discussion of the experimental results including the performance of the model in the standard dataset.

Chapter 5 concludes this thesis with a summary of research findings and outcomes. Also, the limitations and recommendations for future research are presented as well.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

This chapter explores the different deep-learning models developed for the detection of abnormalities from CXR images over the past decade. With a detailed overview, it analyzes how these models detect and classify different diseases. Besides, careful comparative analysis of these DL techniques in radiology and their performance and limitations are investigated to unfold the scope of future development of DL models for clinical analysis and decision support systems.

2.2 Present State of DL Models

The evolution of DL models is presented before analyzing their detection and classification performance with chest radiography. The evolution of DL in the field of medical imaging is figuratively represented by Fig. 2.1.

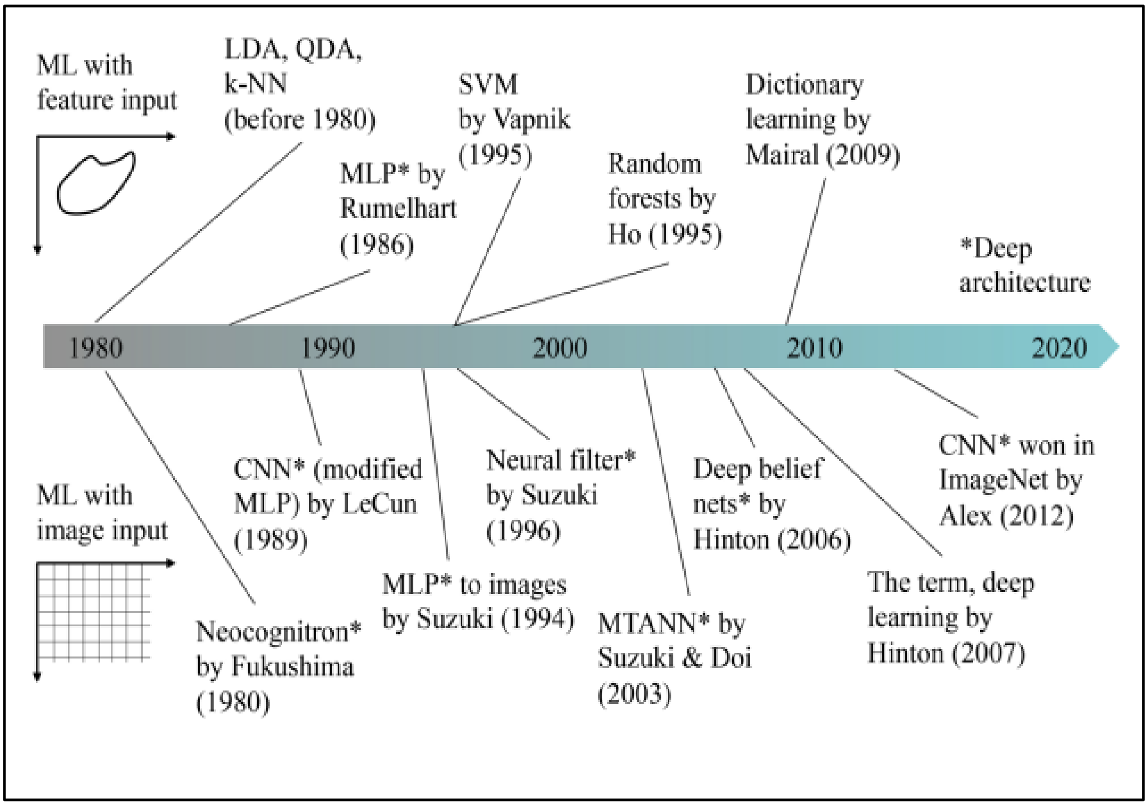

ML with feature input
LDA, QDA, k-NN (before 1980)
SVM by Vapnik (1995)
Dictionary learning by Mairal (2009)
MLP* by Rumelhart (1986)
Random forests by Ho (1995)
*Deep architecture
1980
1990
2000
2010
2020
ML with image input
CNN* (modified MLP) by LeCun (1989)
Neural filter* by Suzuki (1996)
Deep belief nets* by Hinton (2006)
CNN* won in ImageNet by Alex (2012)
MLP* to images by Suzuki (1994)
Neocognitron* by Fukushima (1980)
MTANN* by Suzuki & Doi (2003)
The term, deep learning by Hinton (2007)

Fig. 2.1: The evolution of DL in the field of medical imaging.

Artificial Intelligence is a branch of computer science that enable machines to perform tasks requiring human-like intelligence. The goal of artificial intelligence is achieved by Machine Learning (ML), a subset of artificial intelligence that enables computer systems with learning capabilities to carry out tasks using data automatically and without manual programming. Similarly, deep learning, a subset of ML, primarily deals with the extraction of useful deeper by using artificial neural networks with interconnected nodes simulating neurons of the human brain [21]. Deep Learning is thus considered as the image input machine learning or image-based ML. Deep Learning began gaining popularity starting in 2013. Before that, feature-based ML or ML with feature input used to be dominant in the field of artificial intelligence. Before 1980, even before ML, classical classifiers like Quadratic Discriminant Analysis (QDA), Linear discriminant analysis (LDA), and a K-Nearest Neighbor (k-NN) classifier were employed for classification [22]. A Multi-Layer Perceptron (MLP) was subsequently proposed by Rumelhart and Hinton in 1986 [23]. Later, Vapnik proposed a Support Vector Machine (SVM) algorithm in 1995 which became the most popular classifier for quite some time with publicly available code on the Internet [24]. Successive development of ML models includes random forests by Ho et al. in 1995 [25] and dictionary learning by Mairal et al. in 2009 [26].

Over time, a number of ML with image input (image-based ML) techniques were proposed before the term “deep learning” was introduced in the artificial intelligence field. DL models can be tracked since 1980 from the Neocognitron by Fukushima [27] which was simplified in 1989 by LeCun et al. [28] and then proposed a CNN. Suzuki et al. [29] made use of an MLP to cardiac images in a convolutional way in 1994. In 2003, Suzuki et al. proposed a Massive Training Artificial Neural Network (MTANN) for the classification of patterns [30] and the separation of specific patterns from other patterns in X-ray images in 2006 [31]. In 2006, Hinton et al. proposed Deep Belief Network (DBN) [32] and introduced the term ‘‘deep learning’’ a year later.

Unfortunately, because of the complexity of the underlying optimization problem and the limitations of the computational technology available at the time, deep learning was not widely acknowledged until late 2012. Later in 2012, Convolution Neural Network won the ImageNet Large Scale Visual Recognition Challenge, an annual worldwide picture classification challenge, gaining significant recognition and offering breakthrough performance even exceeding humans in various cases [33].

In recent years, three major variables have enabled to gain the popularity of DL architectures in almost every domain: (a) the availability of enormous amounts of labeled data, (b) the affordable and efficient parallel computing resources, and (c) the advancement in training strategies and architectures [34].

2.3 DL Models for CXR Image Analysis

This section summarizes various DL models for the automatic detection of abnormalities on chest X-ray images. The common abnormalities that those models address are: Pneumonia detection, COVID-19 detection, Tuberculosis detection, and other multiple abnormalities detection. The study of DL models for these specific sets of CXR abnormalities detection is presented in this section.

2.3.1 Pneumonia Detection

Along with the availability of large public chest X-rays data sets, the introduction of DL techniques has proven to yield rapid advancement in pneumonia detection.

In 2019, O. Stephen et al. introduced a CNN model that was trained entirely from scratch to classify and detect pneumonia from CXR images [35]. In this work, in contrast to other methods that simply rely on transfer learning approaches or conventional handcrafted techniques, they designed a CNN model entirely from scratch to extract features and classify input CXR image if it is pneumonia infected or normal. They rearranged the original dataset consisting of 5,856 AP CXR images into 3,722 images for the training set and 2,134 images for the validation set. Also, several data augmentation techniques were employed to artificially increase the volume of the dataset and to address the overfitting problems. They designed a 12-layered sequential CNN model which was trained to obtain a training accuracy of 95.31% and a validation accuracy of 93.73%. Although they have developed the CNN model from scratch, the study is limited by the depth of data, limited findings, and poor classification accuracy for a two-class classification problem.

In 2020, R. Jain et al. demonstrated six CNN models to identify pneumonia using CXR images out of which two models incorporate two and three convolutional layers respectively, and four other models are ResNet50, VGG19, VGG16, and Inception-v3 pre-trained models [36]. In this work, they used 5216 CXR images for training and 624 images for testing obtaining validation accuracy of 85.26% in model 1, 92.31% in model 2, 87.28%, 88.46%, 77.56%, and 70.99% in VGG16, VGG19, ResNet50, and Inception-v3 pre-trained models respectively. Although they have trained several models, they have not properly tuned hyper-parameters as well as fine-tuned the parameters of the pre-trained models, thus, resulting in poor performance in the limited dataset.

In 2021, M. M. Eid et al. presented the application of pre-trained CNNs for extracting features and SVM for Pneumonia classification from CXR images [37]. In this work, they combined residual neural networks (ResNet-50) for feature extraction and SVM for classification. A total of 5,863 CXR images were used which had been grouped randomly into 60% images for training, 20% for validation, and 20% for testing the model. Here, they have achieved an accuracy of 98.03% which is comparatively superior to other related research works.

In 2022, O. A. Fagbuagun et al. presented an approach using CNN and transfer learning techniques for the accurate detection of Pneumonia from CXR images [38]. In this work, they have used a total of 5,856 CXR images out of which 5,232 were used for training and 624 images for testing the model. They have designed a model using Inception-V3 as their pre-trained base model for feature detection and added a new two-class classifier output layer to the end of the network. Here, the model achieved a training accuracy of 95.66% on the training set while an accuracy of 88.14% and an f1-score of 87% on the test images. This indicates that they have not properly addressed the overfitting issues and fine-tuning of the pre-trained model resulting in poor performance in the test dataset.

In 2022, A. Mabrouk et al. presented a computer-aided detection of Pneumonia using ensemble learning based on pre-trained CNN models [39]. They used three well-known pre-trained CNN models namely DenseNet169, MobileNetV2, and Vision Transformer all pre-trained in the ImageNet database. These models were then fine-tuned using the CXR dataset which was combined for final classification. In this work, a total of 5,856 CXR images were used out of which 624 images were used for testing the model. Here, they achieved an accuracy of 93.91% and an F1-score of 93.88% on the test dataset.

In 2022, E. Ayan et al. proposed a method based on the CNN ensemble model for the automatic diagnosis of Pneumonia where they developed a computer-aided Pneumonia detection system [40]. In this work, they trained seven popular CNN models (ResNet-50, VGG-16, VGG-19, Inception-V3, MobileNet, Xception, and SqueezeNet) pre-trained on the ImageNet dataset with fine-tuning strategies on the CXR dataset. Out of seven, the three most successful ones were chosen for the ensemble method. Here, they used a total of 5,856 CXR images where 5,232 images were used for training while 624 were used for testing the model. In model training, ResNet-50, Xception, and MobileNet models achieved superior results with accuracy of 94.42%, 95.03%, and 94.87% which were then selected for the ensemble model giving the accuracy of 95.83% on the test dataset.

In 2023, M. W. Kusk et al. performed an experiment to train and validate the CNN model for the classification of Pneumonia on CXR images acquired at different levels of image noise [41]. In this work, they used a dataset of 5856 AP CXR images where Gaussian noise with zero mean was included in the CXR images at 5 different noise variance levels, representing the decreasing exposure in the image. This dataset was distributed into 80% for training, 10% for testing, and 10% for validation. A custom sequential CNN architecture was trained with six different data conditions including the original dataset and five other datasets with added noise at different levels. Here, they observed that there is no decrease in performance from the original dataset without noise to the five other datasets with added noise at different levels. The accuracy for the original dataset was 90.2% while the accuracy for noise-added datasets ranged from 96.8% - 97.6% on the test dataset. Thus, it was concluded that no performance drop of CNN was observed for added noise compared to the original dataset indicating the potential of decreasing radiation dose to the patients.

In summary, these studies have proven that the implementation of DL techniques whether it is custom-designed CNNs, or transfer learning-based models can detect generic pneumonia on chest X-ray images with satisfactory accuracy. In addition, techniques such as input image pre-processing, and data augmentation with properly tuned hyperparameters in model training enhance the performance of the deep learning models.

2.3.2 COVID-19 Detection

After the COVID-19 pandemic and millions of confirmed cases all over the world, the fast detection of the disease with minimal diagnosis error has become a crucial task among the research community. There has been a massive increase in research regarding chest X-ray images along with DL algorithms for early COVID-19 detection. During the early phase of the COVID-19 pandemic, there were limited numbers of labeled COVID-19 cases, and thus, transfer learning using popular pre-trained classifiers was common in practice.

In 2020, T. Ozturk et al. presented a model for the automatic detection of COVID-19 from CXR images [42]. In this work, they used the DarkNet-19 model as a classifier for the You Only Look Once (YOLO) real-time object detection system instead of building a model from scratch. They modified the original DarkNet architecture using fewer layers and gradually increasing the number of filters in convolution layers. Here, they used a total of 1,125 CXR images where 80% of the images were used for model training and 20% for testing the model. They achieved an accuracy of 98.08% for binary classifications (COVID-19 and No-Findings) and 87.02% accuracy for three classes classification (COVID-19, No-Findings, and Pneumonia). Although, the model performed well in binary classification, however, the performance decreased in the three-class classification task which might be limited due to the less volume of training dataset used in this work.

In 2021, T. Rahman et al. explored the impact of image enhancement techniques and lung segmentation on COVID-19 detection from CXR images [43]. In this work, they proposed a modified U-Net model for lung segmentation and then the performance of seven different CNN models including six pre-trained networks (ResNet101, ResNet50, ResNet18, InceptionV3, ChexNet, and DenseNet201) and one shallow CNN model developed from scratch was examined on plain and segmented lung CXR images where the images were subjected to five different image enhancement techniques (Balance Contrast Enhancement Technique (BCET), Histogram Equalization (HE), image complement, Contrast Limited Adaptive Histogram Equalization (CLAHE), and gamma correction). Here, a total of 18,479 CXR images were used out of which 80% of images were employed for model training and 20% for testing the model. In this work, they observed that the gamma correction-based enhancement technique outperformed other techniques in identifying COVID-19 from the plain and segmented CXR images. However, it was observed that the classification performance is slightly better in plain CXR images as compared to the segmented lung images. They achieved the highest accuracy of 96.29% using the ChexNet model on gamma-corrected plain CXR images and 95.11% accuracy using the DenseNet201 model on gamma-corrected segmented lung images for three class classification task (COVI-19, non-COVID lung opacity, and normal).

In 2022, J. L. Gayathri et al. used a combination of InceptionResnetV2 and Xception pre-trained model for feature extraction, sparse autoencoder for dimensionality reduction, and Feed Forward Neural Network (FFNN) for detection of COVID-19 in CXR images [44]. In this work, a total of 504 COVID-19 images and 542 non-COVID-19 images were used out of which 90% of images were used for training and 10% for testing the model. Here, they achieved an accuracy of 95.78% in the test dataset from this model.

In 2022, S. Sanket et al. presented a CNN-based model for the detection of COVID-19 intending to assist physicians throughout the diagnostic procedure in high workload conditions [45]. In this work, they proposed the CovCNN model which is a deep-CNN-based architecture. Here, they included 219 COVID-19-positive images and 438 non-COVID-19 images out of which 80% images were used for training and 20% images for testing the model. Also, they were able to achieve the highest classification accuracy of 98.4% using their CovCNN model in this work.

In 2022, A. Kumar et al introduced a Computer Aided Diagnosis (CAD) system combining Graph Convolutional Networks and CNNs for detecting COVID-19 in a CXR image [46]. In this work, they designed and evaluated the performance of a custom-made deep learning architecture SARS-Net to detect and classify CXR images for COVID-19 diagnosis. Here, a total of 13,975 CXR images were employed out of which 90% images were used for training and validation and the rest 10% images for testing the model. Also, the proposed SARS-Net model achieved an accuracy of 97.6% on the test set which is promising compared to other relevant works.

From all these papers, it can be concluded that the COVID-19 pandemic has intensified the research into its detection using a diverse number of deep learning models including custom-designed CNNs, transfer learning, and ensembles. These studies have proven that the implementation of deep learning techniques whether it is custom-designed CNNs, or transfer learning-based models can detect generic pneumonia on chest X-ray images with satisfactory accuracy. However, there is still much work to be done to integrate multi-classifier models working in a clinical environment.

2.3.3 Tuberculosis Detection

There is a large number of research works based on various DL techniques for the automatic detection and classification of Tuberculosis from input CXR images. Most of these works make use of several pre-trained CNN models for feature extraction tasks and then add a classifier for predictions.

In 2022, S. I. Nafisah and G. Muhammad proposed an automatic Tuberculosis detection system using advanced DL models [47]. Different CNN models were used in this work and their classification performance was compared using the publicly available CXR dataset. In this work, a total of 1098 CXR images were used where 80% of images were used for training and the rest 20% for testing the model. Here, they employed a transfer learning approach using five pre-trained models (ResNet50, Inception, Xception, MobileNet, and EfficientNetB3) for the automatic detection of Tuberculosis from CXR images. They performed classification tasks for both segmented lung CXR images and raw CXR images, also with and without augmentation in both cases. In this experiment, they were able to obtain the highest accuracy in augmented and lung-segmented CXR images with an average accuracy of 98.7% in the two-class classification task. Although they achieved superior classification tasks, the experiment is limited to generalization ability due to a limited number of datasets.

In 2022, V. Acharya et al. introduced AI-assisted Tuberculosis detection and classification from CXR using a DL normalization-free network model [48]. In this work, they trained ImageNet fine-tuned NFNets for classification tasks and utilized the Score-Cam algorithm to emphasize the areas in the CXR image for accurate inference on the diagnosis. Here, they utilized a total of 3500 CXR images which were split into a 4:1 ratio for training and testing the model. The proposed model achieved an average accuracy of 96% for binary classification demonstrating its importance as a secondary decision tool for assisting radiologists.

In 2022, E. Showkatian et al. proposed a DL-based automatic detection of Tuberculosis disease in CXR images by training a CNN model from scratch [49]. Also, they compared its performance with several other pre-trained CNNs as well. In this work, a total of 800 CXR images were augmented to generate a larger data volume with 2040 CXR images in the training set, and 120 CXR images in both the testing and validation sets. Here, the proposed CNN architecture achieved an accuracy of 87% which was observed to be less than the classification accuracy obtained from pre-trained models like Exception, ResNet50, and VGG16.

In 2023, S. S. Guia et al. employed a DL model for Tuberculosis detection from CXR images using two pre-trained VGG16 and VGG19 models [50]. In this work, a total of 7000 CXR images were used which were divided into training, testing, and validation sets in the ratios 60%, 20%, and 20% respectively. Here, they created an architecture by combining two pre-trained models and adding a dense binary classifier. Also, the proposed architecture was able to achieve an accuracy of 99% on the test set for the binary classification task.

In summary, it can be said that the detection of pulmonary tuberculosis has advanced substantially with the introduction of various DL models. Several papers showed that pre-trained models outperform in most cases. Also, employing the data augmentation technique played a vital role in increasing the accuracy of the model. However, further works require the implementation of these models in clinical settings, integration into multi-classifier systems, and interaction with human observers.

2.3.4 Multiple Abnormalities Detection

Generally, the multi-class abnormality classification requires larger training data sets. As of today, there are several research works on multi-class abnormalities detection models from CXR images. These models are based on the datasets which are created by combining datasets from multiple sources.

In 2020, A. I. Khan et al. proposed CoroNet, a Deep CNN model to automatically detect Pneumonia and COVID-19 infection from CXR images [51]. In this study, they proposed a model based on Xception architecture pre-trained on the ImageNet dataset and trained the model by collecting CXR images for four class classification tasks (COVID-19 vs Pneumonia bacterial vs Pneumonia viral vs normal). Here, they used only 310 normal, 330 Pneumonia bacterial, 310 Pneumonia viral, and 284 COVID-19 CXR images for training. Four equal sets were created by randomly dividing the training set out of which 3 sets were used to train and the remaining set was used for validation. Here, the proposed CoroNet model achieved an average accuracy of 89.6% for the 4-class classification task.

In 2022, A. Musha et al. proposed a model based on YOLOv2 with ResNet architecture for COVID-19 and Pneumonia detection from CXR images [52]. In this work, they collected 3788 CXR images out of which 80% images were used for training and 20% images for testing the model. Here, the proposed model achieved a maximum overall classification accuracy of 97.3% for three class classification (COVID-19 vs Pneumonia vs normal) tasks.

In 2022, F. Bayram and A. Eleyan proposed fusion-based DL for feature extraction and classification by implementing a multi-stream CNN model for COVID-19 and Pneumonia detection on CXR images [53]. A total of 3886 CXR images were used including 1200 COVID-19, 1341 normal, and 1345 Pneumonia images. The dataset was divided into five subsets for fivefold cross-validation. In this study, a multi-stream CNN was designed with three inputs corresponding to the grayscale CXR image, the corresponding local binary pattern, and histograms of oriented gradient features of CXR images. Each of these three inputs was passed in parallel through the feature extraction module consisting of five feature extraction layers composed of a convolution layer, a batch normalization layer, a ReLu activation layer, an average pooling layer, and a dropout layer. The features obtained from these three parallel layers were concatenated and passed to the classification module with three outputs for normal, COVID-19, and Pneumonia cases. Here, the proposed model achieved an average accuracy of 97.76% in the three-class classification task.

In 2022, M. Loey et al. proposed a Bayesian-based optimized DL model to detect COVID-19 and Pneumonia from CXR images [14]. The proposed model is composed of two main components; the first one uses CNN for feature extraction and the second is a Bayesian-based optimizer used for tuning CNN hyperparameters. In this work, they collected a total of 10,848 CXR images including 3616 COVID-19, 3616 normal, and 3616 Pneumonia case images. The dataset was split into three different sets comprising of training, validation, and test sets in the ratio 80%, 10%, and 10% respectively in one set, 70%, 10%, and 20% respectively in the second set, and 60%, 10%, and 30% respectively in the third set. Also, the proposed model achieved the highest accuracy of 96% for a three-class classification task (COVID-19 vs Pneumonia vs normal) in a 10% test split dataset.

In 2023, I. Kanjanasurat et al. combined CNN and Recurrent Neural Network (RNN) models by substituting fully connected layers of CNN with an RNN for the diagnosis of COVID-19 and Pneumonia from CXR and Computed Tomography (CT) images [54]. In this work, the strength of CNNs was employed to extract features, and RNNs were utilized to calculate dependencies and classification based on extracted features. Here, pre-trained CNN models like DenseNet121, ResNet152V2, and VGG19 were combined with LSTM and Gated Recurrent Unit (GRU)-RNN models. Also, a total of 16,210 images were used out of which 9271 were CXR images (2545 normal, 3789 Pneumonia, and 2937 COVID-19 images) and 6939 were CT images (2259 normal, 2365 Pneumonia, and 2315 COVID-19 images). They approximately used 65% of all the image data for training, 20% for testing, and 15% for validation. Out of three CNN-RNN networks, ResNet152V2 with GRU model performed the best achieving 93.37% accuracy for three class classification tasks (COVID-19 vs Pneumonia vs normal).

In summary, the multi-classifiers are greatly advancing rapidly over time showing superior performance in comparison to human operations in many cases. However, due to the more complex and difficult task of classification into more classes, these DL models generally exhibit lower performances than DL models classifying only one form of pathology. This can be also related to the heterogeneity of the testing data set due to overlapping features from several pathologies resulting in misclassification.

2.4 Discussion

The major developments of DL models for CXR image analysis discussed above can be broadly grouped into four categories: (a) CNN-based related studies, (b) transfer learning-based studies, (c) ensemble model-based studies, and (d) other neural network-based studies. CNN-based studies are based on designing and developing CNNs architectures for the detection and classification of diseases in CXR images. Transfer Learning based studies include the application of the pre-trained deep learning models that make use of the information and knowledge learned from the past training itself. The new set of models is then used to put this training to use. Ensemble model-based studies are based on making use of multiple related but different analytical deep-learning models, then synthesizing the result into a single score to improve predictive accuracy. Finally, other neural network-based studies include various research works that are related to image-based ML models supporting deep learning architecture. Table 2.1 presents the performance comparison of different DL techniques that are reviewed in this section for several abnormalities detection in CXR investigating their limitations at the same time.

Table 2.1: Performance comparison of different DL techniques for several abnormalities detection on CXR images

Research	**Dataset**	**Method Used**	**Output Classes**	**Accur-acy(%)**	**Limitations**
A. Pneumonia Detection					
(O. Stephen et al., 2019)	5,856 CXR images	CNN from scratch	2	93.7	Only two class classification model which is trained on a few datasets with lower accuracy performance.
(R. Jain et al., 2020)	5,216 CXR images	Pre-trained CNN	2	92.3	Only two class classification model which is trained on a few datasets with lower accuracy performance. Additionally, it employs the use of pre-trained model without fine tuning.
(M. M. Eid et al., 2021)	5,863 CXR images	Pre-trained CNN with SVM	2	98.1	Only two class classification model which is trained on a few datasets. Also, employs pre-trained model.
(O. A. Fagbuagun et al., 2022)	5,856 CXR images	Pre-trained CNN	2	88.1	Only two class classification model which is trained on a few datasets with lower accuracy performance. Also, it employs the use of pre-trained model without fine tuning.
(A. Mabrouk et al., 2022)	5,856 CXR images	Ensemble learning based on pre-trained CNN models	2	93.9	Only two class classification model which is trained on a few datasets with lower accuracy performance. Also, it employs the use of pre-trained model without fine tuning.
(E. Ayan et al., 2022)	5,856 CXR images	Ensemble learning based on pre-trained CNN models	2	95.8	Only two class classification model which is trained on a few datasets. Also, employs pre-trained model.

(M. W. Kusk et al., 2023)	5,856 CXR images	Custom sequential CNN	2	96.8 – 97.6	Only two class classification model which is trained on a few datasets.
B. COVID Detection					
(T. Ozturk et al., 2020)	1,125 CXR images	DarkNet-19 model	2	87	Only two class classification model which is trained on a few datasets.
(T. Rahman et al., 2021)	18,479 CXR images	Pre-trained CNN models	3	95.1	Only three class classification model built using pre-trained models.
(J. L. Gayathri et al., 2022)	1,046 CXR images	Pre-trained CNN models	2	95.8	Only two class classification model which is trained on a few datasets. Also, employs pre-trained model.
(S. Sanket et al., 2022)	657 CXR images	CovCNN model based on deep-CNN architecture	2	98.4	Only two class classification model which is trained on a few datasets.
(A. Kumar et al., 2022)	13,975 CXR images	SARS-Net (custom DL architecture from scratch)	3	97.6	Only three class classification model which is trained on a few datasets.
C. Tuberculosis Detection					
(S. I. Nafisah et al., 2022)	1,098 CXR images	Pre-trained CNN models	2	98.7	Only two class classification model which is trained on a few datasets. Also, employs pre-trained model.
(V. Acharya et al., 2022)	3,500 CXR images	ImageNet model fine-tuned NFNets	2	96	Only two class classification model which is trained on a few datasets. Also, employs pre-trained model.
(E. Showkatian et al., 2022)	2,160 CXR images	ConvNet (CNN model from scratch)	2	87	Only two class classification model which is trained on a few datasets with lower accuracy performance.
(S. S. Guia et al., 2023)	7,000 CXR images	Pre-trained CNN models	2	99	Only two class classification model. Also, employs pre-trained model.
D. Multiple Abnormalities Detection					
(A. I. Khan et al., 2020)	1,234 CXR images	Pre-trained CNN models	4	89.6	Only four class classification model which is trained on a few datasets with lower accuracy. Also, employs pre-trained model.

(A. Musha et al., 2022)	3,788 CXR images	YOLOv2 with residual network architecture	3	97.3	Only three class classification model which is trained on a few datasets. Also, employs pre-trained model.
(F. Bayram et al., 2022)	3,886 CXR images	Fusion based model implemented by a multi-stream CNN	3	97.7	Only three class classification model which is trained on a few datasets.
(M. Loey et al., 2022)	10,848 CXR images	Bayesian-based optimized DL model	3	96	Only three class classification model which is trained on a few datasets.
(I. Kanjanasurat et al., 2023)	(9,271 CXR & 6,939 CT)	Combination of pre-trained CNN and RNN	3	93.4	Only three class classification model with lower accuracy performance. Also, employs pre-trained model.

2.5 Chapter Summary

Existing DL methods for the detection of abnormalities in CXR images, their performance, and limitations are comprehensively investigated in this chapter. DL techniques have demonstrated their great potential for medical image analysis to be a leading research domain in detecting various abnormalities in CXR images and play a vital role as a decision support system in radiological examinations. However, there can be several improvements to enhance the performance of these models and further work is required in the multi-classifier implementation of these models in clinical setting environments.

CHAPTER 3
A NEW CXRNet MODEL

3.1 Introduction

This chapter presents a new CNN model, the **CXRNet,** for the classification of input CXR images into one of five classes namely Cardiomegaly, COVID, Pneumonia, Tuberculosis, and Normal. The materials and methods employed in modeling the proposed new mode for the classification of input CXR images are discussed in this chapter. This chapter captures the sources and nature of datasets, the preparation, and processing of datasets before feeding them into the CXRNet model followed by the model's architecture and tuning of various hyper-parameters for attaining the objectives of this research work. The workflow for classifying the input CXR images is finally presented with the necessary technical details and diagrams. The general flow diagram of the working process of this research work is shown in Fig. 3.1 where the workflow in carrying out the training and testing procedure is illustrated.

3.2 Dataset

A total of 13,619 frontal CXR images are used in this research work for training and testing the model. The dataset includes images acquired from various publicly accessible sources contributing to 5 different CXR class categories which are enlisted as follows:

(i) The primary dataset employed in this research work was acquired from Kaggle repositories consisting of a total of 5863 CXR images and contributing into two main categories namely Pneumonia and Normal classes [55]. These images are originally obtained from Guangzhou Women and Children's Medical Center, Guangzhou, China.

(ii) The COVID category dataset used in this research work was also collected from Kaggle repositories contributing a total of 3616 COVID-positive CXR images [56]. This database is developed from the Italian Society of Medical and Interventional Radiology (SIRM) COVID-19 DATABASE [57], Medical Imaging Databank of the Valencia Region (BIMCV) BIMCV-COVID19+ dataset [58], and several other works in GitHub [59].

(iii) The Tuberculosis category dataset used in this study was again collected from Kaggle repositories contributing a total of 700 Tuberculosis CXR images [60]. This was publicly made available from the database developed by a group of researchers from Qatar University, Qatar, and the University of Dhaka, Bangladesh together with their Malaysian counterparts and medical doctors from Hamad Medical Corporation and Bangladesh.

(iv) The Cardiomegaly category dataset used in this study was collected from the clinical PACS database at the National Institutes of Health Clinical Center (NIHCC) [61]. This is the largest publicly accessible CXR dataset to date with 14 different disease labels, however, only a total of 1093 Cardiomegaly labeled CXR images are used in this research work.

The detailed composition of the dataset used in this thesis work is also presented in Table 3.1. Thus, the CXR images collected from various sources are first grouped into one of five different classes producing a total of 1093 Cardiomegaly, 3616 COVID, 3937 Normal, 4273 Pneumonia, and 700 Tuberculosis cases of CXR images. These images were originally made available in various resolutions. Some of the CXR image samples for each category are shown in Fig. 3.2.

Table 3.1: The detailed composition of the dataset collected from various sources

Data Composition	**Sources**						**Total**
	Guangzhou Medical Center	**SIRM**	**BIMCV**	**GitHub**	**NIHCC**	**Kaggle**	
Cardiomegaly					1093		1093
COVID		119	2474	1023			3616
Normal	1583					2354	3937
Pneumonia	4273						4273
Tuberculosis						700	700

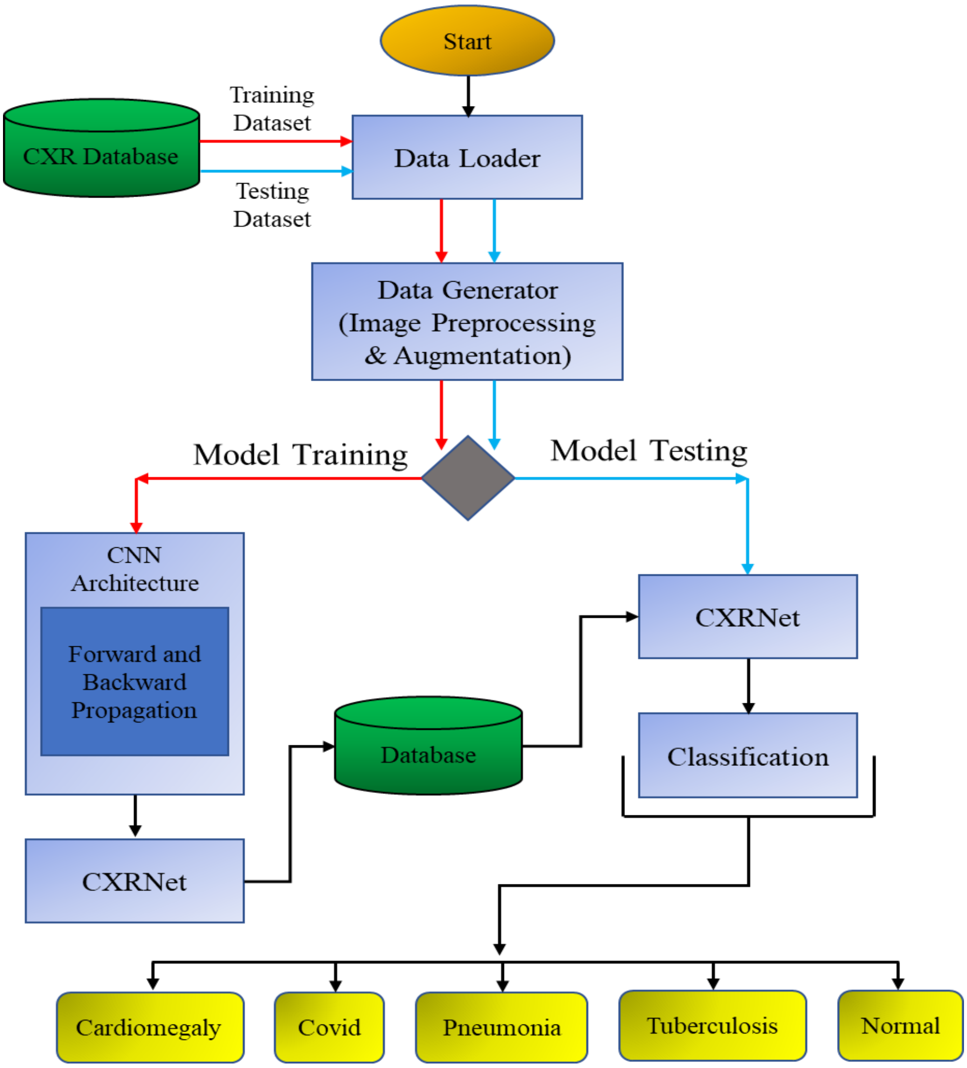
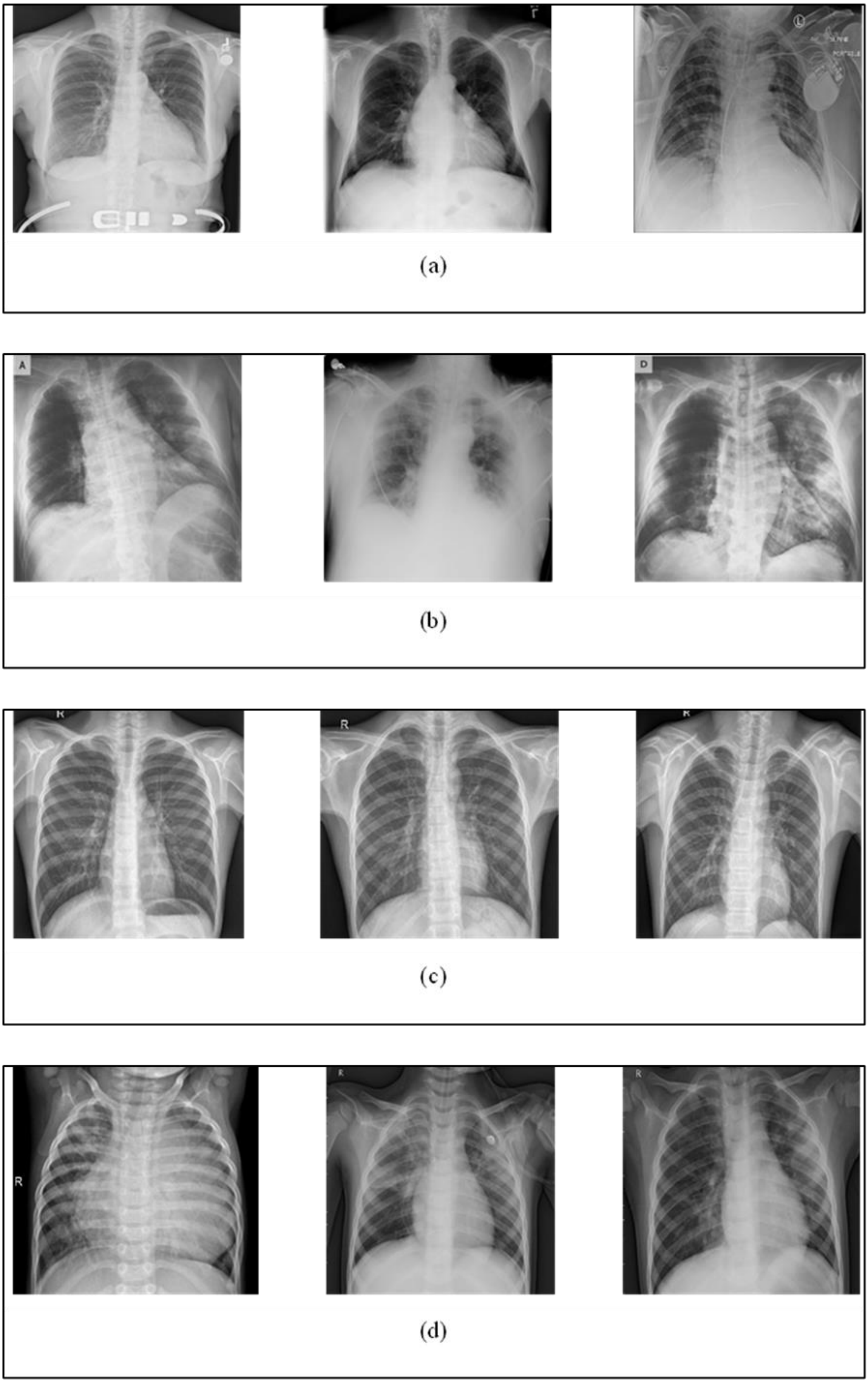


Fig. 3.1: The general flow diagram of the working process of this research work.

In this research work, the data acquisition task is performed keeping in mind to address the following limitations which are common in most of the research works.

(i) The dataset consists of 5 different categories including 4 common chest abnormalities which address the problem of limited findings with 2 or 3 class classification research works thus making it more suitable for real-time adoption of this model in clinical settings.

(ii) The dataset is acquired from various publicly accessible sources thus overcoming the issue of the limited generalizability across datasets as most of the research works are based on a dataset from a single source. This in turn makes the model produce more robust and reliable performance in real-time clinical environments.



(a)

(b)

(c)

(d)

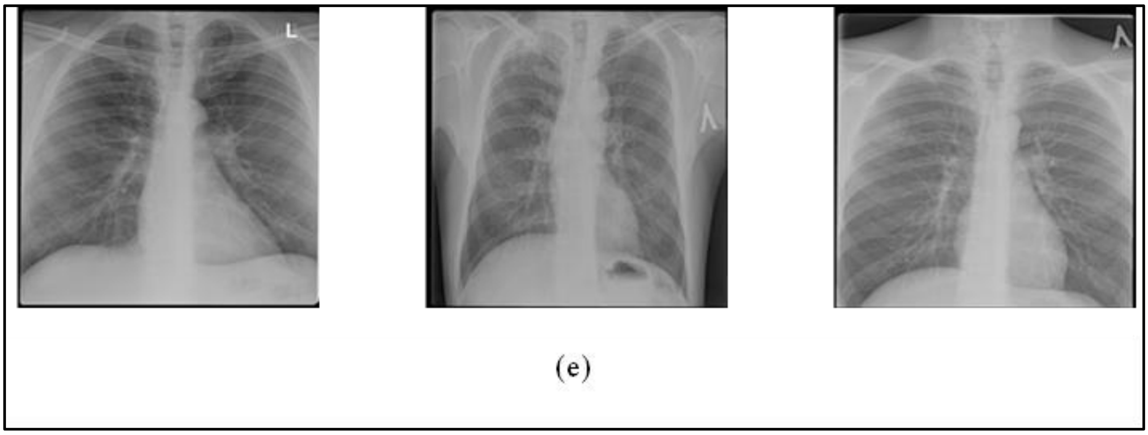


Fig. 3.2: Chest Radiographs samples; (a) Cardiomegaly, (b) COVID, (c) Normal, (d) Pneumonia, and (e) Tuberculosis.

3.3 Data Pre-processing and Augmentation

The CXR images in the dataset are originally 8-bit RGB format (3-channels) images and made available in varying resolutions ranging from 299 × 299 to 512 ×512 to 1024 × 1024 to 2583 × 2916. Hence, image pre-processing is an important step that is performed to prepare data suitable for passing into the CNN model. In this research work, several image pre-processing techniques are implemented using the Keras image preprocessing module which is discussed as follows.

3.3.1 Rescaling

The dataset used in this work is of unsigned integer type with pixel values ranging from 0 to 255. Thus, pixel rescaling is performed to rescale the pixel values from the range of 0 to 255 to the range 0 to 1 which is also known as pixel normalization. This can be achieved by dividing each pixel value by 255 which is represented in Equation (3.1) The pixel normalization ensures that each input image parameter has a similar data distribution and thus making the convergence of loss function faster and speeding up the training process. Also, it helps to reduce the exploding and vanishing gradient problem as during the process of model training, the kernels (weights) are multiplied, and the biases are added to the normalized input feature values.

$$normalize\_images = \frac{original\_\ pixel}{255} \tag{3.1}$$

3.3.2 Resizing

CXR images used in this research work are of various dimensions, however, all the images have to be resized into a shape that is equal to the input layer of the CNN model. Hence, making the decision for the shape of the input layer and resizing the images into that shape is one of the important pre-processing tasks before passing the data into the network. It is governed by two important factors; (a) computation cost which includes memory requirement, and processing time, and (b) model performance which includes accuracy and losses in the model. By taking a smaller image size, it will downscale the number of input features thus reducing memory requirements and overall processing time. However, reducing the image size introduces the loss of image information which might in turn affect the performance of the model. On the other hand, taking a larger image size will preserve the image information which will support the model's performance; however, this will demand higher memory requirements and processing time. Regardless, increasing the image size after a certain value will not decidedly improve the model performance, however, will substantially add up computational cost in the model architecture. Thus, it is important to identify the image size where increasing the size from that instance does not significantly improve the performance for the price of an increase in computation cost. In this research work, the optimum shape of the input layer is set to be 224 × 224, hence, the images are also resized into this shape using the bilinear interpolation method. This method is one of the techniques to find the unknown pixel value at point (*x*, *y*) using the linear interpolation method from the known pixel values at four points which is represented in Equation (3.2).

$$f(x,y) = \frac{1}{(x_2 - x_1)(y_2 - y_1)} \begin{bmatrix} x_2 - x & x - x_1 \end{bmatrix} \begin{bmatrix} f(Q_{11}) & f(Q_{12}) \\ f(Q_{21}) & f(Q_{22}) \end{bmatrix} \begin{bmatrix} y_2 - y \\ y - y_1 \end{bmatrix} \quad (3.2)$$

Where, $f(Q_{11})$ is pixel value at point (x_1, y_1), $f(Q_{12})$ is pixel value at point (x_1, y_2), $f(Q_{21})$ is pixel value at point (x_2, y_1), and $f(Q_{22})$ is pixel value at point (x_2, y_2).

3.3.3 Augmentation

A CNN by itself cannot address rotation and scaling problems if not provided with such samples during the training of the model [62]. Hence, training a CNN needs a huge amount of generalized data to avoid the overfitting problem and to obtain a reliable result [63]. However, the working database might not contain a large volume of generalized data at all times. In such a situation, rather than gathering new data, data augmentation techniques can be utilized to generate an increased number of datasets by augmenting the existing data. Data augmentation can greatly broaden the range of data that are available for training the models. Also, Image augmentation is very crucial when the dataset is imbalanced [43]. Hence, in this research work, various data augmentation strategies are implemented, as listed in Table 3.2, to address the data imbalance problem, prevent the overfitting problem, and improve the generalization abilities of the proposed CNN architecture. The mathematical expression for new rotated coordinates (x_{new}, y_{new}) corresponding to original coordinates (x, y) is represented in Equation (3.3). Also, the mathematical expression for image shear mapping operation is represented in Equation (3.4) where there is linear mapping to displace each point in the image in a fixed horizontal direction by an amount proportional to its y-coordinate.

$$\begin{pmatrix} x_{new} \\ y_{new} \end{pmatrix} = \begin{pmatrix} x * \cos\theta + y * \sin\theta \\ y * \cos\theta - x * \sin\theta \end{pmatrix} \tag{3.3}$$

Where, θ is the rotation angle

$$\begin{pmatrix} x_{new} \\ y_{new} \end{pmatrix} = \begin{pmatrix} x + m * y \\ y \end{pmatrix} \tag{3.4}$$

Where, *m* is a fixed parameter and called it shear factor.

Table 3.2: Data augmentation techniques

Data Augmentation Techniques	**Values**
Rescale	1.0/255
Rotation_range	5
Shear_range	0.2
Zoom_range	0.2
Horizontal_flip	True

In this work, a total of 13,619 frontal CXR images are collected and the distribution of images in each category is represented in Fig. 3.3. It indicates that there is an imbalanced data distribution among five categories. Hence, there is a need for balancing the dataset to prevent the model from becoming more biased towards the majority class. In this work, the oversampling technique is implemented by increasing the number of samples of minority classes using augmentation techniques to balance the overall dataset. The distribution of the dataset after applying the data augmentation to increase the number of samples of minority class is shown in Table 3.3. This data distribution is later used in training and testing the performance of the CNN model.

Table 3.3: Data distribution in each category after oversampling the minority classes using data augmentation strategies (Balanced dataset)

Categories	**No of Samples**
Cardiomegaly	4,207
COVID-19	4,224
Normal	4,225
Pneumonia	4,273
Tuberculosis	4,200
Total	**21,129**

3.4 CXRNet Architecture

CXRNet is a model based on CNN architecture proposed in this research work for the classification of input CXR image into one of five different classes namely Cardiomegaly, COVID, Normal, Pneumonia, and Tuberculosis cases. CNN architecture is implemented in this research work because it is built on feed-forward neural network topologies for automatic feature extraction utilizing the inherent properties of images and then the classification of images based on the extracted features. Moreover, a CNN requires substantially less pre-processing than conventional classification algorithms [64].

The general block diagram of the proposed CXRNet model is shown in Fig. 3.4. It is a 16-layer CNN architecture that is implemented using two different Keras model APIs.

First, the Functional model is created to implement three parallel convolution layers of kernel sizes 3 × 3, 5 × 5, and 7 × 7 respectively. The augmented CXR image is passed through the input layer and then to the three parallel convolution layers where the outputs of these three layers are then concatenated and passed as input to the sequential model. The sequential model is based on six basic components, namely the convolution layer, activation function, pooling layer, dropout layer, flatten layer, and dense layer. These elements are utilized in various layers, and they have their respective functionality in the CNN architecture. The detailed architecture of the purposed CXRNet model is illustrated in Fig. 3.5 while the summary of the model is presented in Table 3.4. In this architecture, there is a total of 2,827,909 trainable parameters that are learned during model training and will be used in the classification of the input CXR images.

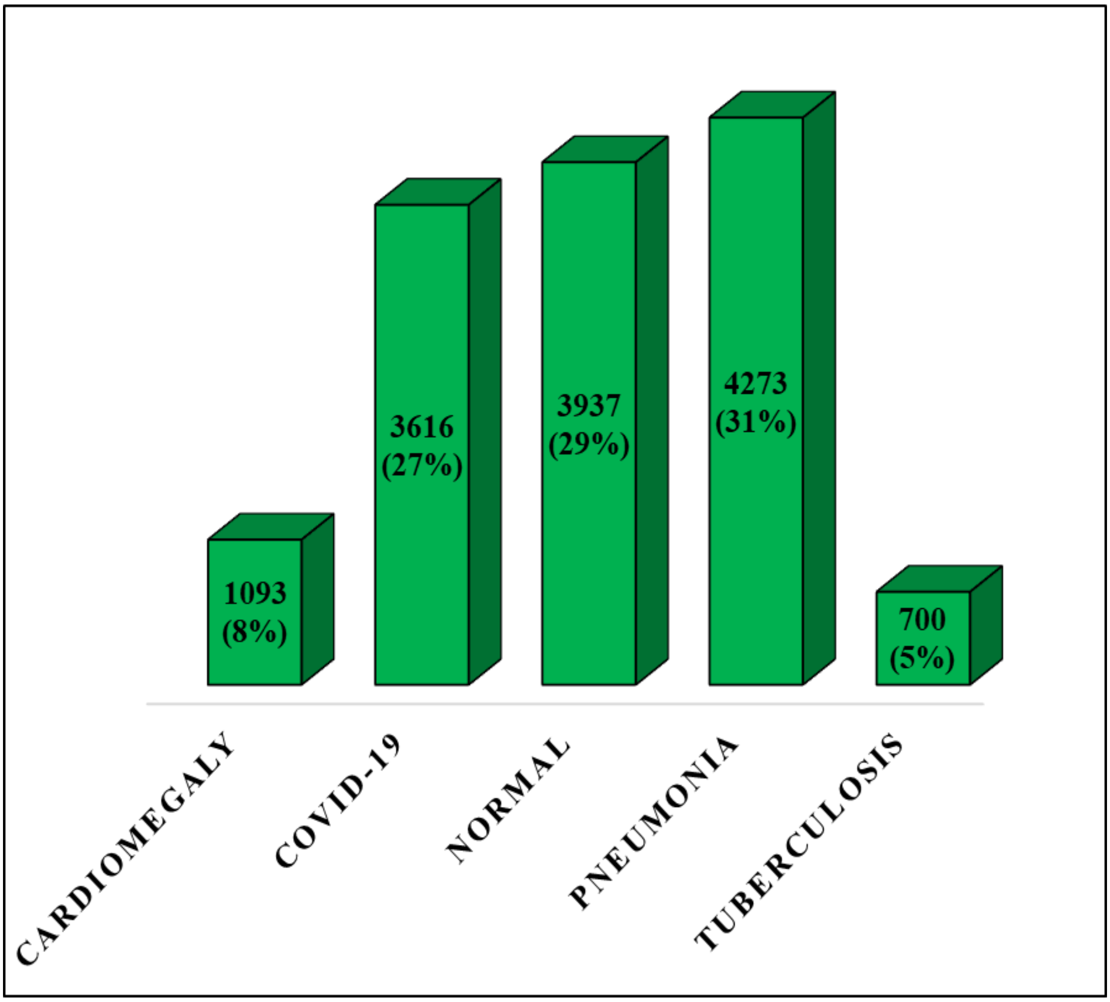


Fig. 3.3: The original distribution of images in each category collected from various sources.

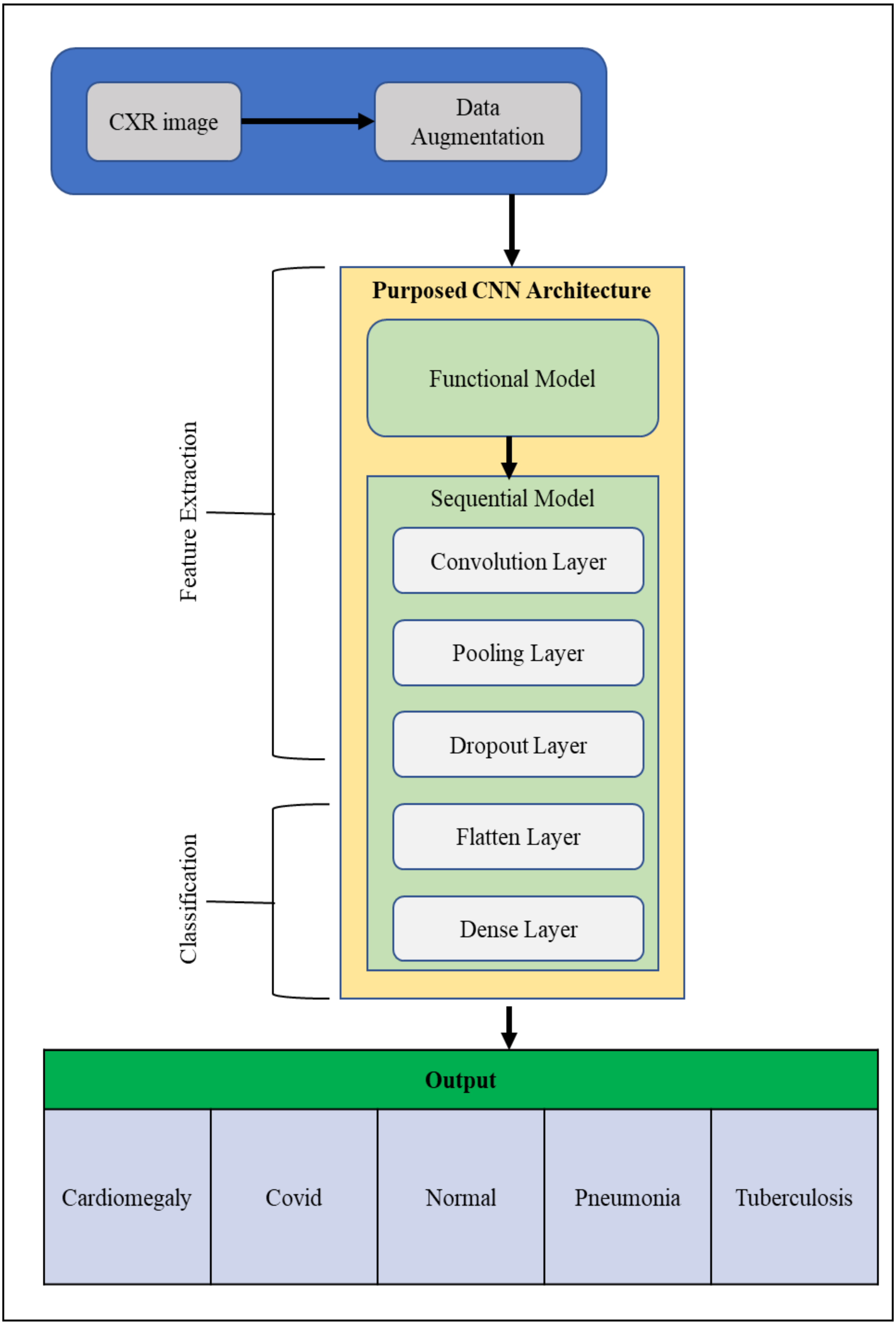


Fig. 3.4: The general block diagram of the proposed CXRNet model.

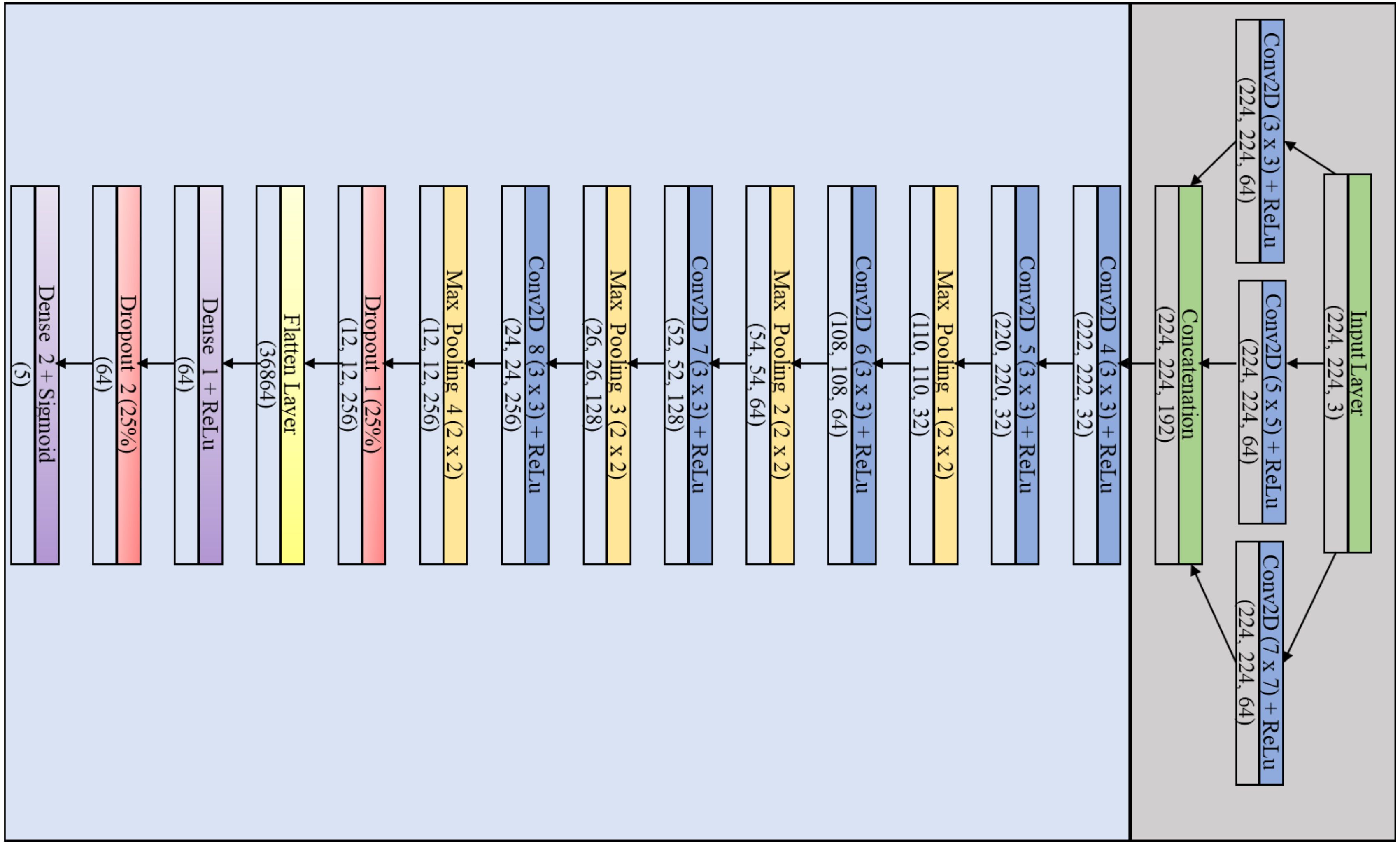


Fig 3.5: The detailed architecture of the proposed CXRNet model.

Table 3.4: Summary of the proposed CXRNet architecture with output shape and parameters for each layer

Layer (Type)	**Output Shape**	**Parameters**
input (Input)	224 × 224 × 3	0
model (Functional)	224 × 224 × 192	16128
conv2d_4 (Conv2D)	222 × 222 × 32	55328
conv2d_5 (Conv2D)	220 × 220 × 32	9248
max_pooling2d_1 (MaxPooling2D)	110 × 110 × 32	0
conv2d_6 (Conv2D)	108 × 108 × 64	18496
max_pooling2d_2 (MaxPooling2D)	54 × 54 × 64	0
conv2d_7 (Conv2D)	52 × 52 × 128	73856
max_pooling2d_3 (MaxPooling2D)	26 × 26 × 128	0
conv2d_8 (Conv2D)	24 × 24 × 256	295168
max_pooling2d_4 (MaxPooling2D)	12 × 12 × 256	0
dropout_1 (Dropout)	12 × 12 × 256	0
flatten (Flatten)	36864	0
dense_1 (Dense)	64	2359360
dropout_2 (Dropout)	64	0
output (Dense)	5	325
Total Params	2,827,909	
Trainable Params	2,827,909	
Non-Trainable Params	0	

The detail of the various layers and elements of the proposed CXRNet model are discussed below.

3.4.1 Input Layer

The input layer, which is the first layer of the proposed CXRNet model, has an input shape of (224, 224, 3), which corresponds to the shape of the input CXR image to be fed into the network.

3.4.2 Convolution Layer

The convolution process takes place in this layer, where the feature maps from the previous layer convolute with kernels and is passed through the ReLu activation function configuring the output feature maps. The mathematical expression for the convolution operation is given in Equation (3.5).

$$y_j^l = f\left(\sum_{i \in N_j} y_i^{l-1} * M_{ij}^l + a_j^l \right) \tag{3.5}$$

Where, y_j^l represents the *j^th^* output feature map of the *l^th^* layer, *f()* is a non-linear function, N_j is the input map selection, y_i^{l-1} is the i^{th} input map of $(l\text{-}1)^{th}$ layer, $M_{ij}^l$is the kernel for the input *i* and output map *M* in the j^{th} layer, and a_j^l is the bias associated with the j^{th} output map.

The convolution layer is also called the feature extraction layer as it extracts features using convolution filters whose parameters are learned through model training. Since the width and height of each kernel are configured to be less than the input feature maps, each neuron in the activation map is only linked to a limited local area of the input volume. In comparison to normal feed-forward layers, the convolution layers perform better for image input DL models because of the following reasons.

(i) **Parameter Sharing**: The filter parameters are shared for all local positions since the activation map is acquired by performing convolution operation between the filter and the input. This will reduce the number of parameters hence reducing the computation efforts and good generalization.

(ii) **Sparsity of Connections**: Each neuron in the activation map is only connected to a small local area of the input volume since the size of each kernel is configured to be smaller than the input.

(iii) **Translation Invariance**: The convolution operation is independent of the object feature position in an image.

In CXRNet architecture, there are eight different convolution layers of various kernel sizes and filter numbers (*N*). The early convolutional layers in the network process an image and come up with extracting low-level features like edges, while the convolutional layers deeper in the network can detect complex features such as corners, objects, etc. When an input image is subjected to a convolutional layer, each kernel is convolved across the width and height of the input image, giving a 2D feature map of that filter.

Hence, in this architecture, 32 feature maps are created for each parallel convolution layer when the input images are passed through the parallel convolutional layers. Then, the output of these parallel convolution layers is concatenated generating 96 feature maps. Similarly, corresponding feature maps are created as per the number of filters in respective convolutional layers.

3.4.3 Pooling Layer

In the CNN, the pooling layer is often used to downsample the outputs which in turn reduces the size, thus making the speedy computation, preventing overfitting, and reducing the memory requirement as well [65]. There are different types of pooling layers, including average pooling, max pooling, and sum pooling. Average pooling calculates the average for each patch of the feature map, max pooling selects the largest value from the feature map, and sum pooling calculates the sum of all feature map elements. Among them, max pooling is the most prevalent and frequently used pooling layer. This is so that the feature map containing the most notable features of the previous feature map can be summarized by the maxpooling procedure. In CXRNet, four max-pooling layers are used each after convolutional layers. These max-pooling layers are each of (2, 2) pool size and (2, 2) stride. By selecting the largest value from the input window of (2, 2) pool size for each channel of input, it downsamples the input along with spatial dimension (height and width) as represented in Fig. 3.6. Thus, the spatial shape of the resulting output from the max pooling layer is given in Equation (3.6).

$$\textit{output\_shape} = \textit{floor}((\textit{input\_shape} - \textit{pool\_size})/\textit{strides}) + 1 \tag{3.6}$$

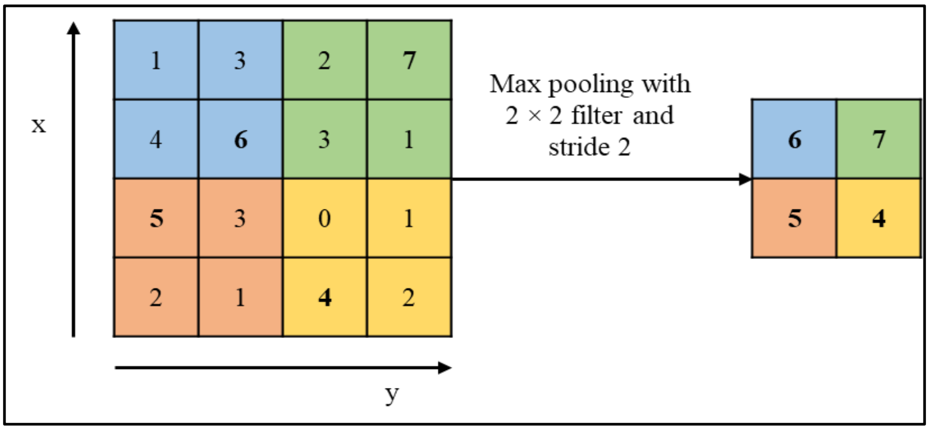


Fig. 3.6: Maxpooling operation illustration.

3.4.4 Dropout Layer

The term “dropout” describes dropping out units or ignoring units (i.e., neurons) at random in a neural network during the training phase. During each step of model training, this layer randomly sets input units to zero value with a set frequency. This will eliminate those nodes and remove all the incoming and outgoing links from that node resulting in a much diminished network. However, those inputs not set to zero are scaled up by 1/(1-rate), keeping the total sum of all inputs constant [66]. In this CNN architecture, two dropout layers are used with dropout rates of 25%. Thus, the random 25% inputs to these layers are eliminated at each step during model training. Also, the effect of the implementing dropout layers in the CXRNet performance is experimented and observed in this research work. The use of a dropout layer is one of the techniques that help to reduce overfitting issues during model training. This can be explained by the following reasoning.

(i) The overfitting issue may result due to assigning higher weight values for certain features thus losing the generalization capability for other input feature maps. However, using dropout, the NN doesn’t rely on any one feature map as the input can drop out at random. Thus, it will force the NN to spread out weight parameters enhancing the generalization capabilities and addressing the overfit issues.

(ii) The use of a dropout layer results in much smaller and diminished networks which correspond to the simpler network. Thus, the simpler network is inherently tolerant of overfitting issues.

3.4.5 Flatten Layer

The flatten layer is used to translate the multidimensional data value input into a single dimension. This is necessary before passing data into a fully connected dense layer. Thus, a flatten layer is used before the dense layer in this CNN architecture where the output from the previous layer of shape (12 × 12 × 256) is translated into a single dimension with a total of 36,864 nodes. The flattening operation of a two-dimensional input vector is shown in Fig 3.7.

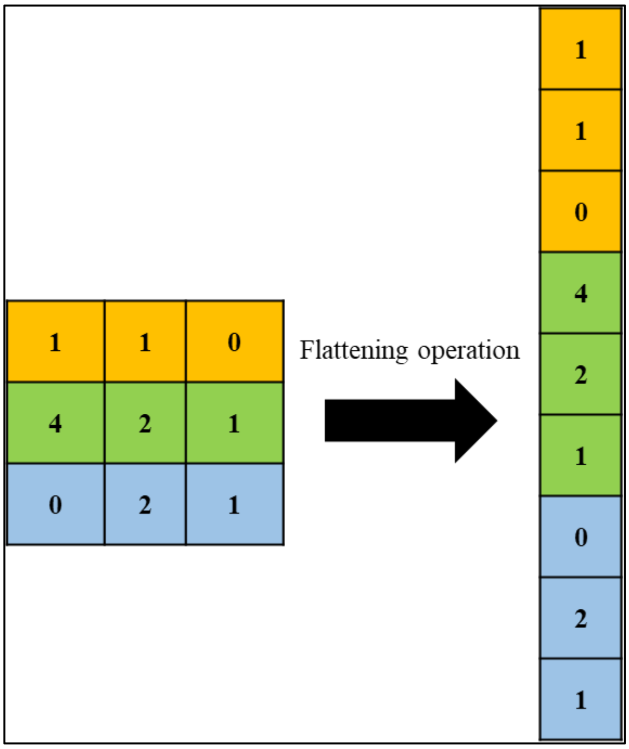


Fig. 3.7: The flattening operation of two-dimensional input vector.

3.4.6 Dense Layer

The dense layer is also known as the fully connected layer. It is a crucial part of CNNs, which has been proven to be extremely successful in recognizing and classification of input images. The operation in a typical CNN architecture begins with convolution and pooling layers, breaking down the image input into features. Then, the result is fed into a fully connected dense network that is responsible for the final classification decision task. based upon the features broken down from convolution and pooling layers.

A fully connected dense layer is simply a feed-forward neural network. The example of a fully connected dense layer is shown in Fig. 3.8. The operation of the dense layer is represented in Equation (3.7). In this research work, two dense layers are implemented after flatten layer, where the second dense layer is used for classification with five output classes for Cardiomegaly, COVID, Normal, Pneumonia, and Tuberculosis cases.

$$output=activation(dot(input,\ kernel)\ +bias) \tag{3.7}$$

Where, the *Kernel* is a weights matrix created by the layer, activation is the element-wise activation function supplied as the activation argument, and bias is a bias vector.

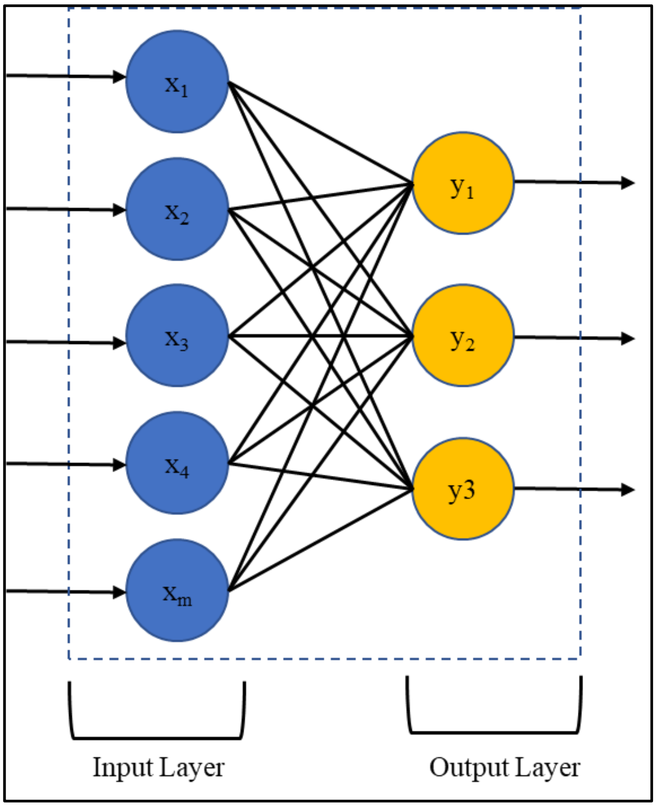


Fig. 3.8: An example of a fully connected layer in a deep neural network.

3.4.7 Activation Function

As real-world problems are non-linear, activation functions are used to introduce non-linearity in the model. Various forms of activation functions have been used to transform data into a non-linear form. However, in this model, two different types of activation functions are implemented which are discussed as follows.

3.4.7.1 ReLU

The Rectified Linear Unit is the most popular and commonly used activation function in hidden layers of a DL neural network architecture because of its faster and simplicity of computation thus speeding up the training process as well as preventing vanishing gradient problems in model training [67]. The ReLU function returns a zero value if it receives any negative input, otherwise will pass the input value unchanged. This operation is similar to the rectification action performed by a diode; hence, it is named accordingly. The mathematical representation for the ReLU activation function is represented in Equation (3.8) and the line plot is shown in Fig. 3.9.

$$f(x) = \max(0, x) \tag{3.8}$$

Where, *f*(*x*) is the output from ReLU, and *x* is the input value.

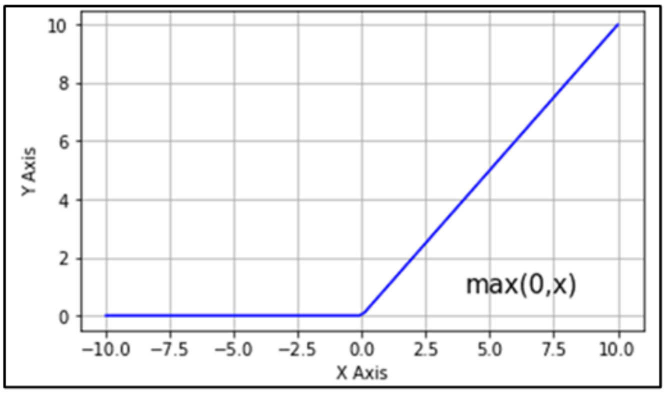


Fig. 3.9: The line plot of ReLU for negative and positive inputs.

3.4.7.2 Sigmoid

It is a type of activation function that maps the input between 0 and 1 values. However, this function can result in a vanishing gradient problem during model training. Hence, it is only used in the output layer only as it is suitable for predicting probability as an output. The mathematical representation for the sigmoid activation function is represented in Equation (3.9) and the line plot is shown in Fig. 3.10.

$$f(x) = 1/(1 + \exp(-x)) \tag{3.9}$$

Where, *f*(*x*) is the output from the sigmoid function, and *x* is the input.

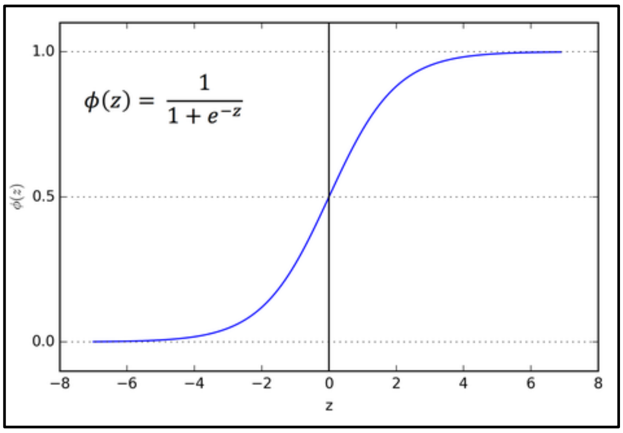


Fig. 3.10: The line plot of the sigmoid function for negative and positive inputs.

3.5 Model Training

In this research work, the balanced dataset is randomly distributed into training and testing sets in various ratios to observe the performance of the CXRNet in different training-testing distributions. In model training, the training dataset is used to learn the trainable parameters of the network. Hence, the majority of the total dataset volume is allocated for model training. Also, a portion of the testing dataset is used as a validation dataset to observe the overfitting issue in model performance during the training process. The use of validation datasets is very important in training large DL neural networks where a large amount of time is required for the training process. Hence, the validation dataset ensures whether the model training is suffering from the overfitting issue or not during training as the model with the overfitting problem will not be able to perform well in the testing dataset. During the training process, if the training accuracy is increasing while there is decreased or no improvement in validation accuracy, then, it is a clear indication of overfitting. In such a situation, the use of a validation dataset provides an early indication to improve the model architecture without the need to wait for the completion of the training process and hence, saving valuable time.

In this research work, the CXRNet model was trained for 50 epochs where the model is iterated over mini-batches of batch-size 32 images. One epoch corresponds when all the training data are passed through the model once and hence the number of iterations required for completing one epoch of training is given by Equation (3.10). The various hyperparameters set for the training process are listed in Table 3.5. These parameters play a vital role in the model training process and the performance of the model. The choice and tuning of these parameters are explained later in this section.

$$iterations\ (i) = length(training\_dataset)/batch\_size \tag{3.10}$$

Table 3.5: The hyper-parameters of the proposed CXRNet model

Hyper-parameters	**Value**
Optimizer	Adam
Loss Function	Sparse Categorical Cross Entropy
Initial Learning Rate	0.001
No. of Epochs	50
Batch Size	32

The overall training process of the proposed CNN model can be summarized in two basic operations which are discussed below.

3.5.1 Forward Propagation

During this process, the pre-processed images in a batch size of 32 images are fed into the input layer of CNN architecture which is then processed in various layers computing outputs in each layer. The output from any node in a layer is the result of two operations where (a) first there is a linear operation between the previous layer input, weights of the layer, and the associated bias for the layer, (b) then there is a non-linear operation by the activation function in each node. In this case, the output from the previous layer is fed as input to the next layer which shall continue until the output of the last dense layer is computed. Thus, the forward propagation process starts with inputting the image batch in the input layer and completes with generating output from the last dense layer which is the result of the sigmoid activation function. This takes place for each iteration in every epoch until the training process is completed.

3.5.2 Backpropagation

Backpropagation is the process of backward propagation of errors in supervised learning of DL neural networks using various optimization algorithms. For a given error function in an ANN, backpropagation computes the gradient of the error function with respect to the network’s weight and bias. As the name suggests, the gradients computation is performed backward across the network, starting with the gradients of the final layer of weights and bias and ending with the gradients of the first layer of weights and bias. Also, a portion of the gradient computation from one layer is used when computing the gradient for the previous layer. As a result, the backward flow of error information enables for efficient computation of gradient at each layer. The computed gradients are then used to update the parameters of the corresponding layer of the network thus optimizing the error function towards minima.

In this model, the outputs from the last layer which is the output of the sigmoid activation function are used to compute the loss function. It is a multi-class model with five output nodes where the output class labels are integer encoded as (a) Cardiomegaly ‘0’, (b)

COVID ‘1’, (c) Normal ‘2’, (d) Pneumonia ‘3’, and (e) Tuberculosis ‘4’ for the computation of sparse categorical cross-entropy. The loss function for a single image is represented in Equation (3.11).

$$L_i = -\log\left(\frac{e^{y_i}}{\sum_j e^{s_j}}\right) \tag{3.11}$$

Where, y_i is the correct image label for input image, s_j is the j^{th} component of the output vector from output layer for j = 0, 1, 2, 3, 4

Again, for a single iteration with a batch size of N = 32, the total loss is represented in Equation (3.12).

$$L = \frac{1}{N}\sum_i L_i \tag{3.12}$$

Where, i = 0, 1, ……., 31

Once the total loss function is computed for an iteration, then the goal of the optimization algorithm is to find the parameters (weights and bias) based on its gradients to minimize this loss function. This task is implemented in this model with the Adam optimization algorithm which is a stochastic gradient descent technique based on adaptive estimation of the first-order and second-order moments. Hence, in this architecture, the weights and biases are updated for each iteration.

3.6 Adam Optimization Algorithm

It is the most popular optimization algorithm which has become common in most DL applications. It is an extension of the stochastic gradient descent technique designed specifically for training DL neural networks. In addition, it is computationally efficient, invariant to diagonal rescaling of gradients, has little memory requirement, and is well suited for larger problems in terms of data and parameters [68]. It is a combination of RMSprop and gradient descent with a momentum optimization algorithm. Adam uses estimations of first and second-order moments of the gradient to adapt the learning rate for each weight of the neural network and hence update network parameters in each iteration during the training process. The mathematical representation for first-order moment and the second-order moment is given in Equation (3.13), Equation (3.14), Equation (3.15), and Equation (3.16).

$$V_{dw} = \beta_1 * V_{dw} + (1 - \beta_1)dw \tag{3.13}$$

Where, V_{dw} is the first-order moment or moving average of the gradient of weight parameter, β_1 is the first-order moment constant (default value = 0.9), and dw is the gradient of the loss function in Equation (3.12) with respect to weight.

$$V_{db} = \beta_1 * V_{db} + (1 - \beta_1)db \tag{3.14}$$

Where, V_{db} is the first-order moment or moving average of the gradient of the bias parameter, and db is the gradient of the loss function in Equation (3.12) with respect to bias.

$$S_{dw} = \beta_2 * S_{dw} + (1 - \beta_2)dw^2 \tag{3.15}$$

Where, S_{dw} is the second-order moment or moving average of the square of the gradient of the weight parameter, and β_2 is the second-order moment constant (default value = 0.999)

$$S_{db} = \beta_2 * S_{db} + (1 - \beta_2)db^2 \tag{3.16}$$

Where, S_{db} is the second-order moment or moving average of the square of the gradient of the bias parameter

Then, the correction in network parameters (weights and bias) in each iteration during the training process is represented in Equation (3.17) and Equation (3.18).

$$weight(new) = weight(old) - \alpha * \frac{V_{dw}}{\sqrt{S_{dw} + \in}} \tag{3.17}$$

Where, α is the learning rate, and $\in$ is a small constant (10^{-8}) to avoid dividing by zero condition.

$$bias(new) = bias(old) - \alpha * \frac{V_{db}}{\sqrt{S_{db} + \in}} \tag{3.18}$$

3.7 Improving and Optimization of CNN Network

The process of designing and training a deep neural network is not definitive and is completed after certain pre-defined steps but rather an iterative process that is repeated after certain changes are adjusted from the analysis of results in the previous iteration. There are several design considerations involving decision-making in the choice of various constraints while designing a CNN. This might include various constraints like the number of layers in the network, number of hidden units in a layer, learning rate, choice of activation functions, choice of optimization algorithms, size of the input layer, size of filters, number of filters, approaches to address bias/variance problem, etc. There is no mathematically established proven method for the choice of these considerations, however, it depends on the nature of the problem, the source and nature of data, and other various factors. In this research work, various previous works were reviewed and then a standard CNN architecture was first developed to address the limitations of previous works in the classification of input CXR images. This standard architecture was then experimented with by training the model. Then, the result or performance of the model was observed and analyzed which provided the basis for tuning the network to address the problem. This process was repeated until a satisfactory performance was achieved giving an efficient CXRNet architecture.

The various techniques that are implemented in this research work for the improvement and optimization of CNN architecture are discussed in this section.

3.7.1 Bias/Variance Problem

During the training of any DL models, the bias/variance problem is the most commonly observed problem which should be properly addressed to achieve an efficient model for real-world applications. In traditional ML applications, it was not possible to address bias or variance problems individually without offsetting the other one, hence, researchers worked on their tradeoff to achieve optimum results. However, with the introduction of DL architecture, there is no more tradeoff between bias and variance as one can be addressed without hurting the another. The bias/variance problem is defined based on three metrics: (a) Bayes error, (b) train error, and (c) validation error. The Bayes error is the lowest possible prediction error that can be achieved for any classifier and is equivalent to the irreducible error. In computer vision, NLP, and speech recognition applications, Bayes error is nearly equivalent to human-level error as humans are very good at natural perception tasks. In a DL model, it is said to have a high bias problem if the training error is large as compared to the Bayes error, and the gap between two errors is also called avoidable bias. If avoidable bias is large, it is necessary to spend time and resources on bias correction. On the other hand, a DL model is said to have a high variance problem if the validation error is large compared to the training error. The high variance problem is also an indication of overfitting of the model, which is performing well for the training dataset, however, is not able to perform well on other datasets that are not part of training the model. In such a situation, it is necessary to implement variance reduction techniques in the DL model. There are four different scenarios of bias/variance problems which are discussed below.

(i) High bias/high variance: Need for both bias correction and variance reduction techniques.

(ii) High bias/low variance: Need for only bias correction techniques.

(iii) Low bias/high variance: Need for only variance reduction techniques.

(iv) Low bias/low variance: Desired condition.

To address the bias/variance problem, it is necessary to know the bias correction and variance reduction techniques. In this research work, a working procedure was followed to address the bias-variance problem while designing an efficient CNN architecture which is shown in Fig. 3.11. The two basic goals of designing a DL neural network are (a) to optimize the cost function or loss in the network (reduce bias), and (b) to enhance the generalizability of the network (reduce variance or overfitting). In this regard, the working procedure was developed where different bias correction techniques and variance correction techniques are addressed during model training process. First, the training accuracy of the model was observed if there is a high bias problem. If there is high bias condition, then several bias correction techniques were implemented, and the process was repeated again and again finally producing high training accuracy. Once the high bias condition was addressed, then the process was repeated to address high variance condition. The developed working procedure is applicable not only in this research work but also can be of significant value in designing any DL networks.

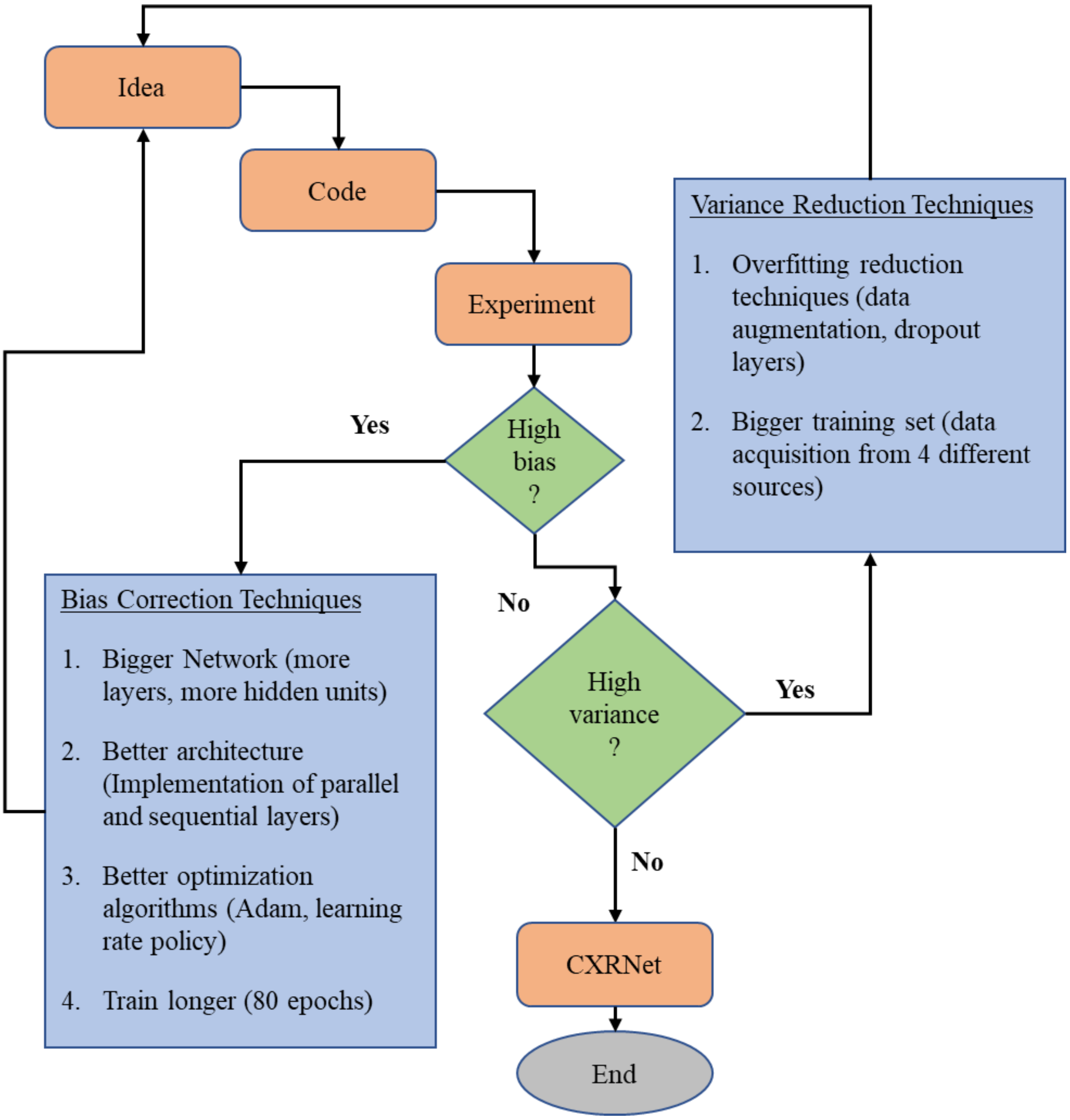

Idea
Code
Experiment
High bias ?
Yes
No
Bias Correction Techniques
1. Bigger Network (more layers, more hidden units)
2. Better architecture (Implementation of parallel and sequential layers)
3. Better optimization algorithms (Adam, learning rate policy)
4. Train longer (80 epochs)
High variance ?
Yes
No
Variance Reduction Techniques
1. Overfitting reduction techniques (data augmentation, dropout layers)
2. Bigger training set (data acquisition from 4 different sources)
CXRNet
End

Fig. 3.11: Working procedure to address bias/variance problem while designing proposed CXRNet architecture.

3.7.2 Input Shape

Image rescaling is one of the important pre-processing steps necessary to rescale the shape of the image as per the input shape of the CXRNet architecture. Hence, the choice of input image shape is a key factor that determines the performance and computational complexity of the model. Using smaller images will reduce the computational cost, however, there will be a loss in information resulting in poor model performance. On the other hand, using larger images will preserve the information, however, it will introduce computational costs (training time, memory requirement, etc.). However, increasing the image shape after a certain point will not result in much difference in model performance compared to added computational cost. Thus, finding the optimum input image shape is crucial in designing the optimum DL model. This can be achieved by training the model starting from a smaller input shape and then increasing the shape observing the performance until there is no significant increase in model performance. In this research work, the process was experimented with by training the model with an image shape starting from 126 × 126 dimensions. It was observed that by increasing the image shape beyond 224 × 224, there is no significant improvement in training performance, hence it is implemented as an input shape for the network. Also, all the CXR images are then reshaped into this shape as the images are originally of different dimensions.

3.7.3 Model Checkpoint

In this research work, a model checkpoint was implemented using the Keras Callbacks library to save the parameters of the CXRNet for the best model performance during the training process. The validation accuracy metric was monitored during each epoch to evaluate the best CXRNet model. During the training of the CXRNet for 50 epochs, the epoch producing the best validation accuracy was saved as the best CXRNet model. Later, the best CXRNet model was loaded for further evaluation of the performance of the model.

3.7.4 Early Stopping

In this research work, the early stopping technique was implemented using the Keras Callbacks library for early breaking of the model training if there is no improvement in validation accuracy by 0.01 step for 10 consecutive epochs. This was implemented to save time and resources during the training process.

3.7.5 Hyperparameters Tuning

Hyperparameters are parameters whose values control the learning process during model training and determine the values of model parameters (weights and bias) during the process. The values of hyperparameters are set before the model training and are said to be external to the model as the values are not changed during model training. Hence, tuning or optimizing the hyperparameter is testing a set of hyperparameter values and observing the output as the measurement of the model performance to discover the optimum values. In DL architecture, there is a number of hyperparameters that can be categorized as follows.

(i) Model hyperparameters: These hyperparameters determine the network structure.

- Size of the network (number of layers, number of hidden units, network topology)
- Kernel (size and type of the filter), etc.

(ii) Algorithm hyperparameters: These hyperparameters affect the speed and quality of the learning process.

- Learning rate
- Batch size
- Number of epochs, etc.

In traditional ML models, various hyperparameter tuning approaches, such as grid search, randomized search, and Bayesian optimization can be implemented for hyperparameter optimization. However, these approaches are less efficient in the case of the DL model with a high-dimensional space of hyperparameter numbers [69]. Hence, in this research work, the priority index for different hyperparameters was set up based on their impact on model performance during the training process. Then the corresponding policy was implemented for tuning hyperparameters with the highest priority index thus reducing the need for tuning the parameters with minimum control over the training process.

(i) The learning rate of the Adam optimization algorithm plays an important role during the model training process and is one of the hyperparameters with the highest priority index. It corresponds to the factor by which there is an update in the network parameters. Its value ranges between 0 and 1. Although a higher learning rate value ensures that there is a larger update in network parameters resulting in faster convergence of the error function, however, there is a high probability of oscillation in the error function as the error converges to minima resulting in no more convergence in the loss function. On the other hand, although a lower value ensures the convergence of the loss function, however, the change is minimal, and it will take a long time to converge the training process. Hence, the learning rate policy is implemented in this research work to tune the value of the learning rate during model training to speed up the training process and address the problem of plateaus in gradient descents. This is achieved by setting up the initial learning rate to a maximum value of 0.001 to speed up the training process. Then, the learning rate was reduced by a factor of ½ if there is no increase in validation accuracy for three consecutive epochs to address the issue of the plateau in gradient descent.

(ii) Model hyperparameters also play a vital role in model performance hence should be assigned a higher priority index in designing any DL architecture. In this work, model hyperparameters such as the number of hidden layers, and the number of hidden units are tuned as per Fig. 3.11 while other hyperparameters like kernel size {(3 × 3) for convolution filter, (2 × 2) for maxpooling}, choice of the activation function (ReLU in hidden layers, sigmoid in output layer), choice of optimization algorithms (Adam), etc. are set as per the standard practices for better model performance.

(iii) The batch size determines the number of input CXR images that are loaded in one forward pass to compute the parameters of the network through backpropagation in a single iteration. A good batch size depends upon the working system like RAM and GPU size as the data is loaded into the memory. Smaller batch size represents a smaller portion of data; hence the learning curve is not always convergence. However, it requires less memory, and the update of parameters is much faster. On the other hand, a larger batch represents a larger portion of data hence, it maximizes the convergence of the learning curve. However, it requires larger memory and the update on parameters is much slower. In practice, batch sizes in the power of 2 are implemented to utilize GPU processing capability at its fullest. In this research work, a batch size of 32 is implemented to exploit the optimum GPU usage and produce good learning curve convergence as well.

CHAPTER 4
RESULTS AND DISCUSSIONS

4.1 Introduction

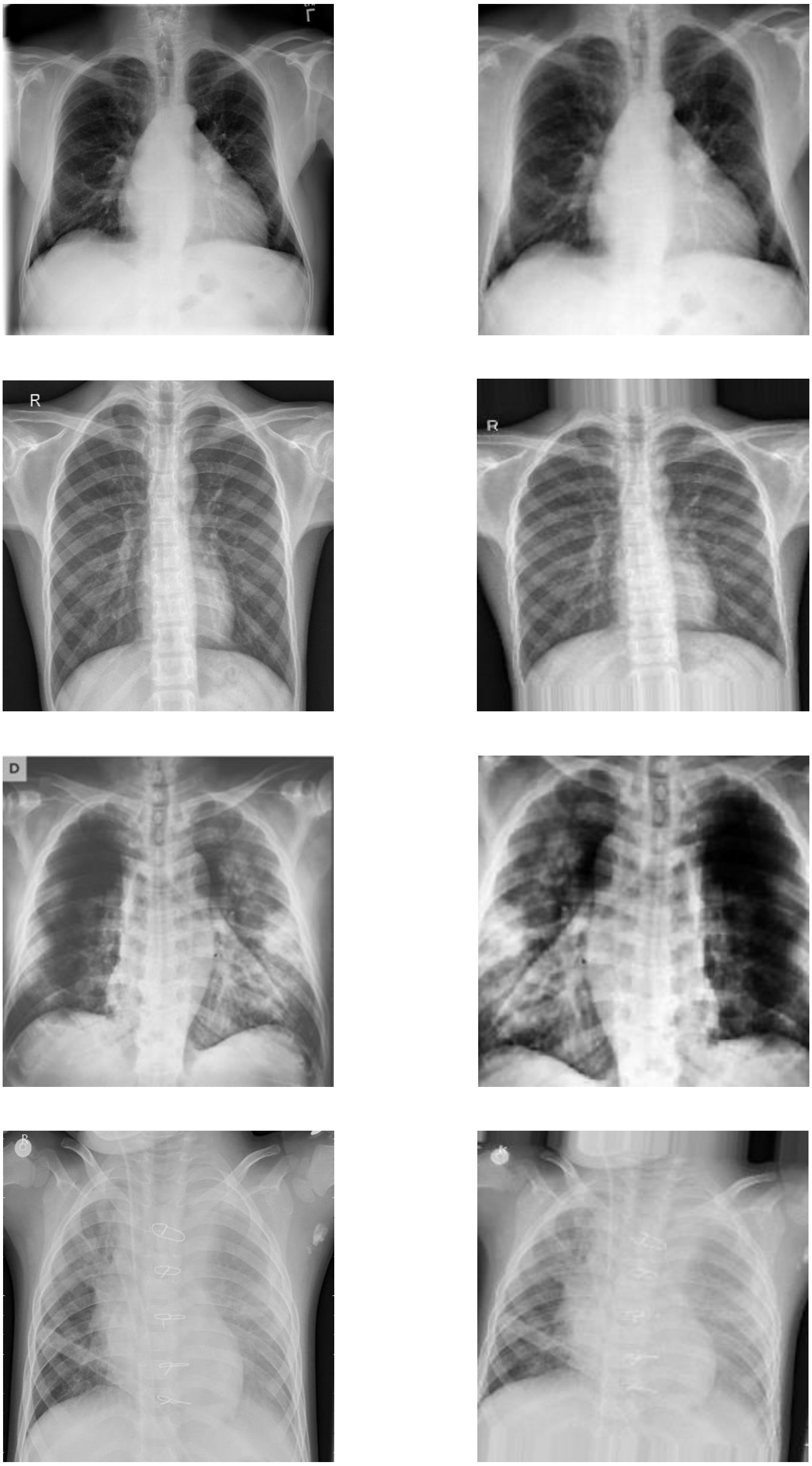
In this chapter, the experimental results obtained from the materials and methods described in Chapter 3 for the classification of input CXR images into five classes namely, cardiomegaly, COVID, normal, pneumonia, and tuberculosis are thoroughly analyzed and discussed. Thus, the results of various tasks performed in this research work are presented and discussed accordingly. Also, a set of evaluation metrics are defined, and the performance of the CXRNet as obtained from the training process is observed on the test datasets. In addition, it presents a comparative analysis of the performance of the proposed CXRNet with other related works. Finally, the strength and opportunities for the real-time adoption of the proposed model as a decision support system for clinical diagnosis of CXR images are presented exploring various challenges and limitations as well.

4.2 Results

This section presents the experimental results obtained from the various tasks performed in this research work as discussed in Chapter 3.

4.2.1 Input Pre-processing and Augmentation

In this research work, a set of image pre-processing techniques including data augmentation techniques as discussed in the previous chapter are implemented to prepare data before passing into the model. Once the images are loaded into the data loader, then the data generator generates the output images executing these pre-processing tasks. The train data generator is implemented in such a way that the generator shall generate new training images executing data augmentation techniques for every next epoch during model training to enhance the generalizability of the model and overcome the overfitting problem as well. The same generator is also used to generate new images of minority classes to create a balanced dataset for this research work. However, the test data generator only implements rescaling and resizing imaging pre-processing techniques to evaluate the performance of the model in real-world datasets without any data augmentation tasks. The samples of images before and after implementing image pre-processing and augmentation tasks are represented in Fig. 4.1.



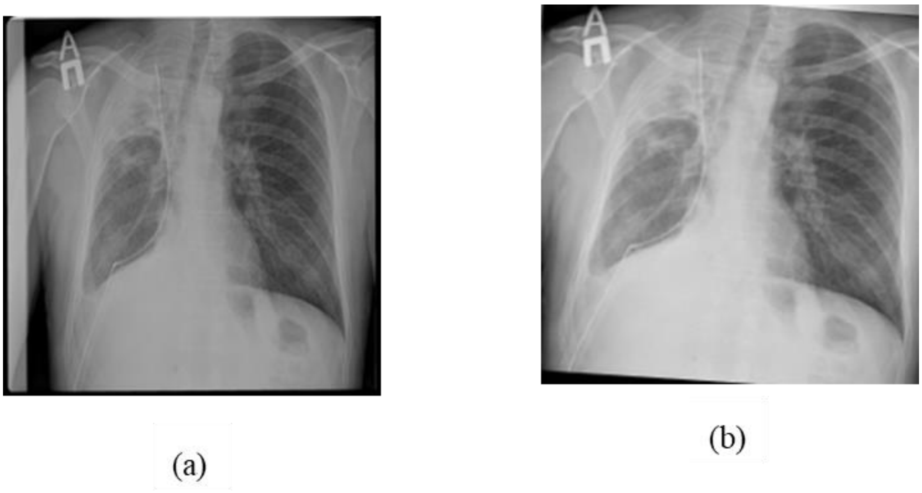


Fig. 4.1: The samples of images before and after image pre-processing and augmentation tasks, (a) before, and (b) after.

4.2.2 Model Training

In this research work, the balanced dataset is randomly distributed into training and testing sets in various ratios to observe the performance of the CXRNet in different training-testing combinations. The distribution of the dataset is represented in Table 4.1 where the training dataset is used for the model training while 50% of the testing dataset is used for model validation during the training process. There are a total of three different data distribution conditions where the model is trained and evaluated for each condition. The performance of the CXRNet is then evaluated on the testing dataset.

Table 4.1: The distribution of the dataset for various training-testing splits

Distribution	**Dataset**	**Labels**					**Total**
		(0)	**(1)**	**(2)**	**(3)**	**(4)**	
70% Train, 30% Test	Training	2,967	2,974	2,987	3,009	3,004	14,941
	Testing	1,240	1,250	1,238	1,264	1,196	6,188
80% Train, 20% Test	Training	3,398	3,386	3,404	3,402	3,381	16,971
	Testing	809	838	821	871	819	4,158
90% Train, 10% Test	Training	3,801	3,834	3,790	3,847	3,776	19,048
	Testing	406	390	435	426	424	2,081

In Fig. 4.2 and Fig 4.3, the learning process of the proposed CXRNet model is represented for the first case where the total dataset is randomly distributed into 70% for training and 30% for testing the model. In this figure, the graph of training set accuracy vs validation set accuracy as well as training loss vs validation loss is presented during model training for 50 epochs. In this case, the proposed CXRNet has achieved the highest training accuracy value of **97.67%** and the highest validation accuracy of **95.17%** during the model training process.

Similarly, in Fig. 4.4 and Fig. 4.5, the learning process of the proposed CXRNet model is represented for the second case where the total dataset is randomly distributed into 80% for training and 20% for testing the model. In this case, the proposed CXRNet has achieved the highest training accuracy value of **98.45%** and the highest validation accuracy of **96.15%** during the model training process.

Lastly, in Fig. 4.6 and Fig. 4.7, the learning process for the third case is represented where the total dataset is randomly distributed into 90% for training and 10% for testing the model. In this case, the proposed CXRNet has achieved the highest training accuracy value of **96.41%** and the highest validation accuracy of **96.2%** during the model training process.

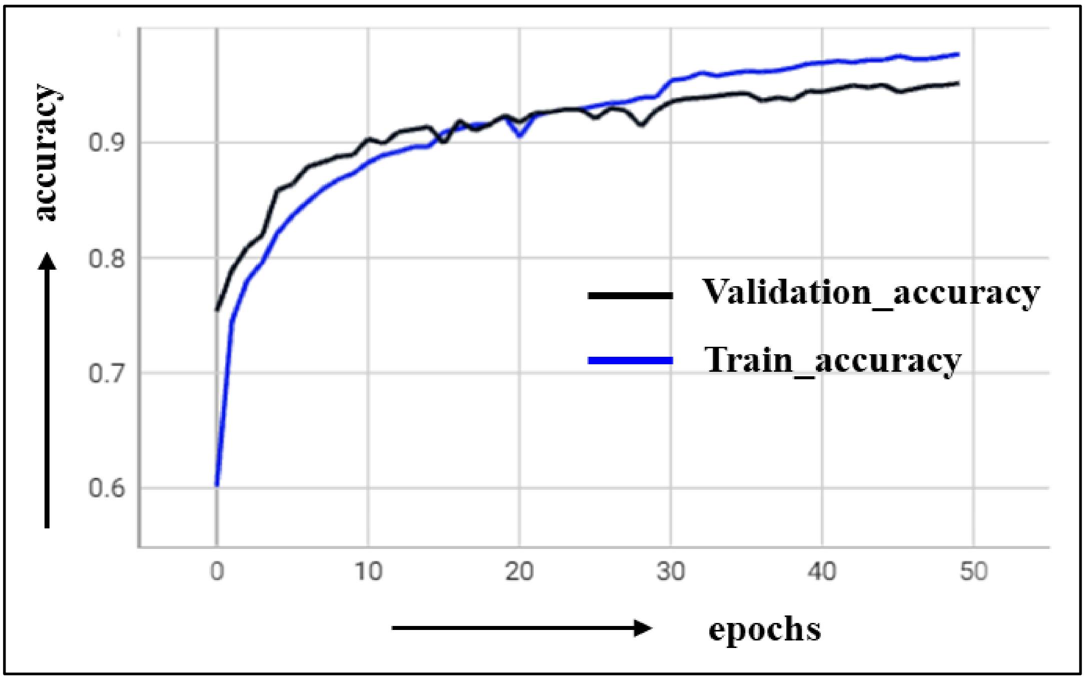


Fig. 4.2: Training accuracy vs validation accuracy curve of proposed CXRNet model for **70% - 30%** data split condition.

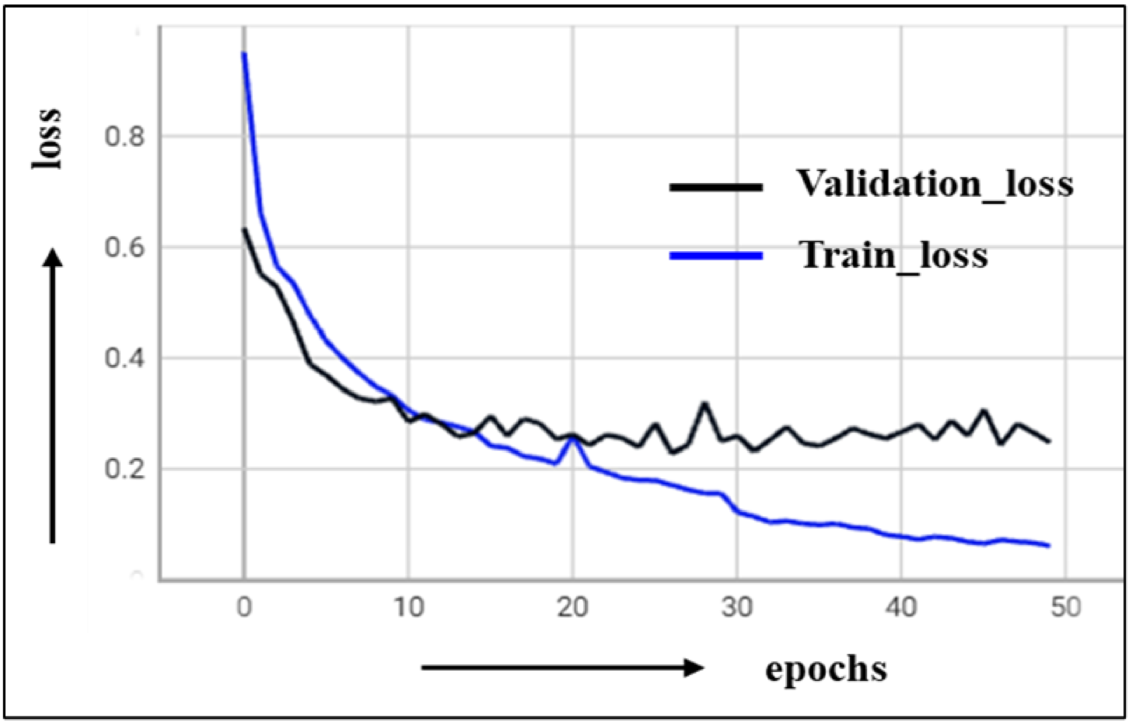


Fig. 4.3: Training loss vs validation loss curve of proposed CXRNet model for **70% - 30%** data split condition.

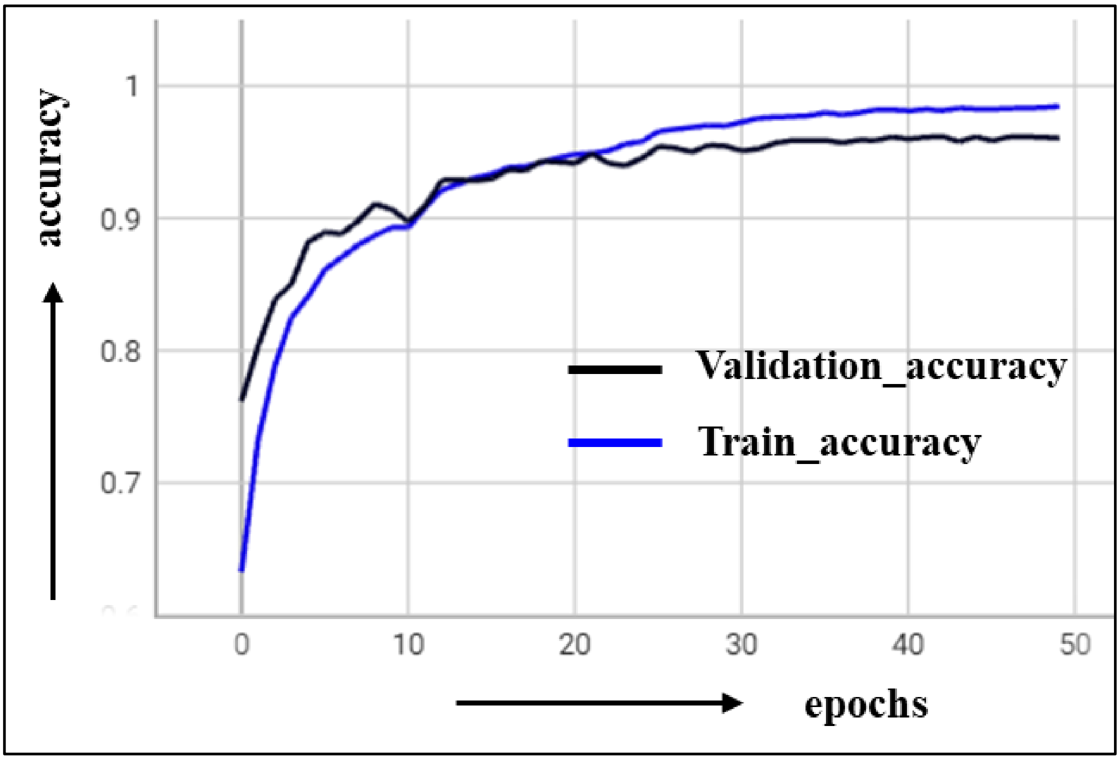


Fig. 4.4: Training accuracy vs validation accuracy curve of proposed CXRNet model for **80% - 20%** data split condition.

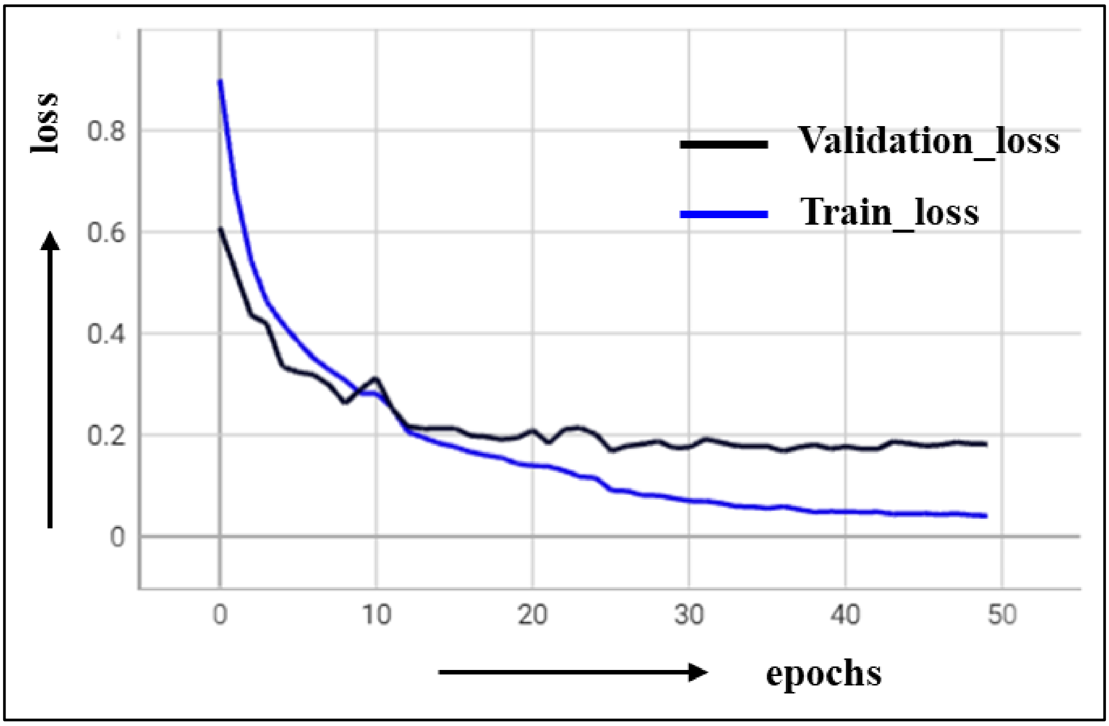


Fig. 4.5: Training loss vs validation loss curve of proposed CXRNet model for **80% - 20%** data split condition.

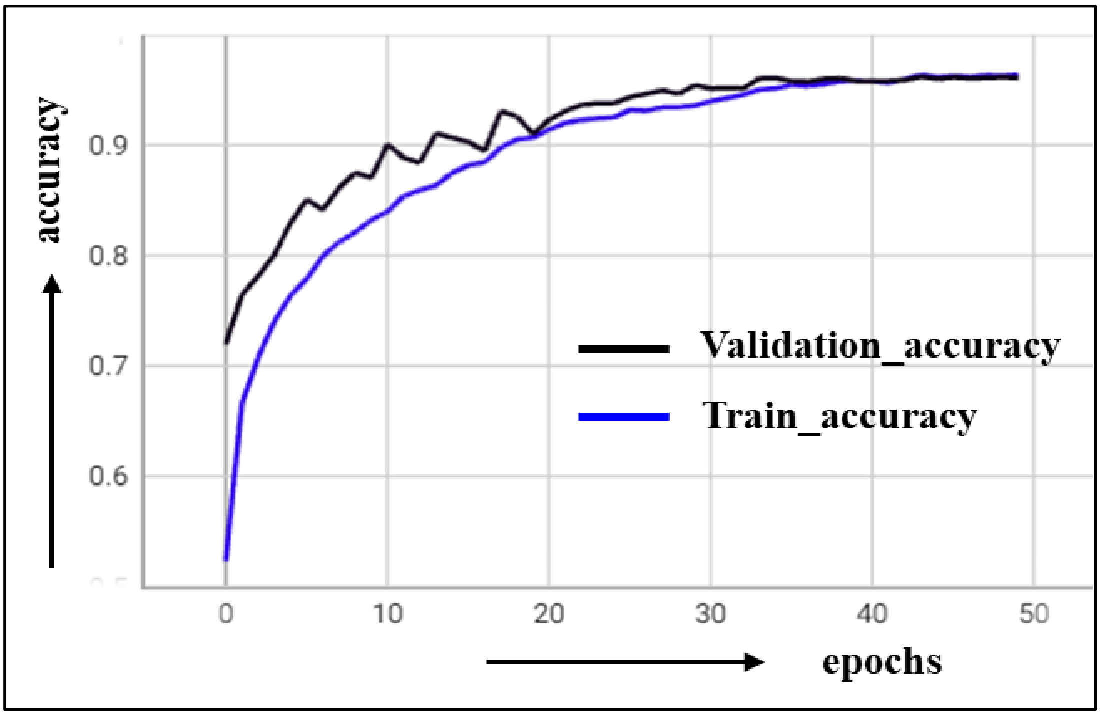


Fig. 4.6: Training accuracy vs validation accuracy curve of proposed CXRNet model for **90% - 10%** data split condition.

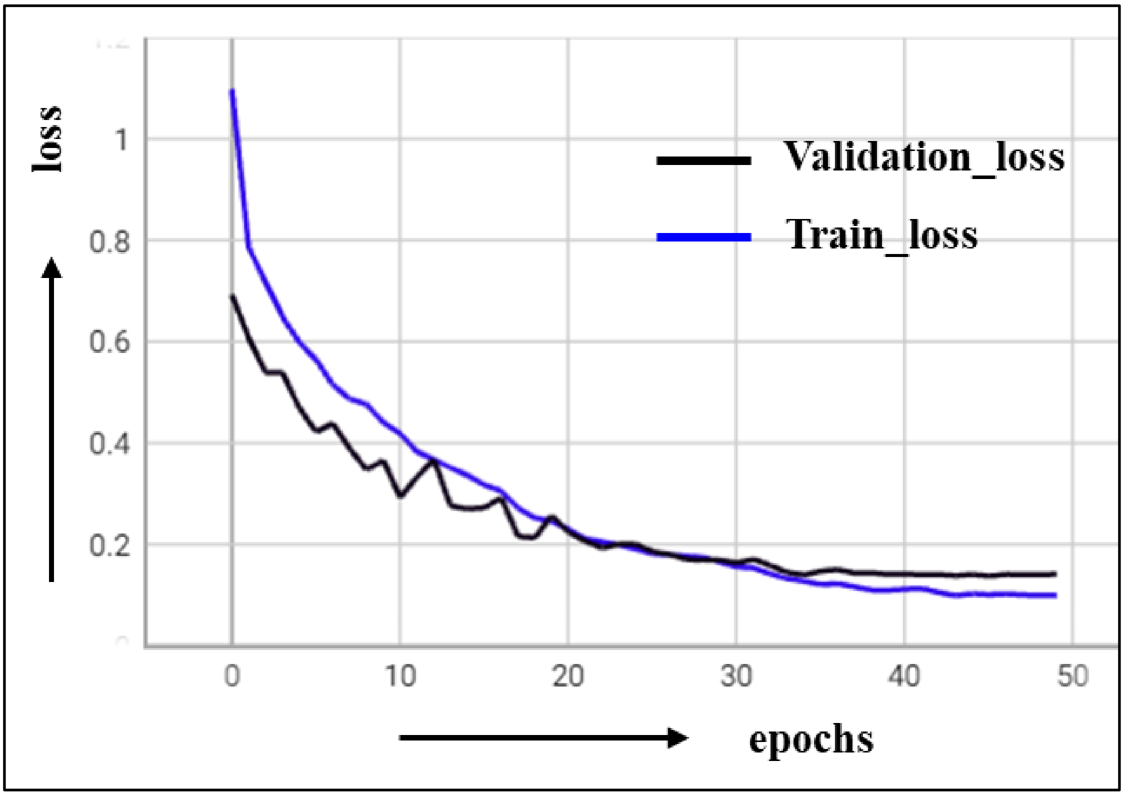


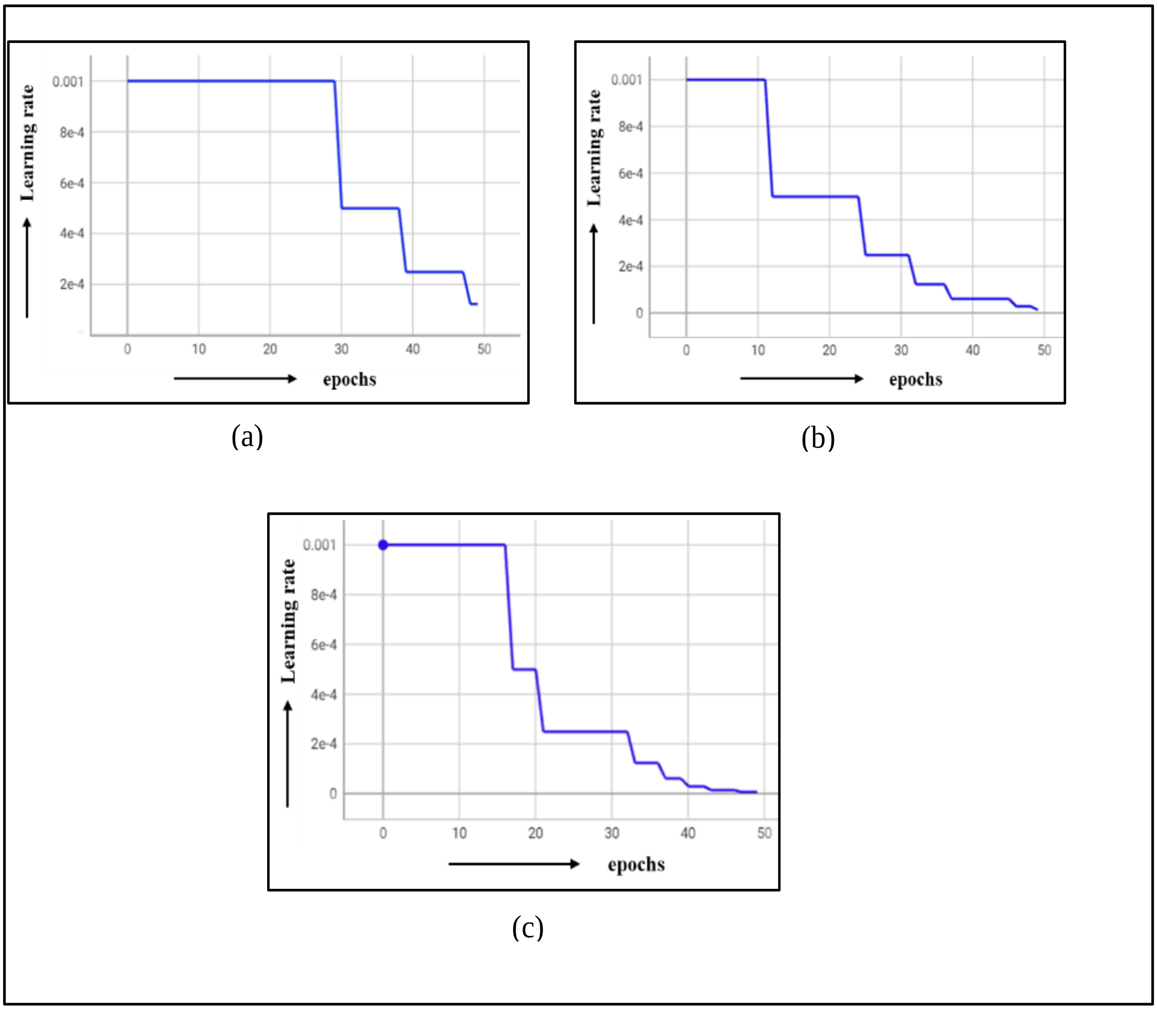
Fig. 4.7: Training loss vs validation loss curve of proposed CXRNet model for **90% - 10%** data split condition.

During the training of the proposed CXRNet model for three different dataset splits, the model achieved an average training accuracy of **97.51%** and an average validation accuracy of **95.84%** indicating the highest level of classification performance. The summary of the results is presented in Table 4.2. This result indicates that increasing the training dataset improves the performance of the model and the generalizability of the model in large data distribution. Also, there is a low bias and low variance in model performance which in turn supports the working procedure developed in this research work to address the bias/variance problem which is common in most of the relevant research works.

Table 4.2: The summary of the results from CXRNet training for different dataset splits

Dataset Split	**Training Accuracy (%)**	**Validation Accuracy (%)**
Case I (70% - 30%)	97.67	95.17
Case II (80% - 20%)	98.45	96.15
Case III (90% - 10%)	96.41	96.2
Average	**97.51**	**95.84**

The graph of the learning rate obtained during the training process for three different cases is presented in Fig. 4.8 which indicates that the learning rate policy has been implemented accordingly to address the problem of plateaus as well.



(a)

(b)

(c)

Fig. 4.8: Graph of learning rate values during model training, (a) 70% - 30% split, (b) 80% - 20% split, and (c) 90% - 10% split.

Also, the impact of dropout layer and data augmentation techniques are experimented with by training the model without applying these techniques. The observation of model performance during the training process is presented in Fig. 4.9 and Fig. 4.10. In these figures, it is observed that there is a huge difference between train and validation accuracy or between train and validation loss. Also, validation loss starts increasing after certain iterations indicating an overfitting problem in the model. These figures support that, the implementation of these techniques in this research work played a vital role to address the overfitting problem that is common in various DL models.

In this research work, to find the appropriate input shape of the proposed CXRNet, the model has been trained for several input shape conditions. The observations for three different input image shapes are shown in Fig. 4.11 where it is observed that increasing image shape beyond 224 × 224 doesn't significantly improve the training performance of the model which is then approved as an appropriate input shape of the CXRNet model.

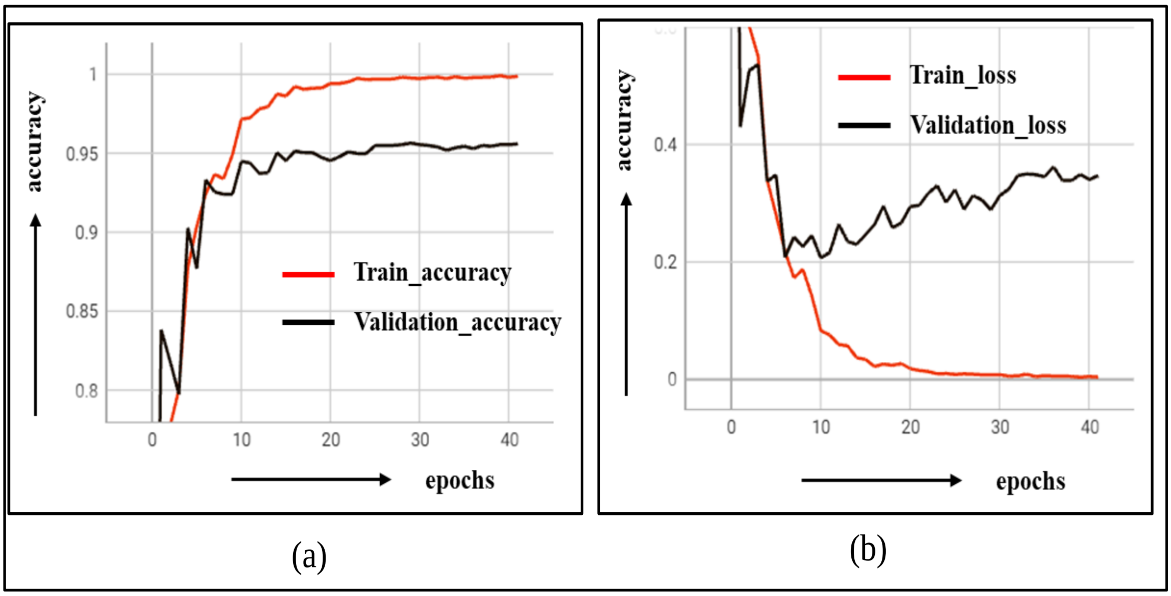


Fig. 4.9: Model performance without data augmentation techniques, (a) accuracy graph, and (b) loss graph.

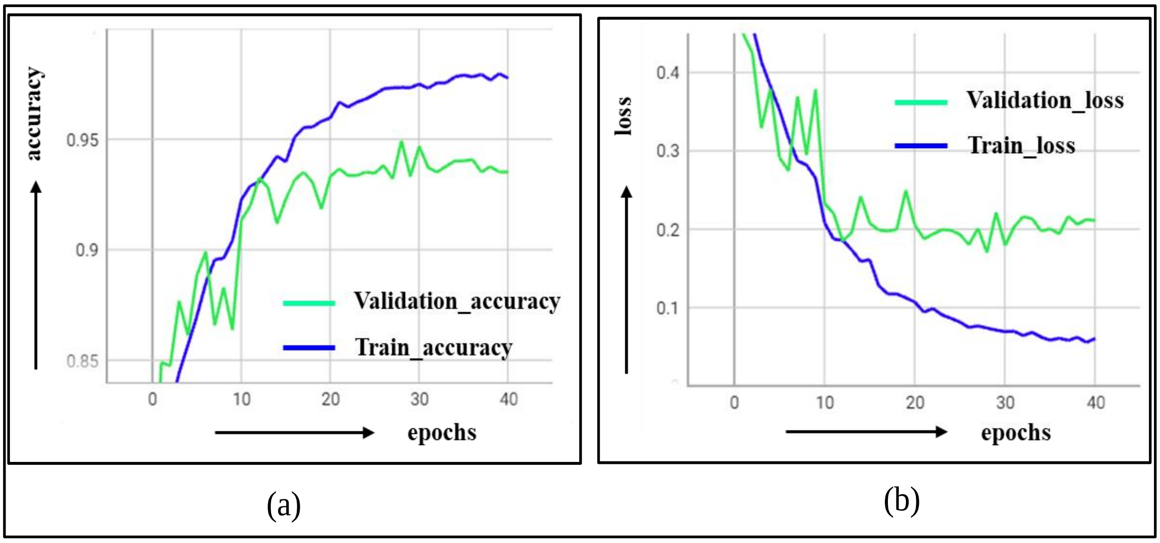


Fig. 4.10: Model performance without dropout layers, (a) accuracy graph, and (b) loss graph.

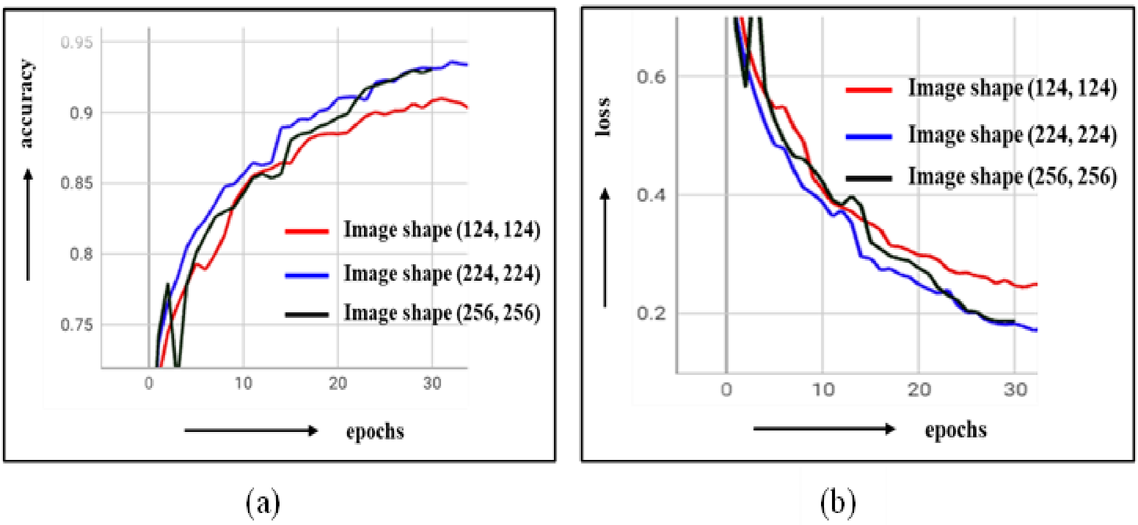


Fig. 4.11: Training performance of the model for different image shapes, (a) accuracy graph, and (b) loss graph.

4.3 Qualitative Evaluation

Once the model training process is completed, it is then necessary to evaluate the classification performance of the model in the test dataset. This is necessary to observe the generalizability of the model in diverse dataset environments and hence to validate the performance for real-time adoption of the model as a decision support system in clinical settings. For this, four common evaluation metrics namely, accuracy, recall, precision, and f1-score are employed to assess the proposed model. To evaluate these metrics, the following terms are defined first.

(i) True Positive (TP): It refers to an object corresponding to the positive class and also predicted to the same class by the model.

(ii) False Positive (FP): It refers to an object corresponding to the negative class, however, predicted as positive by the model.

(iii) True Negative (TN): It refers to an object corresponding to the negative class and also predicted to the same class by the model.

(iv) False Negative (FN): It refers to an object belonging to a positive class, however, predicted as negative by the model.

In correspondence to the above parameters, the four evaluation metrics are represented mathematically by Equations (4.1), (4.2), (4.3), and (4.4).

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{4.1}$$

$$Precision = TP/(TP + FP) \tag{4.2}$$

$$Recall = TP/(TP + FN) \tag{4.3}$$

$$F1 - score = \frac{2}{1/Precision + 1/Recall} \tag{4.4}$$

In this research work, the total dataset is randomly distributed into three different train-test splits, so the model performance is evaluated for three different conditions. The corresponding evaluation metrics and the confusion matrix for each case are also presented accordingly. These metrics represent the performance of the proposed CXRNet model in predicting the input images in one of five different classes. The evaluation metrics of the CXRNet on the test dataset from 70% - 30% data split condition is presented in Table 4.3 while the confusion matrix for this condition is shown in Fig. 4.12. Precision provides the accuracy of the model over total positive label predictions while recall measures the total correct true predictions by the model over total ground truth positives. The F1-score value is simply the harmonic mean of precision and recall value. In this case, an average accuracy of 95%, a precision of 95%, a recall of 95%, and an f1-score of 95% were achieved on the test dataset.

Table 4.3: Experimental performance metrics results of CXRNet on 70% - 30% split case

X-ray Category	Precision	Recall	F1-score	Support
Cardiomegaly	0.91	0.96	0.93	1240
COVID	0.95	0.94	0.95	1250
Normal	0.94	0.89	0.92	1238
Pneumonia	0.97	0.97	0.97	1264
Tuberculosis	0.98	0.98	0.98	1196
Average	**0.95**	**0.95**	**0.95**	**6188**

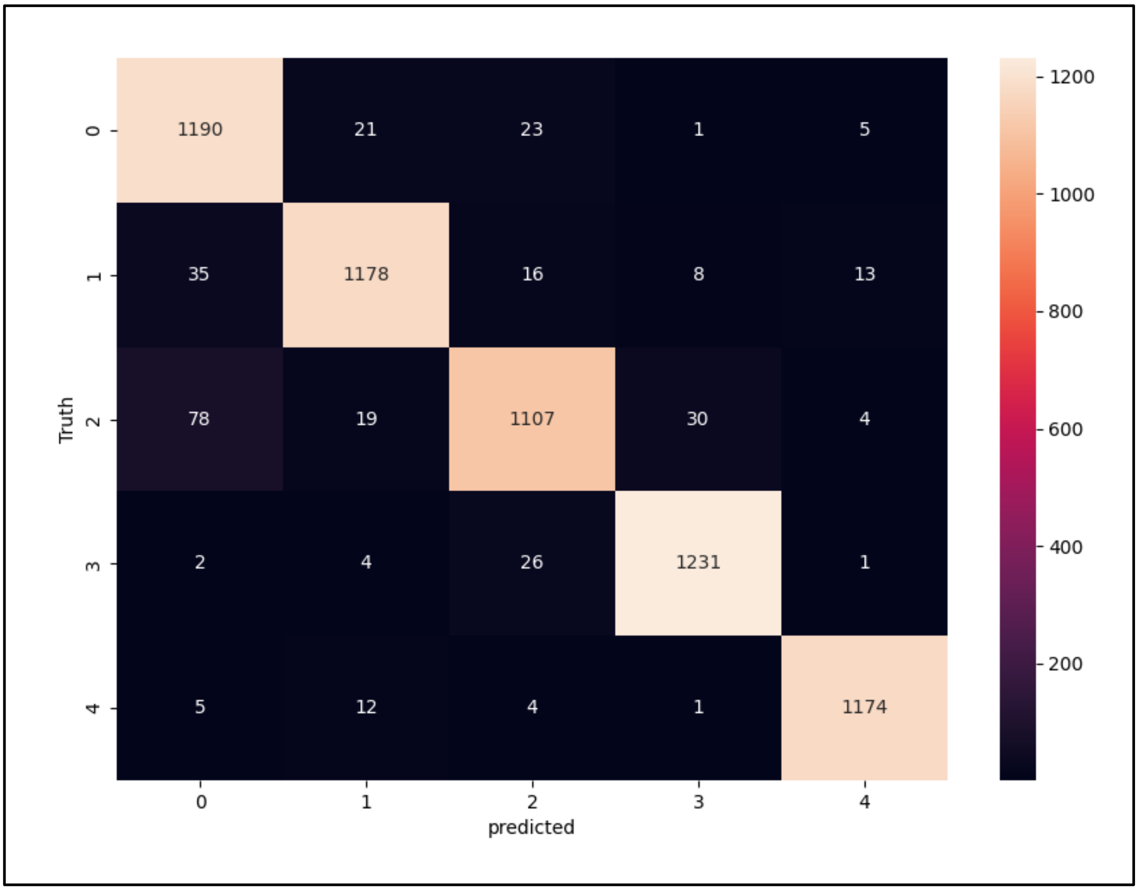


Fig. 4.12: The resulting confusion matrix of the CXRNet on 70% - 30% split case.

Similarly, the evaluation metrics of the CXRNet on the test dataset from 80% - 20% data split condition is presented in Table 4.4 while the confusion matrix for this condition is shown in Fig. 4.13. In this case, an average accuracy of 96%, a precision of 95%, a recall of 95%, and an f1-score of 95% were achieved on the test dataset.

Table 4.4: Experimental performance metrics results of CXRNet on 80% - 20% split case

X-ray Category	Precision	Recall	F1-score	Support
Cardiomegaly	0.95	0.97	0.96	809
COVID	0.95	0.96	0.96	838
Normal	0.87	0.93	0.90	821
Pneumonia	0.98	0.90	0.94	871
Tuberculosis	1.00	0.98	0.99	819
Average	**0.95**	**0.95**	**0.95**	**4158**

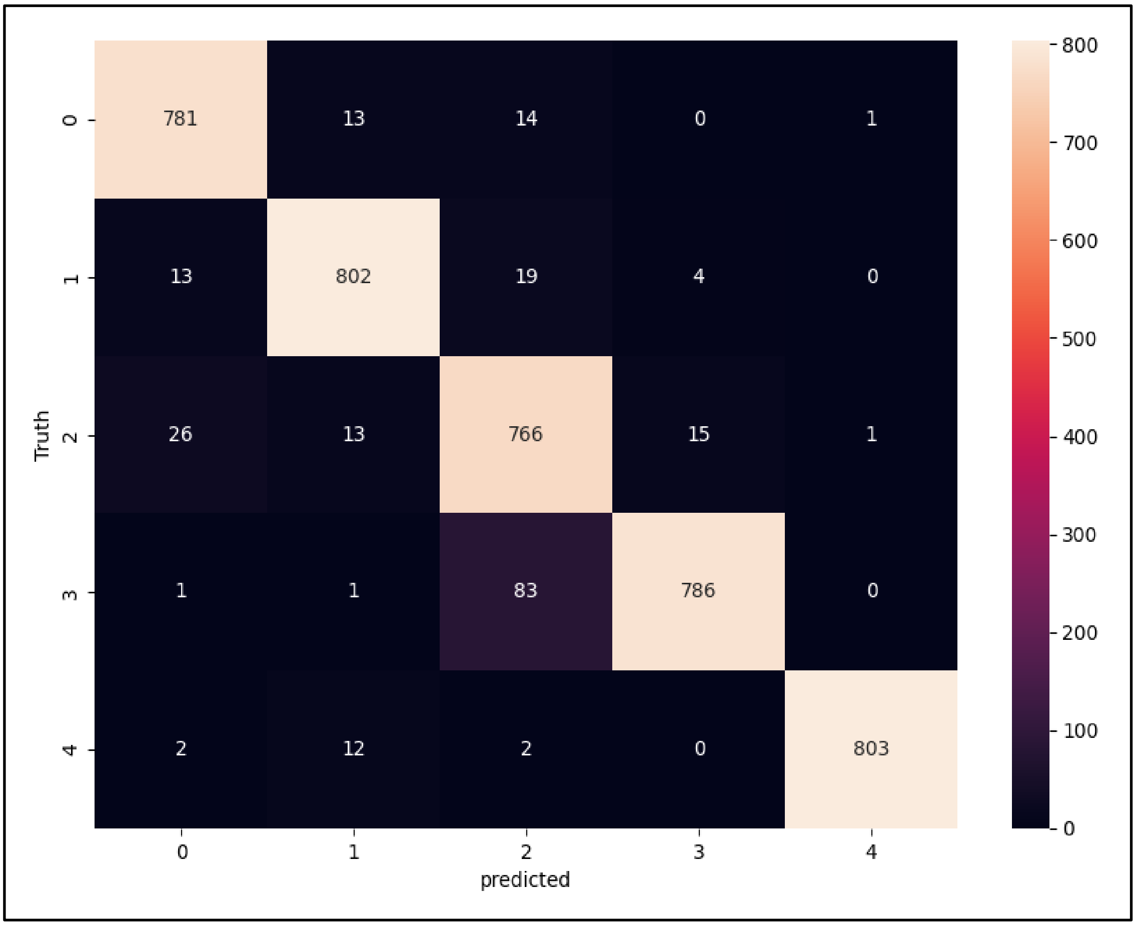


Fig. 4.13: The resulting confusion matrix of the CXRNet on 80% - 20% split case.

In the third case, the evaluation metrics of the CXRNet on the test dataset from 90% - 10% data split condition is presented in Table 4.5 while the confusion matrix for this condition is shown in Fig. 4.14. In this case, an average accuracy of 96%, a precision of 96%, a recall of 96%, and an f1-score of 96% were achieved on the test dataset.

Table 4.5: Experimental performance metrics results of CXRNet on 90% - 10% split case

X-ray Category	**Precision**	**Recall**	**F1-score**	**Support**
Cardiomegaly	0.95	0.96	0.95	406
COVID	0.96	0.95	0.96	390
Normal	0.95	0.93	0.94	435
Pneumonia	0.96	0.98	0.97	426
Tuberculosis	0.99	0.99	0.99	424
Average	**0.96**	**0.96**	**0.96**	**2081**

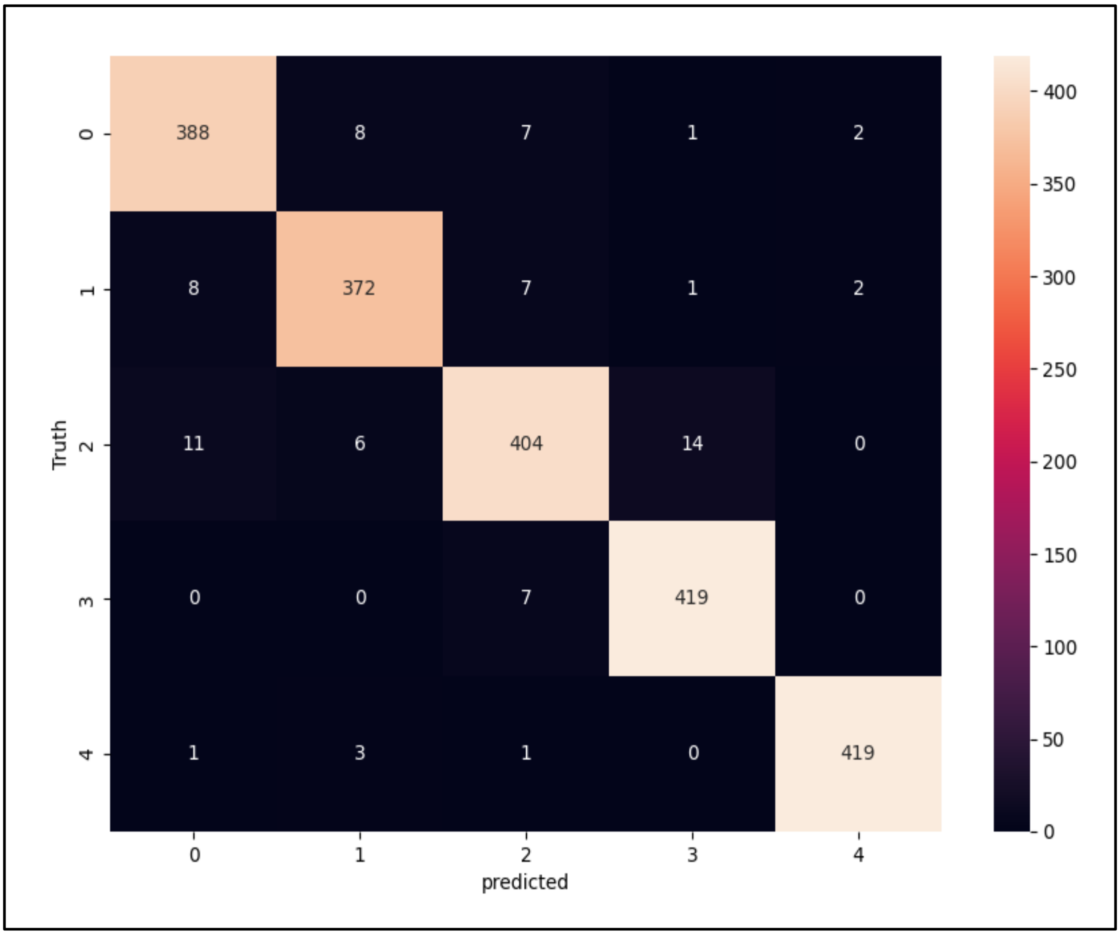


Fig. 4.14: The resulting confusion matrix of the CXRNet on 90% - 10% split case.

The summary of the evaluation metrics of the proposed CXRNet model on three different dataset split conditions is presented in Table 4.6. The model achieved an accuracy of 95.66%, a precision of 95.33%, a recall of 95.33%, and an f1-score of 95.33% on average indicating superior performance in comparison to other relevant multi-class classification research works.

Table 4.6: The summary of evaluation metrics of the CXRNet model

Dataset Split	**Precision (%)**	**Recall (%)**	**F1-score (%)**	**Accuracy (%)**
Case I (70% - 30%)	95	95	95	95
Case I (80% - 20%)	95	95	95	96
Case I (90% - 10%)	96	96	96	96
Average	**95.33**	**95.33**	**95.33**	**95.66**

Also, in this research work, the performance of the proposed CXRNet on binary classification tasks has been investigated as well. For this, the output layer of CXRNet for multi-class classifications is modified into two class classifications by substituting final dense layer with two output nodes and then the model is trained with one of the abnormalities case and normal case CXR images in 80% - 20% split conditions. The evaluation metrics are presented in Table 4.7 while the corresponding confusion matrix is shown in Fig. 4.15. The CXRNet is able to achieve an accuracy of 97.75%, a precision of 97.75%, a recall of 97.75%, and an f1-score of 97.75% on average indicating the superior performance of the model in binary classification tasks as well in comparison to other relevant research works.

Table 4.7: Experimental performance metrics results of CXRNet on the binary classification task

X-ray Category	Precision	Recall	F1-score	Support
A. Cardiomegaly vs Normal				
Cardiomegaly	0.94	0.98	0.96	809
Normal	0.98	0.94	0.96	821
Average	**0.96**	**0.96**	**0.96**	**1630**
B. COVID vs Normal				
COVID	0.97	0.99	0.98	838
Normal	0.99	0.97	0.98	821
Average	**0.98**	**0.98**	**0.98**	**1659**
C. Pneumonia vs Normal				
Pneumonia	0.98	0.97	0.98	821
Normal	0.98	0.98	0.98	871
Average	**0.98**	**0.98**	**0.98**	**1692**
D. Tuberculosis vs Normal				
Tuberculosis	0.99	0.99	0.99	821
Normal	0.99	0.99	0.99	819
Average	**0.99**	**0.99**	**0.99**	**1640**
Overall Average	**0.9775**	**0.9775**	**0.9775**	**6,621**

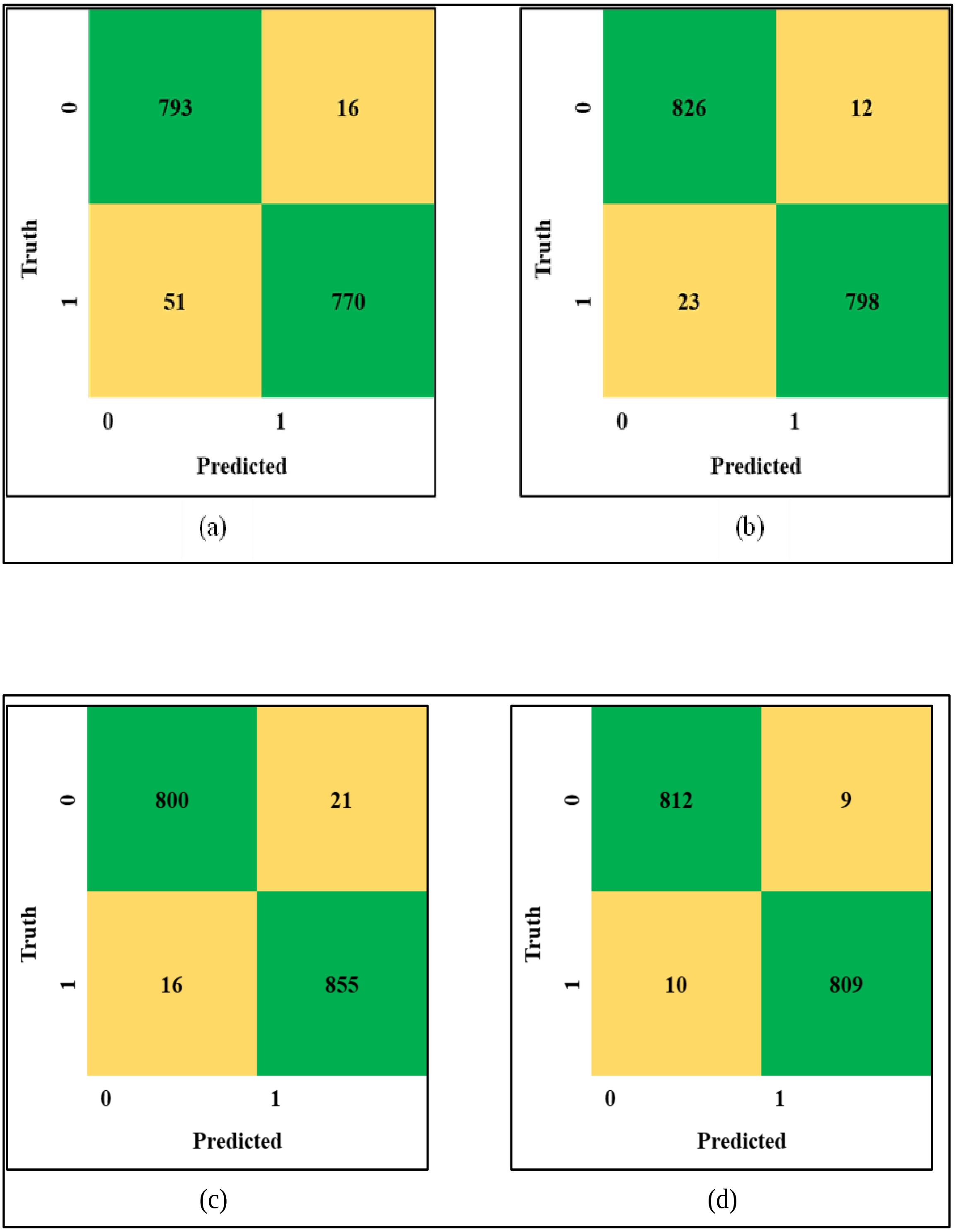


Fig. 4.15: The resulting confusion matrix of the CXRNet for binary classification task, (a) Cardiomegaly vs Normal, (b) COVID vs Normal, (c) Pneumonia vs Normal, and (d) Tuberculosis vs Normal.

4.4 Comparison with Related Works

After understanding the performance of the proposed model on test dataset, it is necessary to compare the performance of the model with other relevant research works for a better understanding of the overall outcome and contribution of this research in the related field. In addition, this outlines the limitations and challenges that are able to address by this research work in the existing relevant field and also discloses the new scopes for improvement as well. Hence, this section provides the comparative analysis on the performance of the proposed CXRNet model with other relevant model that are reviewed in Chapter 2. The comparative analysis of the performance of the proposed CXRNet model with the other relevant research works is presented in Table 4.8.

There are a large number of research works on automatic disease detection based on chest radiography using CNN. Most of these works are based on small-size datasets from single data sources and hence are limited to generalization capability of the model in diverse data source environments. Also, majority of the works are only a few class classification models with only two or three output classes. However, the limited findings of these models are not practical in real time application as a CXR image is used in diagnosis of multiple abnormalities.

Similarly, many of the research works made use of the transfer learning approach by using various pre-trained models for feature extraction. However, these pre-trained models are not only trained with the CXR dataset, hence, there is need for fine tuning with CXR dataset for effective performance. Hence, without fine tuning, transfer learning from the pre-trained model may involve negative transfer when the CNN is pre-trained on data not similar to a chest radiograph.

Hence, this research work addresses these discussed issues and challenges by developing a custom CNN architecture from scratch that can classify input chest radiographs into five different classes. At the same time, the model is trained across a diverse set of data source environment and the performance is validated on real time dataset from different hospitals thus offering the generalizability of the model in real time adoption as an efficient decision support system. Although, the performance of the model is superior to relevant research work, it is still not absolutely perfect.

Table 4.8: Comparison of the proposed CXRNet model with other related research works

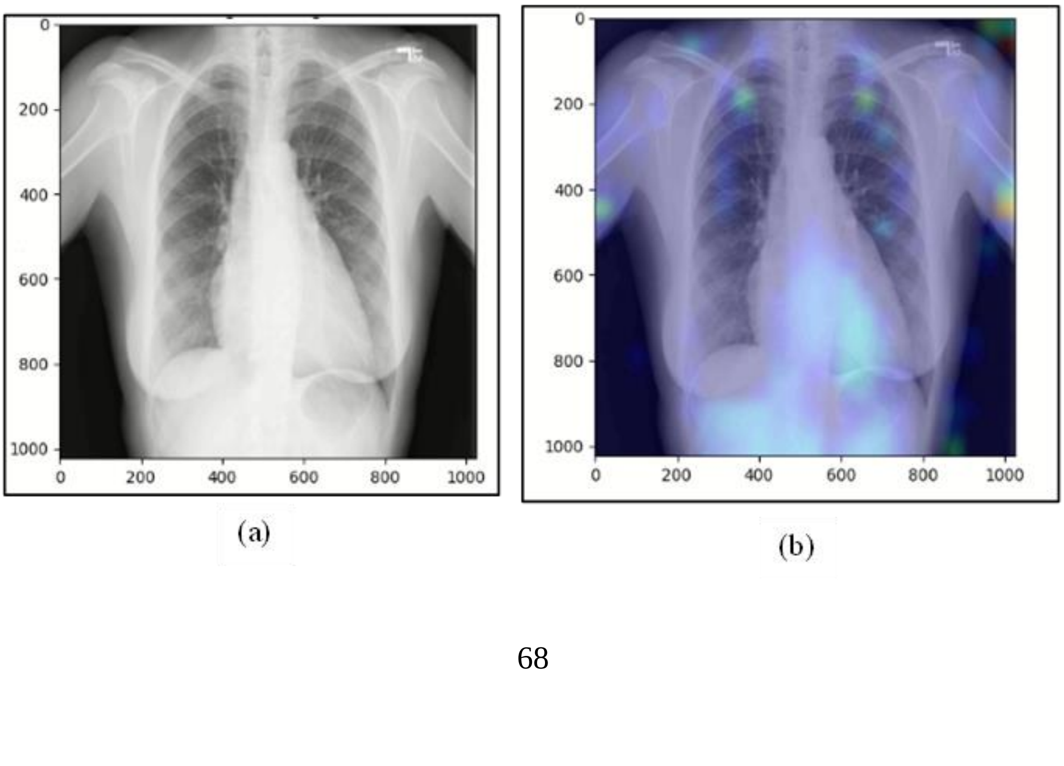
Research	**Dataset**	**Method Used**	**Output Classes**	**Precision(%)**	**Recall(%)**	**F1-Score(%)**	**Accuracy(%)**
A. Pneumonia Detection							
(O. Stephen et al., 2019)	5,856 CXR images	CNN from scratch	Two (Pneumonia vs Normal)	-	-	-	93.7
(R. Jain et al., 2020)	5,216 CXR images	CNN with pre-trained models (VGG16, VGG19, ResNet50, and Inception-v3)	Two (Pneumonia vs Normal)	97	90	93.4	92.3
(M. M. Eid et al., 2021)	5,863 CXR images	Pre-trained CNN(ResNet-50) with SVM	Two (Pneumonia vs Normal)	100	96.4	98.2	98.1
(O. A. Fagbuagun et al., 2022)	5,856 CXR images	Pre-trained CNN (Inception-V3)	Two (Pneumonia vs Normal)	90	85	87	88.1
(A. Mabrouk et al., 2022)	5,856 CXR images	Ensemble learning based on pre-trained CNN models (DenseNet169, MobileNetV2, and Vision Transformer)	Two (Pneumonia vs Normal)	93.7	93	93.4	93.9
(E. Ayan et al., 2022)	5,856 CXR images	Ensemble learning based on pre-trained CNN models (ResNet-50, Xception and MobileNet)	Two (Pneumonia vs Normal)	95.9	95.2	95.5	95.8
(M. W. Kusk et al., 2023)	5,856 CXR images	Custom sequential CNN for dataset at different noise levels	Two (Pneumonia vs Normal)	-	-	-	96.8 – 97.6
Proposed Work	**8,498 CXR images**	**CXRNet for binary classification**	**Two (Pneumonia vs Normal)**	**98**	**98**	**98**	**98**

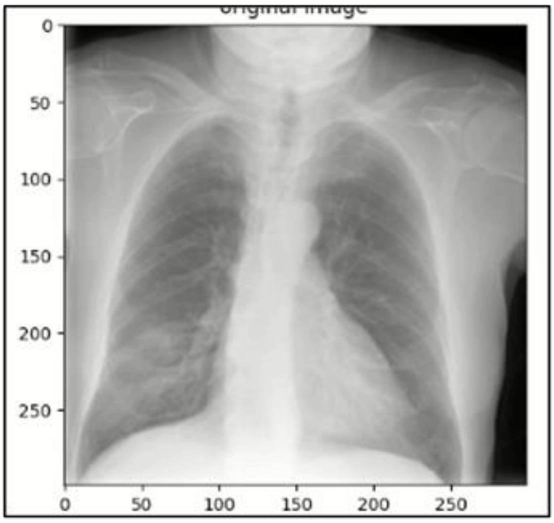
B. COVID Detection							
(T. Ozturk et al., 2020)	1,125 CXR images	DarkNet-19 model as a classifier for the YOLO real time object detection system	Two (COVID vs No findings)	89.9	85.3	87.3	87
(T. Rahman et al., 2021)	18,479 CXR images	U-Net model for lung segmentation and pre-trained model (ChexNet) for classification	Three (COVID vs non-COVID lung opacity vs Normal)	94.5	94.5	94.5	95.1
(J. L. Gayathri et al., 2022)	1,046 CXR images	Pre-trained CNN models (InceptionResnetV2 and Xception)	Two (COVID vs non-COVID)	95.6	95.6	95.6	95.8
(S. Sanket et al., 2022)	657 CXR images	CovCNN model based on deep-CNN architecture	Two (COVID vs non-COVID)	-	-	-	98.4
(A. Kumar et al., 2022)	13,975 CXR images	SARS-Net (custom-made DL architecture)	Three (Normal vs non-COVID vs COVID)	94.5	94.6	94.5	97.6
Proposed Work	**8,449 CXR images**	**CXRNet for binary classification**	**Two (COVID vs normal)**	**98**	**98**	**98**	**98**
C. Tuberculosis Detection							
(S. I. Nafisah et al., 2022)	1,098 CXR images	Pre-trained CNN models (EfficientNetB3)	Two (TB vs no-TB)	98.3	98.3	98.3	98.7
(V. Acharya et al., 2022)	3,500 CXR images	ImageNet model fine-tuned NFNets	Two (Healthy vs TB)	96.3	91.8	93.9	96
(E. Showkatian et al., 2022)	2,160 CXR images	ConvNet (CNN model from scratch)	Two (Healthy vs TB)	88	87	87	87

(S. S. Guia et al., 2023)	7,000 CXR images	Pre-trained CNN models (VGG16 and VGG19)	Two (TB vs normal)	-	-	-	99
Proposed Work	**8,425 CXR images**	**CXRNet for binary classification**	**Two (Tuberculosis vs normal)**	**99**	**99**	**99**	**99**
D. Multiple abnormalities Detection							
(A. I. Khan et al., 2020)	1,234 CXR images	CoroNet (model based on Xception pre-trained architecture)	Four (COVID vs Pneumonia bacterial vs Pneumonia viral vs normal)	90	89.29	89.8	89.6
(A. Musha et al., 2022)	3,788 CXR images	DL model based on YOLOv2 with residual network architecture	Three (COVID vs Pneumonia vs normal)	97.4	97.4	97.4	97.3
(F. Bayram et al., 2022)	3,886 CXR images	Fusion based DL model implemented by a multi-stream CNN	Three (COVID vs Pneumonia vs normal)	97.8	97.8	97.8	97.7
(M. Loey et al., 2022)	10,848 CXR images	Bayesian-based optimized DL model	Three (COVID vs Pneumonia vs normal)	96	96	96	96
(I. Kanjanasurat et al., 2023)	16,210 images (9,271 CXR & 6,939 CT)	Combination of CNN and RNN where CNN used for feature extraction and RNN for classification	Three (COVID vs Pneumonia vs normal)	93.7	93.4	93.5	93.4
Proposed Work	**21,129 CXR images**	**CXRNet (CNN architecture from scratch)**	**Five**	**95.3**	**95.3**	**95.3**	**95.7**

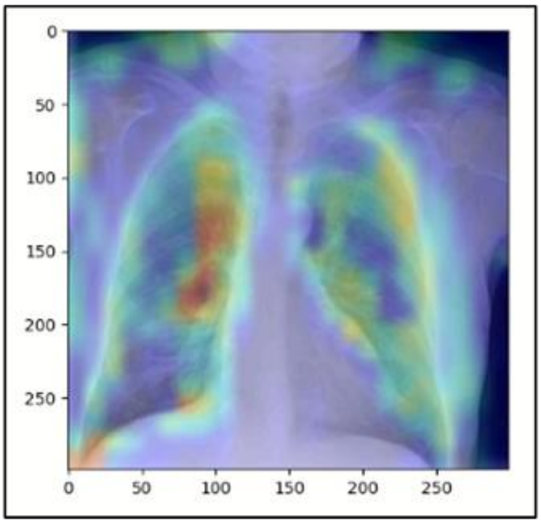
4.5 CXRNet as a Clinical Diagnosis and Decision Support System

The performance of the DL model is always compared to the human-level performance. If an experienced radiologist with 20 years of experience can classify the test dataset with 97% or higher accuracy, then another radiologist with just 5 years of experience may classify the same dataset with 92% or less accuracy. Thus, human performance is always subjected to diagnostic level and other various factors as well. On the other hand, there is a large number of people who do not have access to medical imaging expertise even today. At the same time, even in countries with highly advanced healthcare systems, radiograph interpretation demands a radiologist's time and expertise. However, experienced radiologists are still subjected to human limitations like fatigue and various biases leading to errors during radiograph interpretation. In such a situation, the results presented in this research work demonstrate that a CNN can be used to successfully detect diseases in chest radiographs at a level comparable to or even higher than practicing radiologists. This research work has developed the CXRNet model that can classify input chest radiographs into one of five classes namely Cardiomegaly, COVID, Normal, Pneumonia, and Tuberculosis offering an adequate number of findings for real-time implementation as well. In addition, this work also implemented the output visual aid generating the activation maps depicting discriminative areas from the last convolution layer of the proposed CNN architecture for each class classification. In Fig. 4.16, the input original images and their corresponding activation map output from the last convolution layer of this proposed CNN model are presented.

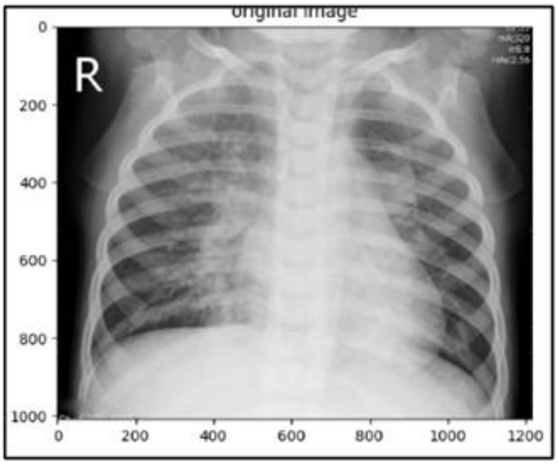



original image
0
50
100
150
200
250
0
50
100
150
200
250

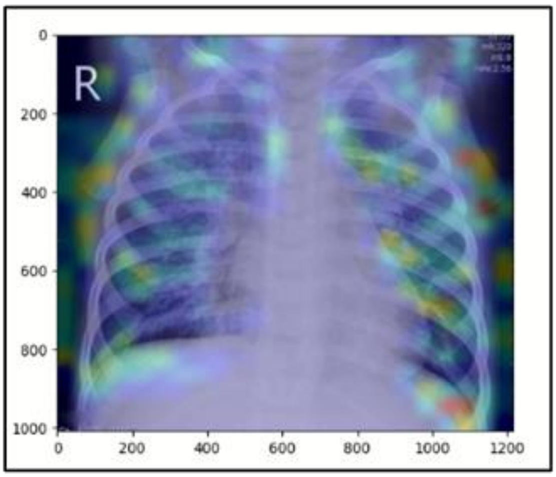
(c)


0
50
100
150
200
250
0
50
100
150
200
250

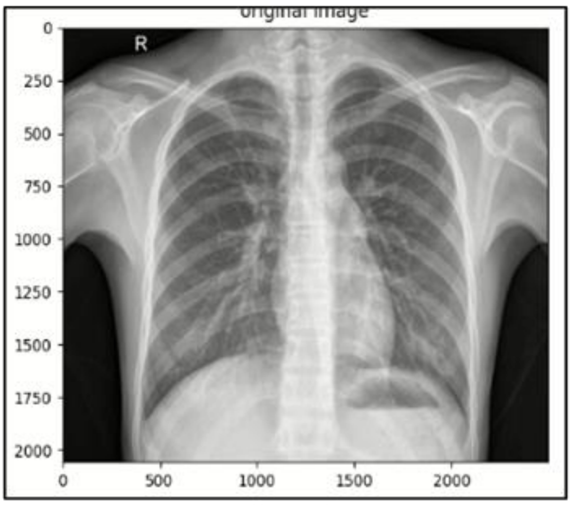
(d)


original image
R
0
200
400
600
800
1000
0
200
400
600
800
1000
1200

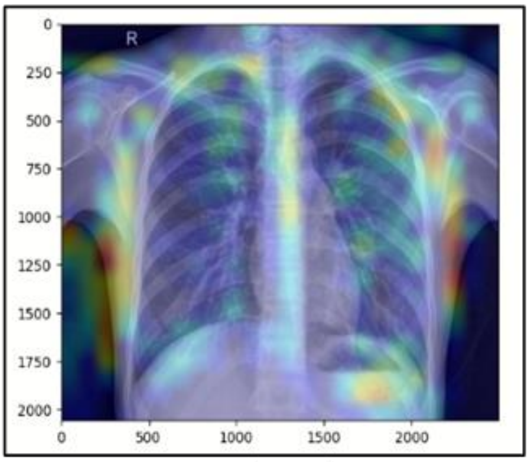
(g)


R
0
200
400
600
800
1000
0
200
400
600
800
1000
1200

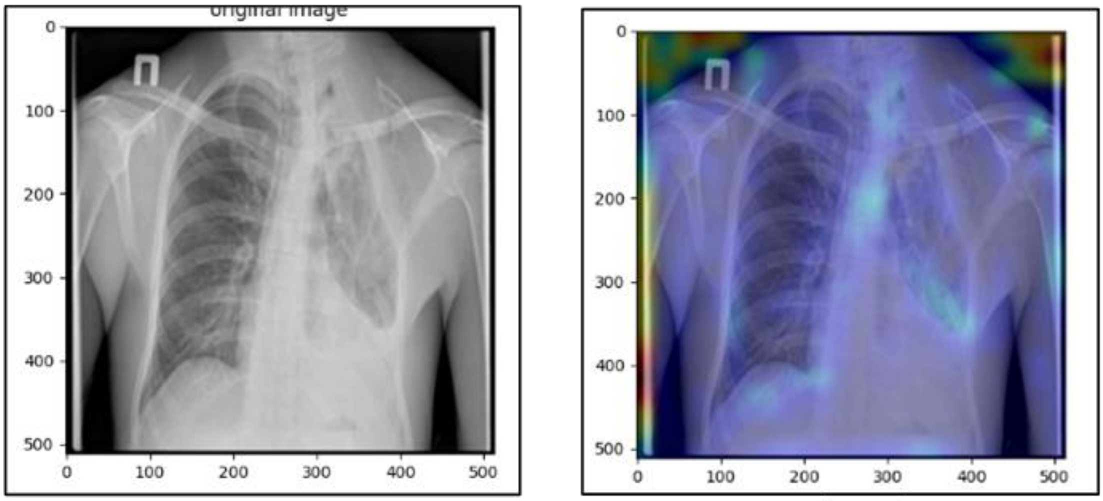
(h)


original image
R
0
250
500
750
1000
1250
1500
1750
2000
0
500
1000
1500
2000

(e)


R
0
250
500
750
1000
1250
1500
1750
2000
0
500
1000
1500
2000

(f)



(i) (j)

Fig. 4.16: Predicted images from the proposed CXRNet model with its corresponding heat-map activated image from last convolutional layer: (a) Cardiomegaly original image, (b) Heat-map activated Cardiomegaly image, (c) COVID original image, (d) Heat-map activated COVID image, (e) Normal original image, (f) Heat-map activated Normal image, (g) Pneumonia original image, (h) Heat-map activated Pneumonia image, (i) Tuberculosis original image, and (j) Heat-map activated Tuberculosis image.

In Fig. 4.17, an illustration of the graphical abstract of this research work in real-time clinical integration for the clinical diagnosis and decision support system is presented. The clinical integration of this system would contribute greatly as a decision support system to patient care by reducing the time to diagnosis and facilitating the access to interpretation of chest radiographs thereby addressing the issues of missed information and misinterpretation as well.

4.6 Discussion

Today, there are large numbers of research works based on DL technologies for the automatic classification of CXR images. However, the real-time adoption of these models in the clinical environment is still a challenge due to various limitations which are well discussed and highlighted in the literature review and problem statement section. These limitations mainly include: (a) lower accuracy or inefficient performance, (b) limited findings, (c) limited generalizability over the dataset, (d) use of pre-trained modes trained without fine-tuning on CXR dataset (e) lack of validation on development set dataset, (f) computationally and economically expensive. In this research, various strategies and techniques are implemented to address these limitations which are discussed in various sections of this research paper. A rigorous iterative approach was pursued to address these limitations in designing an efficient CNN architecture from the scratch with superior model performance on the test data set as well. Likewise, the dataset from various sources was acquired to produce adequate findings including four major abnormalities in the CXR image, also addressing the generalizability of the model across the dataset. Also, the CNN architecture is designed from scratch and is validated on the development set data from hospitals, hence is not computationally and economically expensive for the real-time implementation of the system as a decision support system in the clinical environment.

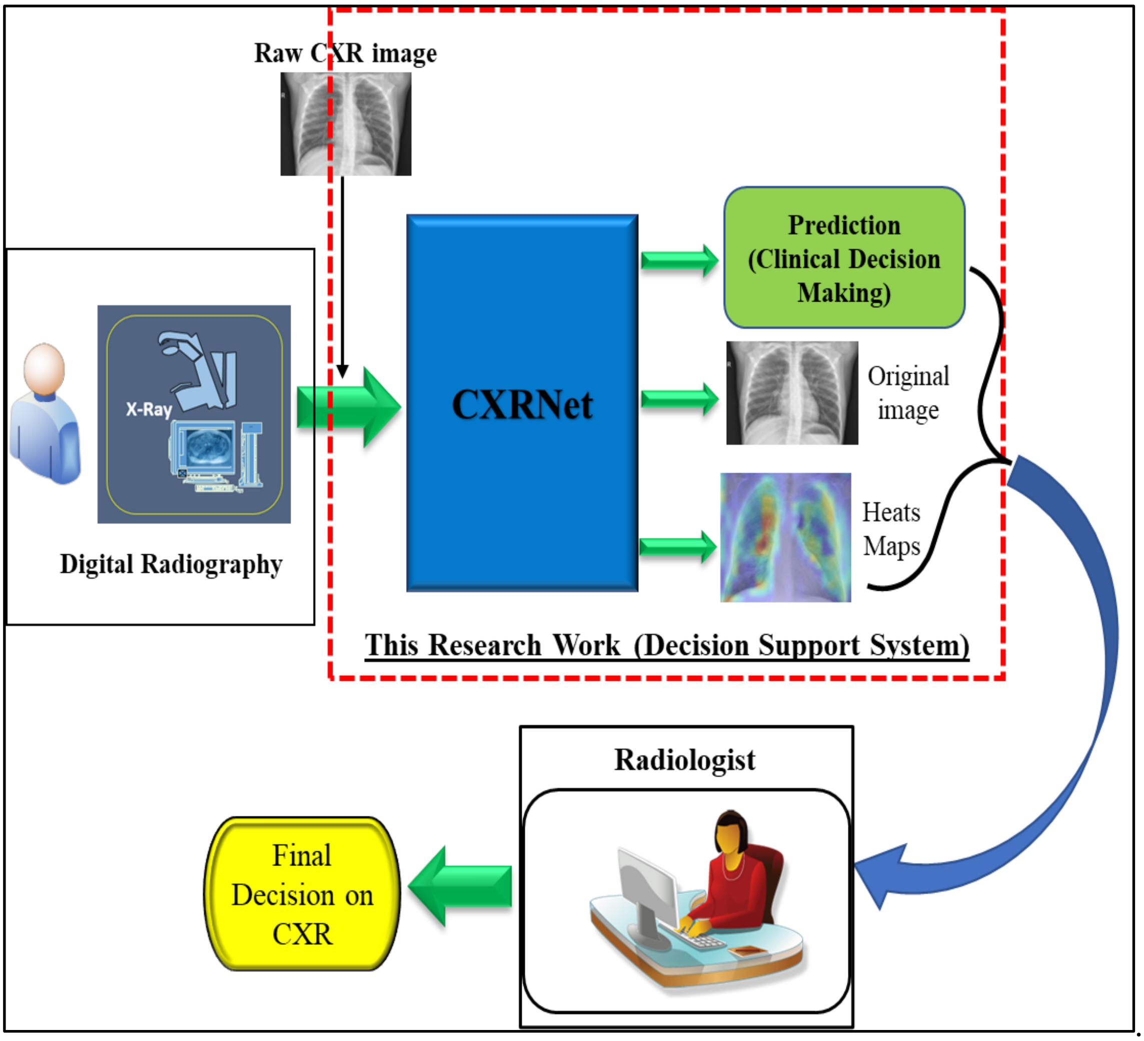


Fig. 4.17: An illustration of graphical abstract of this research work in clinical diagnosis and decision support system.

Although the results are overwhelming and are superior to other related works, it is still not perfect. So, there is still some room for improvement in this model. First, although datasets from four multiple sources are combined, it is still not sufficient enough as the deep learning algorithms tend to work better with the larger dataset. Second, a chest radiograph can be used as a diagnostic tool for a large number of lung diseases, however, this model is capable of only five class classification tasks which can be improved for multiple classes. Third, only frontal radiographs were accessed for developing this model, however, accurate diagnoses require a lateral view as well. The lack of datasets limits the performance of this model for a wide range of disease classifications.

CHAPTER 5
CONCLUSION AND FUTURE WORKS

5.1 Outcomes

Chest radiograph plays a significant role in the diagnosis of various lung diseases, and it is popularly used as a first-line diagnostic tool worldwide. However, the correct interpretation of information while reading a chest radiograph is a major challenge even for experienced radiologists. On the other hand, the majority of the research works based DL techniques for automatic diagnosis of abnormalities in CXR face serious challenges in real-time adaptation in clinical practice due to several limitations. In this regard, this research work plays a vital role to address these issues. The outcomes of this research work are listed below:

(i) An efficient CNN model (CXRNet) is developed from scratch to detect Cardiomegaly, COVID, Normal, Pneumonia, and Tuberculosis cases from input chest radiographs.

(ii) The model has achieved an accuracy of over 95% which is comparatively equivalent or even superior to human-level performance.

(iii) A generalized dataset of five different labels is developed by gathering data from several publicly accessible sources.

(iv) A robust image processing algorithm incorporating various pre-processing techniques and data augmentation techniques is developed and the impact on model performance is observed.

(v) The comparative analysis of the model performance with various relevant research works is performed and also the strength of the proposed CXRNet is highlighted addressing the limitations of other research works.

(vi) Various policies for improving the network performance and optimization algorithms for the model checkpoint, early stopping, and hyperparameters tuning is established.

(vii) It developed a working procedure to design an efficient CNN architecture from scratch and then optimization of the architecture for improving performance by addressing various issues during model training. The techniques, algorithms, and policies implemented in this research work are not only limited to this work but are of great importance to those who are looking forward to developing any DL applications.

5.2 Limitations

The performance of the model developed in this thesis is equivalent or even superior to other related research work. However, it is still not absolutely perfect which is limited due to various constraints. Some of the limitations of this research work include:

(i) The classification of CXR images is limited to five classes, however, in practice, multiple other abnormalities can be detected from CXR interpretation.

(ii) Only frontal radiographs were accessed for developing this model; however, more absolute diagnoses may require lateral view as well. The lack of datasets limits the performance of this model for a wide range of disease classifications.

5.3 Future Work

Today, the application of DL techniques in the medical field is growing rapidly and a large number of research works are carried out to address various issues in this field. In this context, the CNN model developed, and the strategies implemented for network optimization in this thesis provide motivation for future research indicating areas that can be improved for the further development of a robust decision support system. The recommended areas of future work include the following:

(i) Real-time validation, testing, and debugging of the model in clinical practice to implement the model as an efficient decision support system for radiologists.

(ii) Development of a model that could classify multiple pulmonary diseases that can be achieved from the interpretation of the CXR image and the localization of the abnormalities.

(iii) In real-world practice, diagnosis of certain abnormalities requires CXR followed by CT or MRI, hence, integration of CXR with CT and MRI images to expand the domain of decision support systems in real-world applications is a significant research area.

REFERENCES

[1] P. Rajpurkar *et al.*, “Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists,” *PLOS Medicine*, vol. 15, no. 11, p. e1002686, Nov. 2018.

[2] H. Yoo *et al.*, “AI-based improvement in lung cancer detection on chest radiographs: results of a multi-reader study in NLST dataset,” *European Radiology*, vol. 31, no. 12, pp. 9664–9674, 2021.

[3] A. A. Peters *et al.*, “Performance of an AI based CAD system in solid lung nodule detection on chest phantom radiographs compared to radiology residents and fellow radiologists,” *Journal of Thoracic Disease*, vol. 13, no. 5, pp. 2728–2737, 2021.

[4] S. Candemir and S. Antani, “A review on lung boundary detection in chest X-rays,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, no. 4, pp. 563–576, 2019.

[5] C. Qin, D. Yao, Y. Shi, and Z. Song, “Computer-aided detection in chest radiography based on artificial intelligence: A survey,” *BioMedical Engineering Online*, vol. 17, no. 1. 2018.

[6] D. Meedeniya, H. Kumarasinghe, S. Kolonne, C. Fernando, I. D. la T. Díez, and G. Marques, “Chest X-ray analysis empowered with deep learning: A systematic review,” *Applied Soft Computing*, vol. 126, p. 109319, Sep. 2022.

[7] G. Liang and L. Zheng, “A transfer learning method with deep residual network for pediatric pneumonia diagnosis,” *Computer Methods and Programs in Biomedicine*, vol. 187, 2020.

[8] B. C. Bizzo, R. R. Almeida, M. H. Michalski, and T. K. Alkasab, “Artificial intelligence and clinical decision support for radiologists and referring providers,” *Journal of the American College of Radiology*, vol. 16, no. 9, pp. 1351–1356, 2019.

[9] D. A. Moses, “Deep learning applied to automatic disease detection using chest X-rays,” *Journal of Medical Imaging and Radiation Oncology*, vol. 65, no. 5, pp. 498–517, 2021.

[10] S. Pang, Z. Yu, and M. A. Orgun, “A novel end-to-end classifier using domain transferred deep convolutional neural networks for biomedical images,” *Computer Methods and Programs in Biomedicine*, vol. 140, pp. 283–293, 2017.

[11] M. J. Awan, M. H. Bilal, A. Yasin, H. Nobanee, N. S. Khan, and A. M. Zain, “Detection of covid-19 in chest x-ray images: A big data enabled deep learning approach,” *International Journal of Environmental Research and Public Health*, vol. 18, no. 19, 2021.

[12] M. D. B. S. Tam *et al.*, “Augmenting lung cancer diagnosis on chest radiographs: positioning artificial intelligence to improve radiologist performance,” *Clinical Radiology*, vol. 76, no. 8, pp. 607–614, 2021.

[13] Y. Li, “Research and application of deep learning in image recognition,” pp. 994–999, Mar. 2022.

[14] M. Loey, S. El-Sappagh, and S. Mirjalili, “Bayesian-based optimized deep learning model to detect COVID-19 patients using chest X-ray image data,” *Computers in Biology and Medicine*, vol. 142, no. December 2021, p. 105213, 2022.

[15] K. S. Chandrasekar, “Exploring the deep-learning techniques in detecting the presence of Coronavirus in the chest X - ray images : A comprehensive review,” *Archives of Computational Methods in Engineering*, no. 0123456789, 2022.

[16] P. R. A. S. Bassi and R. Attux, “A deep convolutional neural network for COVID-19 detection using chest X-rays,” *Research on Biomedical Engineering*, vol. 38, no. 1, pp. 139–148, 2022.

[17] R. Fan and S. Bu, “Transfer-learning-based approach for the diagnosis of lung diseases from chest X-ray images,” *Entropy*, vol. 24, no. 3, pp. 1–13, 2022.

[18] Z. Huang *et al.*, “Fusion High-Resolution Network for Diagnosing ChestX-ray Images,” *Electronics*, vol. 9, no. 1, p. 190, 2020.

[19] H. Fujita, “AI-based computer-aided diagnosis (AI-CAD): the latest review to read first,” *Radiological Physics and Technology*, vol. 13, no. 1. pp. 6–19, 2020.

[20] A. Bustos, A. Pertusa, J. M. Salinas, and M. de la Iglesia-Vayá, “PadChest: A large chest x-ray image dataset with multi-label annotated reports,” *Medical Image Analysis*, vol. 66, p. 101797, 2020.

[21] N. E. M. Khalifa, M. Loey, F. Smarandache, “SS symmetry within the lack of chest COVID-19 X-ray dataset : A novel detection model based on GAN and deep,” 2020.

[22] K. Suzuki, “Overview of deep learning in medical imaging,” *Radiological Physics and Technology*, vol. 10, no. 3, pp. 257–273, 2017.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature 1986 323:6088*, vol. 323, no. 6088, pp. 533–536, 1986.

[24] S. R. Sain, “The nature of statistical learning theory,” *Technometrics*, vol. 38, no. 4, pp. 409–409, Nov. 1996.

[25] T. K. Ho, “Random decision forests,” *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 1, pp. 278–282, 1995.

[26] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” *ACM International Conference Proceeding Series*, vol. 382, 2009.

[27] K. Fukushima and S. Miyake, “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition,” pp. 267–285, 1982.

[28] Y. LeCun *et al.*, “Backpropagation applied to handwritten zip code recognition,” *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[29] S. KENJI, H. ISAO, I. KAZUO, and N. MICHIO, “Recognition of coronary artery stenosis using neural network on DSA system.,” *電子情報通信学会論文誌 D-2*, vol. 77, no. 9, pp. 1910–1916, 1994, Accessed: Aug. 17, 2022. [Online]. Available: https://jglobal.jst.go.jp/en/detail?JGLOBAL_ID=200902199013995284.

[30] K. Suzuki, S. G. Armato, F. Li, S. Sone, and K. Doi, “Massive training artificial neural network (MTANN) for reduction of false positives in computerized detection of lung nodules in low-dose computed tomography,” *Medical Physics*, vol. 30, no. 7, pp. 1602–1617, Jul. 2003.

[31] K. Suzuki, H. Abe, H. MacMahon, and K. Doi, “Image-processing technique for suppressing ribs in chest radiographs by means of massive training artificial neural network (MTANN),” *IEEE Transactions on Medical Imaging*, vol. 25, no. 4, pp. 406–416, Apr. 2006.

[32] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation,* vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems,* vol. 25, 2012, Accessed: Aug. 18, 2022. [Online]. Available: http://code.google.com/p/cuda-convnet/.

[34] P. M. Cheng *et al.*, “Deep learning: An update for radiologists,” *Radiographics,* vol. 41, no. 5, pp. 1427–1445, 2021.

[35] O. Stephen, M. Sain, U. J. Maduh, and D. U. Jeong, “An efficient deep learning approach to Pneumonia classification in healthcare,” *Journal of Healthcare Engineering,* vol. 2019, 2019.

[36] R. Jain, P. Nagrath, G. Kataria, V. Sirish Kaushik, and D. Jude Hemanth, “Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning,” *Measurement: Journal of the International Measurement Confederation,* vol. 165, p. 108046, 2020.

[37] M. M. Eid and Y. H. Elawady, “Efficient Pneumonia detection for chest radiography using ResNet-based SVM,” *European Journal of Electrical Engineering and Computer Science,* vol. 5, no. 1, pp. 1–8, 2021.

[38] O. A. Fagbuagun, O. Nwankwo, S. A. Akinpelu, and O. Folorunsho, “Model development for pneumonia detection from chest radiograph using transfer learning,” *Telkomnika (Telecommunication Computing Electronics and Control),* vol. 20, no. 3, pp. 544–550, 2022.

[39] A. Mabrouk, R. P. D. Redondo, A. Dahou, M. A. Elaziz, and M. Kayed, “Pneumonia detection on chest X-ray images using ensemble of deep convolutional neural networks,” *Applied Sciences (Switzerland),* vol. 12, no. 13, 2022.

[40] E. Ayan, B. Karabulut, and H. M. Ünver, “Diagnosis of pediatric Pneumonia with ensemble of deep convolutional neural networks in chest X-ray images,” *Arabian Journal for Science and Engineering,* vol. 47, no. 2, pp. 2123–2139, 2022.

[41] M. W. Kusk and S. Lysdahlgaard, “The effect of Gaussian noise on pneumonia detection on chest radiographs, using convolutional neural networks,” *Radiography,* vol. 29, no. 1, pp. 38–43, 2023.

[42] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, “Automated detection of COVID-19 cases using deep neural networks with X-ray images,” *Computers in Biology and Medicine*, vol. 121, p. 103792, Jun. 2020.

[43] T. Rahman *et al.*, “Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images,” *Computers in Biology and Medicine,* vol. 132, no. March, p. 104319, 2021.

[44] J. L. Gayathri, B. Abraham, M. S. Sujarani, and M. S. Nair, “A computer-aided diagnosis system for the classification of COVID-19 and non-COVID-19 pneumonia on chest X-ray images by integrating CNN with sparse autoencoder and feed forward neural network,” *Computers in Biology and Medicine*, vol. 141, no. November 2021, p. 105134, 2022.

[45] S. Sanket, M. V. R. Sarobin, L. J. Anbarasi, J. Thakor, U. Singh, and S. Narayanan, “Detection of novel coronavirus from chest X-rays using deep convolutional neural networks,” *Multimedia Tools and Applications*, vol. 81, no. 16, pp. 22263–22288, 2022.

[46] A. Kumar, A. R. Tripathi, S. C. Satapathy, and Y. D. Zhang, “SARS-Net: COVID-19 detection from chest x-rays by combining graph convolutional network and convolutional neural network,” *Pattern Recognition*, vol. 122, p. 108255, 2022.

[47] S. I. Nafisah and G. Muhammad, “Tuberculosis detection in chest radiograph using convolutional neural network architecture and explainable artificial intelligence,” *Neural Computing and Applications*, vol. 6, 2022.

[48] V. Acharya *et al.*, “AI-assisted Tuberculosis detection and classification from chest X-rays using a deep learning normalization-free network model,” *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[49] E. Showkatian, M. Salehi, H. Ghaffari, R. Reiazi, and N. Sadighi, “Deep learning-based automatic detection of tuberculosis disease in chest X-ray images,” *Polish Journal of Radiology*, vol. 87, no. 1, pp. 118–124, 2022.

[50] S. S. Guia, L. Abdelkader, M. Kara, and M. Hammoudeh, "Tuberculosis detection using chest X-ray image classification by deep learning," *Research Square*, 2023.

[51] A. I. Khan, J. L. Shah, and M. M. Bhat, "CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images," *Computer Methods and Programs in Biomedicine*, vol. 196, p. 105581, Nov. 2020.

[52] A. Musha, A. Al Mamun, A. Tahabilder, J. Hossen, B. Jahan, and S. Ranjbari, "A deep learning approach for COVID-19 and pneumonia detection from chest X-ray images," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 4, pp. 3655–3664, 2022.

[53] F. Bayram and A. Eleyan, "COVID-19 detection on chest radiographs using feature fusion based deep learning," *Signal, Image and Video Processing*, vol. 16, no. 6, pp. 1455–1462, 2022.

[54] I. Kanjanasurat, K. Tenghongsakul, B. Purahong, and A. Lasakul, "CNN–RNN Network Integration for the Diagnosis of COVID-19 Using Chest X-ray and CT Images," *Sensors*, vol. 23, no. 3, p. 1356, 2023.

[55] "Chest X-Ray Images (Pneumonia) | Kaggle." https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia (accessed Jun. 04, 2022).

[56] "COVID-19 Radiography Database | Kaggle." https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database (accessed Jun. 04, 2022).

[57] "COVID-19 DATABASE - SIRM." https://sirm.org/category/senza-categoria/covid-19/ (accessed Feb. 07, 2023).

[58] "BIMCV-COVID19 – BIMCV." https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/#1590858128006-9e640421-6711 (accessed Feb. 07, 2023).

[59] E. Khan, M. Z. U. Rehman, F. Ahmed, F. A. Alfouzan, N. M. Alzahrani, and J. Ahmad, "Chest X-ray classification for the detection of COVID-19 using deep learning techniques," *Sensors 2022, Vol. 22, Page 1211*, vol. 22, no. 3, p. 1211, Feb. 2022.

[60] "Tuberculosis (TB) Chest X-ray Database | Kaggle." https://www.kaggle.com/datasets/tawsifurrahman/tuberculosis-tb-chest-xray-dataset (accessed Jan. 21, 2023).

[61] "CXR8 | Powered by Box." https://nihcc.app.box.com/v/ChestXray-NIHCC (accessed Jan. 21, 2023).

[62] E. Ayan and H. M. Ünver, "Diagnosis of pneumonia from chest X-ray images using deep learning," *2019 Scientific Meeting on Electrical-Electronics and Biomedical Engineering and Computer Science, EBBT 2019*, pp. 2–6, 2019.

[63] M. Olayemi, "*Tuberculosis diagnosis from Pulmonary chest X-ray using deep learning*," ," *University of Kwazulu-Natal*, pp. 11-55, 2022.

[64] N. Krishnamoorthy, K. Nirmaladevi, T. Kumaravel, K. S. Sanjay Nithish, S. Sarathkumar, and M. Sarveshwaran, "Diagnosis of Pneumonia Using Deep Learning Techniques," *2022 2nd International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies, ICAECT 2022*, vol. 6, no. 2, pp. 48–68, 2022.

[65] Y. Zhang, J. Gao, and H. Zhou, "Breeds Classification with Deep Convolutional Neural Network," *ACM International Conference Proceeding Series*, pp. 145–151, 2020.

[66] F. Yang, H. Zhang, and S. Tao, "Simplified multilayer graph convolutional networks with dropout," *Applied Intelligence*, vol. 52, no. 5, pp. 4776–4791, 2022.

[67] S. Hayou, A. Doucet, and J. Rousseau, "On the impact of the activation function on deep neural networks training," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 4746–4754, 2019.

[68] "tf.keras.optimizers.Adam | TensorFlow Core v2.9.1." https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam (accessed Jun. 05, 2022).

[69] W.-C. Yeh, Y.-P. Lin, Y.-C. Liang, C.-M. Lai, and X.-Z. Gao, "Simplified Swarm Optimisation for the Hyperparameters of a Convolutional Neural Network," *arXiv*, pp. 1–26, 2021.

LIST OF PUBLICATIONS

1. **Bipin Thapa Magar**, Kumar Shrestha, and Md. Asadur Rahman, "CNN-Based Clinical Diagnosis and Decision Support System for Chest X-ray," 2nd International Conference on Mechanical Engineering and Applied Sciences, Military Institute of Science and Technology (MIST), Dhaka 1216, Bangladesh (2022).