Jimma University

Jimma Institute of Technology

School of Biomedical Engineering

Biomedical Imaging Stream

Lung Diseases Classification from Chest X-Ray Images using Deep Learning

A Thesis submitted to School of Graduate Studies of Jimma Institute of Technology, Jimma University, in Partial Fulfillment of the requirements for the Degree of Master of Science in Biomedical Engineering (Biomedical Imaging)

By: Fethya Seid

Advisors:

Main Advisor: Gizeaddis Lamesgin (Ph.D.)

Co-Advisor: Abel Worku Tessema (MSc.)

February, 2020 G.C.

Jimma, Ethiopia

Declaration

I declared this research with the title of ***"Lung Diseases Classification from Chest X-Ray Images using Deep Learning"*** as my original work and I assure it with my signature.

Name: Fethya Seid Sign________________ Date___________________

On behalf of the School of Biomedical Engineering at Jimma Institute of Technology, we the advisors of this research with the title of ***"Lung Diseases Classification from Chest X-Ray Images using Deep Learning"*** and I, the evaluator, confirm that this research is approved as MSc. Thesis for the student.

Advisors

Name: Dr. Gizeaddis Lamesgin (Ph.D.) Sign_______________ Date___________________

Name: Mr. Abel Worku (M.Sc) Sign_______________ Date___________________

Internal Examiner

Name: _______________________ Sign_______________ Date___________________

External Examiner

Name: _______________________ Sign_______________ Date___________________

Chair

Name: _______________________ Sign_______________ Date___________________

Abstract

Lung diseases are disorders in the lung that affect proper functioning of the breathing system. The top five prevalent lung diseases that are the leading cause of death include Chronic Obstructive Pulmonary Disease, Lung Cancer, Pneumonia, Tuberculosis, and Pneumothorax. Diagnosis of lung diseases is usually performed through visual inspection of chest X-ray images, especially in the developing world. This is time consuming, tedious, and subjected to inter and intra-observer variability which may lead to misdiagnosis.

In this research, a method for automatic, accurate and reliable classification of the top five lung diseases from chest X-ray radiograph images is proposed using a deep learning approach. The data required for training, validation and testing the system was collected from online National Institute of Health chest X-ray14 dataset repository and the local data was acquired from Jimma University Medical Center radiology department. Deep learning approach based on Xception model was used for multi class classification task. All the images have been pre-processed prior to feeding to the model. The system has been developed using Python 3.7 programing language. A graphical user interface has been developed for ease of use and implementation. An accuracy, sensitivity and specificity of 97.3%, 97.2%, and 99.4%, respectively, have been achieved for multi-class classification using the proposed algorithm. The system takes only an average of 1 minute to provide the diagnosis result.

The developed system will have a great impact in reducing the diagnosis errors imposed by the manual visual inspection method and can be used as a decision support system for physicians, especially those in low resource setting where both the expertise and the means is in scarce.

Keywords: - Lung disease, Chest X-ray, Image processing, Multi-class Classification, Deep Learning, Xception

Acknowledgments

First of all I would like to thank Allah for helping me in all my life and blessing me much more than I deserve. I also want to extend my heartfelt gratitude to my advisors Dr. Gizeaddis Lamesgin and Mr. Abel Worku Tessema for their unlimited effort in this thesis and showing me the ways to perform scientific research.

Secondly, I also want to thank Mr. Mohammed Aliy, Mr. Kokeb Dese, Ms. Genet Tadesse and Mr. Boaz Berhanu for the valuable advice and guidance throughout the work, and Dr. Eliyas Kedir for helping me in getting the data and giving valuable advice regarding diagnosis of lung diseases at Jimma University Medical Center. I would also like to acknowledge Jimma University, Jimma Institute of Technology, School of Biomedical Engineering for providing me the opportunity to study Master's program.

Finally, I would like to thank my families and my friends for their unfailing support and encouragements throughout my years of study.

List of Abbreviations

AI	Artificial Intelligence
ADAM	Adaptive Moment Estimation
ANFIS	Adaptive Neuro Fuzzy Inference
CAD	Computer-Aided Detection
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
COPD	Chronic Obstructive Pulmonary Diseases
CT	Computed Tomography
CXR	Chest X Ray
DCNN	Deep Convolutional Neural Network
DL	Deep Learning
DR	Digital Radiography
FL	Fuzzy Logic
GPU	Graphical Processing Unit
JUMC	Jimma University Medical Center
LLC	Limited Liability Company
MF	Membership Function
ML	Machine Learning
MRI	Magnetic Resonance Imaging
NN	Neural Network

NCD	Non Communicable Diseases
NIH	National Institutes of Health
RB	Rule Based
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Decent
TB	Tuberculosis
TN	True Negative
TP	True Positive
WHO	World Health Organization

Table of Contents

Declaration i

Abstract ii

Acknowledgments iii

List of Abbreviations iv

List of Figures ix

List of Tables x

CHAPTER ONE 1

1. Introduction 1

1.1. Background of the Study 1
1.2. Lung Anatomy 1
1.3. Lung Diseases and Risk Factors 2
1.4. Existing Lung Disease Diagnosis Techniques 3
1.4.1. Chest Radiography 3
1.4.2. Manual Interpretation of Chest X-Ray Images 5
1.5. Computer Visions for Chest X-Ray Image Interpretation 6
1.6. Artificial Intelligence for Classification of chest X-Ray Images 7
1.7. Statement of the Problem 10
1.8. Objectives 10
1.8.1. General Objective 10
1.8.2. Specific Objectives 11
1.9. Motivation 11
1.10. Significance of the Thesis 11
1.11. Scope of the Thesis 12

CHAPTER TWO 13

2. Related works 13

2.1. Review of Automatic Diagnosis of Lung Diseases from Chest X-Ray Images 13

3. Materials and Methods 18

3.1. Image Data Collection 18
3.2. Image Pre-processing 20
3.2.1. Image De-noising 20
3.2.2. Image Enhancement 22
3.2.3. Image Augmentation 24
3.3. Image Classification 24
3.3.1. Transfer learning 25
3.4. The Xception Model 26
3.4.1. Network Configuration for Xception 30
3.4.2. Data preparation 31
3.4.3. Training Xception Model for chest X-ray images Classification 31
3.4.4. Performance Analysis of the Xception Model 32
3.5. Graphical User Interface (GUI) 33
3.6. Software and Hardware Materials used for this Research 33
CHAPTER FOUR 35
4. Results and Discussion 35
4.1. Results 35
4.1.1. Pre-processing 35
4.1.2. Training Xception for Classification of Multiple Lung Diseases 38
4.1.3. Performance of the Xception Model on Multi Class Classification Task 40
4.1.4. Implementation Using GUI 42
4.2. Discussion 43
CHAPTER FIVE 46
5. Conclusion and Recommendations 46
5.1. Conclusion 46
5.2. Recommendations 46
REFERENCES 48
APPENDICES 56
Appendix A: Approval Letter from JUMC 56
Appendix B: Implementation Code for Noise Removal and Enhancement 57

Appendix C: Implementation Code for Classification.. 57
Appendix D: Implementation Code for GUI 61

List of Figures

Figure 1-1:Unsophisticated anatomy of lungs 2

Figure 1-2: Interpretation of chest X-ray images.. 5

Figure 1-3: Taxonomy of AI from hand crafted ML to DL.. 7

Figure 1-4: Architecture of ANN... 8

Figure 1-5: Structure of NN 8

Figure 1-6: Performance of deep learning with respect to the amount of data.................. 9

Figure 3-1: General block diagram of the methodology of the research. 18

Figure 3-2: Sample images from the datasets 19

Figure 3-3: The median filter 22

Figure 3-4: Contrast limited adaptive histogram equalization (CLAHE)......................... 23

Figure 3-5: Comparative analysis of different models. 26

Figure 3-6: The convolutional layer of the Xception model... 27

Figure 3-7: Data split scheme used in this study 31

Figure 3-8: General layout of the developed GUI 33

Figure 4-1: Result of de- noising. 35

Figure 4-2: Effects of histogram equalization variations on image 36

Figure 4-3: Effects of AHE... 37

Figure 4-4: Result of CLAHE.. 37

Figure 4-5: Augmentation result 38

Figure 4-6: Training and validation accuracy using Xception model............................... 39

Figure 4-7: Classification accuracy 41

Figure 4-8: Sensitivity and Specificity of the model. 41

Figure 4-9: Implementation of GUI 42

List of Tables

Table 1-1: Lung diseases and their chest X-ray findings 4

Table 3-1: The total amount of dataset used for the study.. 19

Table 3-2: Comparison of different pre-processing methods. 21

Table 4-1: Description of datasets used for training, validation and testing 39

Table 4-2: Confusion matrix... 40

CHAPTER ONE

1. Introduction

1.1. Background of the Study

Lung disease is a disorder affecting the lungs and causing breathing problems. According to European lung white book [1], which is the first comprehensive survey on lung respiratory health, respiratory diseases cause more than 4 million deaths annually worldwide, and lung cancer is the first most common cause of respiratory death, followed by Chronic Obstructive Pulmonary Disease (COPD), lower respiratory infections (pneumonia), and Tuberculosis (TB). The major risk factors are tobacco smoking, environmental factors and life events such as, mode of delivery, type of feeding and antibiotic use [2].

According to the World Health Organization (WHO) chronic pulmonary lung diseases, cardiovascular diseases, cancer, and diabetes, are the cause for the death of 34% of total population of Ethiopia [3]. Ethiopia is one of the countries with high prevalence of lung diseases or diseases of respiratory systems like TB, and pneumonia. TB is one of the leading causes of death and health-related socio-economic problems worldwide. Ethiopia ranks seventh among the world's top 22 countries severely affected by TB [4]. Pneumonia is the number one infectious killer of children, and Ethiopia is among 15 top pneumonia high risk countries. An estimated number of 3,370,000 children are encounter pneumonia in every year [5].

1.2. Lung Anatomy

Lung is one of the main organs of the respiratory system in human's anatomy which undergoes the exchange of oxygen and carbon dioxide. This is done across a very large epithelial surface area about 70 square meters that is highly permeable to gases. It is a paired organs and pyramid shape structure connected to the trachea by the right and left bronchi, bordered by the diaphragm, enclosed by the pleurae, which are attached to the mediastinum. It is made up of smaller units called lobes. Fissures separate these lobes from each other. The right lung consists of superior, middle, and inferior lobes. The left lung consists of two lobes: the superior and inferior lobes. A bronchopulmonary segment is a division of a lobe, and each lobe houses multiple bronchopulmonary segments. Each segment receives air from its own tertiary bronchus and is supplied with blood by its own artery [6]. Figure 1.1 shows the gross anatomy of lungs.

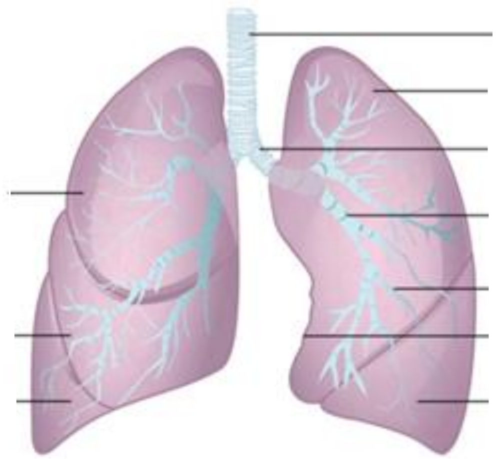

Figure 1-1:Unsophisticated anatomy of lungs [6].

1.3. Lung Diseases and Risk Factors

Lung diseases are problems in the lung that affects proper functioning of its work. Among lung diseases lung cancer, COPD, pneumonia, TB, and pneumothorax, are the most common once.

Lung carcinoma is the other name for lung cancer, also known as malignant lung tumor which is characterized by uncontrolled growth of lung tissue cells. Most cases (85%) of the lung cancer are related with long-term tobacco smoking. But in some cases (10–15%) the causes are idiopathic. And these cases are related with a combination of genetic factors and exposure to radon gas, asbestos, second-hand smoke, or other forms of air pollution [7].

COPD is a chronic inflammatory lung disease that is characterized by air flow limitation, obstructed airflow from the lungs and blocked airways that makes breathing difficult. It includes emphysema (can be defined as destruction and enlargement of the lung alveoli) and chronic bronchitis (a condition of chronic cough and phlegm). Long-term exposure to irritating gases or particulate matter and most often from cigarette smoke may expose the individual for this disease. COPD is one of the major risk factors for heart disease, lung cancer and a variety of other conditions [8].

Pneumonia is a bacterial, viral or other organismic disease that causes infection in one or both lungs. Infection is relatively common among adults and pneumonia incidence is highest where immune systems are compromised. It causes inflammation in the air sacs in the lungs, which are called alveoli. The alveoli get filled with fluid or pus, making it difficult to breathe [9].

TB is the other type of lung affecting diseases and, it can also spread to other parts of the body, like the brain and spine. The causing agent for TB is Mycobacterium tuberculosis [9].

Pneumothorax is a lung disease which is characterized by an abnormal leakage of air in the space between the lung and the chest wall. In some of cases the amount of air in the chest rises when a one-way valve is formed by an area of damaged tissue, may cause a tension pneumothorax. It is caused by the development of leak in the visceral pleura, such that air escapes from the lung during inspiration, which leads to an increase pressure of air with in the pleural cavity and decrease the cardiac output [10].

Generally the causes for diseases of the respiratory systems are tobacco use, exposures to indoor and outdoor air pollutants, allergens, occupational exposure, and to a lesser extent other chronic diseases, unhealthy diet, obesity, overweight intake and physical inactivity [11].

1.4. Existing Lung Disease Diagnosis Techniques

Methods commonly used to assess lung diseases includes: pulmonary function test using Spirometer for COPD, blood and sputum analysis for pneumonia and TB, CT scan for lung cancer, and chest X-ray imaging for all. In lung disease identification, radiologists integrate their medical knowledge with chest X-ray images to get the nature and pathological appearances of lung diseases and choice on treatment options [8]. Currently in Ethiopia, lung diseases are diagnosed by visual interpretation of chest radiography (CXR) images. It is the common and standard way in diagnostic techniques.

1.4.1. Chest Radiography

Chest radiograph, usually known as chest X-ray, produce images of the heart, lungs, blood vessels, airways, and bones of the chest and spine. It is used to diagnose conditions affecting the chest, its contents, and nearby structures. Chest X-rays are one of the most widely used diagnostic tools compared to, Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), Computed Tomography (CT), and Nuclear medicine, because of their low cost and ease of use in under developing countries. Chest X-ray contains a huge amount of information about a patient’s healthiness. Though, properly interpreting of chest X- ray images is always a major challenge for the radiologists. These challenges are mainly due to the overlapping of tissue structures in chest X-ray images. For instance, when the contrast between the lesion and the surrounding tissue is very low or when the lesion overlaps the ribs or large pulmonary blood vessels [12].

Lung diseases can be displayed in chest X-ray images in the form of cavitation, consolidation, infiltration, blunted costophrenic angle, and small broadly distributed nodules [12]. Table 1.1 below shows lung diseases with effects on the lung and their chest X-ray findings, and also Figure 1.2 displays interpretation of chest X-ray images for six diseases.

Table 1-1: Lung diseases and their chest X-ray findings [8].

Type of diseases	**Effect**	**X-ray findings**
Lung cancer	They can attack nearby tissues and form tumors. It can start anywhere in the lungs and affect any part of the respiratory system.	Appearance of lung nodules
COPD	Air sacs and Airway lose their elastic quality. Lung tissue is destroyed	Enlarged lungs, irregular air pockets and flattened diaphragm.
Pneumonia	Destruction of lung parenchyma	Infiltration or consolidation
Tuberculosis	The infection destroys patients' lung tissue	Cavitation
Pneumothorax	Producing all or a part of the lung to collapse. Air usually enters this space.	Collapsed lung

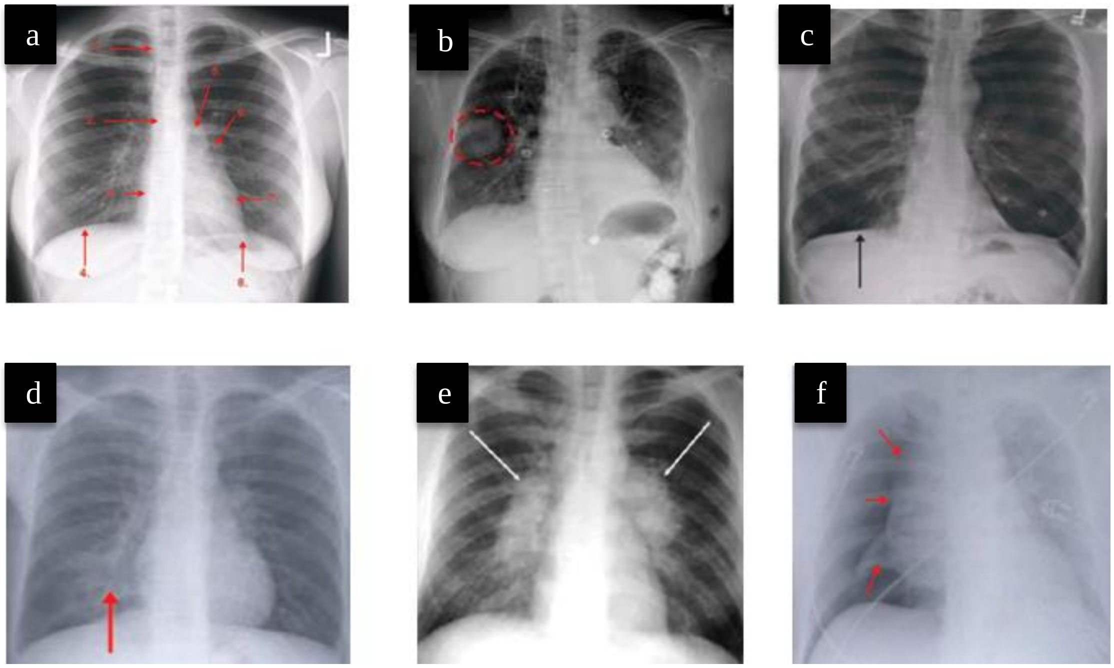

Figure 1-2: Interpretation of chest X-ray image for (a) Normal, (b) Lung cancer, (c) COPD, (d) Pneumonia, (e) TB and (f) Pneumothorax with corresponding findings nodules, enlarged lung, consolidation, cavitation and collapsed lung respectively [8].

1.4.2. Manual Interpretation of Chest X-Ray Images

Chest X-ray image inspection requires uniform and logical technique with long duration of time. A systematic examination takes place by reviewing all parts of the image in step by step manner and aims to identify all possible abnormalities on the chest X-ray images. Even if there is no one best standard approach, step by step assessment is the commonly used method for chest X-ray images. On X-ray images, air appears black, fat appears gray, soft tissues and water appear as lighter shades of gray, bone and metal appear white. The denser the tissue, the whiter it will appear on X-ray whereas, less dense tissues appear radiolucent, dark on the image [13]. Similarly, when viewing the lungs, starting at the apex or base, it always compares each region (upper, mid and lower) of the lungs from side to side. This process is continued until all areas of the image have been systematically viewed. Additionally, hidden areas: such as apical areas underlying the clavicles, perihilar and paratracheal regions, retrocardiac and infra diaphragmatic areas should be careful examined. However, a complete examination with a consistent manner is essential than the order in which areas of the image are viewed. This is especially significant in images with obvious abnormalities that may catch the eye and that may prevent other more subtle abnormalities from being noticed unless the entire image is methodically evaluated. Generally radiologists inspect chest X-ray images by visualization of features mentioned on Table 1-1.

Visual inspection of chest X-ray images is error prone, and is a great challenge since it requires high level of expert knowledge. The task is very time consuming, it also lead to inter or intra observer variability again the result may be inaccurate, it gives false positive result. This means due to the complex nature of chest X-ray image, manual examination leads to inconsistent and subjective reports causing misdiagnosis.

1.5. Computer Visions for Chest X-Ray Image Interpretation

Computer vision is an interdisciplinary field of science that concerns on how computers can be made to gain high-level understanding from digital images and videos. It works to make it possible to see, identify and process images in the same way that human vision does, and concerned with the automatic extraction, analysis and understanding of useful information from a single image or a sequence of images. Generally, computer vision tries to automate tasks in the same way that human visual system can do [14].

Image processing is one part of computer vision. Computer vision system uses image processing algorithms to analyze the image that is detected by computer. The main difference is in the goal of the process, but not in the methods, that means the goal for image processing is to do some transformation on an image to improve enhancement and readability, whereas the goal of the computer vision focuses on trying to do human vision including learning, being able to make inferences and take actions based on visual input.

The use of computers helps radiologists in the acquisition (e.g. X-ray, CT, MRI, and US) and reporting of medical images. More recently, computer programs have been developed and approved for use in clinical practice that aid radiologists in detecting potential abnormalities on diagnostic radiology exams. This application has been termed as Computer Aided Diagnosis (CAD). It is a system that assists doctors in the interpretation of medical images to decrease observational errors and thus the false negative rates. The CAD systems commonly involves the following steps: image pre-processing, extracting regions, extracting ROI features, and classifying diseases according to the features.

The wide application of chest X-rays and the complexity of the reading leads to CAD systems. CAD system are powerful and accessible for chest X-ray based diagnosis of lung diseases, and it can help doctors to detect suspicious lesions that are easily missed, thus improve the accuracy of the diagnosis [12].

1.6. Artificial Intelligence for Classification of chest X-Ray Images

Artificial intelligence (AI), sometimes called machine intelligence, is technology that enables a computer to think like human beings. It learns to perform a given task in away the human brain learns. Machine learning (ML) is a subset of AI which can be used to develop several algorithms to perform certain tasks. Neural Network (NN) is a subfield of ML and this subfield has produced Deep Learning (DL), by increasing the number of layers, algorithms which are capable of handling very complex tasks [15]. Figure 1.3 shows the hierarchical taxonomy of AI.

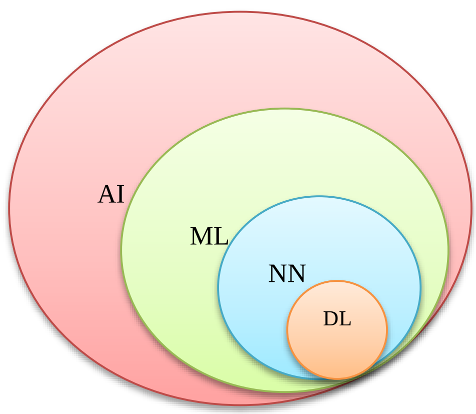

Figure 1-3: Taxonomy of AI from hand crafted ML to DL [15].

An Artificial Neural Network (ANN) is creating simulated brain that is inspired by the biological nervous systems, and it is a simple mathematical model of the brain which is used to process nonlinear relationships between inputs and outputs in parallel. An ANN consists of input layers, hidden layers, and output layers. The input layer receives the input value and the next hidden layer is a set of layers between input and output layers that consists of convolutional layers, pooling layers, fully connected layers, and normalization layers. The last layer of NN is the output layer that produces a given output [15]. Figure 1.4 shows the general architecture of ANN.

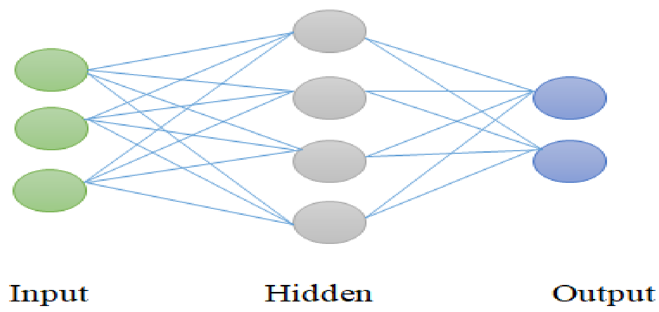

Figure 1-4: Architecture of ANN: ball like shapes are the nodes (neuron) of the network and lines indicate the layer of the network [15].

Neural network is the algorithm used to recognize the relationship between the input and output; it has two important features, the architecture (neurons and their connection) and the weights (the value of the layers). Neurons perform the mathematical functions that combine weights and inputs, adding bias and finally apply activation function to change the value into probability. Figure 1.5 below describes the structure of NN with input, neurons and output value.

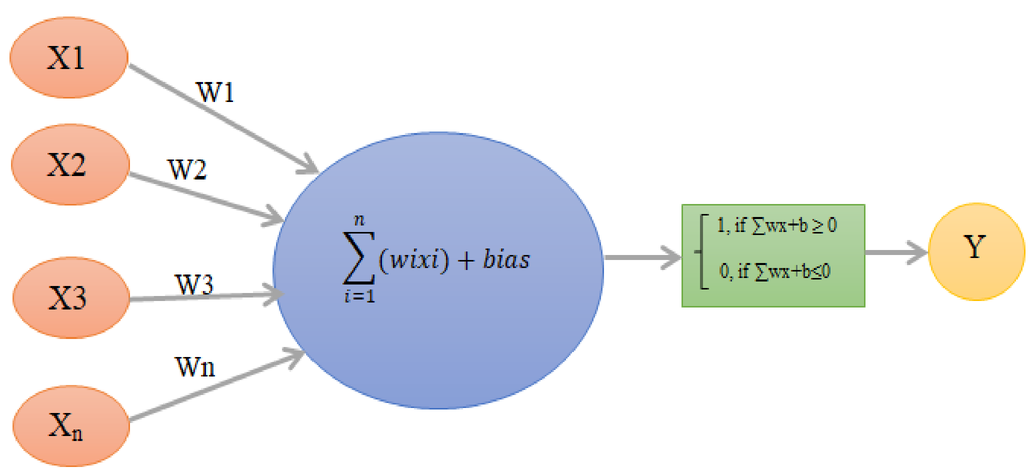

Figure 1-5: Structure of NN contains input values (X), weight of the layer (W), nodes, activation function and output (Y) [16].

During the training phase, the network learns different features to predict a given output. To evaluate how good the network learns a given feature loss is calculated using loss function. This is used to optimize and improve the performance of the NN. DL is a subset of artificial neural networks that contain multiple layers this makes the network very deep. An increase in network layer allows learning very similar features easily. As a result DL becomes dominant in various computer vision tasks and is fascinating interest across a variety of fields, including radiology.

DL consists of several hidden layers in between the input and output layer which allows for many stages of non-linear information processing units with hierarchical architectures to be present that are exploited for feature learning and pattern recognition. Especially for big data analytics DL has a better performance than machine learning because there is no need to apply handmade feature extractions and depth of the network helps to generalize well [15]. Figure 1.6 represents the DL and ML ability to generalize large amount of dataset.

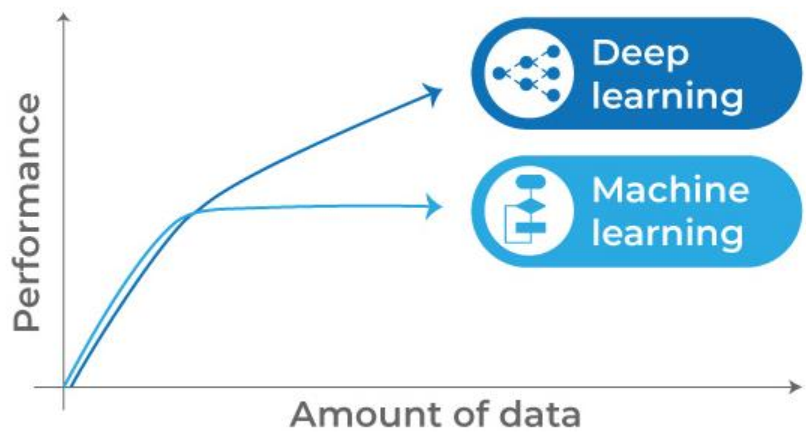

Figure 1-6: Performance of deep learning with respect to the amount of data, DL has an ability to generalize well for large dataset [15].

DL is used for a variety of tasks, a popular use is for classification and is popular for its pattern recognition abilities, which are needed for pattern recognition and decision-making and DLs are robust classifiers with the ability to generalize and make decisions from large and somewhat ambiguous input data. DL is modeling mechanisms particularly skilled in solving nonlinear problems. Generally DL algorithms with very deep layers are very essential for chest X-ray images classification because the features on the chest X-ray images are similar, so to differentiate the features accurately, the depth of the layer is the major parameter. Deep learning outsmarts traditional CAD systems (rule based, hand crafted ML) because of its ability to execute feature of an object by itself, learn categories incrementally through its hidden layer architecture, it's powered by massive amounts of data, and solve the problem end to end. From the state-of-the-art deep learning algorithms used in medical image classification Xception performs way better than other models in X-ray image detection [17]. Recently Xception is radically applied in medical image classification, for instance Breast Cancer Diagnosis, skin cancer classification, and diabetic retinopathy detection.

1.7. Statement of the Problem

Many people suffer from chronic respiratory conditions globally due to the toxic effects of biomass fuel consumption, outdoor air pollution and tobacco smoke. The complex nature of manifestation of lung diseases on chest X-ray images makes the diagnosis process of lung cancer, COPD, pneumonia, TB and pneumothorax problematic through visual inspection. This manual diagnosis method is highly dependent on the knowledge and experience of the experts assessing the images which may lead to subjective diagnosis errors and cause other medical complications on patients. Limited number of expert radiologists and resource, especially in the developing world, makes the challenge severe, time consuming and tedious task.

Several CAD systems have been proposed to address the shortcoming of the manual system. However, most literatures are found to have limited accuracy which is intensive in medical areas, computational expensiveness, small dataset usage, and problems on performance analysis of the model (most studies compare the performance of the algorithm with that of radiologist rather than other models). Moreover, most literatures do not consider of the lung diseases. In addition, there is no or little attention given to automate the diagnosis system in developing countries, especially in Ethiopia.

Therefore, lack of automatic, simple and reliable lung disease classification system is the great challenge in today's health care. As a result of this, a sensitive, accurate and automatic system is required to classify lung diseases from chest X-ray images and help to reduce the heavy workloads of radiologists and reduce misdiagnosis.

1.8. Objectives

1.8.1. General Objective

The main objective of this thesis is to design and develop an automated system that can classify lung diseases including Lung cancer, COPD, Pneumonia, TB, and Pneumothorax from chest X-ray images using Deep Learning.

1.8.2. Specific Objectives

The specific objectives of this thesis are:

- To perform pre-processing on the collected chest X-ray images.
- To fine-tune Xception model for 6 class classification.
- To classify lung diseases using Xception model.
- To evaluate the performance of the system.
- To develop Graphical User Interface for ease of use of model implementation.

1.9. Motivation

I preferred to do my thesis on lung disease classification using chest X-ray images because of several reasons; the main reason is that, lung disease is the most frequent cause of death in patients in all ages. Beside chest radiographs are widely used in the evaluation and diagnosis of lung diseases. The motive to automate the manual diagnosis techniques starts by interviewing a radiologist and the challenges that they face during the diagnosis time like, misdiagnosis of diseases that happens due to shortage of high level expertise and the fact that the diagnosis procedure is time consuming and tedious. Even in Jimma University Medical Center there is only one expert radiologist to manage all imaging modalities. In addition to this the occurrence of the disease is frequent enough in X- ray reports to provide reasonable test set, and Chest radiography is easily affordable medical imaging modality. Even in low resource areas, modern digital radiography (DR) machines are available. These make the problem a hot issue and motivated this thesis research.

1.10. Significance of the Thesis

This study mainly focuses in designing and developing an automated system that can diagnose lung diseases based on an accurate and reliable algorithm. It will also provide easy diagnosis system for physicians. This multi class classification helps to classify different diseases of the respiratory system.

The system reduces the misdiagnoses of the diseases due to over fatigue of radiologists, reduces diagnosis time, determines the type of the disease and contributes towards on time and proper treatment. Additionally, the approach offers a simple user friendly system for diagnosing the diseases.

1.11. Scope of the Thesis

The scope of this research was to diagnose and classify lung diseases into 6 classes based on the analysis of chest X-ray images of patients. The images were acquired from the chest of patients with posterior anterior position.

CHAPTER TWO

2. Related works

Manual decision about status of a disease from chest X-ray image is subjective and the result might vary from expert to expert. To improve this, CAD systems has arrived. In diagnosing lung disease, researchers have been doing a lot by developing several computer-aided diagnosis systems to improve the health care services. Different image processing and classification of chest X-ray images have been proposed in the literature. In this section some of recent automated classification methods are discussed.

2.1. Review of Automatic Diagnosis of Lung Diseases from Chest X-Ray Images

Due to complexity of formalizing the judgment for automatic detection of lung pathologies based on images, most researchers use CAD systems. It requires creation of a reasonable training set which consists of images with labeled lung diseases. The wide application of chest X-rays and the complexity of reading them make CAD systems a hot research topic. The concept of CAD for chest X-rays has made fast progress from Rule-based (RB) prediction to ML approaches and up graded to, now, DL [16, 17, 18].

Several studies have been done to diagnose lung abnormalities using rule –based systems [19, 20, 21, 22, 23] and classify chest X-ray images as normal or abnormal based on the knowledge that is represented in the form of if-then rules and Membership Functions (MF). Rule based system is a mathematical tool for dealing with the uncertainty and the imprecision typical in the medical field. The reasoning is based on compositional rule of inference and the knowledge of specialists is important to determine the parameters. Farahani et al. [21] proposed four modules: working memory, knowledgebase, inference engine and user interface to classify lung cancers using fuzzy rule based expert system. Moreover, a smart system for the early detection of lung diseases is developed using Adaptive Neuro Fuzzy Inference (ANFIS) and waterfall model [22]. ANFIS that has been built for this application consists of four inputs and one output, with 81 rules. At the ANFIS development stage, several ANFIS architectures have been tested using MF type and different optimization methods. In addition to this, implementation and applications of a Belief Rule Based Expert System (BRBES) [23], that allows the measurement of the level of suspicion of TB by taking into account of its various signs and symptoms has been developed. It is tracked by the presentation of the knowledge-based construction as well as a description of the BRBES's interface. The knowledge-based construction consists of developing a BRB tree by identifying the necessary antecedent and consequent attributes and an interface can be defined as a media, facilitating the users to interact with the system. This system requires more memory and processing time.

The RB system demands deep knowledge of the domain as well as a lot of manual work, is time consuming, generating rules for a complex system is challenging and tedious, has less learning capacity in that: the system will generate the result as per the rules so the learning capacity of the system by itself is much less, and if an application that we want to build is too complex, building the RB system can take lot of time and analysis. Complex pattern identification is a challenging task in the RB approach.

ML computing approaches enable us to build a more powerful intelligent decision-making system by using the advantages of an artificial neural network for chest X-ray image classification [17, 24, 25, 26]. It is suitable for classification of chest X-ray images by which features are extracted from complete image or, more typically, regions in the image, and a computer is trained to classify feature vectors. Studies have been done to classify chest X-ray images using SVM classifier based on two cascaded classifiers [26] to classify the CXR image as TB and non TB. In this study combinations of various classifiers can be found more often than a single classifier. Likewise, in order to increase efficiency of classifiers, dimension of feature set is reduced by means of principal component analysis, kernel principal component analysis, linear discriminant analysis, nonlinear discriminant analysis or sequential feature selection.

ANN has been also employed for lung and pneumonia classification [19]. Even though, the method detects both lung cancer and pneumonia diseases, by using Gabor filter as a feature extractor ANN for classifier with improved accuracy, it can classify only two diseases. And the source of the data specified make it less applicable. Another methods using Feed forward neural network [27] was also proposed by enhancing the classification from binary to multi class lung cancer, pneumonia, and TB using statistically and geometrical features extracted from X-ray images. But the source as well as the amount of the data and also the accuracy was not specified. SVM based classification with texture features [28] was proposed to solve the problem related with accuracy, and it achieved 96%. The problem was solved by using joint PCA-SVM classification [29].

Those studies reported that the ML approach in chest X-ray images classification based on extracted features affects classification performance, and it has limitations like each application needs to be specially trained, require large amounts of hand-crafted, structured training data, learning must generally be supervised: training data must be tagged, and do not learn incrementally or interactively.

In addition to the above RB and ML methods, several studies have been carried out using the recent state of the art DL based approach for abnormality detection in chest X-ray images was introduced [20] by using the three neural networks GoogLeNet, InceptionNet, and ResNet to diagnose lung diseases. However, the classification was binary (normal or abnormal), and the accuracy was not specified. Residual Inception module has been used to classify TB and pneumonia [30]. The system offered accuracy of 96% being still binary classification task and it lacks further classification and, amount of data was not exactly specified, and the method is not clear enough. Another study has also been proposed for use in chest pathology identification using deep feature selection with non-medical training using CNN that was trained with ImageNet to classify the pattern of the diseases [31]. However dataset was very small (93 chest X-ray images), and the accuracy was not indicated on the paper. Furthermore CNN has been used to classify general opacity, diffuse lung opacity, cardiomegaly, abnormal hilar with an accuracy of 86.7% [32]. This work tried to classify any abnormality found on chest X-ray images, and has relatively low accuracy. In addition to this DCNN has been used to classify 8 pathologies [33]. However the accuracy was not indicated in the paper. On the other hand, recent studies have revealed pneumonia detection task using CNN model trained from scratch. This system achieves 93% accuracy but it fails to categorize other lung diseases which have high prevalence and the source of dataset was not clearly specified [34].

CheXNet model, a dense convolutional neural network trained with ImageNet which consists of a classification branch and an attention branch has been introduced to classify 14 thoracic diseases [35]. The study used chest X-ray images from chest X-ray14 online datasets for classifying pneumonia and it was extended to classify 14 thoracic diseases. But, the outcome of the study was not clearly defined that means the training accuracy, validation accuracy, specificity and sensitivity of the model ware not defined, and it only compares the algorithm with the radiologists using F1score. Similarly, CheXNeXt which is an improved version of CheXNet which also consists of pre-trained DenseNet CNN that has been introduced to classify 14 thoracic diseases using chest X-ray 14 publicly available images [36]. The authors considered both lateral and frontal images to train their system. Even if they perform specificity and sensitivity analysis for each disease type, still the model validation was not clearly defined. On top of this, multi-task classification on 12 different thoracic abnormalities classification has been done using dense net pre-trained model [37]. However, performance of the model not well specified. Another study [38] tried to classify four main thoracic diseases using DenseNet and it can localize the diseased area. The authors used only experimental data, and the result may not be applicable on real data with multiple abnormalities.

Apart from this, a new DL algorithm known as Lunit has been proposed in the [39], and the algorithm was designed to classify chest radiographs of patients with four major thoracic diseases. But, the study has several limitations. First, the study was performed at a single institution (labeling was done at the institution) and thus it is unknown whether the performance of the algorithm is reproducible in different institutions or not. Second, because of the retrospective nature of the study design, the effect of the algorithm on a real-time clinical workflow was not evaluated. Third, it compares the performance of the algorithm with that of on-call radiology residents, rather than experienced radiologists

Even though several automated systems have been developed to minimize human intervention and errors related with human knowledge gaps, still there are many problems that remained unsolved. The algorithms in most studies have not yet been fully validated and there is a problem on performance analysis of the models (almost all the studies compare the performance of the algorithm with that of radiologists rather than other models). Moreover, most studies were carried out on pneumonia and pulmonary tuberculosis even though there are various pathologies and abnormalities in real-world clinical practice. Generally, most of the recent works focused on designing binary classification systems, the validation accuracy of most papers was not specified, small number of datasets were used, and some of the studies done in this area achieved limited accuracies.

In this study, it has been proposed an accurate, reliable and automatic system with improved accuracy, which classifies multiple independent (Lung cancer, COPD, Pneumonia, TB, and Pneumothorax) lung diseases. The study introduced the most recent model of computer vision technology over the lung disease classification including image processing algorithms

CHAPTER THREE

3. Materials and Methods

In this chapter, different procedures that are developed for the achievement of the research outcome will be briefly demonstrated including the experimental setup, materials, and methodology of the research. Figure 3.1 shows the overall block diagram of the procedure used in this study.

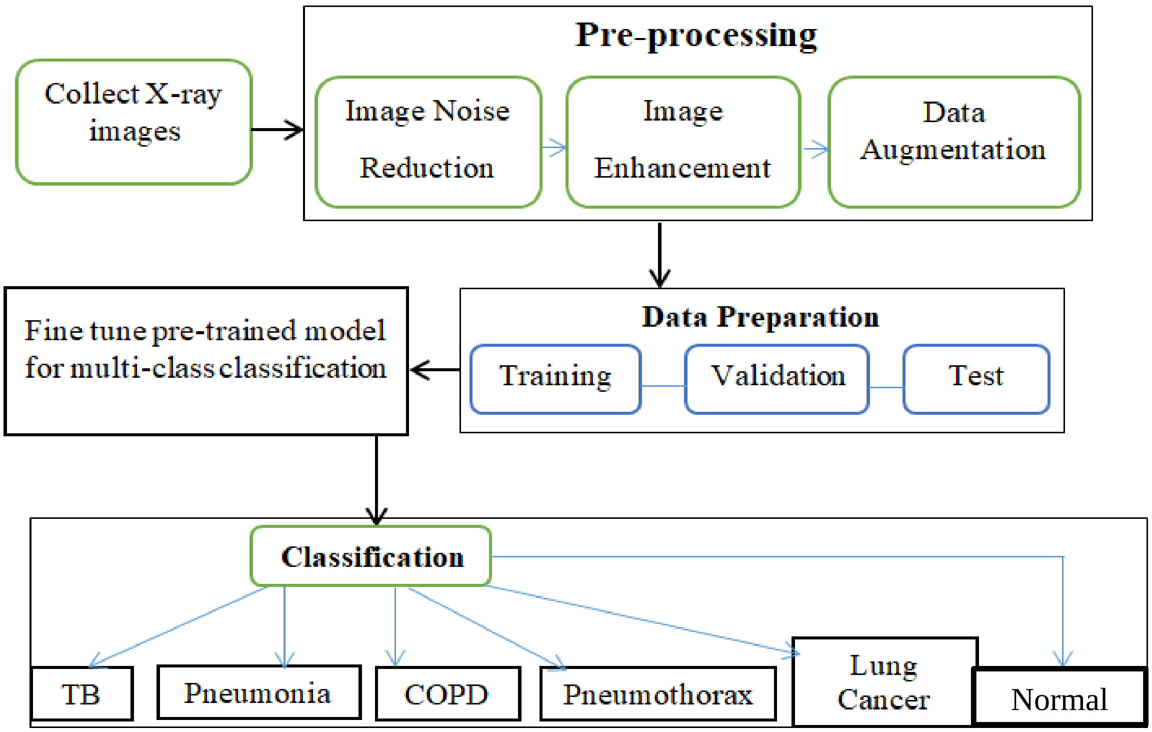

Figure 3-1: General block diagram of the methodology of the research.

3.1. Image Data Collection

The first step in this research was collecting proper sample images. Local chest X-ray images and online datasets were collected to train, validate and test the model. 11,716 images were collected from National Institute of Health (NIH) dataset repository [40] with known labels.

Additionally 443 images were collected from Jimma University Medical Center (JUMC) radiology department cross checked with the medical history of the patients for proper labeling of the diseases with the age range of 10-55 years. The total amount of data with its corresponding class is illustrated on Table 3-1.

Table 3-1: The total amount of dataset used for the study

Diseases type	**Data from NIH**	**Data from JUMC**
Lung cancer	1823	12
COPD	717	9
Pneumonia	3875	153
TB	1680	110
Pneumothorax	2038	19
Normal	1583	140
Total	**11,716**	**443**

A total of 12,159 CXR image data was collected from online database and JUMC radiology department. Figure 3.2 represents sample data of the images.

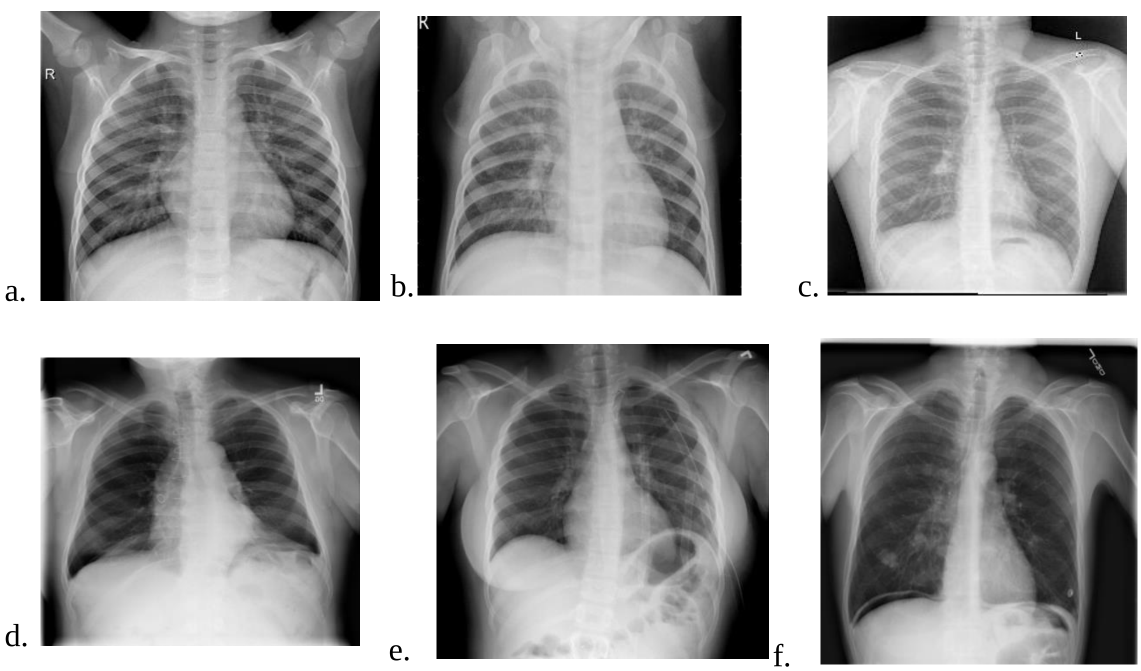

Figure 3-2: Sample images from the datasets. (a) Normal, (b) Pneumonia, (C) TB, (d) COPD, (e) Pneumothorax and (d) Lung cancer

3.2. Image Pre-processing

The main purpose of pre-processing is to enhance the quality of the images. The quality of the pre-processing method has a large influence on the performance of the classification task, which is used to boost the precision and interpretability of an image [41]. In chest X-ray image analysis, pre-processing is done to remove noises from the image and enhance the image for extracting essential information for further analysis.

Noise is undesired information that affects the quality of data or image. In X-ray radiography, usually images are in the form of gray scale. Besides, the statistical nature of X-ray emission implies that not every pixel will exactly detect the same photon to others, some pixels have more X-ray and appear darker, where as some pixels get fewer X-ray photons and appear brighter. However, this distribution of pixels is random and it has a shaded appearance which results in a salt and pepper type of distribution. Generally X-ray images are dominantly affected by salt and pepper known as impulse noises. In addition to salt and pepper noises, X-ray images are affected by speckle and Poisson noises [42]. The purpose of image pre-processing was, therefore, to remove unwanted objects and noises from the chest X-ray images so that it becomes ready for the subsequent image processing tasks. Before processing the images, the pixels intensity was adjusted as images have different contrast values.

3.2.1. Image De-noising

There are various de-noising techniques including Weiner filter, Gaussian filter, Median filter, and Mean filter. Median filter is the most effective filters used for removal of salt and pepper noises as well as speckle and Poisson noises on X-ray images [41, 42]. The de-noising step was required to reduce the noises in the X-ray images and perform a more accurate diagnosis of lung disorders by improving the quality of the images. The comparison shows the median filter proves better than the other filters to deal with noises in X-ray images. Table 3.2 below shows comparison of different de-noising techniques.

Table 3-2: Comparison of different pre-processing methods [43, 44].

S.no.	Method	Advantage	Disadvantage
1	Median filter	It is popular non-linear filter for removing salt and pepper noises, speckle, and Poisson noises.	It is difficult to treat analytically the effect of a median filter.
2	Mean filter	Easy to implement. Useful for removing salt and pepper noises.	If images are highly noisy it can remove image details.
3	Gaussian filter	Useful for removing noises and blur from the image.	If it is used alone it may blur the edge.
4	Wiener filter	Most important linear filter to remove noises and to de-blur images	If the amount of blur is very high, the wiener filter may not be much effective.

Median filter

The median filter (MF) is the most popular nonlinear filter used for removing noises while preserving edges, and it is particularly effective at removing 'salt and pepper' type noise, speckle, and Poisson noises [45]. It works by moving filter window pixel by pixel, replacing each pixel with the median value of neighboring pixels. Median value is calculated by first sorting all the pixel values from the window into numerical order, and then replacing the pixel being considered with the middle pixel value. The original value of the pixel is included in the computation of the median. Median filters are quite popular because, for certain types of random noise, they provide excellent noise- reduction capabilities, with considerably less blurring than linear smoothing filters of similar size [46]. The median is a more robust average than the mean and so a single very unrepresentative pixel in a neighborhood will not affect the median value significantly. Figure 3.3 demonstrates implementation of a 3 × 3 median filter.

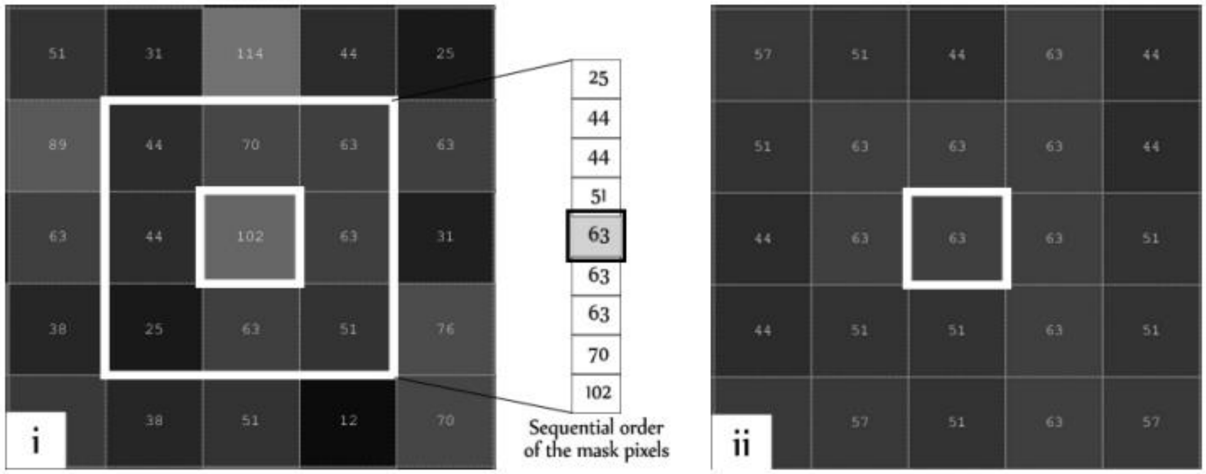

Figure 3-3: Replacing the gray value of the center pixel with the median of the mask. The larger white square represents the filter mask and the center white square represents the pixel that needs to be replaced (i). The output image after the center pixel value, 102, is replaced by the median of the mask, 63, (ii).

3.2.2. Image Enhancement

After the noise removal process was done, further enhancement was performed using Histogram processing (histogram equalization), which is one way of image enhancement procedures. In this study, the histogram equalization was used to increase the content of visual information. Histogram equalization (HE) is one of the common methods used for image enhancement. This technique enhances the contrast of images by which the range of the histogram is increased. It accomplishes this by effectively spreading out the most frequent intensity values; it is the uniform distribution of gray-level of an image stretching out the intensity range of the image. This method usually increases the global contrast of images when its usable data is represented by close contrast values. This allows for areas of lower local contrast to gain a higher contrast and it may produce images which do not look as natural as the input ones. Redistributing the highest peaks over the lowest histogram peaks using HE over enhances the lowest peak gray-levels or/and under enhances the highest peak gray-levels and leads to a totally different output images. To overcome these drawbacks different brightness preserving techniques are used for image enhancement. Adaptive histogram equalization (AHE) resolves the limitation of HE by reducing the problem domain locally assuming the regions (tiles) have somewhat ideal distribution of histogram at least, the big difference between the highest and the lowest histogram peak will be reduced by a considerable degree. Even though AHE improves the contrast of the image under question better than the standard HE, still it has its own drawback. Most of the time background pixels share majority of the count; they might over enhance the background noise [47].

Contrast Limited Adaptive Histogram Equalization (CLAHE) is the improved version of AHE. CLAHE operates on a small region on a gray scale image, and its contrast is limited [48]. It was developed to prevent the overall amplification of noise and affecting other wanted regions that adaptive histogram equalization can cause [49]. These technique follows three procedures, mainly (CLAHE is depicted on Figure 3.4) first, the image need to be partitioned into tiles (rectangular contextual regions). Then, the histogram equalization enhancement technique is applied on each and every tile of the image. Lastly, to avoid the visibility of boundaries between tiles, bilinear interpolation is computed at the boundary pixels.

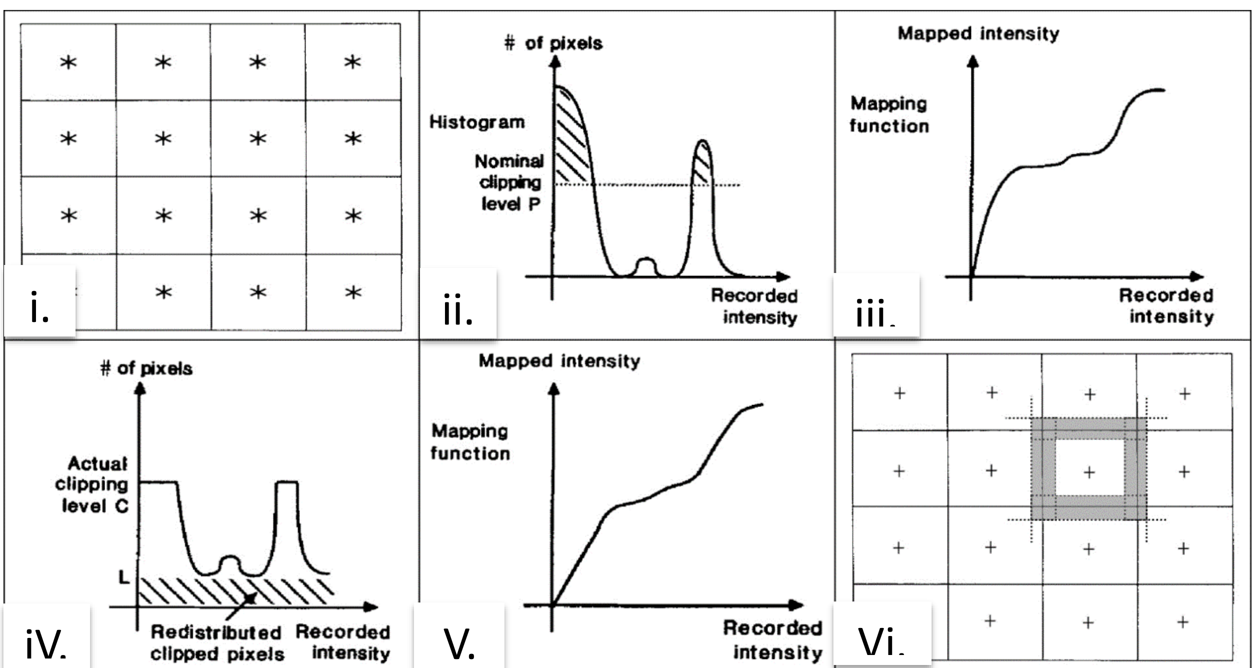

Figure 3-4: Contrast limited adaptive histogram equalization (CLAHE): contextual regions or tiles (i), histogram of the original image and its clip limit (ii), cumulative histogram before clipping (iii), clipped histogram (iv), cumulative histogram after clipping (v) and linear interpolation of tile border pixels (vi).

3.2.3. Image Augmentation

Limited data is a major challenge in deep learning models for image classification. Often, imbalanced classes can be the related difficulty; while there may be sufficient data for some classes, equally important, but under sampled classes will suffer from poor class-specific accuracy. There are many ways to address complications associated with limited data and imbalanced classes. Image augmentation is one useful technique that can increase the size of the training set without acquiring new images [50]. Data augmentation that includes rotation, flipping and random cropping, has been employed to increase the number of data. As a result of this, important and large amount of data can be reproduced representing the same class as the original image to allow the neural network to learn features of an image properly. In this thesis work, after data augmentation is applied on the images, the total amount of dataset increased from 12,159 to 17,751 images.

In addition to this, the images were scaled to uniform size as needed by the model that classifies the images, and to decrease the computational time. As a result, all data were resized to 299 × 299 image sizes from the original image size of 1066 × 1066.

3.3. Image Classification

The act or process of categorization or grouping of images/objects inside images into categories according to established criteria like features is known as classification. Image classification refers to a process in computer vision that can set apart the image into the prearranged category based on the features on the image [51]. That means taking an input (like a picture) and outputting its class. Image classification came into existence for decreasing the gap between the computer vision and human vision by training computer with the input data. Image classification in machine learning consists of feature extraction module that extracts the important features such and a classification module that classify based on the features extracted. The main limitation of machine learning is, there is a separate feature extraction and classification process, and also it can only extract certain set of features on images and unable to extract differentiating features from the training set of data. This disadvantage is resolved by using deep learning [52].

Deep learning is a name for an algorithm in machine learning like neural network but the difference in DL is the network architectures, and the feature extraction is done automatically in the different layers of the network. This means there is no handcraft feature extraction step before classification. It has deeper layers, which means it has a lot of hidden layers. So, its performance is better than neural network to classify images.

Deep learning is preferable for chest X-ray image classifications due to the fact that image features are similar for different diseases and in order to classify diseases properly, it requires a very deep network [53]. Convolutional neural network (CNN) is one type of deep learning, which is highly applicable in medical image classification [50,51]. The current state of diagnostic performance by deep learning algorithms for medical imaging compared with healthcare professionals shows deep learning models matching or exceeding humans in diagnostic performance [56]. When a network detect images, the network learns the feature in a step by step manner, the first layer detect edges, the second layer some basic shapes, and it goes into deeper complex features and finally the network classifies the pattern based on the features. Deep learning techniques includes: CNN, recursive neural network, recurrent neural network and pre-trained network. Those are highly applicable in medical image classification problems.

3.3.1. Transfer learning

Transfer learning is a term that refers to passing weight values of a trained neural network to another new neural network, so that building and training a network from scratch will be avoided. This can be achieved by using pre-trained models (trained on large dataset) and thus it saves computational time and computational power. There are two major ways that transfer learning used for re-purposing: Fine tuning a CNN and CNN as a fixed feature extractor. Fine tuning a CNN use the weight of the pre-trained model instead of randomly initializing them, and train like normal. Fine tuning techniques can be done either by training all the entire model or train some layers and leave others frozen. CNN as a fixed feature extractor method is usually done by maintaining all the weights of the CNN with the frozen convolutional base and training only the final layer.

Different potential models are available for lung disease classification from X-ray images. Among them, VGG (VGG-11, -13, -16, and -19), Inception (Inception-v1, Inception-v2, Inception-v3, Inception-v4), ResNet (ResNet -18, -34, -50, -101), Xception, and DenseNet (DenseNet -121, -169, and -201) types of pre-trained models are mostly used. In order to perform a comparison between the models, multiple performance indicators are useful for a comprehensive benchmark of models, specifically, classification accuracy, model complexity (by counting the total amount of learnable parameters), memory usage (the memory allocated for the network model and the memory required while processing the batch), computational complexity (the computational cost), and inference time (same batch size). Generally Xception pre-trained model (extreme version of Inception family) is less complex with top accuracy rate [57]. Model complexity is the major consideration because of limited infrastructures like memory space and GPU. Figure 3.5 below shows the performance of Xception is high compared to other models.

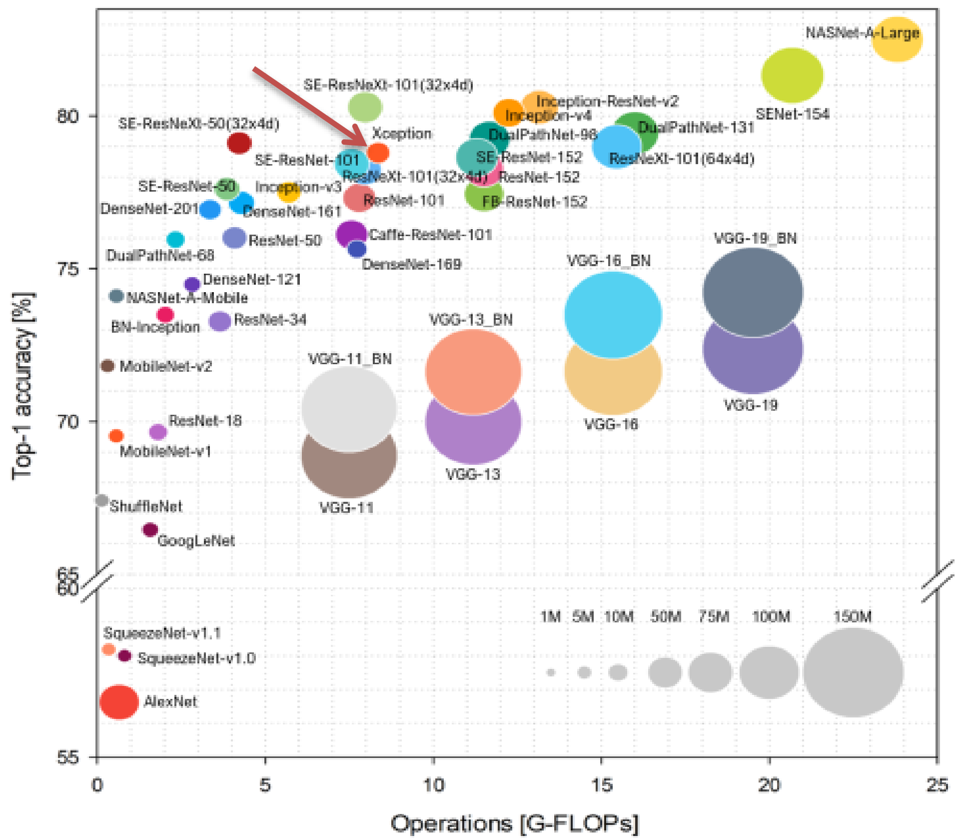

Figure 3-5: Comparative analysis of different models. The Figure illustrates Xception model found on the top of the others with accuracy and the size of the shape indicates the weight of the model, demonstrated Xception being the smallest weighted model [57].

3.4. The Xception Model

Xception stands for Extreme Inception, and is proposed by Google researchers Francois Chollet in April 2017 [58]. It is an extension of the inception architecture which replaces the standard

Inception modules with depth wise separable convolutions. It is even better than Inception family. It is the latest version of convolutional neural network and one of the smallest weighted models. And also it is 71 layers deep and requires an image input size of 299×299.

Xception is a deep convolutional neural network architecture that is made up of modified depthwise separable convolution blocks and Maxpooling layers with residual connections. Xception reduces the high computational power and expensive operation of the standard convolution by using modified depthwise separable convolution. Modified depthwise separable convolution has two steps, to break it down into little parts, point wise convolution followed by depthwise convolution. The first step perform convolution to a single input channel at a time, unlike a standard convolution that applies convolution on all channels. The second step performs linear combination of each of this layer. This reduces the number of parameters in a convolution, and computational power because multiplication is an expensive operation relative to addition, so the main aim is reducing the multiplication, that means the number of connections are fewer and the model is lighter [58]. Basically modified depthwise separable convolution removes the presence of non-linearity. Figure 3.6 briefly describes the separable convolutions for Xception model.

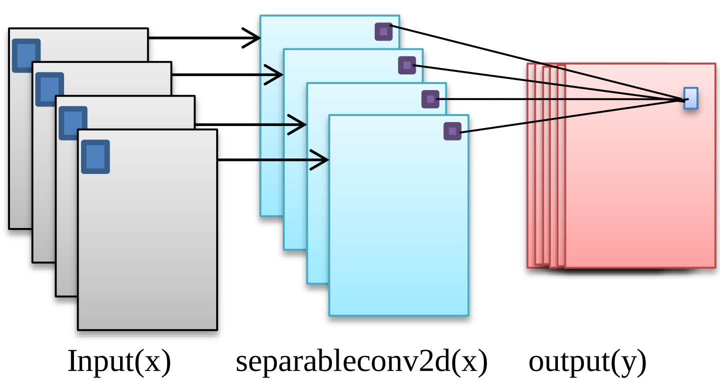

Figure 3-6: The convolutional layer of the Xception model, first applies point wise convolution on the input and then apply depth wise convolution for each [59].

Maxpooling layer is a pooling layer that provides dimensionality reduction of an image size which is width and height of feature maps. This reduces the computational power requirement of the system. After performing the initial convolution with 36 convolutional layers, all of which have linear residual connections around them, max-pooling the network will begin with residual blocks. Residual connection known as skip connection is used to fix the problem of degradation problem and the inability of a deep neural network [60]. Degradation problem is a problem caused by the network depth, increasing the number of layers causes the accuracy will start to saturate and eventually degrade. The shallower networks perform better than the deeper counterparts that have few more layers added to them. So, skip these extra layers is the solution for degradation problem. It has batch normalization and factorization. Batch normalization allows each layer of a network to learn by itself a little bit more independently of other layers, and factorization is reducing the number of connections without decreasing the network efficiency.

Generally Xception convolutional neural network has two parts; the first is the convolutional base, 36 convolutional layers forming the feature extraction of the network, which is composed of the modified depthwise separable convolutional and pooling layers. And the convolutional layer is responsible for extracting the features from an image. Whereas, the second part of the model is the classifier. It is composed of the fully connected layers having 1000 neurons (ImageNet class output). In the current thesis study Xception pre-trained model was adapted and fine-tuned to classify chest X-ray images into multiple classes.

Hyper-parameters of the network

Hyper-parameters are parameters which need to be set before training a network to get a fine result and once they are set they cannot be changed during training. Choosing appropriate hyper-parameters is important in the success of neural network architectures. It can be divided into two categories: Optimizer hyper-parameters (related to the optimization and training), and Model Specific hyper-parameters (related to model structure) [61].

In this thesis, optimizer hyper-parameters which include epoch, batch number, learning rate, and model hyper-parameters that includes number of layers, hidden unit and hidden layers have been used to get good performance of the system. These includes, choosing the right optimizer, adjusting the learning rate, choosing the appropriate activation function and choosing the proper loss function.

Batch size

Batch number represents the number of images that can be processed in one full iteration. It has an effect on the resource requirements of the training process, speed, number of iterations and depending on the GPU resource. It is possible to limit the batch size from

1 to any number of images. The batch size has an effect on the convergence of losses due to the number of images processed in one iteration. The typical strategy is to start from smaller batch sizes, increasing gradually until the loss converges to acceptable values [61]. In the case of this research, the batch size used was 32.

Epoch

The epoch of a network is determined by batch number. Subdivision and size of the training set refers to one complete cycle (model sees all images one time) used in the training. It is a single training cycle, which contains many steps depending on the data and configuration. After all training samples are finished or seen by the network, one epoch has elapsed, and then, the second set will be started at second epoch. In short, one epoch represents one full cycle of the network analyzing all training samples. In this study, epoch was set to 50.

Learning Rate

It represents how much weight of the network is adjusted with respect to the loss gradient. If it is small, then the speed of convergence will be small, but the loss will be more stable. If the value is larger, then the speed along the downward direction is higher but the training loss will be not stable, so electing proper learning rate according to the situation is very important. It is recommended to start by higher learning rate, then decreasing gradually to make the loss stable and converge to the ideal value. In this thesis, learning rates from 0.01, 0.001, 0.0001 and 0.00001 were used. If the learning rate is too low, the model will miss the important patterns in the data; conversely, if it is high, it may have collisions. In the current study 0.01 learning rate offered best performance [61].

Optimizer

Optimizers are set of rules used to change the parameters of the neural network such as weights and learning rate in order to reduce the losses and to provide the most accurate results possible. There are different optimizers for optimization techniques including Stochastic Gradient Decent (SGD), adaptive gradient (AdaGrad), adaptive delta (AdaDelta), Adaptive Moment Estimation (ADAM) and Root Mean Square Propagation (RMSProp). ADAM is a replacement for SGD for training deep learning models. ADAM combines the best properties of the AdaGrad and

RMSProp algorithms [62]. It has ability to get loss of the network easily and fast convergent algorithm of optimization.

Activation function

Activation function refers to functional mappings between the inputs and response variable. The most appropriate activation function is Rectified Linear Unit (ReLU). Training a network by using ReLU would be faster, and it is more biologically inspired [63]. Mainly ReLU removes the problem of vanishing gradient.

Loss function

Loss function or cost function known as error function is a function that maps an event or values of one or more variables onto a real number intuitively representing some cost associated with the event. It is dependent on the model's parameter. Loss functions play important role in network training [64]. An optimization problem seeks to minimize a loss function.

3.4.1. Network Configuration for Xception

The network is configured according to the data size, number of class, nature and performance of the model by using specific arrangement of the layers and nodes in the network (architecture) of Xception. The activation function for each layer is ReLu because it does not activate all the neurons at the same time. Dropout introduces regularization within the network to reduce over-fitting. This ultimately improves generalization by randomly skipping some units or connections with a certain probability. As an optimizer ADAM optimizer was chosen for its best performance in terms of speed to converge and accuracy [61]. Learning rate was 0.1 and reduced manually two times by 1/10 until value accuracy plateaus. Loss function for multi class classification was categorical crossentropy. On top of it flatten layer followed by Softmax function was used for multi-class classification model. It returns the probabilities of each class. Before training is started the model is useful to define two callbacks. That is model checkpoint and early stopping. Model checkpoint: when training requires a lot of time to achieve a good result, often many iterations are required. In this case, it is better to save the best performing model only when an epoch that improves the metrics ends. Early stopping: sometimes, during training we can notice that the generalization gap (i.e. the difference between training and validation error) starts to increase, instead of decreasing. This is a symptom of over fitting that can be solved and often a practical and efficient solution is to stop training when the generalization gap is getting worse.

3.4.2. Data preparation

Before feeding the data to the network, dataset split was done for training and testing. Split data into train: test randomly; well-known rule of splitting the data is 80–20 percent training and testing sets respectively [65]. Among the 20%, 10% was used for validation, and 10% for test. A training set is used to train the network while a validation set is used to monitor the model performance during the training process, to fine tune hyper-parameters, and perform model selection. Finally a test set is used once in order to evaluate the performance of the final model. Figure 3-7 shows the percentage of the dataset.

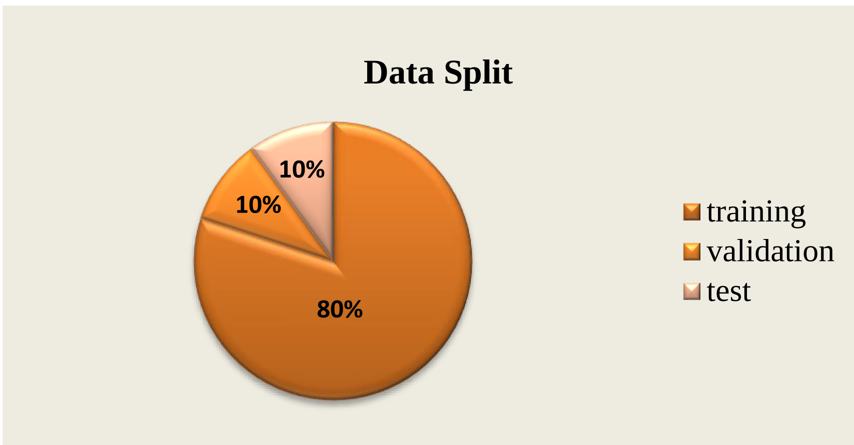

Figure 3-7: Data split scheme used in this study

3.4.3. Training Xception Model for chest X-ray images Classification

Since the Xception model was pre-trained on non-medical images (ImageNet), fine tuning has been done in this thesis. The model was used as a feature extractor and classifier. That is, both its architecture and weight value (from ImageNet) were adapted. Considering the dataset used in this research, since the dataset is different from the pre-trained model's dataset (ImageNet and chest X-ray image) of non-medical image. Initial lower layers of the network learn very generic features from the pre-trained model. To achieve this, initial layers weights of pre-trained models were frozen and not updated during the training. Higher layers are used for specific features. Higher layers of pre-trained models are trainable or fine-tuned. Finally, by using soft-max regression technique, the probability distribution of each possible class was found and based on the most probable class the images were classified accordingly. For the time of training, the following hyper-parameters have been selected: Batch = 32, Learning rate= 0.01, Epoch = 50. Then depending on the status of the training or convergence of the error, hyper-parameters were changed, especially the batch and learning rate. As the number of steps increased, the learning rate was decreased and batch number was increased to make the loss more stable and converge to the smallest point possible. Loss values were calculated via forward propagation and learnable parameters were updated via back propagation by optimizers.

3.4.4. Performance Analysis of the Xception Model

A test set consisting of 295 images which was 10% of the total image data for one class were given as input for the model to test the performance of the model. Using unseen dataset, the model performance was tested using a confusion matrix. The system saves the best result for the model during training, and after training is completed the model performance was analyzed with the saved weight. By using confusion matrix, the performance measure parameters like, accuracy, sensitivity and specificity were calculated using the formula shown in Equations (3.1, 3.2, and 3.3). Classification accuracy is defined as the percentage of test set samples that are correctly classified by the model. Sensitivity is the ratio of correctly classified right samples to true positive samples and specificity is defined as the ratio of correctly classified negative sample to true negative samples [66].

$$\text{Accuracy} = \frac{TN+TP}{TN+TP+FN+FP} x\ 100\% \quad (3.1)$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} x 100\% \quad (3.2)$$

$$\text{Specificity} = \frac{TN}{TN+FP} x 100\% \quad (3.3)$$

Where, TP = True positive

FP= False positive

TN = True negative

FN= False negative

To generalize the above mentioned methodology part, this research divided the process broadly into 4 stages. Each stage requires a certain amount of time to execute, loading and pre-processing data, defining model architecture, training the model, and evaluation of model performance.

3.5. Graphical User Interface (GUI)

Graphical user interface (GUI) has been developed for our system for ease of use of our technique. To layout the graphics, Qt designer was used, command line utility was also used to convert the design to python equivalent. In this system the GUI contains the components 'Browse button' which is used to browse the image from the saved file, 'Camera selection button' that enables the user to select the camera, 'Process button' used to process the image. 'Result display button' that makes the system to display the result in text and image form, and the 'Exit' button is used to stop the program. Figure 3.8 below represents a snap shot of the developed GUI for this system.

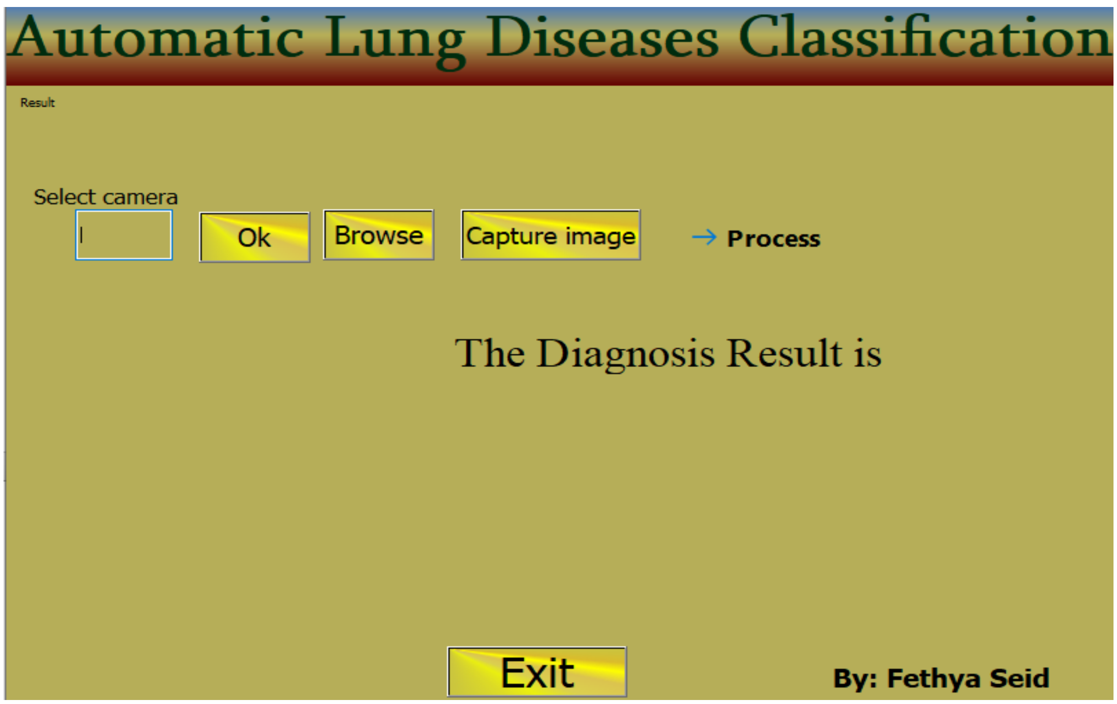

Figure 3-8: General layout of the developed GUI

3.6. Software and Hardware Materials used for this Research

To conduct this research the main materials that were used include: Toshiba Core(TM) i3-8250 CPU @ 1.60GHz 1.80GHz Laptop having 4GB RAM, 64 bit operating system, x64-based processor, window 10. Google Collaborator with free GPU of 12 GB for 12 hours used from cloud computing. Python 3.7.3 software with different modules like Keras, Tensorflow, Sklearn, Pre-processing step and classification with deep learning were performed using python libraries such as OpenCV and TensorFlow for training and testing the whole system. Jupyter notebook editors were used to write and execute the codes offline.

CHAPTER FOUR

4. Results and Discussion

4.1. Results

This chapter presents the results of the pre-processing, training, validation measurement, and evaluation on the test dataset. First, results obtained from pre-processing algorithm are presented followed by results from the training stage. Classification performance of the model based on the performance evaluation metrics which are accuracy, sensitivity, and specificity are presented afterwards.

4.1.1. Pre-processing

4.1.1.1. De-noising

.

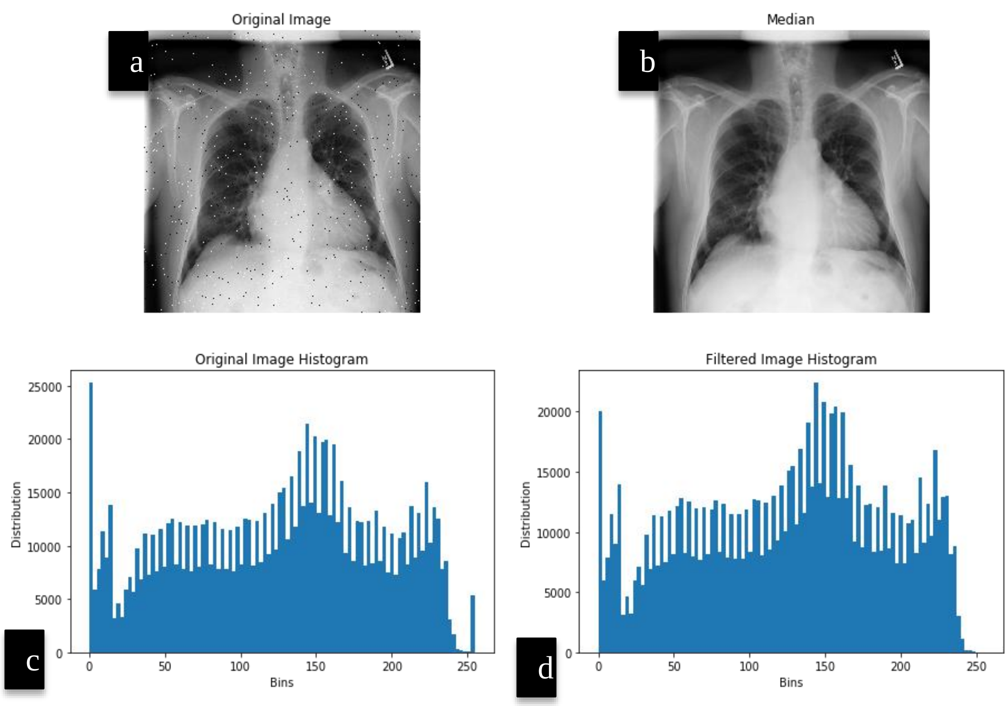

Figure 4-1: Result of de- noising. (a) Original image, (b) Filtered image (c) Histogram plot for original image and (d) Histogram plot for filtered image

4.1.1.2. Enhancement

After removing salt and pepper noise from the CXR images, histogram equalization was applied to enhance the images. First conventional histogram equalization was applied as a result the whole image was enhanced including the lesions. This effect can destroy the important features exhibited by the various abnormalities and leads to a totally different output image. Figure 4.2 illustrates the scenario. AHE improves the contrast of the images better than the standard HE. Most of the time, since background pixels share the majority of the count, they might over enhance the background noise as demonstrated on Figure 4.3.As a result, AHE images suffer from noise problems.

The remedy to the noise is the improved version CLAHE. The scenario is illustrated on Figure 4.4. CLAHE control the pixel value of the output image by clipping the highest histogram peaks and then equally redistribute the clipped pixels over the remaining gray-level ranges.

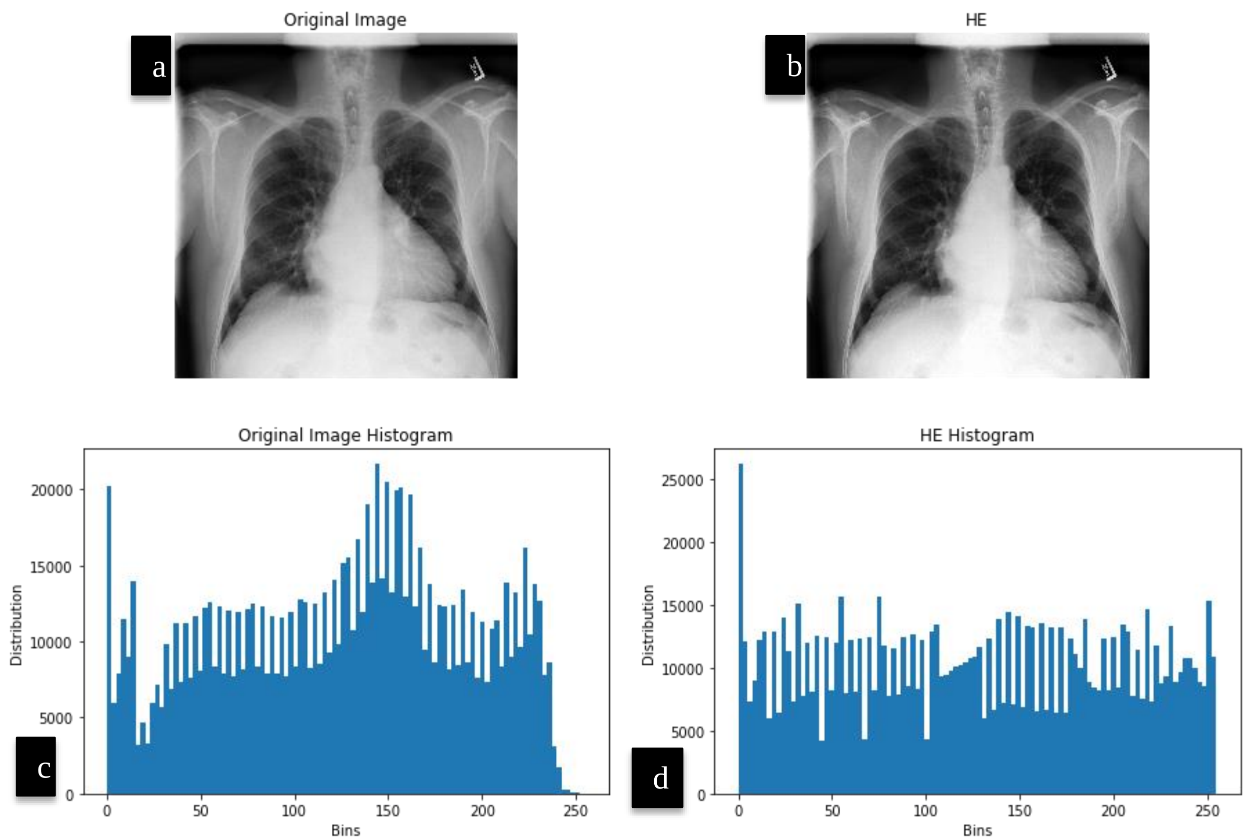

Figure 4-2: Effects of histogram equalization variations on image: (a) Filtered image, (b) Equalized image, (c) Histogram plot for Filtered image and (d) Histogram plot for Equalized image

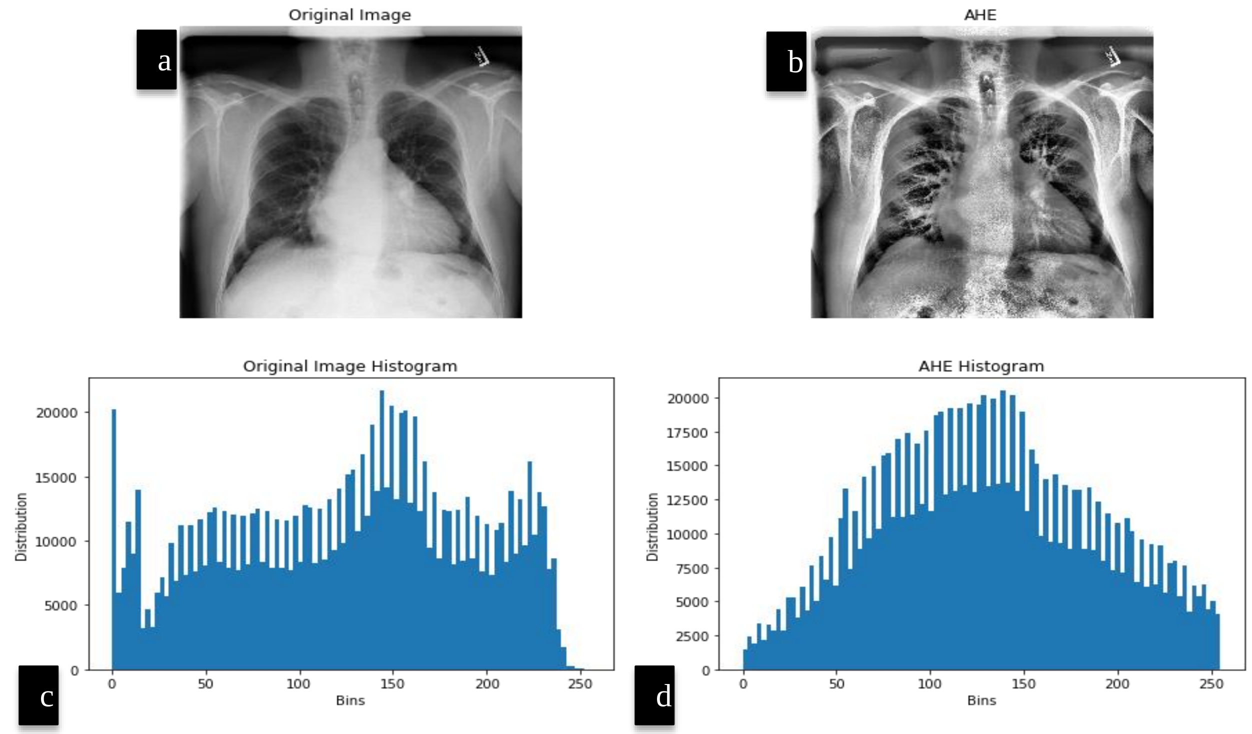

Figure 4-3: Effects of AHE : (a) Filtered image, (b) Equalized image with AHE, (c) Histogram plot for Filtered image and (d) Histogram plot for Equalized image.

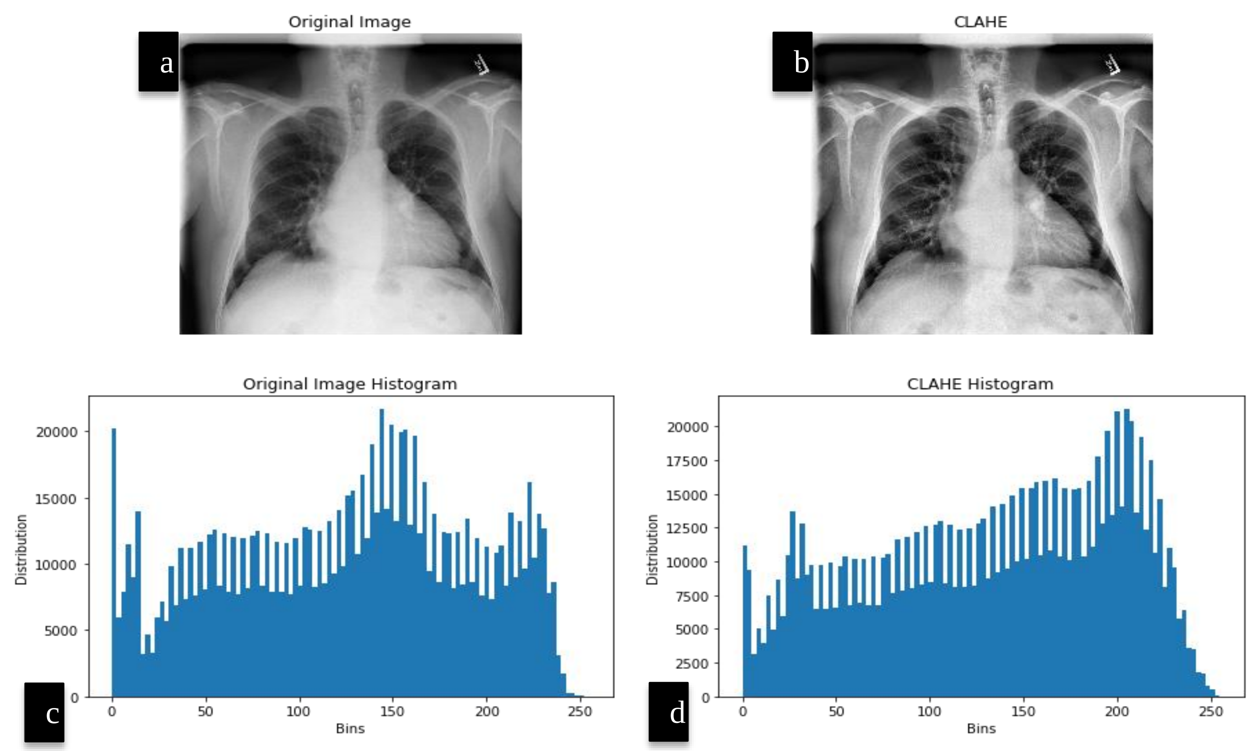

Figure 4-4: Result of CLAHE (a) Filtered image, (b) Equalized image, (c) Histogram plot for filtered image and (d) Histogram plot for Equalized image.

4.1.1.3. Image Augmentation

Figure 4.5 shows sample results from data augmentation. In this research, data augmentation is basically used to remove the problem of class imbalance. Among the augmentation methods, rotation of 45%, 180%, 270% were applied. Both the validation and training data were augmented. After augmentation a total of 17,751 CXR images were obtained. Therefore, the number of images per class becomes equivalent.

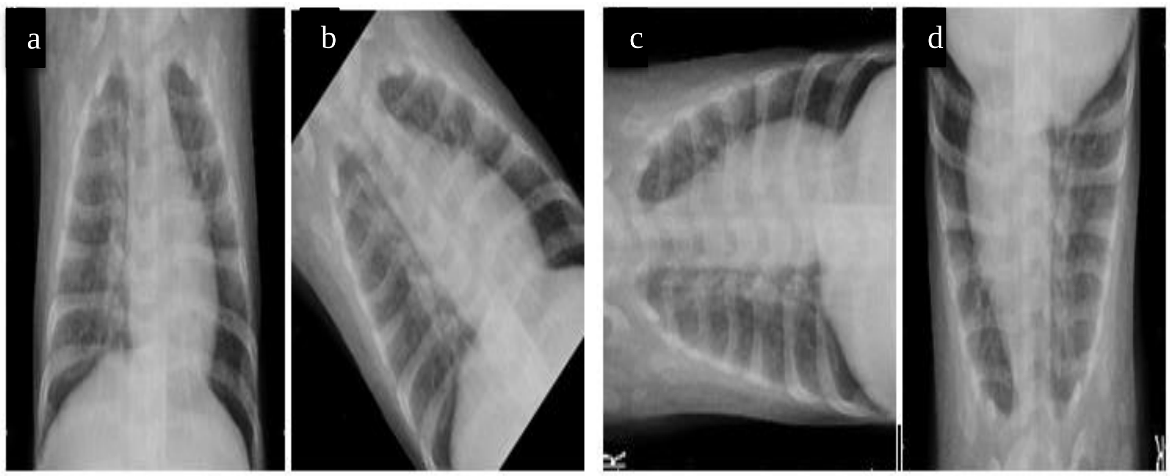

Figure 4-5: Augmentation result (a) Original image, (b) 45% rotated image, (c) 270% rotated image, (d) 180% rotated image

4.1.2. Training Xception for Classification of Multiple Lung Diseases

In this thesis, multi class classification model has been used to classify input images into six classes of normal, pneumonia, TB, COPD, pneumothorax, and lung cancer. The model was trained using the training dataset for all of the six classes having 2360 images per class which is 80% of the image, and validated with the validation dataset of 1770 images. The number of data used for training and validating the system is shown in Table 4.1.

Table 4-1: Description of datasets used for training, validation and testing Xception model

	Normal	**TB**	**Pneumonia**	**COPD**	**Pneumothorax**	**Lung cancer**	**Total**
Training Set	2360	2360	2360	2360	2360	2360	14160
Validation Set	295	295	295	295	295	295	1770
Test Set	295	295	295	295	295	295	1770
Total	2950	2950	2950	2950	2950	2950	**17700**

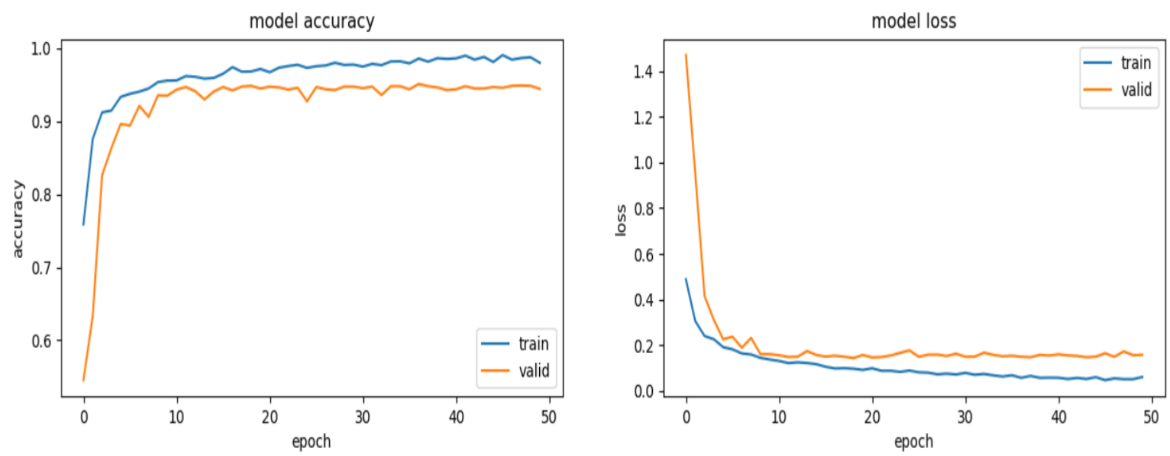

Figure 4-6: Training and validation accuracy using Xception model

Figure 4-6 demonstrates the curve plots of the training and validation scores as the model converges which can be observed from the decrease in loss and validation loss with epochs. Also it is able to reach above 92% validation accuracy in just 10 epochs. The epoch size is 50, learning rate of the model is 0.01. This Learning curve shows the rate of improvement in learning and validating a task as a function of time.

4.1.3. Performance of the Xception Model on Multi Class Classification Task

The performance of the model was evaluated for different performance metrics such as classification accuracy, sensitivity and specificity. Confusion matrix (Table 4.2) was used to obtain the classifier accuracy across different classes.

The test set contains 295 images per class, which is 10% of the total data used. Different accuracies, specificity and sensitivity are obtained for all classification.

According to Table 4.2, 15 COPD images were miss-classified, 6 of them as pneumothorax and 9 of them are as lung cancer. And from pneumothorax class 8 images are misclassified, 5 of them as COPD and 3 of them as lung cancer. From lung cancer class 22 images were misclassified 13 of them as COPD and 9 of them as pneumothorax. Generally, the model gives classification accuracy of 97.3%, average sensitivity 97.2% and average specificity 99.4%.

Table 4-2: Confusion matrix

	Normal	TB	Pneumonia	COPD	Pneumothorax	Lung Cancer	Sensitivity (%)
Normal	295	0	0	0	0	0	100
TB	0	295	0	0	0	0	100
Pneumonia	0	3	292	0	0	0	98.9
COPD	0	0	0	280	6	9	94.9
Pneumothorax	0	0	0	5	287	3	97.2
Lung Cancer	0	0	0	13	9	273	92.5
Specificity (%)	100	99.7	100	98.7	98.9	99.1	Accuracy(%) = 97.3

From the confusion matrix, accuracy of each class was also calculated. The average classification accuracy of the whole classes was 97.3 %. The result shows that the developed model appears promising. Figure 4.7 below shows the accuracy achieved with the developed model for each class. Additionally From the confusion matrix, specificity and sensitivity of each class were also calculated. Figure 4.8 below shows the specificity and sensitivity achieved with the developed model for each class. The sensitivity of TB and pneumonia shows high values. The specificity is generally good for overall classes. The average specificity computed for all classes is 99.4% and the average sensitivity computed for all classes is 97.2%.

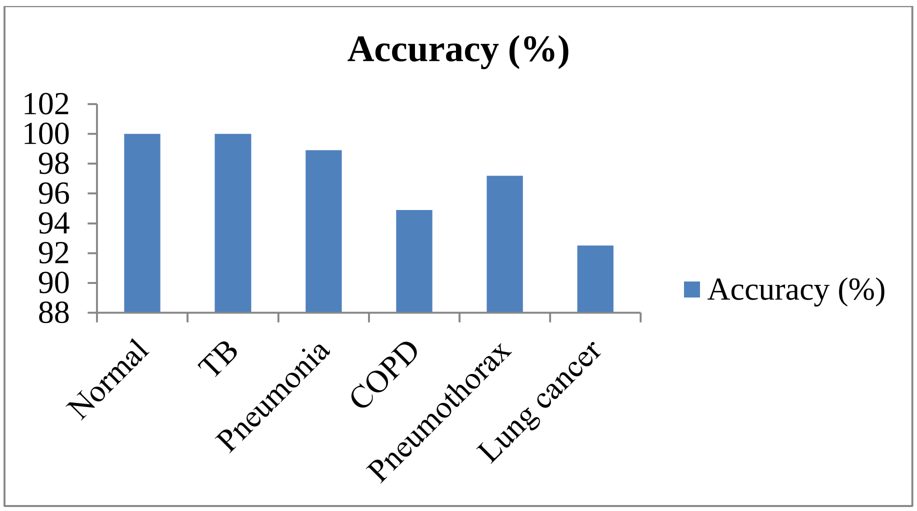

Figure 4-7: Classification accuracy

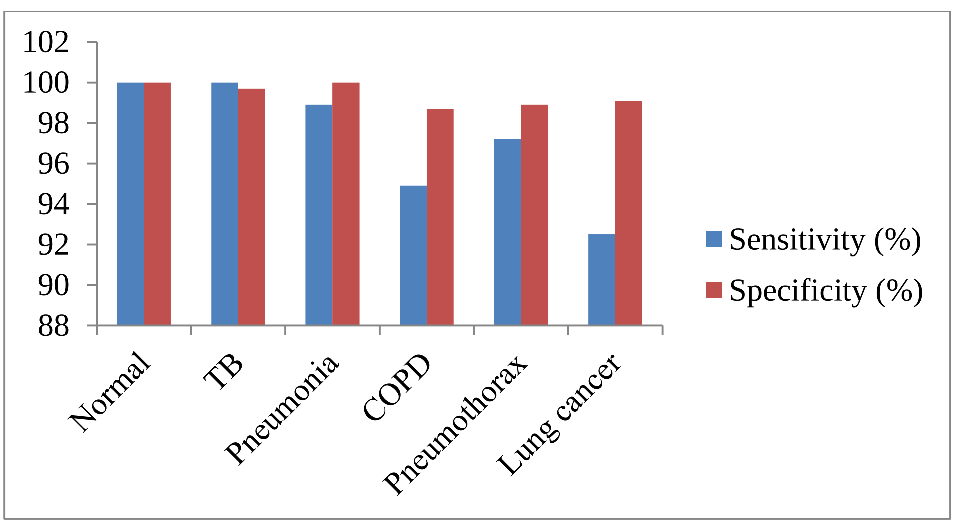

Figure 4-8: Sensitivity and Specificity of the model.

4.1.4. Implementation Using GUI

The proposed GUI first request for username and password in order to access the facilities provided by the system as shown in Figure 4.9 on the first column. Once a valid username and password are provided, the registration interface is displayed. After registration, the user proceeds to capture or browse the necessary image using the browse button. Pressing process button to pass the image to pre-processing step and then to the classification model. Finally the diagnosis result appears on the output button. A general representation of the processes is shown on Figure 4.9.

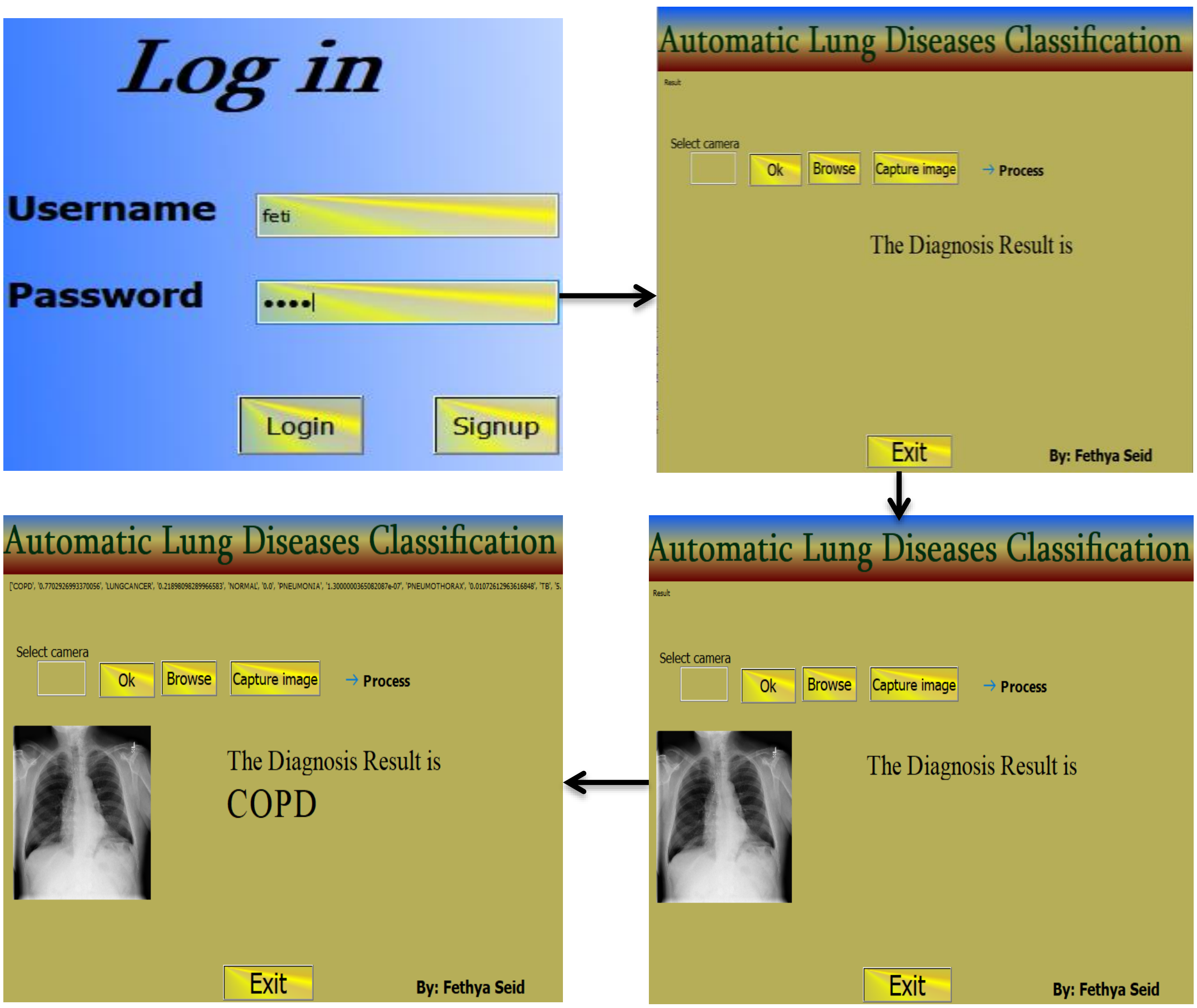

Figure 4-9: Implementation of GUI

4.2. Discussion

Among many lung diseases, the five conditions (Lung cancer, COPD, pneumonia, TB, and pneumothorax) primarily contribute to the global burden of respiratory diseases. They cause four million deaths annually [1]. Lung diseases can be caused by infection, smoking tobacco, breathing in secondhand tobacco smoke, radon, asbestos, or other forms of air pollution. Early diagnosis of these diseases can create an opportunity to get a proper treatment; meanwhile can contribute in reducing other complications because of the diseases. The common approach to diagnose these diseases is through visual inspection of the chest X-ray images. However, the current diagnosis method of lung diseases from chest X-ray images is tedious, time-consuming and subjected to inter and intra-observer variability. Because of the complex nature of chest X-ray radiography images and the limited number of radiologists, accurate and reliable classification of diseases remains challenging. Current classification of lung diseases using AI systems performs well with the state of the art DL.

In this research multi-class classification scheme for top prevalent lung diseases has been developed to diagnose and classify the type of diseases easily using DL. DL approach based on Xception model was used for classification. All the images have been pre-processed prior to feeding to the model. Median filter was used to remove noises from chest X-ray images, and CLAHE to enhance the contrast of the images. A graphical user interface has been developed for ease of use. The system takes only an average of 1 minute to provide the diagnosis result. The performance of the algorithm was also evaluated using a confusion matrix. Sensitivity, specificity, and accuracy were used as performance metrics.

The result obtained from de- noising employed in this study showed that X-ray images affected by salt and pepper noises are properly filtered using the most popular non-linear filter which is Median filter. The effects have seen clearly on the image and histogram plot. From the Figure 4-1 we can observe that an image containing dark pixels in bright regions and bright pixels in dark regions and eliminated on the second image. The method preserved the edges well in addition to removing noise.

Next, image enhancement using histogram equalization was done, enhancing an image which has highly uneven histogram distribution using global histogram equalization lead to unrealistic output image whose contrast is worse than the input image (Figure 4.2). In the input image, some gray-level values have excessively high histogram peak while some gray-level values have the smallest histogram peak. Redistributing the highest peaks over the lowest histogram peaks using HE over enhances the lowest peak gray-levels or/and under enhances the highest peak gray-levels and leads to a totally different output image. Figure 4-2: (a-d) illustrates the scenario. And also AHE effects are visualized clearly on Figure 4.3.

From result presented on Figure 4.5, data augmentation was applied to remove class imbalance problems. Many medical image classification tasks have class imbalance issues as, only a small labeled training set is available due to shortage of labeled dataset and there exists a high imbalance ratio between rare class and common class, which prevents the model from achieving high classification accuracy [67]. So in this study the problem of class imbalance is rectified using data augmentation. All six classes have equal number of training as well as testing images.

As we can see from Figure 4.6 the Tensorboard's graph known as learning curve explain training and validation accuracy as well as loss of the Xception model. It is dual learning curve (training and validation) of the model during training on both the training and validation datasets. During training of the model, the current state of the model at each epoch of the training algorithm can be evaluated. It can be evaluated on the training dataset to give an idea of how well the model is learned. And also evaluation on the validation dataset gives an idea of how well the model is generalized. It indicates the good fit of the model using loss of training and validation. Training loss decreases to a point of stability and then validation loss decreases to a point of stability and has a small gap with the training loss.

In order to show the separability performance of diseases and to determine most effective diseases, the model was tested with separate dataset. As shown in Table 4.2 Xception model give different classification accuracies, sensitivities and specificities for all six classes. The model gives less specificity for lung cancer and COPD relative to other classes, which shows that more false positive detection appeared in these class compared to other class, because evaluation of lung nodules on chest X-rays is often difficult due to characteristics of lung lesions including size, density and location. Computer-aided detection system to detect lung nodules has not been widely accepted and utilized because of high false positive rates [68]. DL methods having very deep layers help to improve detection by reducing the number of false positives reads. So this system is provides better diagnosis result than others. Sensitivity was presented for all classes (measure of positive rate). Generally, Xception performed over all classification accuracy of 97.3%, average sensitivity 97.2% and average specificity 99.4%. The results show that there are only few false positives. Also, the accuracy, sensitivity and specificity of the proposed system are high. Figure 4.7 and 4.8 shows the accuracy, specificity and sensitivity of the system.

The study presented in this thesis, improved accuracy and sensitivity for lung disease classification compared to other works. Faes et al [69] proposed binary and multi class classification of medical image using deep learning. They show multi class classification tasks of the chest X-ray images is low compared to binary classification tasks for chest X-ray images. The main reasons include the complex appearance of pathologies in X-ray projection images, this study proved that multi class classification of chest X-ray images is challenging. Other study conducted by Guendel et.al [37] on multi class classification on chest X-ray images, uses CNN model of deep learning achieving limited accuracy of 88% to classify 12 abnormalities on the chest X-ray images. However, in this study by using an appropriate deeper network and by reducing the number of abnormalities, from 12 to 5 (considering the most prevalent diseases), increase the classification accuracy to 97.3% which overcome the problem and challenges mentioned in the study. Other studies investigated deep learning based automatic detection algorithms for classification of chest X-ray with pneumonia [70] and pulmonary tuberculosis[71]. However, the algorithms used had limited clinical utility, as there are various pathologies and abnormalities other than pneumonia and pulmonary tuberculosis in real-world clinical practice. Additionally, algorithms in those studies have not yet been fully validated in unseen (unknown) datasets [56,57], limiting the generalizability of results.

The main goals of this thesis were, to diagnose five most prevalent diseases from chest X-ray images with improved classification performance, to introduce new model (Xception) which is applicable in other medical image classification having outsmarted performance and to consider particularly major thoracic diseases that account for most lung abnormalities observed on chest X-ray images. Generally, improved classification accuracy has been obtained. The developed GUI also allows user friendly implementation of the proposed system.

In Ethiopia most of the health centers/ health facilities found in rural areas uses the conventional X-ray machine. Hence, this automated system can play a significant role in filling the gap of limited radiologists in such areas

CHAPTER FIVE
5. Conclusion and Recommendations

5.1. Conclusion

This research employed Xception model of convolutional neural network, which is currently the state-of-the-art in image classification performing faster and more accurate than other image classification networks. The study first showed that image pre-processing has a significant impact on classification performance in identifying different types of lung diseases. Generally, using the Xception model over all classification accuracy of 97.3%, average sensitivity 97.2% and average specificity 99.4% have been obtained. This system can play a significant role in supporting radiologists and physicians during the diagnosis procedures. Besides, the system is accurate, reliable and easily implementable implying that it can reduce the workload and diagnostic errors especially in developing countries like Ethiopia where there is scarcity in number of radiologists.

The key contribution of this thesis includes multi class classification system with improved accuracy for diagnosis of most prevalent disease of the respiratory system considering the diseases prevalence. The study employed recent model for multi class classification of chest X-ray images. Ease of use of the system is also addressed by developing a simple Graphical User Interface. In Ethiopia most of the health centers/ health facilities found in rural areas face the problem of limited number of radiologists. Hence, this automated system can play a significant role in filling the gap of limited radiologists in such areas.

5.2. Recommendations

This thesis proposed an algorithm that performs multi class classification task to classify top five prevalent diseases of the respiratory system. The study was able to achieve an adequate result regarding multiclass classification. However, there is a limitation on labeled data set of the top five prevalent diseases for multi label classification task. For the future, the study could be extended to multi label classification task which is classification of more than two diseases on one image. In addition, further study can be done to implement web-based real-time diagnosis system for multiple lung diseases and 3D CNN to improve the sensitivity of cancerous tissue. This study is limited with computing infrastructures like GPU, storage memory, and others.

However, possible efforts ware exerted to overcome all the above constraints and to accomplish the desired work successfully.

REFERENCES

[1] G. J. Gibson, R. Loddenkemper, B. Lundbäck, and Y. Sibille, "Respiratory health and disease in Europe: The new European Lung White Book," *Eur. Respir. J.*, vol. 42, no. 3, pp. 559–563, 2013.

[2] D. K. Klugman, "Why continued neglect of pneumonia threatens the achievement of health goals," *IEEE Xplore,* 2018, p. 45.

[3] A. Misganaw, D. H. Mariam, A. Ali, and T. Araya, "Epidemiology of major non-communicable diseases in Ethiopia: A systematic review," *J. Heal. Popul. Nutr.*, vol. 32, no. 1, pp. 1–13, 2014.

[4] D. S. Abebe, G. Bjune, G. Ameni, D. Biffa, and F. Abebe, "Prevalence of pulmonary tuberculosis and associated risk factors in Eastern Ethiopian prisons," *Int. J. Tuberc. Lung Dis.*, vol. 15, no. 5, pp. 668–673, 2011.

[5] T. Abuka, "Prevalence of pneumonia and factors associated among children 2-59 months old in Wondo Genet district, Sidama zone, SNNPR, Ethiopia.," *Curr Pediatr Res*, vol. 21, no. 1, pp. 19–25, 2017.

[6] J. K. Lindsay M. Biga, Sierra Dawson, Amy Harwell, Robin Hopkins, *Anatomy and Physiology Volume 3 of 3*, vol. 2. Textbook Equity, 2014.

[7] M. Mustafa, A. J. Azizi, E. IIIzam, A. Nazirah, S. Sharifa, and S. Abbas, "Lung Cancer: Risk Factors, Management, And Prognosis," *IOSR J. Dent. Med. Sci.*, vol. 15, no. 10, pp. 94–101, 2016.

[8] J. L. Dennis Kasper , Anthony Fauci , Stephen Hauser, Dan Longo, J. Larry Jameson, *Harrison's Principles of Internal Medicine*, 19th ed. McGraw-Hill Education / Medical;, 2015.

[9] D. R. Brenner, J. R. McLaughlin, and R. J. Hung, "Previous lung diseases and lung cancer risk: A systematic review and meta-analysis," *PLoS One*, vol. 6, no. 3, pp. 1–10, 2011.

[10] M. S. Kalkat, “Spontaneous pneumothorax,” *Tips Tricks Thorac. Surg.*, vol. 2928, no. May, pp. 463–474, 2018.

[11] E. R. S. Sheffield, *The Global Impact of Respiratory Disease. Second Edi, World Health Organization*. 2017.

[12] C. Qin, D. Yao, Y. Shi, and Z. Song, “Computer-aided detection in chest radiography based on artificial intelligence: A survey,” *Biomed. Eng. Online*, vol. 17, no. 1, pp. 1–23, 2018.

[13] S. Meseret, “Screening Chest X-Ray Interpretations and Radiographic Techniques IOM GUIDELINES,” *IOM Guidel.*, pp. 2–0, 2015.

[14] B. Halalli and A. Makandar, “Computer Aided Diagnosis - Medical Image Analysis Techniques,” *Breast Imaging*, p. 26, 2018.

[15] M. Z. Alom *et al.*, “A state-of-the-art survey on deep learning theory and architectures,” *Electron.*, vol. 8, no. 3, pp. 1–67, 2019.

[16] R. Vijay, “Understanding Neural Networks - Towards Data Science.”, https://towardsdatascience.com/understanding-neural-networks-19020b758230, Accessed on: 08, Jan, 2019

[17] E. Ayan, “Diagnosis of Pneumonia from Chest X-Ray Images using Deep Learning,” *IEEE Rev. Biomed. Eng., vol. 2*, pp. 0–4, 2019.

[18] S. Candemir and S. Antani, “A review on lung boundary detection in chest X-rays,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 14, no. 4, pp. 563–576, 2019.

[19] S. Pattar, “Detection and Classification of Lung Disease – Pneumonia and Lung Cancer in Chest Radiology Using Artificial Neural Network,” *Int. J. Sci. Res. Publ.*, vol. 5, no. 1, pp. 2250–3153, 2015.

[20] C. Tataru, D. Yi, A. Shenoyas, and A. Ma, “Deep Learning for abnormality detection in Chest X-Ray images,” *ResearchGate*, May, vol. 6, no. 4, 2017.

[21] F. V. Farahani, M. H. F. Zarandi, and A. Ahmadi, “Fuzzy rule based expert system for diagnosis of lung cancer,” *Annu. Conf. North Am. Fuzzy Inf. Process. Soc. - NAFIPS*, vol. 2015–Septe, 2015.

[22] N. Hasanah, F. Arifin, D. Irmawati, and Muslikhin, “Smart System for Lung Disease Early Detection,” *J. Phys. Conf. Ser.*, vol. 1140, no. 1, 2018.

[23] M. S. Hossain, F. Ahmed, Fatema-Tuj-Johora, and K. Andersson, “A Belief Rule Based Expert System to Assess Tuberculosis under Uncertainty,” *J. Med. Syst.*, vol. 41, no. 3, 2017.

[24] J. Hamidzadeh, R. Javadzadeh, and A. Najafzadeh, “Fuzzy Rule Based Diagnostic System For Detecting The Lung Cancer Disease,” *J. Renew. Nat. Recources Bhutan*, vol. 3, no. 1, pp. 147–157, 2015.

[25] A. Badnjevic, L. Gurbeta, and E. Custovic, “An Expert Diagnostic System to Automatically Identify Asthma and Chronic Obstructive Pulmonary Disease in Clinical Settings,” *Sci. Rep.*, vol. 8, no. 1, pp. 1–9, 2018.

[26] A. N. Zakirov, R. F. Kuleev, A. S. Timoshenko, and A. V Vladimirov, “Advanced Approaches to Computer-Aided Detection of Thoracic Diseases on Chest X-Rays,” vol. 9, no. 88, pp. 4361–4369, 2015.

[27] E. Pune and V. I. T. Pune, “Automatie Deteetion ofMajor Lung Diseases U sing Chest Radiographs and Classifieation by Feed-forward Artifieial Neural Network,” *ResearchGate*, vol. 5, no. 1 pp. 1–5, 2016.

[28] G. P. Melendez and M. Cordel, “Texture-based detection of lung pathology in chest radiographs using local binary patterns,” *BMEiCON 2015 - 8th Biomed. Eng. Int. Conf.*, 2016, Philippines.

[29] A. M. Santos and A. M. P. Canuto, “Using semi-supervised learning in multi-label classification problems,” *Proc. Int. Jt. Conf. Neural Networks*, pp. 10–15, 2012.

[30] M. Haloi, K. R. Rajalakshmi, and P. Walia, “Towards Radiologist-Level Accurate Deep

Learning System for Pulmonary Screening," *ResearchGate*, 2018.

[31] Y. Bar, I. Diamant, L. Wolf, S. Lieberman, E. Konen, and H. Greenspan, "Chest pathology identification using deep feature selection with non-medical training," *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.*, vol. 6, no. 3, pp. 259–263, 2018.

[32] C. Brestel and M. Cohen-sfaty, "RadBot-CXR : Classification of Four Clinical Finding Categories in Chest X-Ray Using Deep Learning," *Med. Imaging with Deep Learn.*, no. Midl 2018, pp. 1–9, 2018.

[33] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017–Janua, pp. 3462–3471, 2017.

[34] R. H. Abiyev and M. K. S. Ma'aitah, "Deep Convolutional Neural Networks for Chest Diseases Detection," *Journal of Healthcare Engineering*, vol. 2018. pp. 1–11, 2018.

[35] P. Rajpurkar *et al.*, "CheXNet : Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning," *IEEE Rev. Biomed. Eng.* vol. 4, no. 1pp. 3–9, 2017.

[36] P. Rajpurkar *et al.*, "Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists," *PLoS Med.*, vol. 15, no. 11, pp. 1–17, 2018.

[37] S. Guendel *et al.*, "Multi-task Learning for Chest X-ray Abnormality Classification on Noisy Labels," *IEEE Xplore* pp. 1–10, 2019.

[38] E. J. Hwang *et al.*, "Development and Validation of a Deep Learning-Based Automated Detection Algorithm for Major Thoracic Diseases on Chest Radiographs," *JAMA Netw. open*, vol. 2, no. 3, p. e191095, 2019.

[39] E. J. Hwang *et al.*, "Deep Learning for Chest Radiograph Diagnosis in the Emergency Department.," *Radiology*, vol. 3, no. 1, pp. 191225, 2019.

[40] National Institutes of Health - Clinical Center, "NIH Chest X-ray Dataset of 14 Common Thorax Disease Categories," 2017. [Online]. Available: ttps://www.nih.gov/news-events/news-releases/nih-clinical-center-provides-one-largest-publicly-available-chest-x-ray-datasets-scientific-community. [Accessed: 04-Jan-2020].

[41] D. A. Pitaloka, A. Wulandari, T. Basaruddin, and D. Y. Liliana, "Enhancing CNN with Preprocessing Stage in Automatic Emotion Recognition," *Procedia Comput. Sci.*, vol. 116, pp. 523–529, 2017.

[42] P. Subbuthai, K. Kavithabharathi, and S. Muruganand, "Reduction of types of Noises in dental Images," *Int. J. Comput. Appl. Technol. Res.*, vol. 2, no. 4, pp. 436–442, 2013.

[43] K. K. Bharathi, S. Muruganand, and A. Periasamy, "Digital Image Processing Based Noise Reduction Analysis of Digital Dental Xray Image Using MATLAB,"Jour of Adv Research in Daynamical and Control system vol. 2, no. 2, pp. 198–203, 2014.

[44] S. U. Khan, M. Ishaq, N. Ullah, C. R. A. Ahamd, and I. Ahmed, "A Novel Algorithm for Removal of Noise from X-Ray Images.," *ResearchGate*, vol. 2, no. 2, 2016.

[45] S. J. Attia, "Enhancement of Chest X-ray Images for Diagnosis Purposes," *ResearchGate*, vol. 6, no. 2, pp. 2014–2017, 2016.

[46] R. Srinivas and S. Panda, "Performance Analysis of Various Filters for Image Noise Removal in Different Noise Environment," *Int. J. Adv. Comput. Res.*, vol. 3, no. 4, pp. 2249–7277, 2013.

[47] J. Das, B. Das, J. Saikia, and S. R. Nirmala, "Removal of Salt and Pepper Noise Using Selective Adaptive Median Filter," *IEEE Xplore*, vol. 1, no. 1,2016.

[48] S. X. Yang, "Contrast Limited Adaptive Histogram Equalization-Based Fusion in YIQ and HSI Color Spaces for Underwater Image Enhancement," *Int. J. Pattern Recognit. Arti¯cial Intell.*, vol. 32, no. 7, pp. 1–26, 2018.

[49] P. T. K.koonsanit, S.Thongvigitmanee, N.Pongnapang, "IMAGE ENHANCEMENT ON DIGITAL X-RAY IMAGES USING N-CLAHE X-ray CT and Medical Imaging

Laboratory (CTI) Biomedical Electronics and Systems Research Unit (BESRU) National Electronics and Computer Technology Center (NECTEC), Thailand Faculty of Medical,” *ResearchGate*.

[50] A. Mikołajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” *2018 Int. Interdiscip. PhD Work.*, pp. 117–122, 2018.

[51] W. Gaul *et al.*, *Studies in Classification , Data Analysis , and Knowledge Organization Managing Editors*. 2019.

[52] S. Regina Lourdhu Suganthi, M. Hanumanthappa, and S. Kavitha, “Event Image Classification using Deep Learning,” *ICSNS 2018 - Proc. IEEE Int. Conf. Soft-Computing Netw. Secur.*, vol. 7, pp. 614–617, 2018.

[53] R. Singh *et al.*, “Deep learning in chest radiography: Detection of findings and presence of change,” *PLoS One*, vol. 13, no. 10, pp. 1–12, 2018.

[54] R. H. Abiyev and M. K. S. Ma’aitah, “Deep Convolutional Neural Networks for Chest Diseases Detection,” *J. Healthc. Eng.*, vol. 2018, 2018.

[55] A. Maier, C. Syben, T. Lasser, and C. Riess, “REVIEW A gentle introduction to deep learning in medical image processing,” *Zeitschrift fßr Medizinische Phys.*, vol. 29, no. 2, pp. 86–101, 2019.

[56] X. Liu *et al.*, “Articles A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging : a systematic review and meta-analysis,” *Lancet Digit. Heal.*, vol. 1, no. 6, pp. e271–e297, 2019.

[57] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, “Benchmark analysis of representative deep neural network architectures,” *IEEE Access*, vol. 6, pp. 64270–64277, 2018.

[58] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017–Janua, pp. 1800–1807, 2017.

[59] S.-H. Tsang, "Review: Xception — With Depthwise Separable Convolution, Better Than Inception-v3 (Image Classification)." vol. 3, no. 2,2018.

[60] M. S. Hanif and M. Bilal, "Competitive residual neural network for image classification," *ICT Express*, no. xxxx, pp. 1–10, 2019.

[61] Dr. Adrian Rosebrock, *Deep Learning for Computer Vsion with Python*, 1st Editio. PYIMAGESEARCH, 2017.

[62] E. M. Dogo *et al.*, "Optimization Algorithms on Convolutional Neural Networks," *2018 Int. Conf. Comput. Tech. Electron. Mech. Syst.*, pp. 92–99, 2018.

[63] K. Eckle and J. Schmidt-Hieber, "A comparison of deep networks with ReLU activation function and linear spline-type methods," *Neural Networks*, vol. 110, pp. 232–242, 2019.

[64] Y. Srivastava, V. Murali, and S. R. Dubey, "A Performance Comparison of Loss Functions for Deep learning," *Indian Inst. Inf. Technol.*, 2018.

[65] S. Lall and S. Boyd, "Stabilization of Networked Control Systems with Sparse Observer-Controller Networks," IEEE transactions on automatic control, vol. 60, no. 6, June 2015.

[66] A. Baratloo, M. Hosseini, A. Negida, and G. El Ashal, "Part 1: Simple Definition and Calculation of Accuracy, Sensitivity and Specificity.," *Emerg. (Tehran, Iran)*, vol. 3, no. 2, pp. 48–9, 2015.

[67] C. Zhang, "Medical image classification under class imbalance." Iowa State University, p. 102, 2019.

[68] Y. Sim *et al.*, "Deep Convolutional Neural Network–based Software Improves Radiologist Detection of Malignant Lung Nodules on Chest Radiographs," *Radiology*, no. 11, p. 182465, 2019.

[69] L. Faes *et al.*, "Automated deep learning design for medical image classification by health-care professionals with no coding experience: a feasibility study," *Lancet Digit. Heal.*, vol. 1, no. 5, pp. e232–e242, 2019.

[70] O. Stephen, M. Sain, U. J. Maduh, and D. U. Jeong, “An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare,” *J. Healthc. Eng.*, vol. 2019, 2019.

[71] S. Ali *et al.*, “Development and validation of a deep learning-based automatic detection algorithm for active pulmonary tuberculosis on chest radiographs,”springer, vol. 2, no. 2, p. 35, 2018.

[72] Z. Li *et al.*, “Thoracic Disease Identification and Localization with Limited Supervision,” *Adv. Comput. Vis. Pattern Recognit.*, vol. 1, pp. 139–161, 2019.

[73] K. H. Reinert and J. H. Rodgers, “Deep learning at chest radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks,” *Radiology*, pp. 61–98, 2017.

APPENDICES

Appendix A: Approval Letter from JUMC

Approval Letter from JUMC

Jimma University Medical Center (JUMC), Radiology Department would like to approve that the chest X-Ray image data set is collected from the department for the research title "**Multiple Lung Disease Classification from Chest X-Ray Images using Deep Learning''** which is conducted by Fethya Seid and the data were labeled with the help of her clinical collaborator Dr. Elias Kedir. Apart from labeling the data, she has been working with her clinical collaborator in different aspects of the research including advice from the medical perspective and during testing and verifying the system that she has developed.

Name: Dr. Eliyas Kedir

Sign:

Date: 03/01/2022

Elias Kedir (Dr.)
Head, Department of
Radiology

Appendix B: Implementation Code for Noise Removal and Enhancement

```
import cv2

import numpy as np

from matplotlib import pyplot as plt

import os

path1 = 'D:\Data\Datasetedited\Dall'

path2 = 'D:\Data\Datasetedited\Dallprocessed'

listing = os.listdir(path1)

processed = []

for i in range(len(listing)):

    img=cv2.imread(os.path.join(path1,listing[i]),0)

    medfilt = cv2.medianBlur(img,3)

    clahe=cv2.createCLAHE(clipLimit=2.0,tileGridSize=(8,8))

    cl_img=clahe.apply(medfilt)

    processed.append(cl_img)

for i in range(len(processed)):

    os.chdir(path2)

    cv2.imwrite(listing[i], processed[i])
```

Appendix C: Implementation Code for Classification

Importing all the necessary libraries

```
import os
import numpy as np
import pandas as pd
import random
import cv2
import matplotlib.pyplot as plt
```

```
import keras.backend as K
from keras.models import Model,
from keras.layers import Input, Dense, Flatten, Dropout,
BatchNormalization
from keras.layers import Conv2D, SeparableConv2D, MaxPool2D,
LeakyReLU, Activation
from keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau,
EarlyStopping
import tensorflow as tf
```

Creating the Dataset

```
train_generator = data_generator.flow_from_directory(
        '/content/drive/My Drive/Datasetedited/Dtraining',
        target_size=(image_size, image_size),
        batch_size=batch_size_training,
        class_mode='categorical')

validation_generator = data_generator.flow_from_directory(
        '/content/drive/My Drive/Datasetedited/Dvalidation',
        target_size=(image_size, image_size),
        batch_size=batch_size_validation,
        class_mode='categorical')
```

Define some constants for later usage

```
num_classes =6
channels = 1
image_resize =299
objective_function = 'categorical_crossentropy'
```

```
loss_metrics = ['accuracy']
num_epochs = 50
early_stop_patience = 30

steps_per_epoch_training = 15
steps_per_epoch_validation = 2

batch_size_training = 32
batch_size_validation =32
```

Build the fine-tuned pre-trained model

```
from tensorflow.python.keras.applications import Xception
from tensorflow.python.keras.layers import Dense
base_model = Xception(include_top=False, weights='imagenet', inp
ut_shape=(299,299, 1))
# create a custom top classifier
x = base_model.output
x = GlobalMaxPooling2D()(x)
predictions = Dense(num_classes, activation='softmax')(x)
model = Model(inputs=base_model.inputs, outputs=predictions)
for layers in xception model.layers:
        layers.trainable=True
    for i, layer in enumerate (model.layers):
        if i< 51:
            layer.trainable=False
        else:
            layer.trainable=True=['Layer Type', 'Layer Name',
'Layer Trainable']
model.compile(optimizer=ADAM(lr=0.01,momentum=0.9),
loss='categorical_crossentropy', metrics=['accuracy']
```

Fit the data in to the model and run the model

```
fit_history = model.fit_generator(
```

```
        data_generator.flow_from_directory(
        '/content/drive/My Drive/Datasetedited/Dtraining',
        target_size=(image_size, image_size),
        batch_size=batch_size_training,
        class_mode='categorical'),
        steps_per_epoch= 15,
        epochs = 50,
        validation_data=data_generator.flow_from_directory(
        '/content/drive/My Drive/Datasetedited/Dvalidation',
        target_size=(image_size, image_size),
        batch_size=batch_size_validation,
        class_mode='categorical'),
        validation_steps= 2,
        callbacks=[ModelCheckpoint(filepath = '/content/drive/My
 Drive/Datasetedited/model_wieghts.h5', monitor = 'val_loss', sa
ve_best_only = True, mode = 'auto'), EarlyStopping(monitor = 'va
l_loss', patience = early_stop_patience)]
)
```

Saving the weights of the fine-tuned model

```
model.save_weights('model_wieghts.h5')
model.save('model_keras.h5')
```

Evaluation using saved model

```
class_names = ['COPD','LUNGCANCER','NORMAL', 'PNEUMONIA', 'PNEUM
OTHORAX','TB']
import tensorflow as tf
from tensorflow.keras.applications import xception
from keras.models import load_model
#model = tf.keras.models.load_model('model_keras.h5')
model.load_weights('model_wieghts.h5')
model.evaluate_generator(validation_generator, steps=None, max_q
ueue_size=10, workers=1, use_multiprocessing=False)
pred = model.predict_generator(validation_generator)
predicted = np.argmax(pred, axis=1)

print('Confusion Matrix')
```

```
cm = confusion_matrix(testing_generator.classes, np.argmax(pred,
 axis=1))
plt.figure(figsize = (30,20))
sn.set(font_scale=1.4) #for label size
sn.heatmap(cm, annot=True, annot_kws={"size": 12}) # font size
plt.show()
print()
print('Classification Report')
print(classification_report(testing_generator.classes, predicted
, target_names=class_names))
```

Appendix D: Implementation Code for GUI

```
from PyQt4 import QtCore, QtGui
import sqlite3
import cv2
import shutil
from PIL import Image, ImageEnhance
from subprocess import call


try:
    _fromUtf8 = QtCore.QString.fromUtf8
except AttributeError:
    def _fromUtf8(s):
        return s

try:
    _encoding = QtGui.QApplication.UnicodeUTF8
    def _translate(context, text, disambig):
        return QtGui.QApplication.translate(context, text, disambig,
_encoding)
except AttributeError:
    def _translate(context, text, disambig):
        return QtGui.QApplication.translate(context, text, disambig)

class Ui_MainWindow1(object):

    def closeIt(self):
        print('Close button pressed')
        import sys
        sys.exit(0)

    def M(self):
        exit_code = call("C:\Python35\python.exe LungXception.py",
```

```
shell=True)
        file33 = open('s.txt','r+')
        result = file33.read()
        print result
        file33.close()
        if result == 'PNEUMOTHORAX':
            self.label11.setText("PNEUMOTHORAX")

        if result == 'NORMAL':
            self.label11.setText("NORMAL")

        if result == 'LUNGCANCER':
            self.label11.setText("LUNGCANCER")

        if result == 'COPD':
            self.label11.setText("COPD")

        if result == 'PNEUMONIA':
            self.label11.setText("PNEUMONIA")

        if result == 'TB':
            self.label11.setText("TB")

        file1 = open("m.txt","r+")

        print "Output of Read function is "
        r = file1.read()
        rr = r.split('/')
        self.label1.setText(str(rr))
        print rr
        print('Working on The model integration')

    def takePhoto(self):
        # cam = cv2.VideoCapture(0)

        ret, img = self.cam.read();
        if ret:
            self.gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            cv2.imwrite('bio5' + '.png', self.gray)
            cv2.imshow('window', img);

            im = Image.open('bio5.png')
            enhancer = ImageEnhance.Brightness(im)
            enhanced_im = enhancer.enhance(3)
            enhanced_im.save('ebio5.png')
```

```
            im = Image.open('bio5.png')
            enhancer = ImageEnhance.Contrast(im)
            enhanced_im = enhancer.enhance(1)
            enhanced_im.save('eebio5.png')

            # cv2.waitKey(1);

        pixmap = QtGui.QPixmap("ebio5.png")
        pixmap = pixmap.scaledToHeight(400)
        self.ima1.setPixmap(pixmap)



    def image_fun(self):
        select = str(self.lineEdit.text())
        print (select)

        if select == "1":
            self.cam = cv2.VideoCapture(0)

            while (True):
                ret, img = self.cam.read();
                if ret:
                    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
                    cv2.imshow('window', img);
                if (cv2.waitKey(1) == ord('q')):
                    break

            self.cam.release()
            cv2.destroyAllWindows()
        elif select == "2":
            self.cam = cv2.VideoCapture(2)

            while (True):
                ret, img = self.cam.read();
                if ret:
                    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
                    cv2.imshow('window', img);
                if (cv2.waitKey(1) == ord('q')):
                    break

            self.cam.release()
            cv2.destroyAllWindows()
        elif select == "3":
            print("not implimiented just for wifi camera")
        else:
```

```
            print "Please Enter a valid value !!"


    def openSlot(self):
        # This function is called when the user clicks File->Open.
        filename = QtGui.QFileDialog.getOpenFileName()
        shutil.copy2(str(filename),
                     'D:/finalGui/ftyaMOdel/bio5' + '.png')

        im = Image.open('bio5.png')
        enhancer = ImageEnhance.Brightness(im)
        enhanced_im = enhancer.enhance(1)
        enhanced_im.save('ebio5.png')

        im = Image.open('bio5.png')
        enhancer = ImageEnhance.Contrast(im)
        enhanced_im = enhancer.enhance(1)
        enhanced_im.save('eebio5.png')

        pixmap = QtGui.QPixmap("ebio5.png")
        pixmap = pixmap.scaledToHeight(200)

        self.ima1.setPixmap(pixmap)
        print ("File opened successful!!")



    def setupUi(self, MainWindow):
        MainWindow.setObjectName(_fromUtf8("MainWindow"))
        MainWindow.resize(820, 591)
        MainWindow.setStyleSheet(_fromUtf8("\n"
"background-color:rgb(182, 174, 88);\n"
"QDialog{\n"
"background-color:qlineargradient(spread:pad, x1:0.006, y1:0.597, x2:1, y2:0.563, stop:0
rgba(0, 85, 255, 255), stop:0.9375 rgba(255, 255, 255, 255))\n"
"}\n"
"QLineEdit{\n"
"background-color:qconicalgradient(cx:0, cy:0, angle:135, stop:0
rgba(255, 255, 0, 69), stop:0.375 rgba(255, 255, 0, 69), stop:0.423533
rgba(251, 255, 0, 145), stop:0.45 rgba(247, 255, 0, 208),
stop:0.477581 rgba(255, 244, 71, 130), stop:0.518717 rgba(255, 218,
71, 130), stop:0.55 rgba(255, 255, 0, 255), stop:0.57754 rgba(255,
203, 0, 130), stop:0.625 rgba(255, 255, 0, 69), stop:1 rgba(255, 255,
0, 69))\n"
"}\n"
```

```
"QPushButton{\n"
"background-color:qconicalgradient(cx:0, cy:0, angle:135, stop:0
rgba(192, 225, 44, 69), stop:0.375 rgba(255, 255, 0, 69),
stop:0.423533 rgba(251, 255, 0, 145), stop:0.45 rgba(247, 255, 0,
208), stop:0.477581 rgba(255, 244, 71, 130), stop:0.518717 rgba(255,
218, 71, 130), stop:0.55 rgba(255, 255, 0, 255), stop:0.57754
rgba(255, 203, 0, 130), stop:0.625 rgba(255, 255, 0, 69), stop:1
rgba(255, 255, 0, 69));\n"
"\n"
"}"))
        self.centralwidget = QtGui.QWidget(MainWindow)
        self.centralwidget.setObjectName(_fromUtf8("centralwidget"))
        self.lineEdit = QtGui.QLineEdit(self.centralwidget)
        self.lineEdit.setGeometry(QtCore.QRect(50, 180, 71, 41))
        self.lineEdit.setStyleSheet(_fromUtf8(""))
        self.lineEdit.setObjectName(_fromUtf8("lineEdit"))
        self.label = QtGui.QLabel(self.centralwidget)
        self.label.setGeometry(QtCore.QRect(20, 160, 110, 20))
        self.label.setFont(QtGui.QFont('SansSerif', 13))
        self.label.setObjectName(_fromUtf8("label"))


        self.label1 = QtGui.QLabel(self.centralwidget)
        self.label1.setGeometry(QtCore.QRect(10, 84, 800, 22))
        self.label1.setFont(QtGui.QFont('SansSerif', 7))
        self.label1.setStyleSheet("QLabel#nom_plan_label {color:
yellow}")
        self.label1.setObjectName(_fromUtf8("label"))


        self.label11 = QtGui.QLabel(self.centralwidget)
        self.label11.setGeometry(QtCore.QRect(325, 184, 400, 322))
        self.label11.setFont(QtGui.QFont('Times New Roman', 35))
        self.label11.setObjectName(_fromUtf8("label"))

        self.label33 = QtGui.QLabel(self.centralwidget)
        self.label33.setGeometry(QtCore.QRect(325, 280, 320, 32))
        self.label33.setFont(QtGui.QFont('Times New Roman', 25))
        self.label33.setObjectName(_fromUtf8("label"))

        self.pushButton = QtGui.QPushButton(self.centralwidget)
        self.pushButton.setGeometry(QtCore.QRect(140, 182, 81, 41))
        self.pushButton.setStyleSheet(_fromUtf8("QDialog{\n"
"background-color:qlineargradient(spread:pad, x1:0.006, y1:0.597,
x2:1, y2:0.563, stop:0 rgba(0, 85, 255, 255), stop:0.9375 rgba(255,
255, 255, 255))\n"
```

```
"}\n"
"QLineEdit{\n"
"background-color:qconicalgradient(cx:0, cy:0, angle:135, stop:0
rgba(255, 255, 0, 69), stop:0.375 rgba(255, 255, 0, 69), stop:0.423533
rgba(251, 255, 0, 145), stop:0.45 rgba(247, 255, 0, 208),
stop:0.477581 rgba(255, 244, 71, 130), stop:0.518717 rgba(255, 218,
71, 130), stop:0.55 rgba(255, 255, 0, 255), stop:0.57754 rgba(255,
203, 0, 130), stop:0.625 rgba(255, 255, 0, 69), stop:1 rgba(255, 255,
0, 69))\n"
"}\n"
"QPushButton{\n"
"background-color:qconicalgradient(cx:0, cy:0, angle:135, stop:0
rgba(192, 225, 44, 69), stop:0.375 rgba(255, 255, 0, 69),
stop:0.423533 rgba(251, 255, 0, 145), stop:0.45 rgba(247, 255, 0,
208), stop:0.477581 rgba(255, 244, 71, 130), stop:0.518717 rgba(255,
218, 71, 130), stop:0.55 rgba(255, 255, 0, 255), stop:0.57754
rgba(255, 203, 0, 130), stop:0.625 rgba(255, 255, 0, 69), stop:1
rgba(255, 255, 0, 69));\n"
"\n"
"}"))
        self.pushButton.setFont(QtGui.QFont('SansSerif', 15))
        self.pushButton.setObjectName(_fromUtf8("pushButton"))
        self.pushButton.clicked.connect(self.image_fun)

        self.pushButton_2 = QtGui.QPushButton(self.centralwidget)
        self.pushButton_2.setGeometry(QtCore.QRect(330, 180, 131, 41))
        self.pushButton_2.setStyleSheet(_fromUtf8("QDialog{\n"
"background-color:qlineargradient(spread:pad, x1:0.006, y1:0.597,
x2:1, y2:0.563, stop:0 rgba(0, 85, 255, 255), stop:0.9375 rgba(255,
255, 255, 255))\n"
"}\n"
"QLineEdit{\n"
"background-color:qconicalgradient(cx:0, cy:0, angle:135, stop:0
rgba(255, 255, 0, 69), stop:0.375 rgba(255, 255, 0, 69), stop:0.423533
rgba(251, 255, 0, 145), stop:0.45 rgba(247, 255, 0, 208),
stop:0.477581 rgba(255, 244, 71, 130), stop:0.518717 rgba(255, 218,
71, 130), stop:0.55 rgba(255, 255, 0, 255), stop:0.57754 rgba(255,
203, 0, 130), stop:0.625 rgba(255, 255, 0, 69), stop:1 rgba(255, 255,
0, 69))\n"
"}\n"
"QPushButton{\n"
"background-color:qconicalgradient(cx:0, cy:0, angle:135, stop:0
rgba(192, 225, 44, 69), stop:0.375 rgba(255, 255, 0, 69),
stop:0.423533 rgba(251, 255, 0, 145), stop:0.45 rgba(247, 255, 0,
208), stop:0.477581 rgba(255, 244, 71, 130), stop:0.518717 rgba(255,
218, 71, 130), stop:0.55 rgba(255, 255, 0, 255), stop:0.57754
```

```
rgba(255, 203, 0, 130), stop:0.625 rgba(255, 255, 0, 69), stop:1
rgba(255, 255, 0, 69));\n"
"\n"
"}"))
        self.pushButton_2.setFont(QtGui.QFont('SansSerif', 14))
        self.pushButton_2.setObjectName(_fromUtf8("pushButton_2"))

        self.pushButton_2.clicked.connect(self.takePhoto)

        self.commandLinkButton =
QtGui.QCommandLinkButton(self.centralwidget)
        self.commandLinkButton.setGeometry(QtCore.QRect(490, 180, 185,
41))
        self.commandLinkButton.setStyleSheet(_fromUtf8("QDialog{\n"
"background-color:qlineargradient(spread:pad, x1:0.006, y1:0.597,
x2:1, y2:0.563, stop:0 rgba(0, 85, 255, 255), stop:0.9375 rgba(255,
255, 255, 255))\n"
"}\n"
"QLineEdit{\n"
"background-color:qconicalgradient(cx:0, cy:0, angle:135, stop:0
rgba(255, 255, 0, 69), stop:0.375 rgba(255, 255, 0, 69), stop:0.423533
rgba(251, 255, 0, 145), stop:0.45 rgba(247, 255, 0, 208),
stop:0.477581 rgba(255, 244, 71, 130), stop:0.518717 rgba(255, 218,
71, 130), stop:0.55 rgba(255, 255, 0, 255), stop:0.57754 rgba(255,
203, 0, 130), stop:0.625 rgba(255, 255, 0, 69), stop:1 rgba(255, 255,
0, 69))\n"
"}\n"
"QPushButton{\n"
"background-color:qconicalgradient(cx:0, cy:0, angle:135, stop:0
rgba(192, 225, 44, 69), stop:0.375 rgba(255, 255, 0, 69),
stop:0.423533 rgba(251, 255, 0, 145), stop:0.45 rgba(247, 255, 0,
208), stop:0.477581 rgba(255, 244, 71, 130), stop:0.518717 rgba(255,
218, 71, 130), stop:0.55 rgba(255, 255, 0, 255), stop:0.57754
rgba(255, 203, 0, 130), stop:0.625 rgba(255, 255, 0, 69), stop:1
rgba(255, 255, 0, 69));\n"
"\n"
"}"))
        self.commandLinkButton.setFont(QtGui.QFont('SansSerif', 14))

self.commandLinkButton.setObjectName(_fromUtf8("commandLinkButton"))

        self.commandLinkButton.clicked.connect(self.M)


        self.label_2 = QtGui.QLabel(self.centralwidget)
        self.label_2.setGeometry(QtCore.QRect(600, 540, 181, 31))
```

```
        font = QtGui.QFont()
        font.setPointSize(15)
        font.setBold(True)
        font.setWeight(75)
        self.label_2.setFont(font)
        self.label_2.setObjectName(_fromUtf8("label_2"))
        self.label_7 = QtGui.QLabel(self.centralwidget)
        self.label_7.setGeometry(QtCore.QRect(0, 0, 831, 81))
        font = QtGui.QFont()
        font.setFamily(_fromUtf8("Sylfaen"))
        font.setPointSize(35)
        font.setBold(True)
        font.setWeight(75)
        self.label_7.setFont(font)
        self.label_7.setStyleSheet(_fromUtf8("background-color:qlineargradient(spread:pad, x1:0, y1:0, x2:0, y2:1, stop:0 rgba(0, 85, 255, 255), stop:0.495 rgba(1, 255, 255, 0), stop:0.505 rgba(45, 0, 0, 0), stop:1 rgba(100, 0, 0, 255));\n"
"color:rgb(0, 44, 14)"))
        self.label_7.setObjectName(_fromUtf8("label_7"))
        self.pushButton_3 = QtGui.QPushButton(self.centralwidget)
        self.pushButton_3.setGeometry(QtCore.QRect(230, 180, 81, 41))
        self.pushButton_3.setStyleSheet(_fromUtf8("QDialog{\n"
"background-color:qlineargradient(spread:pad, x1:0.006, y1:0.597, x2:1, y2:0.563, stop:0 rgba(0, 85, 255, 255), stop:0.9375 rgba(255, 255, 255, 255))\n"
"}\n"
"QLineEdit{\n"
"background-color:qconicalgradient(cx:0, cy:0, angle:135, stop:0 rgba(255, 255, 0, 69), stop:0.375 rgba(255, 255, 0, 69), stop:0.423533 rgba(251, 255, 0, 145), stop:0.45 rgba(247, 255, 0, 208), stop:0.477581 rgba(255, 244, 71, 130), stop:0.518717 rgba(255, 218, 71, 130), stop:0.55 rgba(255, 255, 0, 255), stop:0.57754 rgba(255, 203, 0, 130), stop:0.625 rgba(255, 255, 0, 69), stop:1 rgba(255, 255, 0, 69))\n"
"}\n"
"QPushButton{\n"
"background-color:qconicalgradient(cx:0, cy:0, angle:135, stop:0 rgba(192, 225, 44, 69), stop:0.375 rgba(255, 255, 0, 69), stop:0.423533 rgba(251, 255, 0, 145), stop:0.45 rgba(247, 255, 0, 208), stop:0.477581 rgba(255, 244, 71, 130), stop:0.518717 rgba(255, 218, 71, 130), stop:0.55 rgba(255, 255, 0, 255), stop:0.57754 rgba(255, 203, 0, 130), stop:0.625 rgba(255, 255, 0, 69), stop:1 rgba(255, 255, 0, 69));\n"
"\n"
"}"))
```

```
        self.pushButton_3.setFont(QtGui.QFont('SansSerif', 15))
        self.pushButton_3.setObjectName(_fromUtf8("pushButton_3"))

        self.pushButton_3.clicked.connect(self.openSlot)


        self.ima1 = QtGui.QLabel(self.centralwidget)
        self.ima1.setGeometry(QtCore.QRect(15, 230, 300, 250))

        self.pushButton_4 = QtGui.QPushButton(self.centralwidget)
        self.pushButton_4.setGeometry(QtCore.QRect(320, 530, 131, 41))
        self.pushButton_4.setStyleSheet(_fromUtf8("QDialog{\n"
"background-color:qlineargradient(spread:pad, x1:0.006, y1:0.597, 
x2:1, y2:0.563, stop:0 rgba(0, 85, 255, 255), stop:0.9375 rgba(255, 
255, 255, 255))\n"
"}\n"
"QLineEdit{\n"
"background-color:qconicalgradient(cx:0, cy:0, angle:135, stop:0 
rgba(255, 255, 0, 69), stop:0.375 rgba(255, 255, 0, 69), stop:0.423533 
rgba(251, 255, 0, 145), stop:0.45 rgba(247, 255, 0, 208), 
stop:0.477581 rgba(255, 244, 71, 130), stop:0.518717 rgba(255, 218, 
71, 130), stop:0.55 rgba(255, 255, 0, 255), stop:0.57754 rgba(255, 
203, 0, 130), stop:0.625 rgba(255, 255, 0, 69), stop:1 rgba(255, 255, 
0, 69))\n"
"}\n"
"QPushButton{\n"
"background-color:qconicalgradient(cx:0, cy:0, angle:135, stop:0 
rgba(192, 225, 44, 69), stop:0.375 rgba(255, 255, 0, 69), 
stop:0.423533 rgba(251, 255, 0, 145), stop:0.45 rgba(247, 255, 0, 
208), stop:0.477581 rgba(255, 244, 71, 130), stop:0.518717 rgba(255, 
218, 71, 130), stop:0.55 rgba(255, 255, 0, 255), stop:0.57754 
rgba(255, 203, 0, 130), stop:0.625 rgba(255, 255, 0, 69), stop:1 
rgba(255, 255, 0, 69));\n"
"\n"
"}"))
        self.pushButton_4.setFont(QtGui.QFont('SansSerif', 25))
        self.pushButton_4.setObjectName(_fromUtf8("pushButton_4"))

        self.pushButton_4.clicked.connect(self.closeIt)



        MainWindow.setCentralWidget(self.centralwidget)
        self.statusbar = QtGui.QStatusBar(MainWindow)
        self.statusbar.setObjectName(_fromUtf8("statusbar"))
        MainWindow.setStatusBar(self.statusbar)
```

```
        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        MainWindow.setWindowTitle(_translate("MainWindow", "Lung", 
None))
        self.label.setText(_translate("MainWindow", "Select camera", 
None))
        self.label1.setText(_translate("MainWindow", "Result", None))
        self.label11.setText(_translate("MainWindow", " ", None))
        self.label33.setText(_translate("MainWindow", "The Diagnosis 
Result is : ", None))
        self.pushButton.setText(_translate("MainWindow", "Ok", None))
        self.pushButton_2.setText(_translate("MainWindow", "Capture 
image", None))
        self.commandLinkButton.setText(_translate("MainWindow", 
"Process", None))
        self.label_2.setText(_translate("MainWindow", "By: Fethya 
Seid", None))
        self.label_7.setText(_translate("MainWindow", "Automatic Lung 
Diseases Classification", None))
        self.pushButton_3.setText(_translate("MainWindow", "Browse", 
None))
        self.pushButton_4.setText(_translate("MainWindow", "Exit", 
None))


if __name__ == "__main__":
    import sys
    app = QtGui.QApplication(sys.argv)
    MainWindow = QtGui.QMainWindow()
    ui = Ui_MainWindow1()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
```