A DEEP LEARNING APPROACH FOR CLASSIFICATION OF SIX CASES OF CHEST X-RAYS

MSc. THESIS

SULEIMAN MOHAMED ABDI

November, 2024

HARAMAYA UNIVERSITY, HARAMAYA

HARAMAYA UNIVERSITY

POSTGRADUATE PROGRAM DIROCTORATE

A Deep Learning Approach for Classification of Six Cases of Chest X-Rays

MSc. Thesis

Suleiman Mohamed Abdi

College:	**Computing and Informatics**
Department:	**Computer Science**
Major Advisor:	**Wondwossen Mulugeta (Ph.D)**
Co-Advisor:	**Faizur Rashid (Ph.D)**

November, 2024

Haramaya University, Haramaya

HARAMAYA UNIVERSITY

POST GRADUATE PROGRAM DIRECTORATE

I hereby certify that I have read and evaluated this Thesis entitled "**A Deep Learning Approach for Classification of Six Cases of Chest X-Rays**" prepared under my guidance by Suleiman Mohamed Abdi.

I recommend that it be submitted as fulfilling the thesis requirement.

Wondwossen Mulugeta (PhD)		April 04, 2024
Major Advisor	**Signature**	**Date**
Faizur Rashid (PhD)		April 04, 2024
Co-Advisor	**Signature**	**Date**

As a member of the Board of Examiners of the MSc Thesis Open Defense Examination, I certify that I have read and evaluated the Thesis prepared by Suleiman Mohamed Abdi and examined the candidate. I recommend that the thesis be accepted as fulfilling the Thesis requirements for the degree of Master of Science in Computer Science.

Chairperson	**Signature**	**Date**
Internal Examiner	**Signature**	**Date**
External Examiner	**Signature**	**Date**

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

DEDICATION

This thesis is dedicated to my beloved parents, Mohamed A. Gargaare and Nabiiha A. Barkhad, whose unwavering support, encouragement, and love have been a constant source of strength throughout my academic journey and life. Your guidance and belief in me have inspired and sustained me through every challenge, and I am profoundly grateful to have you in my life.

I also dedicate this work to my cherished siblings—Bilan, Warsame, Arafo, Ayan, and Haroun—whose unconditional love and support have always been a beacon of motivation. Your exemplary lives have taught me the value of hard work and perseverance in the pursuit of my dreams. This accomplishment is as much yours as it is mine.

STATEMENT OF THE AUTHOR

By signing below, I hereby confirm that this Thesis represents my original work. I have adhered to all ethical and technical standards of academic scholarship in its preparation, including data collection, analysis, and compilation. Any scholarly material incorporated into the Thesis has been appropriately cited and acknowledged.

This Thesis is submitted as part of the requirements for a Master's degree at Haramaya University. It has been deposited in the Haramaya University Library, where it will be accessible to borrowers in accordance with the library's regulations. I affirm that this Thesis has not been submitted to any other institution for the purpose of obtaining an academic degree, diploma, or certificate.

Short quotations from this Thesis may be used without special permission, provided the source is accurately and fully acknowledged. Requests for permission to use extended excerpts or to reproduce any part of this Thesis, either in full or in part, may be granted by the Head of the School or Department if deemed to serve the interests of academic scholarship. In all other cases, permission must be sought directly from the author.

Name: _____ Signature: _____

Date: _____

Department: _____

BIOGRAPHICAL SKETCH

The author was born on November 17, 1992, in Hargeisa, Somaliland. He began his formal education at 31 May Primary School and later attended Farah Omar Secondary School for his secondary education. Upon completing his high school studies, he enrolled at the University of Hargeisa in 2011, where he earned a Bachelor's degree in Information Technology in September 2015.

After graduation, the author embarked on a dual career path, combining academia and public service. He joined the University of Hargeisa as an Assistant Lecturer, where he contributed to the academic growth of students in the field of information technology. Simultaneously, he worked part-time at the Somaliland Ministry of ICT, dedicating approximately five years to advancing both institutions with his skills and commitment.

In November 2019, the author pursued his passion for higher education by enrolling in the Postgraduate Program at Haramaya University. He commenced his studies in the MSc program in Computer Science, marking a significant milestone in his academic journey and professional development. Through his dedication and perseverance, the author continues to contribute to academia and the field of computer science.

ACKNOWLEDGMENT

First and foremost, I would like to express my deepest gratitude to Allah, whose boundless mercy and guidance have enabled me to accomplish this work. His blessings and assistance have been the foundation of my success, and without His grace, this journey would not have been possible.

I am profoundly thankful to my research advisor, Dr. Wondwossen Mulugeta, for his invaluable guidance and unwavering support throughout this journey, from the inception of the study proposal to the successful completion of this thesis. His insights, encouragement, and expertise have been instrumental in shaping the quality of my work. I also extend my sincere appreciation to my research co-advisor, Dr. Faizur Rashid, for his constructive feedback, continuous support, and valuable suggestions, which greatly enriched my research.

I am deeply indebted to my parents for their unconditional love, encouragement, and support during every phase of my academic journey. Their understanding, compassion, and strength in sharing my challenges during this research have been a constant source of motivation and inspiration.

My heartfelt thanks go to the staff at Hargeisa Group Hospital for their collaboration and facilitation in providing the dataset necessary for my experimental work. Their support played a pivotal role in the successful execution of this study.

Finally, I wish to extend my sincere appreciation to my entire family, relatives, friends, and colleagues for their moral support, encouragement, and understanding throughout this journey. A special note of gratitude goes to my beloved mother, Nabiiha A. Barkhad, my father, Mohamed A. Gargaare, and my siblings—Bilan, Warsame, Arafo, Ayan, and Haroun—for their moral upbringing, constant prayers, and unwavering encouragement during my studies. Their love and belief in me have been a source of immense strength and determination.

This accomplishment would not have been possible without the collective contributions of everyone mentioned, and for that, I am truly grateful.

ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
AUC	Area Under Curve
AUROC	Area Under the Receiver Operating Characteristic
CNN	Convolutional Neural Network
CXR	Chest X-Ray
MAF	Mycobacterium Africanum
MOH	Ministry of Health
MTB	Mycobacterium Tuberculosis
OCT	Optical Coherence Tomography
ROC	Receiver Operating Characteristic
TB	Tuberculosis
ViT	Vision Transformers
WHO	World Health Organization
XAI	Explainable Artificial Intelligence

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Artificial Intelligence (AI), particularly deep learning, is transforming healthcare by enabling automated analysis and diagnosis from medical images, addressing critical challenges such as the shortage of radiologists and the demand for accurate diagnostic systems. This study focuses on the classification of six distinct chest X-ray conditions: Normal, Pneumonia, Tuberculosis, Lung Mass, Rib Fracture, and Enlarged Heart.

Using a dataset of 10,200 chest X-ray images collected from Hargeisa Group Hospital, two pretrained convolutional neural network (CNN) architectures, DenseNet and GoogleNet, were fine-tuned and evaluated. Comprehensive preprocessing, including noise removal, image enhancement, and augmentation techniques, ensured high-quality and balanced training data. The models demonstrated exceptional performance, achieving classification accuracies of 97% and 96%, respectively, surpassing benchmarks in multi-class medical image classification.

Despite these promising results, the study encountered limitations. The dataset size, while sufficient for this research, remains relatively small for broader generalizability. Additionally, processing sensitive personal data required compliance with Somaliland's Data Protection Act, posing challenges in accessing and utilizing X-ray images. These limitations highlight the need for expanded datasets and improved data access protocols for future research.

This research establishes a robust framework for automating chest X-ray diagnostics, empowering radiologists with timely and accurate decision support. The findings contribute to advancing AI-driven solutions for healthcare, addressing both global and region-specific challenges.

Keywords - *Chest X-rays, Medical Imaging, Pneumonia, Tuberculosis, Lung Mass, Rib Fracture, Enlarged Heart, DenseNet, GoogleNet, Deep Learning, Artificial Intelligence.*

1. INTRODUCTION

1.1. Background

Artificial Intelligence (AI) is rapidly transforming medicine, offering unprecedented opportunities to enhance diagnostic accuracy, improve clinical decision-making, and optimize patient care. AI systems are poised to revolutionize healthcare by detecting diseases earlier, improving prognosis, and enabling personalized treatment plans, all while saving time and reducing costs. For instance, AI-driven algorithms are being developed to analyze chest X-rays and histopathology slides, assisting clinicians with worklist management, decision support, and telehealth services. Beyond hospitals, AI technologies facilitate continuous health monitoring for millions of patients, streamlining physician visits and follow-ups at an unprecedented scale (Topol, 2019).

Among the various advancements in AI, Deep Learning (DL), a subfield of Machine Learning, has achieved remarkable success in image classification tasks. Deep Learning leverages neural networks to learn patterns directly from raw data without relying on handcrafted features. Convolutional Neural Networks (CNNs), a specialized type of neural network designed for image analysis, have become the cornerstone of medical image classification due to their ability to extract hierarchical features from complex datasets. Recent improvements in CNN architectures and training methodologies have driven significant progress in medical imaging applications, particularly in radiology (LeCun, Y., Bengio, Y., & Hinton, G., 2015). Pretrained CNN models, such as DenseNet and GoogleNet, have further enhanced diagnostic accuracy by utilizing transfer learning to adapt knowledge from large-scale datasets to domain-specific challenges (Litjens & Van der Laak, 2017).

Chest radiography, the most commonly performed imaging examination globally, is critical for the diagnosis and management of numerous life-threatening conditions, including pneumonia, tuberculosis, and cardiomegaly. However, the interpretation of chest X-rays remains a challenging task, often requiring specialized expertise. Variability in human interpretation, limited access to skilled radiologists, and the growing volume of imaging data pose significant challenges to accurate and timely diagnoses, particularly in resource-constrained settings. Automated chest radiograph classification using DL models at a level comparable to practicing radiologists offers

immense potential to improve workflow prioritization, support clinical decision-making, and enable large-scale screening initiatives in global healthcare systems (Rajpurkar, 2020).

Despite these advancements, applying deep learning to medical image classification is not without its challenges. Training DL models requires large, annotated datasets, which are often difficult to curate due to data privacy regulations and the need for expert labeling. Additionally, variability in image quality, noise, and differences in imaging protocols across healthcare facilities complicate the development of robust algorithms. Addressing these challenges necessitates the use of advanced preprocessing techniques, balanced datasets, and carefully optimized CNN architectures to ensure reliability and generalizability in real-world clinical applications (Benjamens, 2020).

This study focuses on leveraging Deep Learning, specifically CNN architectures, to classify chest X-ray images into six diagnostic categories: Normal, Pneumonia, Tuberculosis, Lung Mass, Rib Fracture, and Enlarged Heart. The motivation behind this research is to improve the classification accuracy of medical X-ray images in large databases, enhancing diagnostic efficiency in hospitals and research centers. By addressing the challenges in applying deep learning to medical image classification, this study aims to contribute to the development of robust AI-driven tools that support radiologists and improve patient outcomes.

1.2. Statement of the Problem

According to the World Health Organization (WHO), the leading causes of morbidity and mortality in Somalia include diarrheal diseases (e.g., cholera), tuberculosis (TB), pneumonia, lung mass, malaria (affecting primarily pregnant women and children under five), and measles (WHO, 2014). The National Health Plan of Somalia (1980–1985) reported that one in three children aged 5 to 9 years were infected with TB, and approximately 1% of the population had active TB (MoH, 1980 - 1985). Among Somali refugees in the Horn of Africa region, a prevalence survey revealed sputum smear positivity rates of 2%–3% (Shears, 2014). Pneumonia, another leading health challenge, kills more than two children every hour in Somalia, accounting for 24% of all under-five mortality (Save the Children, 2017). These statistics highlight the critical burden of chest-related diseases in Somalia, emphasizing the urgent need for improved diagnostic solutions.

Chest X-rays remain the most effective method for diagnosing chest conditions such as pneumonia, tuberculosis, and lung mass (WHO, 2001). However, providing accurate interpretations of X-ray images is inherently challenging due to the need for expert radiologists and supporting patient background information. Additionally, artifacts in medical images can further complicate interpretation, increasing the likelihood of misclassification. These challenges are exacerbated in Somalia and other developing countries, where radiologists and doctors rely heavily on manual visual examinations based on their experience. This process is time-consuming, costly, and subject to variability due to differences in expertise. Compounding the issue, rural areas in Somalia face a severe shortage of trained radiologists, limiting access to timely and accurate diagnosis for patients in need.

The dominance of the six selected chest conditions—pneumonia, tuberculosis, lung mass, rib fractures, enlarged heart, and normal—necessitates targeted diagnostic models in regions like Somalia. Previous studies addressing chest X-ray classification have predominantly focused on uni-classification (e.g., classifying pneumonia vs. non-pneumonia) (Chhikara, 2020), bi-classification (e.g., tuberculosis vs. non-tuberculosis) (Hooda, 2017), or tri-classification (e.g., tuberculosis, pneumonia, and normal) (Abubakar, 2020). While these studies have made significant contributions, they fail to address the multi-classification challenges unique to the African disease spectrum. For example, African-specific tuberculosis strains, such as *Mycobacterium africanum* in West Africa, differ significantly from Euro-American strains, complicating generalizability (De Jong, 2010). Existing multi-label models, such as those detecting 14 chest conditions in datasets like CheXpert and ChestX-ray14, are predominantly developed in high-resource settings and do not account for the unique disease profiles and imaging conditions in Africa.

The shortage of radiologists relative to patient populations further highlights the diagnostic gap. In Somalia, where healthcare resources are limited, one radiologist may serve tens of thousands of patients, creating significant bottlenecks in healthcare delivery. This lack of capacity underscores the need for computer-aided diagnostic systems to supplement clinical workflows. These systems can analyze large datasets, prioritize critical cases, and provide decision support, thereby alleviating the burden on overworked radiologists.

To address these gaps, this study aims to develop a deep learning-based classification model capable of accurately categorizing chest X-rays into six diagnostic categories: normal, pneumonia, tuberculosis, lung mass, rib fractures, and enlarged heart. Unlike previous studies that focused on fewer classes or datasets from high-resource environments, this research leverages a dataset collected from Somalia to account for African-specific disease variations and challenges. The use of convolutional neural networks (CNNs), which have demonstrated efficacy in various classification tasks (Liu, 2017), provides a scalable and efficient solution for multi-class classification. This model not only aims to improve diagnostic accuracy but also addresses the time and cost inefficiencies associated with manual X-ray interpretation.

The development of such a model represents a significant advancement in medical imaging for resource-limited settings, offering the potential to enhance mass screening and improve outcomes for chest-related diseases in Somalia. By focusing on multi-classification in the African disease context, this study contributes to bridging the gap in diagnostic capabilities between high-resource and low-resource settings.

2.2.1. Research Gap

Deep Learning have been used in several recent studies for the detection or classification of chest problems/diseases including pneumonia and tuberculosis by analyzing chest x-ray images. (Chhikara, 2020) explored the possibility of detecting pneumonia from chest x-ray images. (Hooda, 2017) presented a deep learning approach to classify chest x-ray images into TB and non-TB categories with an accuracy of 82.09%. (Tahir, 2020) classified different coronavirus families (SARS, MERS and COVID-19) using transfer learning of various pre-trained CNN models with sensitivity values greater than 90%.

(Almezhghwi, 2021) proposed two artificial intelligence approaches for processing and identifying for twelve chest X-ray diseases. But the things that this proposed study is different from this (Almezhghwi, 2021) study are; the diseases that we're classifying, the dataset and methodology that we are using and finally, the evaluation technique that we are using.

Based on the previous works, studies and models that were studied to work on chest x-ray classification were only concerned about uni-classification such us classifying pneumonia from chest x-ray images (Chhikara, 2020), or bi-classification such us classifying tuberculosis and non-

tuberculosis cases from chest x-ray images (Hooda, 2017), and tri-classification such us classifying tuberculosis, pneumonia and normal from chest x-ray images (Abubakar, 2020).

This proposed deep learning model will classify six different categories including normal, pneumonia, tuberculosis, lung mass, rib fractures and enlarged heart. To the best of my knowledge, no such work regarding the use of deep learning networks on doing hexa-classification from chest x-ray images is reported. This study will be focused on the classification of six cases that are related to the chest using deep learning learning-based technique named CNNs.

1.2.2. Research Questions

1. How do DenseNet and GoogleNet models enhance the accuracy of multi-class classification for chest X-rays compared to traditional methods?
2. What preprocessing and augmentation techniques are most effective in improving the quality and balance of a medical chest X-ray dataset?
3. How does the proposed deep learning framework address diagnostic challenges specific to resource-limited settings, such as those in Somalia?
4. What are the performance metrics (e.g., precision, recall, F1-score, AUROC) of the models in classifying six distinct chest conditions?

1.3. Objectives of the Study

1.3.1. General Objective

The general objective of this study was to implement and evaluate a deep learning models for classifying six distinct chest X-ray conditions.

1.3.2. Specific Objectives

Under the above general objective, the study has the following specific objectives:

I. To collect and preprocess a dataset of chest X-ray images, ensuring quality through noise removal and image enhancement techniques.
II. To explore the classification performance of DenseNet and GoogleNet models, focusing on accuracy, precision, recall, F1-score, and AUROC.
III. To optimize DenseNet and GoogleNet models for multi-class classification of six specific chest X-ray conditions.

IV. To compare the performance of DenseNet and GoogleNet in terms of their suitability for medical imaging tasks.

1.4. Scope of the Study

The scope of this study focuses on the implementation and evaluation of deep learning algorithms, specifically Convolutional Neural Networks (CNNs), for the classification of six diagnostic conditions based on chest X-ray images. The primary goal of this research is to assess the performance of models in accurately identifying and classifying Normal, Pneumonia, Tuberculosis, Lung Mass, Rib Fracture, and Enlarged Heart cases.

The dataset utilized for this study consisted of 10,200 chest X-ray images collected from Hargeisa Group Hospital, the largest government-managed hospital in Somaliland. Of these, 5,100 images represented normal conditions, while the remaining 5,100 images were distributed equally across the five abnormal categories, with 1,020 images per category. The dataset covers X-ray images taken over a specified period, reflecting diverse patient demographics and conditions. However, due to resource and data availability constraints, the dataset only includes adult patients and excludes pediatric cases.

In terms of problem coverage, this study addresses the classification of chest X-ray images into six predefined categories. Broader issues such as the detection of additional conditions or the integration of multi-modal imaging data (e.g., CT scans or MRI) are outside the scope of this research. Similarly, while this study utilizes X-ray images as its primary diagnostic medium, it does not cover broader diagnostic systems that incorporate clinical or laboratory data.

Approach coverage is limited to the application of CNNs, particularly DenseNet and GoogleNet architectures. Other deep learning techniques or hybrid approaches, such as Vision Transformers or ensemble methods, were not included in this study due to time and resource limitations. Furthermore, the study does not extend to the deployment of the developed models in real-world clinical environments, focusing instead on experimental evaluation and model performance analysis.

This focused scope ensures a thorough exploration of CNN-based classification methods while acknowledging the limitations in terms of data diversity, diagnostic breadth, and algorithmic

coverage. These constraints also provide a foundation for future studies to address the gaps identified in this research.

1.5. Significance of the study

This study provides significant contributions to both the medical and academic fields. In Medical Practice and Diagnostics, the proposed deep learning framework assists radiologists by automating the classification of six chest conditions, reducing their workload and improving diagnostic accuracy and speed. It addresses the shortage of trained radiologists in resource-limited regions like Somalia, facilitating mass screenings and timely interventions.

In addition, at the side of Academic and Scientific Contributions, This study bridges a notable gap in medical AI by developing a hexa-classification model tailored to regional disease manifestations, a focus overlooked in existing studies. The findings establish a benchmark for leveraging pretrained CNN architectures (DenseNet and GoogleNet) in multi-class medical imaging tasks.

Finally, at the Global Implications, the study demonstrates the feasibility of using deep learning in underrepresented datasets, offering insights that can be generalized to other low-resource healthcare settings globally. It lays the groundwork for further research into multi-class classification and AI-driven diagnostic tools, especially for diseases prevalent in developing countries.

2. LITERATURE REVIEW

2.1. Concept and Definitions

Chest X-rays are among the most frequently performed diagnostic imaging examinations and are pivotal in the diagnosis and management of various medical conditions. They generate detailed images of the heart, lungs, airways, blood vessels, and chest bones, enabling clinicians to evaluate a wide range of chest-related abnormalities (WHO, 2001). Chest X-rays remain the standard imaging modality for the diagnosis of conditions such as pneumonia, tuberculosis (TB), lung masses, rib fractures, and enlarged hearts. Their widespread availability, low cost, and non-invasive nature make them an indispensable tool in both primary care and specialized settings.

2.1.1. Pneumonia

Pneumonia is an acute respiratory infection that directly impacts the lungs. Normally, the alveoli in the lungs fill with air during breathing, but in pneumonia, these sacs become inflamed and filled with pus and fluid, resulting in painful breathing and reduced oxygen intake (WHO, 2021a). Pneumonia is the leading infectious cause of death among children worldwide, claiming 740,180 lives in children under five years old in 2019. This accounts for 14% of all deaths in this age group and 22% of deaths among children aged one to five. South Asia and sub-Saharan Africa bear the highest burden of pneumonia-related mortality. Despite its severe impact, pneumonia is preventable with simple interventions and treatable with affordable, low-technology care (WHO, 2021a).

2.1.2. Tuberculosis (TB)

Tuberculosis is caused by *Mycobacterium tuberculosis*, a bacterium that primarily infects the lungs. TB spreads through airborne transmission, with infected individuals releasing bacteria into the air when coughing, sneezing, or spitting. Approximately one-quarter of the global population carries a latent TB infection, although only a small fraction develops active TB disease. Individuals with weakened immune systems, such as those with HIV, malnutrition, or diabetes, are at higher risk of developing TB (WHO, 2021b). Symptoms include prolonged coughing, fever, night sweats, and weight loss, which often delay diagnosis and increase transmission risks. Without proper treatment, TB has a mortality rate of 45% among HIV-negative individuals and nearly 100% among HIV-positive individuals (WHO, 2021b).

2.1.3. Lung Mass

A lung mass refers to an abnormal growth in the lung, typically larger than 3 centimeters in diameter. Lung masses can be benign or malignant, with malignant masses often indicating lung cancer. The detection of lung masses on chest X-rays can be challenging due to overlapping features such as lesion size, density, and location. These challenges necessitate the use of advanced diagnostic methods. Computer-aided detection systems have shown promise in improving the identification of lung masses by reducing false positives and providing more accurate readings (Society, 2021). Early detection is crucial for effective treatment, as delays in diagnosis can lead to poor prognoses in malignant cases.

2.1.4. Rib Fracture

Rib fractures are among the most common thoracic injuries, often caused by trauma such as falls or motor vehicle accidents. They can vary in severity, with minor fractures healing with conservative management, while severe cases may result in complications like pneumothorax or damage to internal organs (MayoClinic, 2022). Chest X-rays are the initial imaging modality of choice for diagnosing rib fractures. However, subtle fractures may go undetected, requiring further imaging such as CT scans. Automated diagnostic systems leveraging artificial intelligence have demonstrated the potential to enhance fracture detection, particularly in complex cases, by reducing the variability inherent in human interpretation.

2.1.5. Enlarged Heart (Cardiomegaly)

An enlarged heart, or cardiomegaly, is often a sign of underlying cardiovascular conditions such as hypertension, cardiomyopathy, or valve disorders. Chest X-rays are commonly used to identify cardiomegaly, which is characterized by an increased cardiac silhouette. However, further evaluation with echocardiography or advanced imaging modalities like MRI is typically required to confirm the diagnosis and assess the underlying cause (National Heart, Lung, and Blood Institute, 2021). Early detection of cardiomegaly is vital for timely intervention, as untreated conditions leading to an enlarged heart can result in serious complications, including heart failure.

2.1.6. Medical Imaging and Digital Image Processing

Medical imaging forms the backbone of modern diagnostic medicine, providing clinicians with non-invasive tools to visualize and analyze internal body structures. Among these, chest X-rays stand out as a primary diagnostic tool due to their accessibility and cost-effectiveness (WHO,

2021). In recent years, digital image processing techniques have revolutionized medical imaging by enhancing image quality and diagnostic accuracy. Techniques such as noise reduction, contrast enhancement, and edge detection are commonly employed to improve image interpretability, particularly in low-quality X-rays (Gonzalez & Woods, 2018). These methods play a critical role in ensuring that automated systems can analyze images effectively.

2.1.7. Convolutional Neural Networks (CNNs) in Medical Imaging

Convolutional Neural Networks (CNNs) are a type of deep learning algorithm specifically designed for image analysis tasks. They consist of convolutional layers for feature extraction and fully connected layers for classification (LeCun, Y., Bengio, Y., & Hinton, G., 2015). CNNs have become a cornerstone of medical imaging, enabling automated detection and classification of abnormalities with high accuracy. Fully connected layers in CNNs aggregate extracted features and make predictions based on learned patterns, while convolutional layers allow the network to focus on specific regions of interest within the image (Simonyan, K., & Zisserman, A. , 2015). In the context of chest X-rays, CNNs have shown remarkable success in identifying conditions such as pneumonia, tuberculosis, and cardiomegaly, often outperforming traditional diagnostic methods. Their scalability and adaptability make them an essential tool for advancing automated medical diagnostics.

2.1.8. Deep Learning

Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. It is based on Artificial Neural Network (ANN), which are biologically-inspired computing systems. Today, deep learning algorithms are widely used in image processing, speech recognition, natural language processing, audio recognition, and bio-informatics. Deep learning methods have the main advantage over old-fashioned machine learning algorithms like SVM or shallow neural networks. The main advantage is that deep learning algorithms extract the features in the data by themselves. Therefore, there is no need for human intervention during the training process. Besides, this feature extraction mechanism generates features that are hard for a human to think and implement. Compared to other techniques such as standard machine learning algorithms, deep learning is a preferred option; its performance increases as the data scales. Machine Learning algorithms such as SVM, random forest and etc.,

performance plateaus when data increases in these algorithms (Andrew Ng, 2020) as shown on Figure 1.

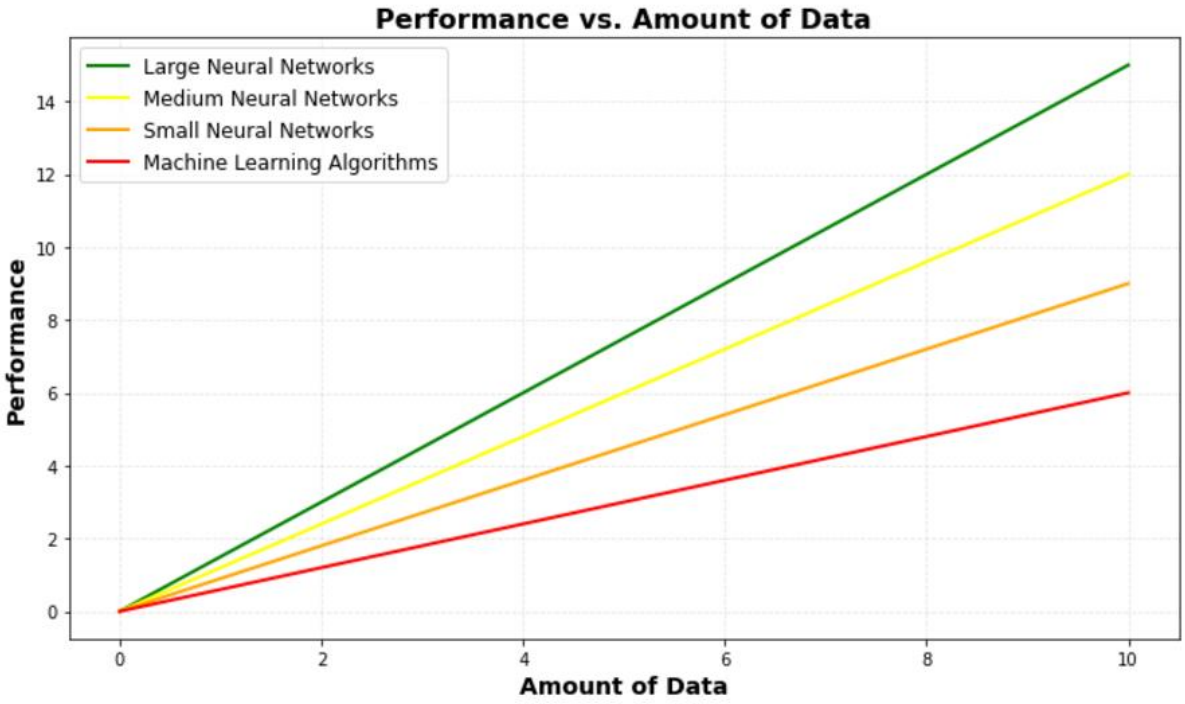

Figure 2.1: Performance of Neural Networks compared to Traditional Learning Algorithms by **(Andrew Ng, 2020)**.

An ANN is an interconnected architecture where there exists an input layer where input data is placed, a hidden layer(s) where artificial neurons are stacked on top of each other and an output layer where the prediction or classification is made (Lee Jacobson, 2014). An ANN is usually a combination of forward and backward propagation techniques. Forward propagation is a technique in which data moves through from the corresponding input layer, hidden layers and output layer sequentially. But back propagation is the opposite of forward propagation, it feeds the network backwards. Back propagation is used to adjust the weight of the neural network after the errors have been computed by forward propagation algorithms. A hidden layer usually consists of a weight which is used in an optimization algorithm such as gradient descent, an activation function such as sigmoid, tanh, relu etc. and a loss function to calculate the loss or error of the function which we use to back propagate to adjust the weights. Activation functions allow or stop neurons from firing into the next layer by comparing it into the activation function thresholds (Matthew and Dnuggets, 2017). Activation functions are used both in the forward and backward propagation

where in the forward propagation an activation function is used to calculate the loss where the output of a function is compared to a real number and in backward propagation they update the parameters of the neural network. Figure 2 shows commonly used activation functions in a neural network.

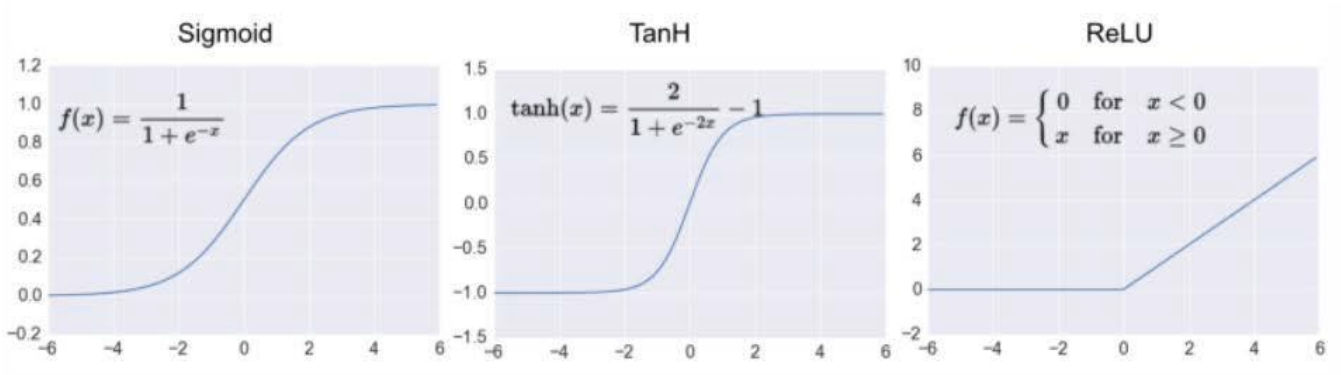

Figure 2.2. Common Activation Functions **(Matthew and Dnuggets, 2017)**

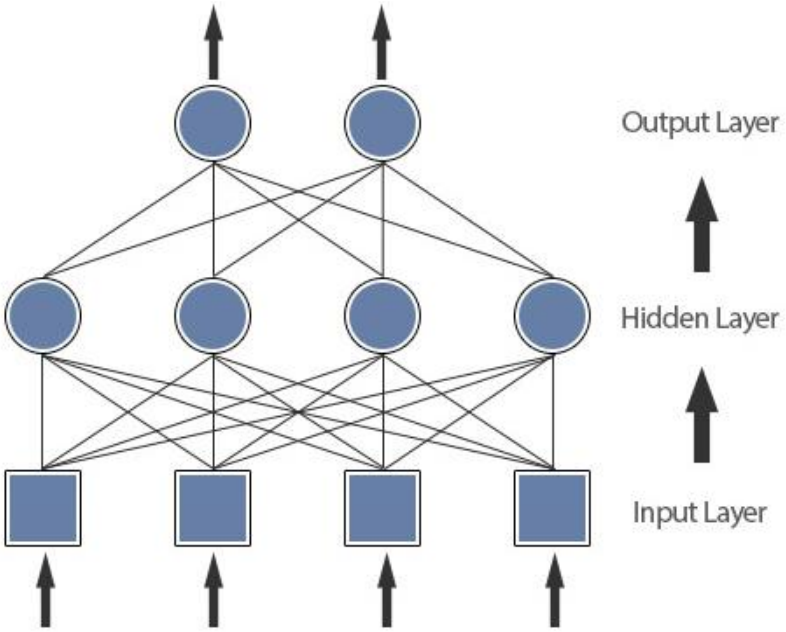

Figure 2.3. Artificial Neural Network also known as Multi-Layer Perceptron **(Lee Jacobson, 2014).**

2.1.8.1. Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of feed-forward ANN. CNNs use very little pre-processing compared to regular neural network models. Network model in CNN learns the features itself by creating a relationship between adjacent nodes. CNNs are generally composed of

convolution layers, pooling layers, and fully connected layers. CNNs use filters to extract the specific patterns, use pooling to help the model to ignore redundant data, and use MLP as a final step to vectorize the resulting data.

2.1.8.2. CNN Architecture

Like a Neural Network, a typical CNN consists of a multiple hidden layer called a convolutional layer where the linear function computes the stride convolutions over an image to extract features. CNN also consists of a pooling layer function such as Max Pool to reduce the size of the image in the neuron and to speed up the computation. Pooling layer does by extracting the features of the neuron image and ignoring the rest, this makes the network more robust. Fully connected layer in CNN, which is like a hidden layer in a neural network where the sum of the outputs of each layer are attend and where each value is an input to the next layer followed by an activation function and an output (Coursera, 2021).

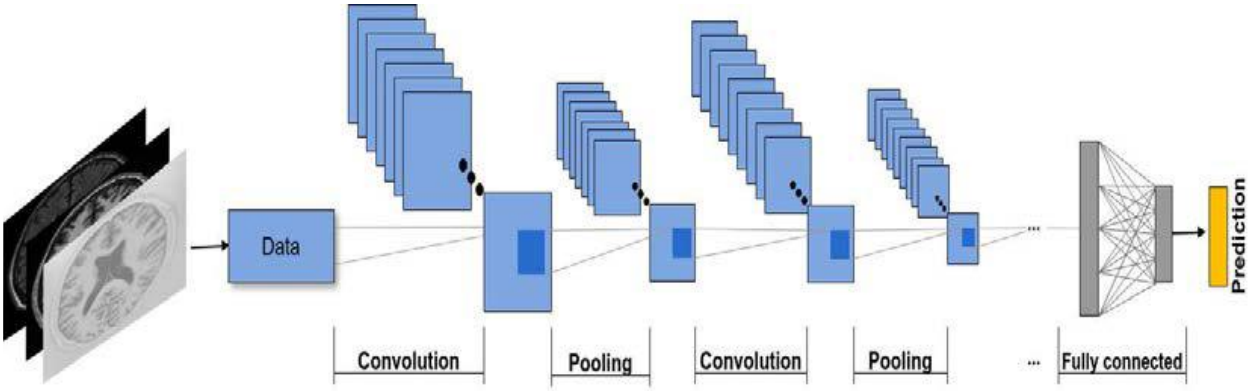

Figure 2.4. Convolutional Neural Network Architecture Example **(Bernal et al., 2018)**.

In a CNN, the linear function that is used is called a convolutional layer. Each neuron in the hidden layer extracts different features by using image processing feature detectors. These features are extracted using a kernel or filter (Francesco, 2019). Figure 2.5 shows an example of how strided convolutions works on an image. The lower is the original image and the upper is the output of the convolutions. Thus, the outcome of the convolutions decreases the dimension of the original image.

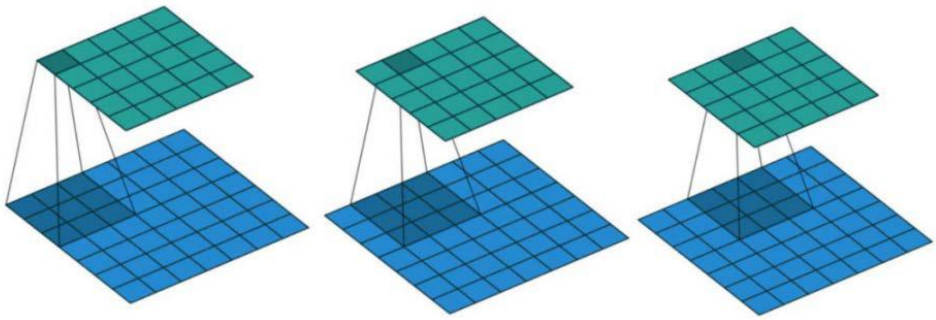

Figure 2.5. Strided Convolution Operation Example **(Francesco, 2019)**.

The pooling layer is work out after the convolutional layer. The reason why pooling included in the architecture is to further reduce the dimensions of the convolutional layer and just extract out the features to make the model more robust. Pooling can be max pooling or average pooling. Max pooling extracts out the highest pixel value out of a feature while average pooling calculates the average pixel value that has to be extracted.

Convolution and pooling layers are used to generate high level features. With the help of a fully connected layer, these high-level features can be multiplied by the weights of the hidden layers (Bernal et al., 2018). A convolutional layer followed by a ReLU and pooling layer can be used multiple times before the fully connected layer. In CNN architecture, earlier convolution operations look for low level features. Instead, later convolution operations try to look for high level features, which are specific to the training data. Therefore, the model will extract more high-level features if there are more convolutional layers in the network.

2.2. Related Work

Medical image classification is a sub-subject of image classification. Many techniques in image classification can also be used on it. Such as many image enhanced methods to enhance the discriminable features for classification (Beutel et al., 2020).

(Almezhghwi, 2021)proposed two artificial intelligence approaches for processing and identifying chest X-ray images to detect chest diseases from such images. They introduced two novel deep learning methods for fast and automated classification of chest X-ray images. First, they proposed the use of support vector machines based on the AlexNet model. Second, they developed support vector machines based on the VGGNet16 method. Combined deep networks with a robust classifier have shown that the proposed methods outperform AlexNet and VGG16 deep learning

approaches for the chest X-ray image classification tasks. The proposed AlexNet and VGGNet based SVM provide average area under the curve values of 98% and 97%, respectively, for twelve chest X-ray diseases.

The author and his team used CNN Model for classifying tuberculosis in chest X Rays. The dataset used in this is obtained from Peruvian partners at "Scio's en Salud". The dataset contains 4701 images in which 453 are labeled as normal and 4258 labeled as abnormal. The final accuracy found after Alex net is about 85.68% a significant improvement from non-shuffle sampling which is 53.02%. Accuracy is achieved 85% only on TB (Liu, 2017).

This (Jeoung, 2019) article used dataset of size 112,120 the images are transformed from 1024x1024 to 224x224 for extracting features from the author of the images used Densenet-169 architecture and pass the output to Support vector machines (SVM) to predict variables achieved accuracy is 80%. They applied different techniques and highest accuracy achieved is 80.

In this article, (Dai, 2017) used the JSRT dataset in this research and the proposed methodology is to identify the x-ray of the patient whether it is normal or not by using organ segmentation techniques and then identify the report using SCAN the main drawback of their work is they use a very small amount of data. Dataset size is too much small mostly used is synthetic data.

The researcher and his fellows used Chest X-ray 14 dataset, containing over 100,000 frontal view X-ray images, published by the National Institutes of Health. The proposed methodology by the researcher is that he used the Chex net algorithm; it is a state-of-the art machine-learning algorithm to detect pneumonia at a level of a human practicing radiologist. It is a 121-layer Convoluted Neural Classification of Pneumonia and Tuberculosis from Chest X-rays 10 Network h. The Chex Net algorithm can identify 14 pathologies from a Chest X-ray. The performance of Chex Net reported an F1 score of 0.435. In this article, (Dai, 2017) used the JSRT dataset in this research and the proposed methodology is to identify the x-ray of the patient whether it is normal or not by using organ segmentation techniques and then identify the report using SCAN the main drawback of their work is they use a very small amount of data. Testing score is quite low in this case (Rajpurkar, 2020).

Former Apple engineer *David W. Dai* worked on detection of Pneumonia and tuberculosis. He used JSRT and Montgomery Chest set. He used very minute dataset and generate synthetic data

and trained model on that synthetic dataset. The issue is in their dataset it is small and mostly synthetic images are added in it (Dai D., 2019a) (Dai D., 2019b).

2.2.1 CNN on Medical Image Classification

With the different CNN-based deep neural networks developed and achieved a significant result on ImageNet Challenger, which is the most significant image classification and segmentation challenge in the image analyzing field (Russakovsky, 2015). The CNN-based deep neural system is widely used in the medical classification task. CNN is an excellent feature extractor, therefore utilizing it to classify medical images can avoid complicated and expensive feature engineering. (Quing, 2014) presented a customized CNN with shallow ConvLayer to classify image patches of lung disease. The authors also found that the system can be generalized to other medical image datasets. Moreover, in other research, it also found that CNN based system can be trained from big chest X-ray film dataset and state-of-art with high accuracy and sensitivity results on their dataset, like Stanford Normal Radiology Diagnostic Dataset containing more than 400,000 CXR and a new CXR database (ChestX-ray8), which consist of 108,948 frontal-view CXR (Wang, 2017).

Moreover, using limited data makes it hard to train an adequate model. Therefore, the transfer learning of CNN is wildly used in medical image classification tasks. (Kermany, 2018) use InceptionV3 with ImageNet trained weight and transfer learning on a medical image dataset containing 108,312 optical coherence tomography (OCT) images. They got an average accuracy of 96.6%, with a sensitivity of 97.8% and a specificity of 97.4%.

The authors also compared the results with six human experts. Most of the experts got high sensitivity but low specificity, while the CNN-based system got high values on both sensitivity and specificity. Moreover, on the average weight error measure, the CNN based system exceeds two human experts. The authors also verified their system on a small pneumonia dataset, including about five thousand images, and achieved an average accuracy of 92.8%, with a sensitivity of 93.2% and a specificity of 90.1%. This system finally may help in accelerating diagnosis and referral of patients and therefore introduce early treatment, resulting in an increased cure rate.

3. RESEARCH METHODOLOGY

3.1. Overview

This study adopts an experimental research methodology to explore the performance of deep learning algorithms in classifying six categories of chest X-ray images. The methodology includes a structured approach to data collection, preparation, and evaluation, aimed at addressing the research objectives. This section outlines the techniques and procedures utilized, providing a comprehensive understanding of the processes involved.

3.2. Research Design

The research design was formulated after identifying the gaps in existing studies, as discussed in Section 1.3. Upon defining the study's main objectives, the experimental methodology was chosen as the most appropriate approach. This decision was based on the lack of prior studies that comprehensively evaluated the efficiency of convolutional neural network (CNN) models for multi-class classification of chest X-ray images.

According to (Schiaffonati and Verdicchio, 2014), an experimental research design is characterized by a systematic set of methods and procedures used to investigate a specific hypothesis. This approach involves conducting experiments under controlled conditions to ensure reliability and validity. Key principles of experimental research include comparison, reproducibility, repeatability, justification, and explanation (Schiaffonati and Verdicchio, 2014).

To achieve the study's objectives, the research was conducted in three main phases:

I. Data Preparation: This phase involved collecting, preprocessing, and augmenting the dataset to ensure high-quality input for the models.
II. Implementation: The deep learning models (DenseNet and GoogleNet) were fine-tuned and trained to classify six distinct chest conditions.
III. Evaluation: The models were rigorously tested using various performance metrics, including accuracy, precision, recall, and F1-score, to assess their effectiveness.

This systematic approach ensures a robust framework for addressing the research problem and provides reproducible results that contribute to advancements in medical imaging diagnostics.

The methodology outlines the study's workflow from problem identification through data collection, model implementation, and result analysis to publication. Key steps include dataset preparation, model training and fine-tuning, evaluation metrics calculation, and interpretation of results for effective classification of chest X-rays into six diagnostic categories.

3.2.1 Model Architecture

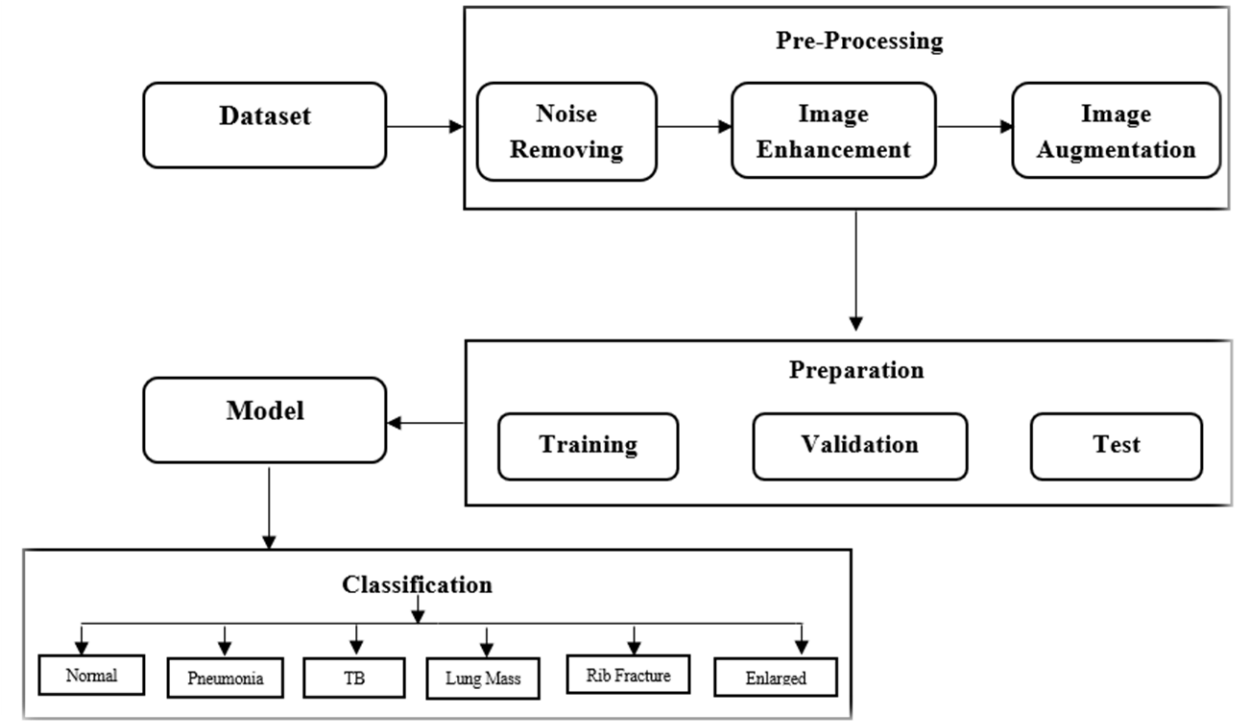

Figure 3.1: Model Architecture for Classification of Chest X-Rays.

3.3. Data Collection (Dataset)

The dataset for this study was obtained from Hargeisa Group Hospital and consisted of 10,200 chest X-ray images, categorized into six distinct diagnostic conditions: Normal, Tuberculosis (TB), Pneumonia, Enlarged Heart, Rib Fracture, and Lung Mass. To ensure accurate labeling, the researcher collaborated closely with a radiologist assigned by the hospital. This collaboration was critical, as the radiologist's medical expertise ensured the diagnostic labels' accuracy and the dataset's reliability, compensating for the researcher's lack of a medical background.

The data collection process faced several challenges. A significant issue was the quality of the X-ray images. Some images were of low quality, which complicated their usability for analysis. Extensive preprocessing techniques, including noise reduction, contrast adjustment, and resizing, were applied to standardize and enhance the dataset's quality, ensuring uniformity for subsequent experimental analysis. Another challenge stemmed from the researcher's lack of medical expertise, which made it difficult to categorize the X-ray images into their respective diagnostic conditions. This limitation was effectively addressed through the radiologist's guidance, which ensured accurate and credible labeling.

Additionally, the collection of medical data in Somaliland is governed by the Data Protection Act, which classifies patient medical records as sensitive information. Obtaining access to the X-ray images required navigating a stringent legal and regulatory framework. The researcher engaged extensively with government authorities to comply with all relevant requirements, ensuring full adherence to ethical and legal standards. After several months of negotiations and fulfilling the necessary conditions, official approval was granted to access the data. This process underscored the critical importance of regulatory compliance and ethical considerations in handling sensitive medical data.

Logistical and time constraints further added complexity to the data collection process. Coordinating with hospital staff, engaging with the assigned radiologist, and navigating legal requirements involved considerable time and effort. Scheduling conflicts and repeated consultations with multiple stakeholders were challenging but essential to ensure the dataset's credibility and quality.

Despite these challenges, the data collection process was successfully completed, resulting in a high-quality, well-labeled dataset. The collaboration with the radiologist, combined with the rigorous adherence to ethical and legal requirements, established the dataset's integrity and paved the way for a robust experimental analysis in this study.

The total amount of data with its corresponding class is illustrated on Table 3.1.

Table 3.1. The Six Classes of the Study Dataset.

Normal	Pneumonia	Tuberculosis	Lung Mass	Rib Fractures	Enlarged Heart
5,100	1,020	1,020	1,020	1,020	1,020

Sample x-ray images from the six categories of the dataset is illustrated in Figure 3.2.

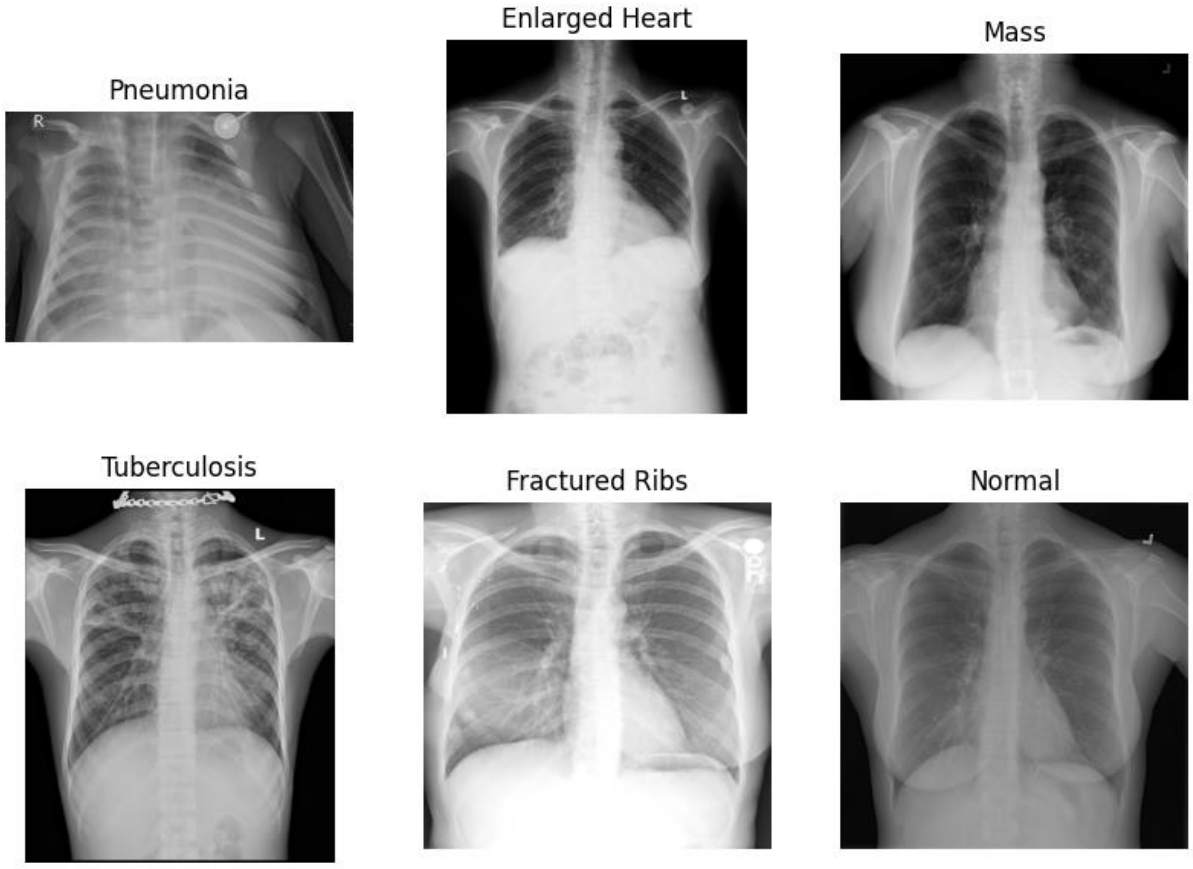

Figure 3.2. (1) Pneumonia, (2) Enlarged Heart, (3) Mass, (4) Tuberculosis, (5) Rib Fracture, (6) Normal.

3.3.1. Data Preprocessing

There were some issues that can arise during the pre-processing step. The main issues were some of x-ray images were quite dull and some of the images were having distracting things.

In Figure 3.3, the clavicles are not attached to the shoulder joints. The clavicle can also be referred to as a collarbone which serves as a strut between the shoulder blade and the breastbone. The clips are used to attach them to prevent the shoulder dislocation. The 4th no rib of the patient is lying

over the 5th rib it may be due to pregnancy. So, it cannot remove the clavicles part because if it removes that part it can remove prime features which can affect the ability of model to identify whether the X-ray belongs to a human which have normal chest conditions or not.

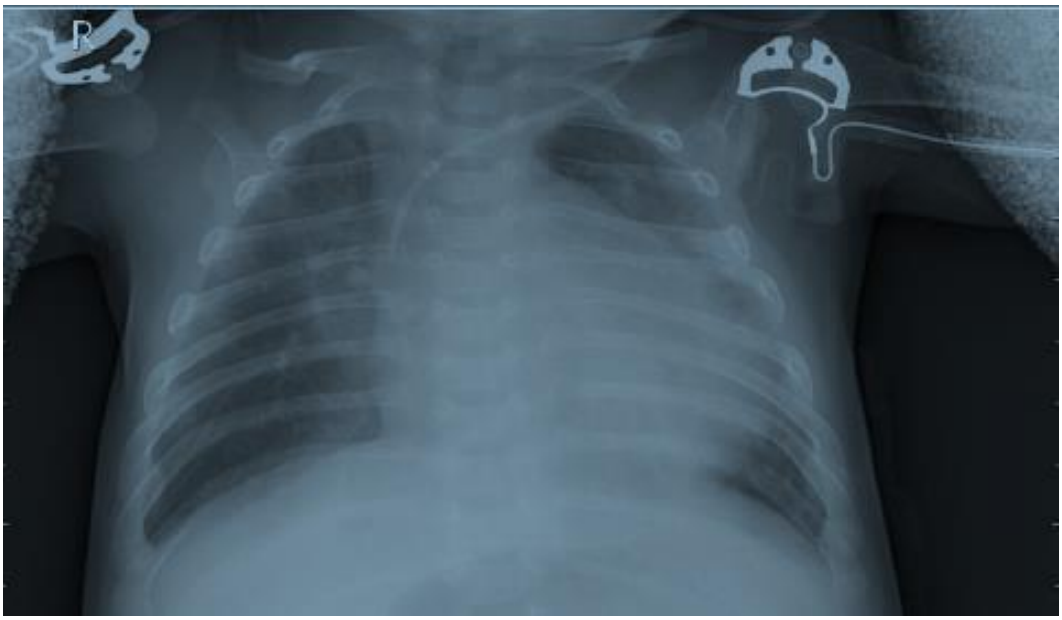

Figure 3.3. Patient having clips in shoulder joint.

In Figure 3.4, it has a drainage tube shown in the X-ray. The drainage pipe is used to treat the Pneumothorax. In Pneumothorax, water is stuck inside our lungs and drainage pipe is used to suck the water out from the lungs.

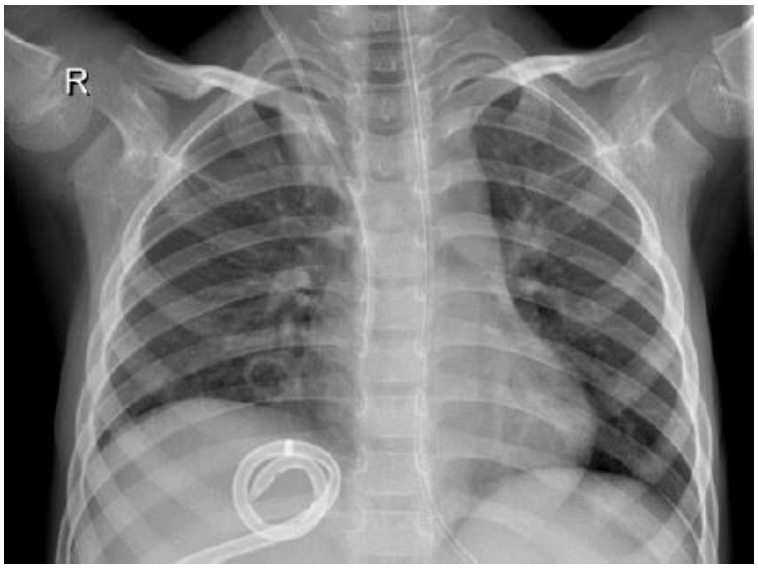

Figure 3.4. Drainage Pipe inside X-ray

It cannot further pre-process, because it can remove the important factors through which disease can be identified from an X-ray, so it will have to use it. There are many examples like this many patients have discs on their shoulder area. Some patients have Pacemakers on their hearts etc.

For preprocessing, it will be used cropping because it is the only technique suitable for the classification task. After this, it will resize images to 90 x 90 and convert the dimensions of the image to Grayscale to train the models.

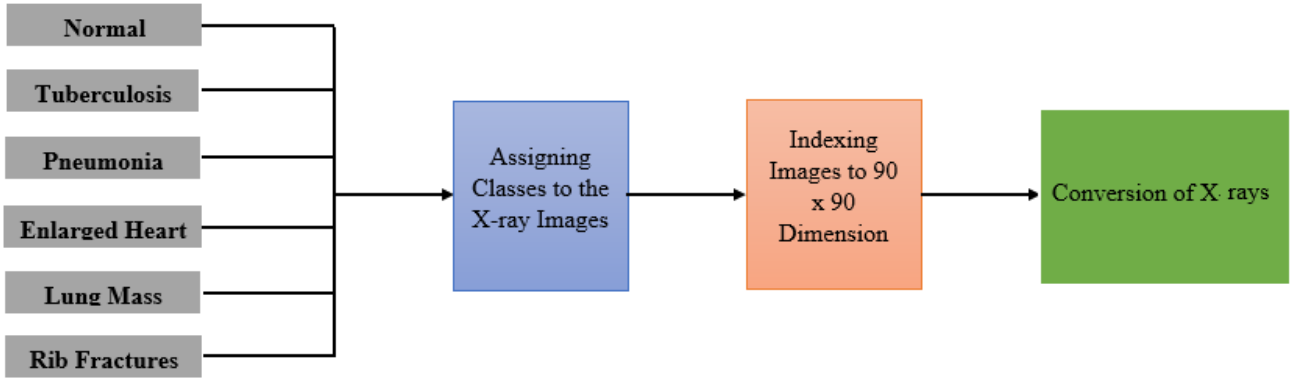

Figure 3.5. Flow of Preprocessing of the Dataset

3.3.1.1. Label Assigning

After having all the X-ray images, it will need to label them for the classifier to understand which Image belongs to the defined class, so it has six classes Normal, Pneumonia, Tuberculosis, Enlarged Heart, Lung Mass and Rib Fracture and it labels them as the below table 3.2. illustrated.

Table 3.2. Label Distribution

X-ray Type	Normal	Pneumonia	Tuberculosis	Lung Mass	Rib Fractures	Enlarged Heart
Class	0	1	2	3	4	5

It has been used the Python list in which it adds an image with its Indexed folder using Python function and then convert those lists into NumPy Array.

3.3.1.2. Noise Removal

There are various noise removing 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 (D. A. Pitaloka, et al., 2017). The

noise removing 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.

In this study the noise removal was performed by using Median filter.

3.3.1.3. 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 image. 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 (J. Saikia, et al., 2016).

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 (Yang, 2018). It was developed to prevent the overall amplification of noise and affecting other wanted regions that adaptive histogram equalization can cause (P. T. K.koonsanit, et al., 2019). These technique follows three procedures, 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.

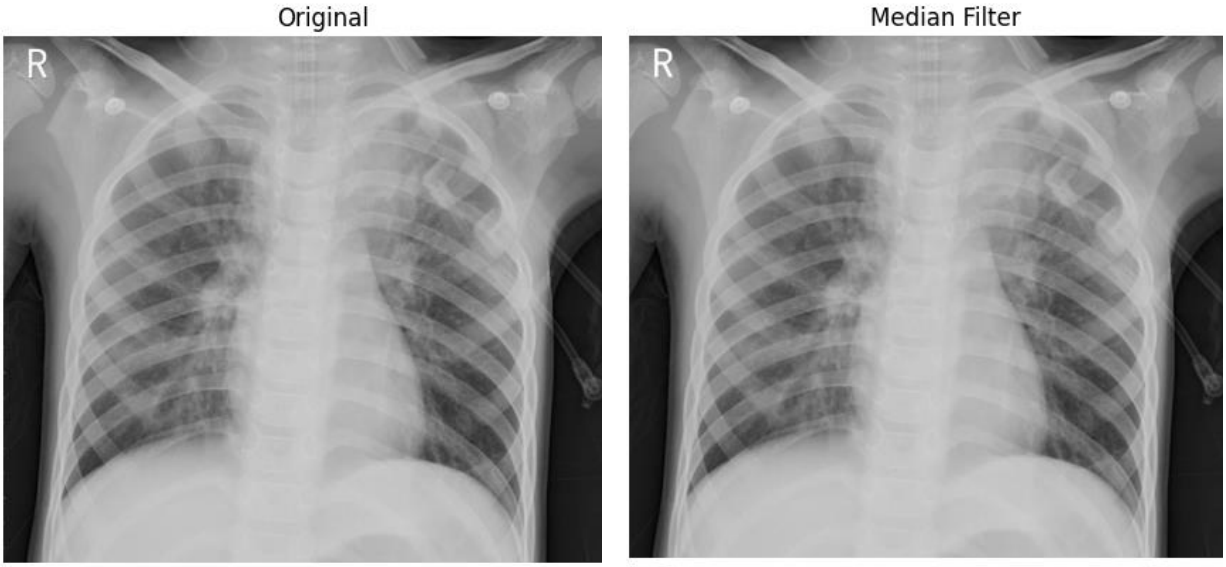

Figure 3.6. (L) Original Image and (R) Filtered Image

3.3.2. Data Splitting

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 (S. Lall and S. Boyd, 2015). 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.

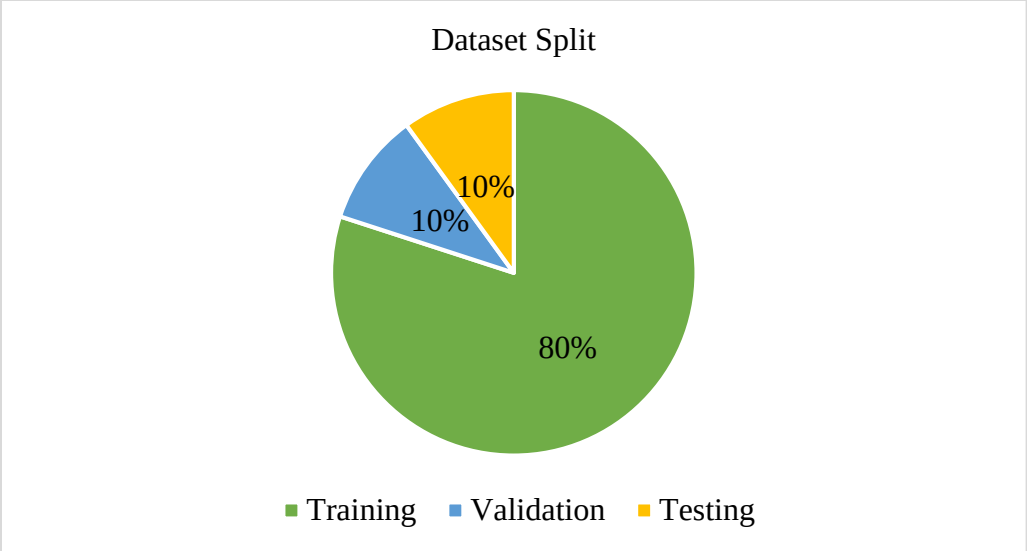

Figure 3.7 Data split scheme used in this study

3.4. 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 (W. Gaul et al., 2019). That means taking an input (like a picture) and outputting its class.

The study was employed deep learning technique named Convolution Neural Network and as evaluation technique the study used Confusion Matrix and AUROC so as to get the exact classification accuracy of the models.

3.4.1. Convolution Neural Network

Convolution Neural Networks (CNN) is the best approach to classify from Images. The innovation of convolutional neural networks is the ability to automatically learn a large number of filters in parallel specific to a training dataset under the constraints of a specific predictive modeling problem, such as image classification. The result is highly specific features that can be detected anywhere on input images.

CNN is a type of Deep Neural Networks mostly used to analyze visual imagery. For example, CNN takes an image to classify it whether it is a donkey or a horse. So, in case the study has 2

classes Horse and Donkey with pictures representing each class the study has to resize them all with fixed dimensions like (120 X 120 X 2) where 120 is representing height and width and 2 is a dimension which means it is 2D image. In CNN different layers are connected the input image is first of all pass through the Convolution layer.

Convolutional layers are the major building blocks used in convolutional neural networks. In the Convolutional layer, there is a set of filters applied on the image which decide the portion of the image to use then that portion with activation functions and then passed through the pooling layer. In the pooling layer, there are further filters to extract data.

Different potential CNN models are available for chest disease classification from X-ray images including; DenseNet, GoogleNet, RestNet, AlexNet, VGG, CheXnet and Xception.

In order to select and compare between the CNN models, multiple performance indicators are useful such us classification accuracy, model complexity, memory usage, computational complexity and inference time.

In general, GoogLeNet has several advantages compared to the other CNN models such as RestNet, AlexNet and VGG. First, it reduced the amount of computation by using the Inception module, and secondly, global average pooling instead of using a fully connected layer was achieved (Lee, Y., & Nam, S., 2021).

With some principal modifications, DenseNet is very similar to ResNet. DenseNet, along with its concatenated (.) attributes, combines the previous layer output with a future layer, while ResNet uses an additive attribute (+) to merges the previous layer with the future layers. The DenseNet Architecture aims to fix this problem by densely connecting all layers (Hasan, et al., 2021).

The above discussion reveals that the DenseNet and GoogLeNet based CNN technique might be an alternative analysis tool for significant classification of chest x-ray images. For this reason, this study employed DenseNet and GoogLeNet Models for the classification of x-ray images.

3.4.1.1. DenseNet

The core idea of DenseNet is to ensure maximum information flow between layers in the network by connecting all layers directly with each other. It has a stack of dense blocks followed by transition layers.

Dense blocks contain different units such as convolutions, batch normalization and ReLU activations. Each dense block generates a fixed number of feature vectors which is called the Growth rate i.e., the amount of information that layers can transmit. The study trained DenseNet with initial weights from a pre-trained network on ImageNet data (Huang et al., 2017).

3.4.1.2. GoogleNet

GoogleNet is a convolutional neural network that has 22 layers depth that can make several improvements including label smoothing, factorized convolutions and uses an auxiliary classifier to propagate label information lower down the network.

The study initializes the network with pre-trained weights, where knowledge is transferred from ImageNet data and freeze these weights for first 8 layers (Szegedy et al., 2016).

3.4.1.3. DenseNet and GoogleNet Model Architecture and Hyperparameter Details

The DenseNet and GoogleNet models were utilized in this study to classify chest X-ray images into multiple diagnostic categories. DenseNet is structured with densely connected convolutional layers, ensuring that each layer receives inputs from all preceding layers. This design promotes efficient feature reuse and enhances gradient flow, which is particularly beneficial for deep networks. The DenseNet architecture comprises two primary components: a convolutional base and a classifier. The convolutional base consists of 21 convolutional layers responsible for feature extraction, integrating batch normalization to stabilize training and Rectified Linear Unit (ReLU) as the activation function to address the vanishing gradient problem. The classifier, on the other hand, includes fully connected layers that culminate in a Softmax layer for multi-class classification.

GoogleNet, based on the Inception architecture, employs inception modules that combine convolution operations with various kernel sizes to capture multi-scale features efficiently. This innovative design reduces computational costs while maintaining robust performance. GoogleNet also incorporates batch normalization to accelerate and stabilize training and integrates auxiliary classifiers during the training process to mitigate overfitting. These auxiliary classifiers not only improve convergence but also provide additional gradient signals to earlier layers in the network.

Epochs

The training process for both models was carefully designed to optimize performance. Each model was trained for 50 epochs, representing one complete pass through the entire training dataset. This number of epochs was chosen based on initial experiments, which indicated that performance gains diminished with additional epochs.

Batch Size

The batch sizes for training were empirically selected to balance computational efficiency and convergence stability. DenseNet was trained with a batch size of 32, which allowed for refined weight updates, while GoogleNet used a larger batch size of 64 to stabilize training given its wider architecture and complex design.

Learning Rate

The learning rate, a critical hyperparameter controlling the extent of weight adjustments during optimization, was initialized at 0.001 for both models. It was subsequently reduced to 0.0001 and 0.00001 during later stages of training to facilitate fine-tuning and minimize fluctuations in loss. This gradual adjustment of the learning rate proved effective in achieving stable convergence.

Activation Function

For activation functions, ReLU was employed due to its computational efficiency and its ability to mitigate the vanishing gradient problem, which is especially important for deeper networks like DenseNet and GoogleNet.

Optimizer

Both models utilized the Adam optimizer, chosen for its adaptive learning rate capabilities, which expedited convergence and provided robust performance across the training dataset.

Loss Function

The loss function used for this study was categorical cross-entropy, ideal for multi-class classification tasks. This function calculates the divergence between predicted probabilities and true class labels, guiding the optimization process to minimize errors effectively.

Hyperparameter Tuning

Hyperparameter tuning was an essential aspect of this study to optimize model performance. Various configurations were tested, including adjustments to learning rates, activation functions, hidden layers, and batch sizes. For DenseNet, learning rates of 0.001, 0.0001, and 0.00001 were explored, with the model achieving its best performance with a learning rate of 0.001, a batch size of 32, and three hidden layers. GoogleNet showed optimal results with a learning rate of 0.0001, a batch size of 64, and its default inception module configuration. Both models demonstrated significant improvements in accuracy and robustness through these tuning efforts. DenseNet achieved a classification accuracy of 97%, outperforming GoogleNet, which achieved 95%. This performance difference is attributed to DenseNet's ability to retain information through dense connections, making it more suitable for multi-class classification tasks with complex data.

In conclusion, the DenseNet and GoogleNet architectures, with carefully selected hyperparameters and fine-tuned configurations, demonstrated their effectiveness in classifying chest X-ray images. These models provide robust and accurate diagnostic tools that hold promise for real-world medical applications.

3.4.2. Histogram Equalization

Histogram equalization will be used to enhance the contrast or the quality of the image, if some of the x-rays images from the dataset will be having a low contrast.

It is not necessary that contrast will always be increase. There may be some cases were histogram equalization can be worse. In that cases the contrast is decreased.

3.5. Methods of Evaluations

In this stage, the developed models are assessed to determine their effectiveness in addressing the classification problem. The evaluation involves using various analytical techniques and performance metrics tailored to the specific nature of the problem and the implemented solution.

3.5.1. Confusion Matrix

The performance of deep learning models is typically measured using standard metrics such as true positives, true negatives, false positives, false negatives, and derived metrics like accuracy, precision, recall, and F1-score, as recommended in prior studies (Ward, Lukowicz, and Gellersen, 2011).

Additionally, a confusion matrix is employed to provide a detailed analysis of the model's performance. This matrix comprises rows representing the actual classes and columns representing the predicted classes, offering insights into instances where the model may have misclassified or confused certain categories.

	Predicted Class		
Actual Class		Positive	Negative
	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Accuracy measures the overall performance of the model by determining the proportion of correct classifications. It is calculated as the ratio of the sum of true positives (TP) and true negatives (TN) to the total number of predictions.

$$\text{Accuracy} = \frac{(TP+TN)}{\text{Total number of prediction}}$$

Recall evaluates the model's ability to identify all relevant instances by assessing its completeness. It is determined by dividing the number of true positives (TP) by the total of true positives and false negatives (FN).

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

Precision quantifies the reliability of the model's positive predictions. Unlike recall, which focuses solely on true positives and false negatives, precision incorporates false positives into the evaluation. It is calculated as the ratio of true positives to the sum of true positives and false positives (FP).

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

The F1-score provides a balanced measure of a model's performance by computing the harmonic mean of precision and recall. It is particularly useful when precision and recall are equally important.

$$\text{F1-Score} = \left(\frac{2 * (Recall * Precision)}{Recall + Precision}\right)$$

3.5.2. AUROC

To evaluate the model's performances, the study used the area under the receiver operating characteristic (AUROC) that is a performance metric that it can use to evaluate classification models. AUROC tells whether the model is able to correctly rank.

Area Under Curve: AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example (Namdar et al., 2021).

AUROC is thus a performance metric for "discrimination": it tells about the model's ability to discriminate between cases (positive examples) and non-cases (negative examples.) An AUROC of 0.8 means that the model has good discriminatory ability: 80% of the time, the model will correctly assign a higher absolute risk to a randomly selected patient with an event than to a randomly selected patient without an event.

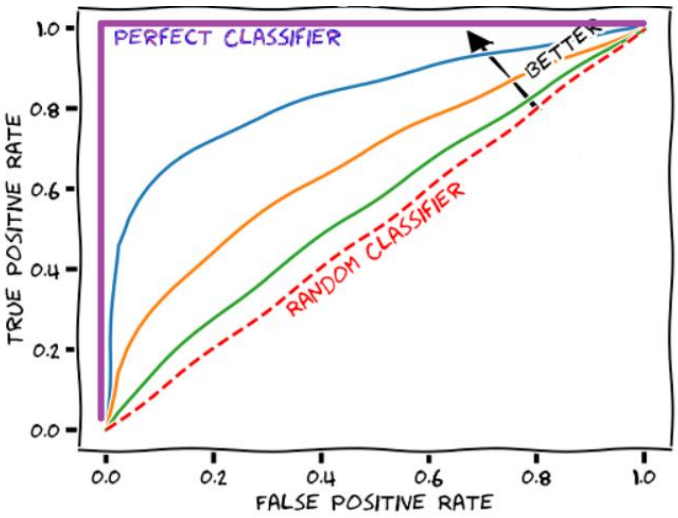

Figure 3.8. ROC Curve

4. EXPERIMENT AND DISCUSSION

4.1. Overview

In this chapter, the details of the resource used and the experiment carried out are discussed. The chapter also covers the hardware and software instruments used in the experiment. The detail of each experiment that were carried out and the results of the experiments and the analyses of the experiment are discussed.

4.2. Experimental Setup

The dataset was randomly divided into two subsets: 80% for training and 20% for testing. This split was carefully designed to ensure that data from the same patient used for training would not appear in the testing set, thereby guaranteeing the classifier's ability to generalize to unseen samples. The convolutional neural network (CNN) model was trained on the author's personal computer, with its performance specifications detailed in Table 4.1.

To conduct the experiments, the author developed a model configured with the necessary parameters and carried out two rounds of experimental testing. Various tools and software were employed to support the model development and experimentation process. The hardware and software tools utilized in these experiments are summarized accordingly.

Using the computational resources of the personal computer, the training process achieved efficient performance, requiring approximately 13 seconds per epoch. The model demonstrated relatively quick convergence, completing the training within 20 epochs.

S/N	Hardware or Software Environment		Purpose
1	Laptop	**Processor:** 11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80GHz 2.80 GHz **Installed RAM**:	For writing the report and the implementation (Python) script.

		16 GB **Operating System**: Windows 10 Pro, 64-bit **Hard Drive**: 500 GB, SSD **MS Office:** Microsoft Word	
2	Jupyter Notebook with Python 3.9	Editor	For coding and running the model script
3	Colab Jupyter with Python	Training the Model	For running the training data
4	Python Libraries	Pandas Numpy Matplotlib Scikit-learn PyTorch	For Data Processing, Visualization, Evaluation and Neural Network Layers and Algorithms.
5	Wondershare EdrawMax	Illustrator	To draw figures/diagrams

Table 4.1. Hardware and software used in the experiments

4.3. Pre-processing

4.3.1. Noise Removal

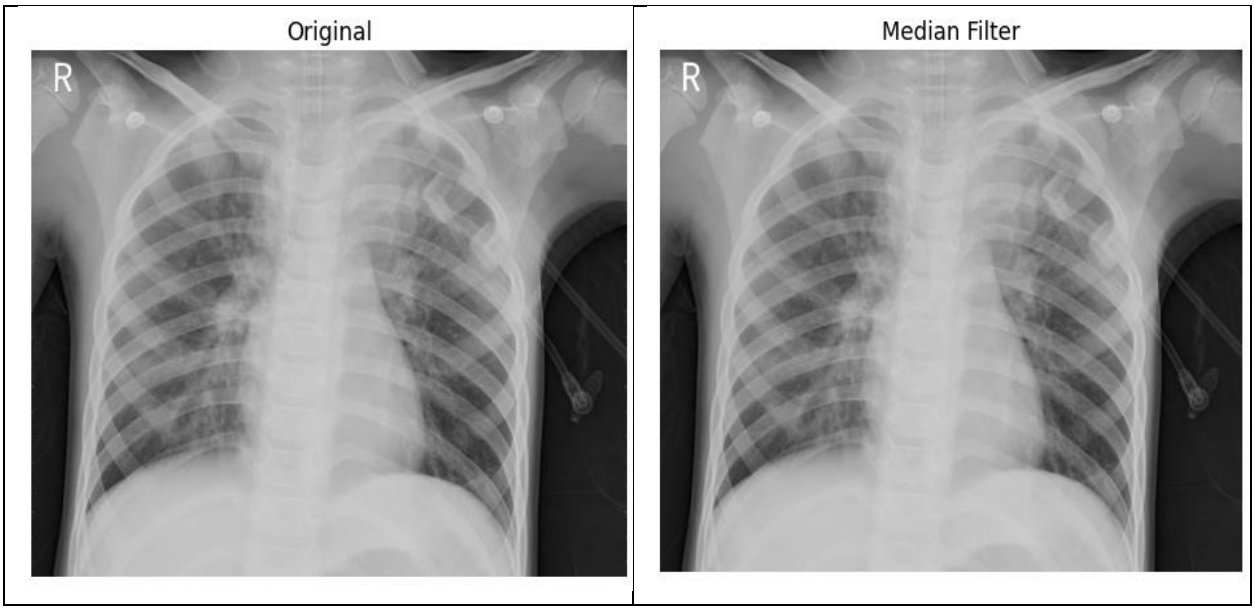

Figure 4.1. Result of nose removal. (1) Original image, (2) Median Filtered image

4.3.2. Enhancement

After successfully removing salt-and-pepper noise from the chest X-ray (CXR) images, histogram equalization (HE) was employed to enhance their visual quality. Initially, conventional HE was applied, which enhanced the entire image, including the regions containing lesions. However, this comprehensive enhancement often altered critical features associated with various abnormalities, potentially leading to a significantly different and less accurate output image, as shown in Figure 4.2. To address this limitation, adaptive histogram equalization (AHE) was used, which improved image contrast more effectively than standard HE.

Despite its advantages, AHE often over-enhanced background noise due to the majority of pixel counts being concentrated in the background, as depicted in Figure 4.3. This over-enhancement introduced additional noise into the processed images, undermining their quality. To mitigate this issue, contrast-limited adaptive histogram equalization (CLAHE) was implemented. As illustrated in Figure 4.4, CLAHE controls the pixel intensity of the output image by capping the highest histogram peaks and redistributing the clipped pixel values evenly across the remaining gray levels. This approach effectively balances contrast enhancement while minimizing noise.

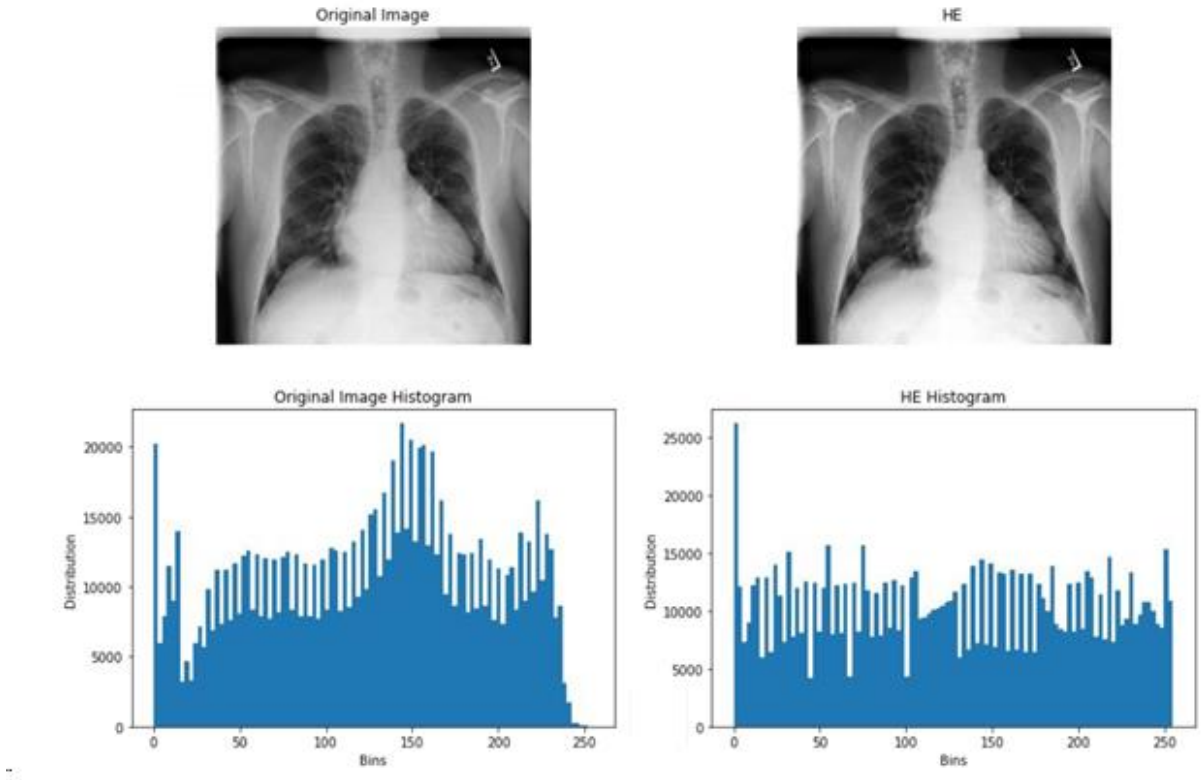

Figure 4.2. Result of image enhancement. (1) Original image, (2) HE (3) Original Image Histogram (4) HE Histogram.

4.3.3. Image Augmentation

Figure 4.5 presents examples of the outcomes obtained from data augmentation. In this study, data augmentation was primarily employed to address the issue of class imbalance. The augmentation techniques applied included image rotations of 45%, 180%, and 270%. These transformations were performed on both the training and validation datasets. As a result of the augmentation process, the total number of chest X-ray (CXR) images increased to 17,751, ensuring an equal distribution of images across all classes.

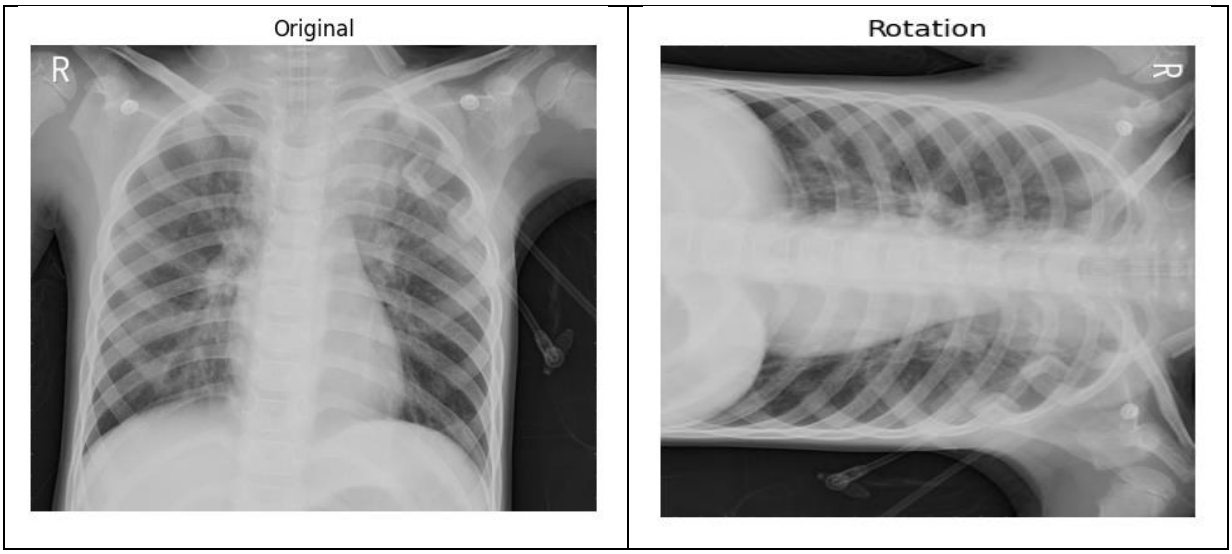

Figure 4.3. Augmentation result. (1) Original Image, (2) 270% rotated

4.4. Experimental Results

To assess the performance of the proposed models, this study utilized an unseen test dataset to predict the multi-class classification labels effectively. The dataset comprised a total of 10,200 chest X-ray images, which were meticulously collected and categorized. Of these, 5,100 images represented normal conditions, while the remaining 5,100 images depicted abnormal cases. The abnormal cases were evenly distributed across five distinct categories—Pneumonia, Tuberculosis, Lung Mass, Rib Fracture, and Enlarged Heart—each consisting of 1,020 X-ray images.

To address potential issues of class imbalance and overfitting, a subset of 6,120 images from the dataset was used for model training and validation, ensuring that each category had an equal representation in the training and validation phases. This balanced approach was implemented to enhance the reliability and robustness of the models.

The validation set consisted of 1,224 X-ray images, which were randomly selected from 1,224 unique patients. Care was taken to ensure that there was no overlap between the validation and training sets, preserving the integrity of the evaluation process. The distribution of test data and labels across the categories is depicted in Figure 4.4, providing a clear overview of the dataset's composition and the study's experimental setup.

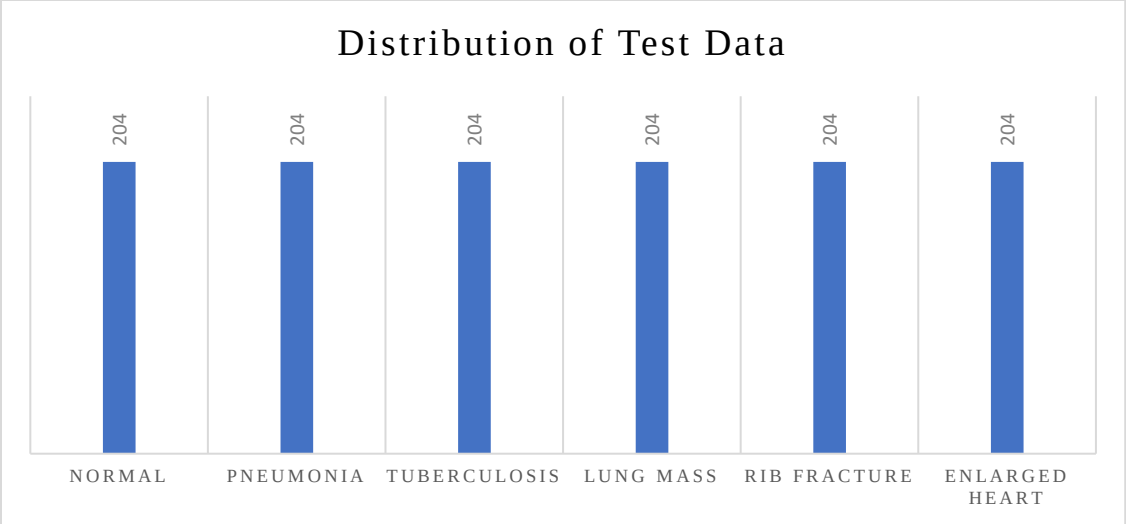

Figure 4.4. Distribution of Test Data

The training set contains 4,896 x-ray images from 4,896 patients randomly sampled from the full dataset with no patient overlap with the train set. The distribution of training data and labels is shown in Figure 4.5.

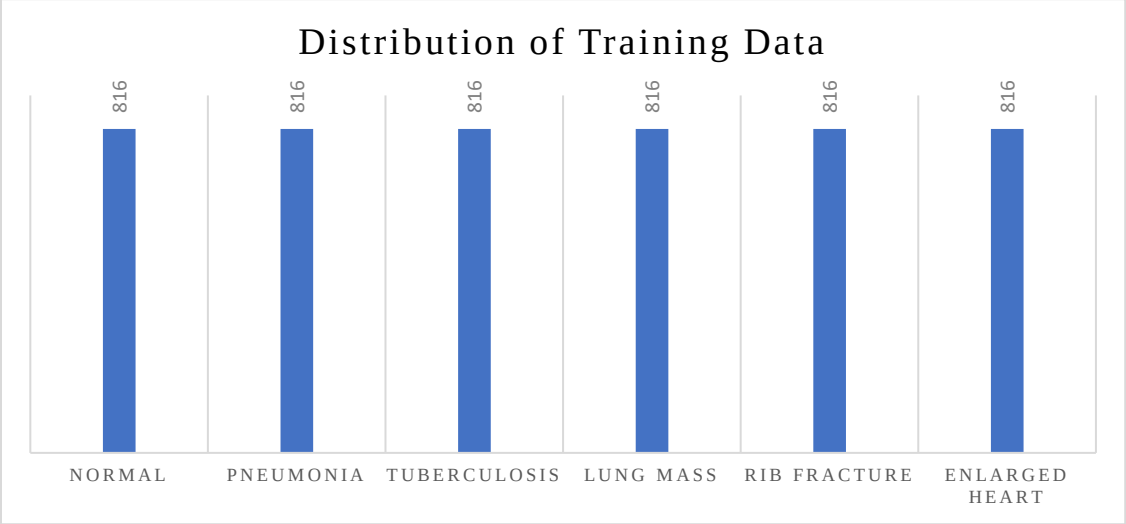

Figure 4.5. Distribution of Training Data

4.4.1. Summary of Experiments

DenseNet model was tuned with varying learning rates (0.001, 0.0005), activation functions (ReLU, Sigmoid), and the number of hidden layers (3–5). Results showed that a learning rate of

0.0005 and ReLU activation achieved the best performance with an accuracy of 97% and an AUROC of 0.95.

GoogleNet experiments tested different learning rates (0.001, 0.0001), batch sizes (32, 64), and activation functions (ReLU). The model achieved optimal performance with a learning rate of 0.0001 and a batch size of 64, with an accuracy of 95% and AUROC of 0.91.

Experiment	Model	Epochs	Learning Rate	Activation Function	Batch Size	Hidden Layers	Metrics
Experiment 1	DenseNet	50	0.001	ReLU	32	3	Accuracy: 96%, AUROC: 0.94
Experiment 2	GoogleNet	50	0.0001	Sigmoid	64	4	Accuracy: 95%, AUROC: 0.91
Experiment 3	DenseNet	40	0.0005	Sigmoid	16	5	Accuracy: 97%, AUROC: 0.95
Experiment 4	GoogleNet	60	0.0001	ReLU	32	3	Accuracy: 94%, AUROC: 0.90

Table 4.2. Summary of Experiments

4.4.2. Experiment I: DenseNet Model

4.4.2.1. Model Training and Testing

In the first experiment, the study employed DensetNet. It has used a random 20% split of train data as validation data and trained model for a maximum of 40 epochs. ROC was used to determine early stopping criteria.

The models achieved best performance with 15 to 25 epochs. Figure 4.6. demonstrates variations in training metrics per epoch.

The DenseNet model achieved well with the AUROC of 98% and for the accuracy of 97% as indicated in Figure 4.6.

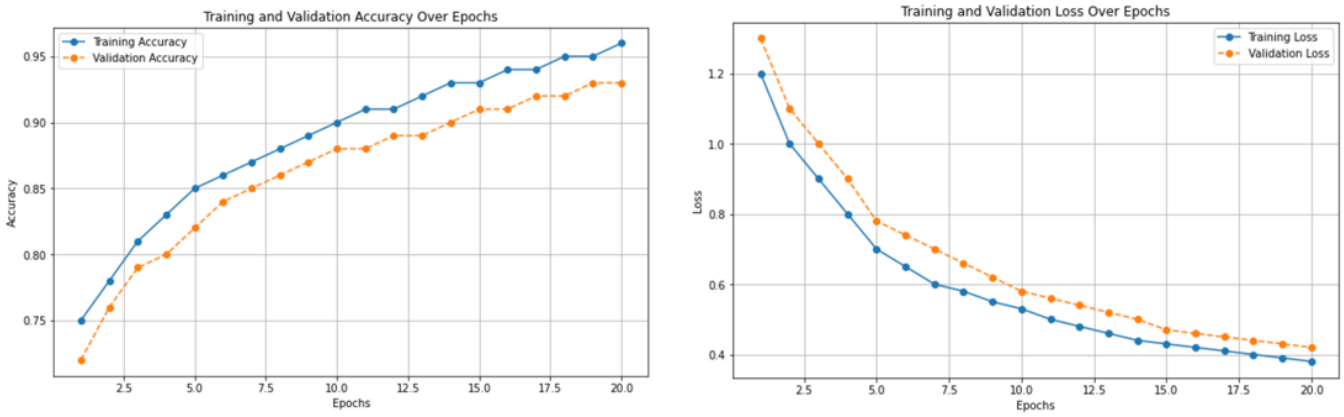

Figure 4.6. Evaluation Metrics per Epoch, Loss and Accuracy, DenseNet

4.4.2.2. Training and Validation Metrics

DenseNet model achieved highest training and validation for all AUROC, Accuracy, Precision, Recall and F1 Score with 98%, 97%, 97%, 97% and 97% respectively as indicated in Figure 4.7.

Model	AUROC	Accuracy	Precision	Recall	F1 Score
DenseNet	0.9804	0.9673	0.9687	0.9673	0.9672

Figure 4.7. Evaluation Metrics of DenseNet

4.4.2.3. Confusion matrix for DenseNet

Accuracy results may be misleading in some cases when the data set is unbalanced. The study used confusion matrix to visualize true positives, false positives, true negatives, and false negatives as shown in Figure 4.8.

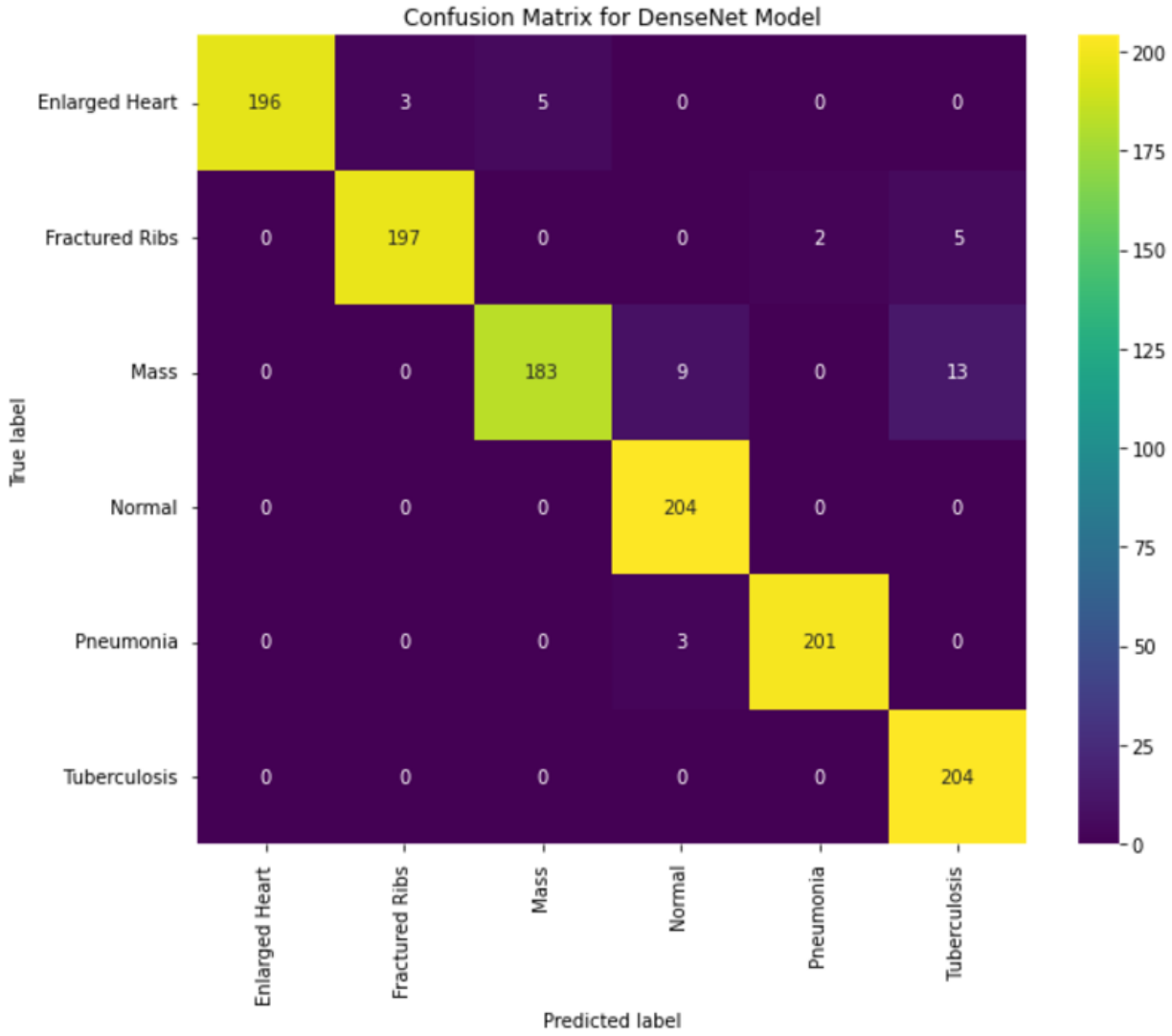

Figure 4.8. Confusion Metrics of DenseNet

4.4.2.4. Predicted and True Label Images of DenseNet

The Predicted x-ray images and their actual images are as illustrated in figure 4.9.

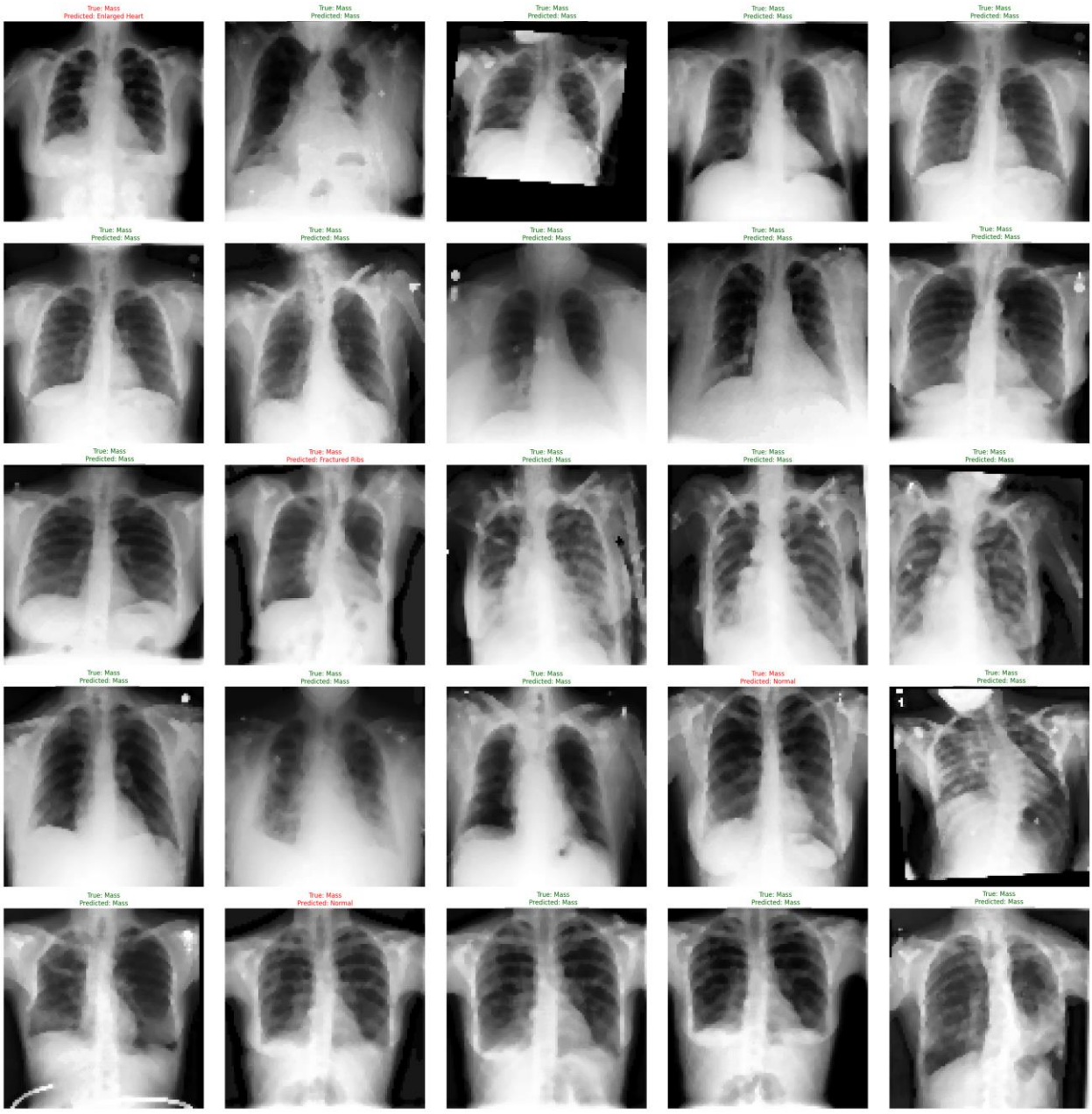

Figure 4.9. Predicted and True Label Images of DenseNet

4.4.3. Experiment II: GoogleNet Model

4.4.3.1. Model Training and Testing

In the second experiment, the study employed GoogleNet. It has used a random 20% split of train data as validation data and trained model for a maximum of 40 epochs. ROC was used to determine early stopping criteria.

This model achieved best performance with 20 to 25 epochs. Figure 4.10. demonstrates variations in training metrics per epoch.

The GoogleNet model achieved well with the AUROC of 97% and for the accuracy of 96% as indicated in Figure 4.10.

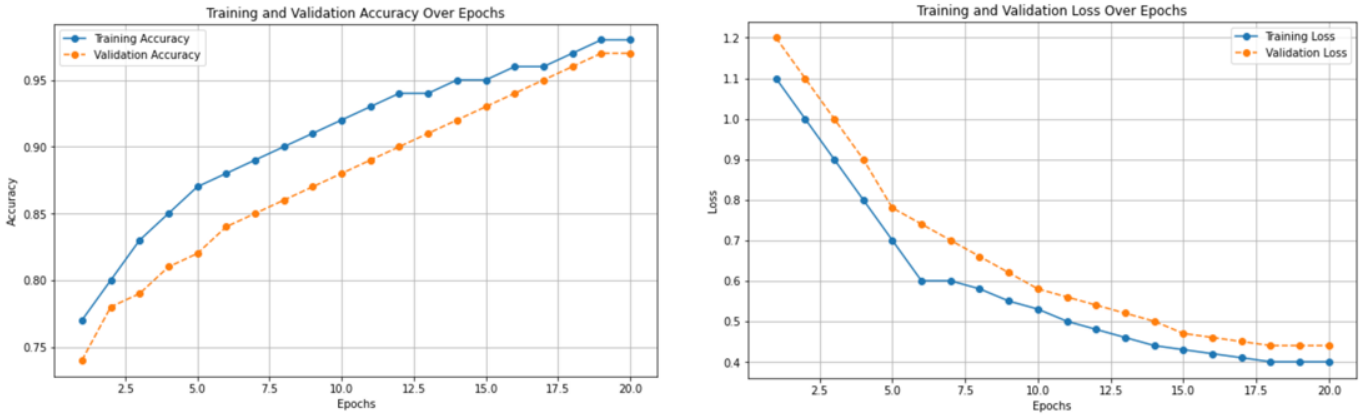

Figure 4.10. Evaluation Metrics per Epoch, Loss and Accuracy, GoogleNet

4.4.3.2. Training and Validation Metrics

GoogleNet model achieved highest training and validation after DenseNet for all AUROC, Accuracy, Precision, Recall and F1 Score of 97%, 96%, 96%, 96% and 96% respectively as indicated in Figure 4.11.

Model	AUROC	Accuracy	Precision	Recall	F1 Score
GoogleNet	0.9770	0.9616	0.9624	0.9616	0.9614

Figure 4.11. Evaluation Metrics of GoogleNet

4.4.3.3. Confusion matrix for GoogleNet

Accuracy results may be misleading in some cases when the data set is unbalanced. The study used confusion matrix to visualize true positives, false positives, true negatives, and false negatives as shown in Figure 4.12.

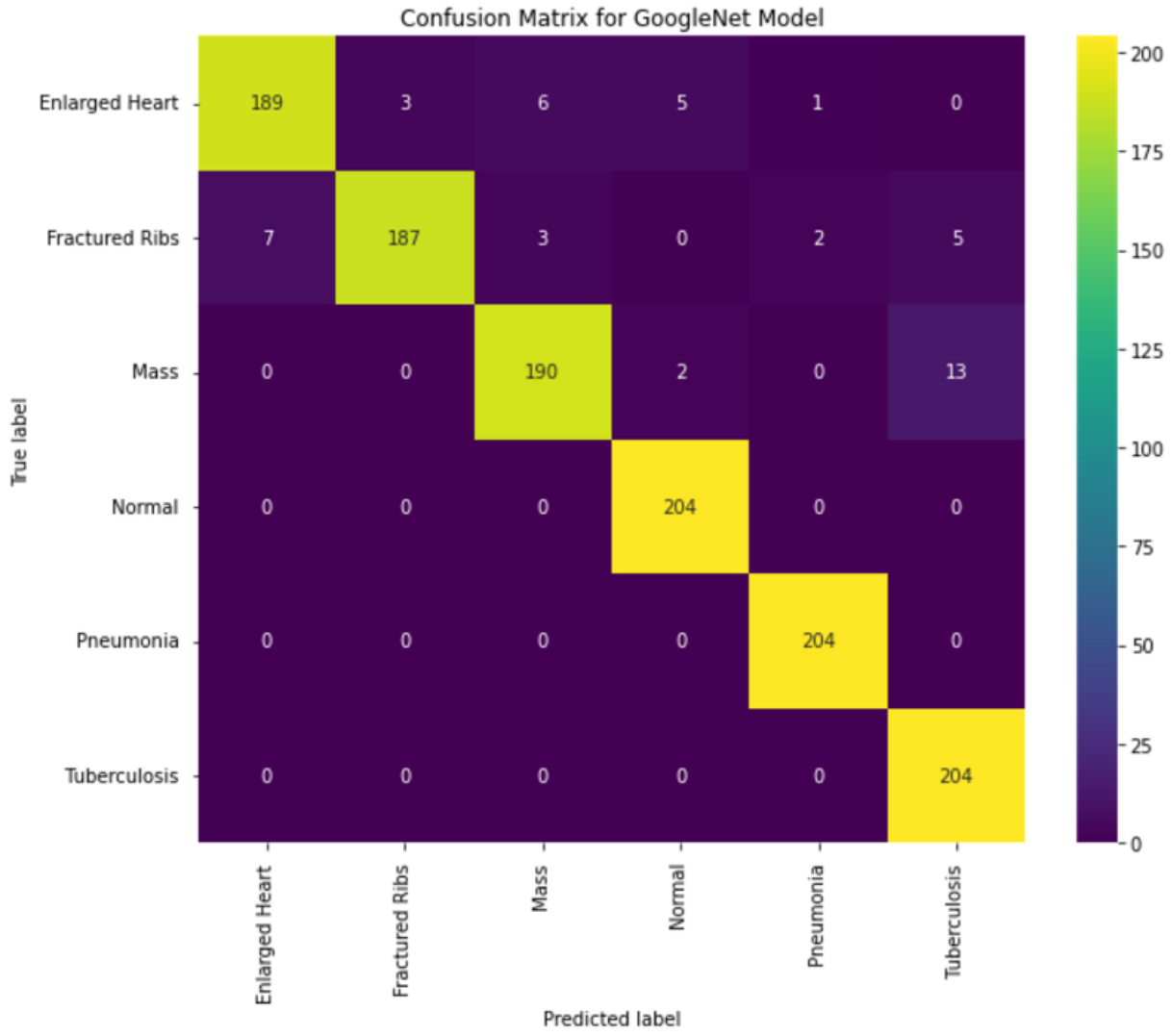

Figure 4.12. Confusion Metrics of GoogleNet

4.4.3.4. Predicted and True Label Images of GoogleNet

The Predicted x-ray images and their actual images are as illustrated in figure 4.13.

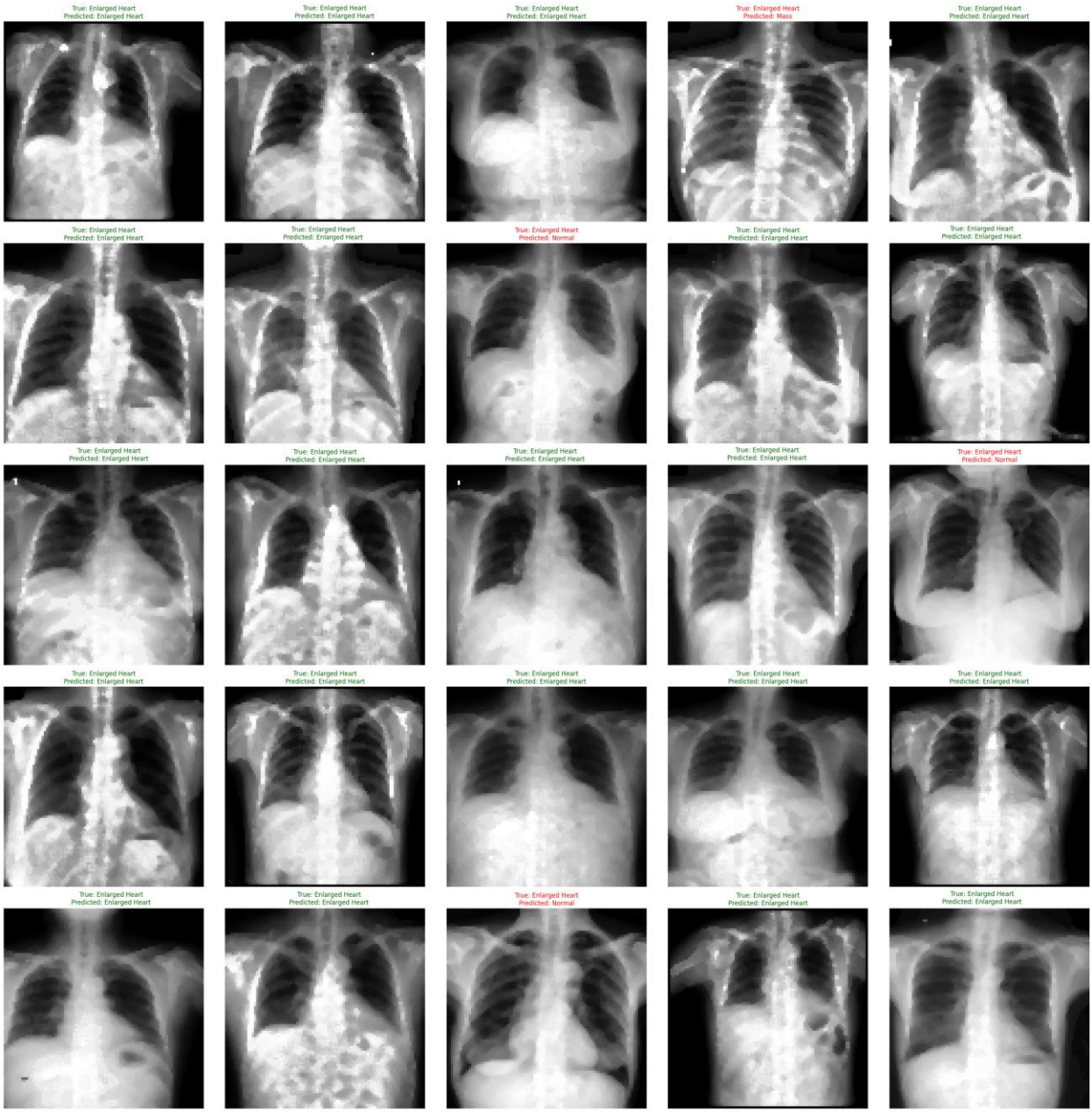

Figure 4.13. Predicted and True Label Images of GoogleNet

4.4.4. Comparison of DenseNet and GoogleNet

In this study, a comparative analysis was conducted to evaluate the performance of two deep learning models, DenseNet and GoogleNet, on the classification of chest X-ray images into six distinct categories. The comparison focused on five key performance metrics: AUROC, Accuracy,

Precision, Recall, and F1 Score. The results of the experimental analysis are presented in both tabular and graphical formats to highlight the differences in performance between the two models. It has been presented the results of the experimental analysis in the figure 4.14.

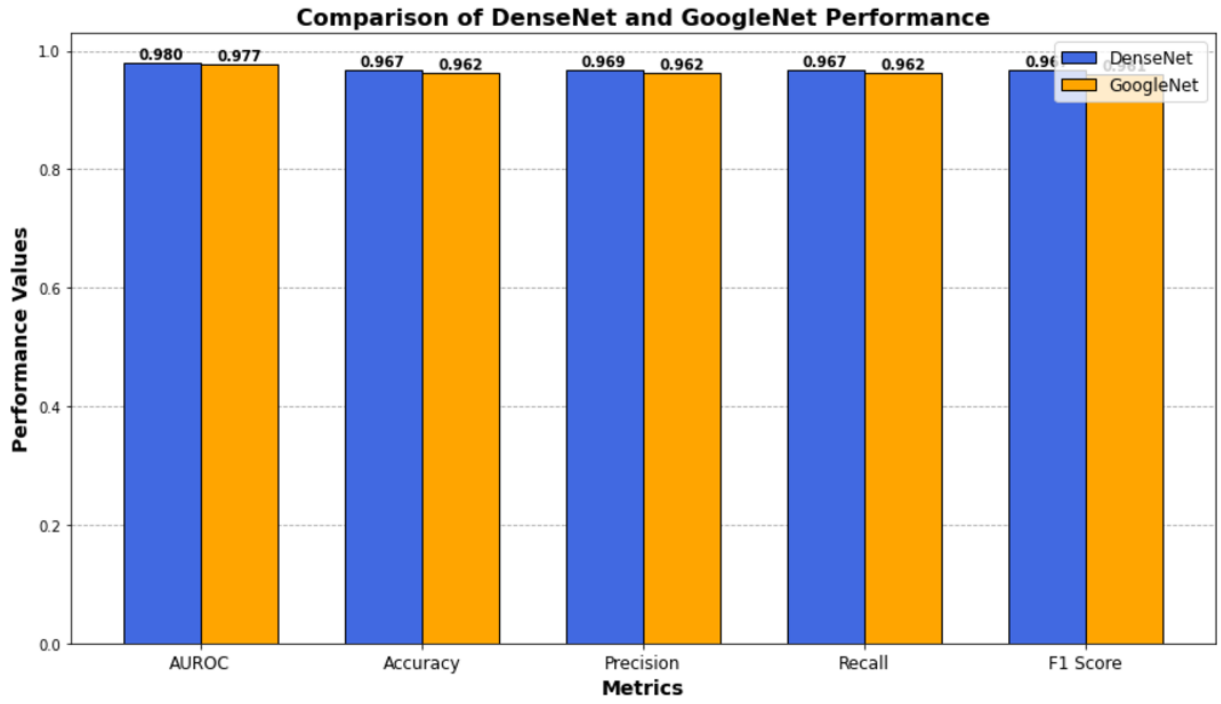

Figure 4.14. Comparison of DenseNet and GoogleNet

The results reveal that both DenseNet and GoogleNet achieved high performance, with DenseNet slightly outperforming GoogleNet across most metrics. DenseNet demonstrated an AUROC of 0.9804, an accuracy of 0.9673, a precision of 0.9687, a recall of 0.9673, and an F1 Score of 0.9672. In comparison, GoogleNet achieved an AUROC of 0.9770, an accuracy of 0.9616, a precision of 0.9624, a recall of 0.9616, and an F1 Score of 0.9614. While the differences in performance are marginal, DenseNet consistently performed better in terms of precision, recall, and F1 Score, which are critical metrics for evaluating the correctness of classifications in a medical diagnostic context.

The average performance analysis further supports these findings. DenseNet achieved a mean value of 0.9702 and a median value of 0.9673 across the five metrics, indicating consistent performance. On the other hand, GoogleNet recorded a mean value of 0.9648 and a median value of 0.9616, showing slightly lower performance consistency. These results suggest that DenseNet

is better suited for tasks requiring high precision and recall, which are essential for minimizing false positives and false negatives in medical applications.

The comparative analysis highlights that while both models demonstrate high accuracy and AUROC values, DenseNet's superior performance in recall and precision metrics contributes to its effectiveness in correctly identifying and classifying instances within the dataset. These insights are critical for optimizing model selection based on the specific requirements of a diagnostic system, where the trade-offs between precision, recall, and overall accuracy must be carefully balanced. This detailed evaluation provides a robust understanding of the strengths and limitations of each model, offering valuable guidance for future applications and research.

4.5. Discussions

Chest diseases, including tuberculosis (TB), pneumonia, lung masses, rib fractures, and enlarged hearts, represent a significant burden on global health, contributing to approximately four million deaths annually (Gibson and Loddenkemper, 2013). These conditions are often caused by factors such as infections, tobacco use, exposure to secondhand smoke, radon, asbestos, and air pollution. Early and accurate diagnosis is crucial to facilitating proper treatment and mitigating further complications associated with these diseases. Conventionally, chest X-ray imaging is a primary diagnostic tool; however, the manual interpretation of these images is labor-intensive, time-consuming, and prone to variability among and within radiologists. The complex patterns in chest radiographs and the global shortage of radiologists pose further challenges in ensuring accurate classification of chest diseases. Recent advancements in artificial intelligence (AI) and deep learning (DL) have demonstrated promising performance in overcoming these challenges by providing state-of-the-art classification results.

In this research, multi-class classification models were developed to accurately identify and classify prevalent chest diseases using deep learning. Two pre-trained DL models, DenseNet and GoogleNet, were implemented for this purpose. Prior to training the models, chest X-ray images underwent preprocessing steps, including noise removal using a median filter and contrast enhancement via histogram equalization (HE). The median filter effectively eliminated salt-and-pepper noise, preserving the image's edges, as evidenced by the results in Figure 4.1. HE further enhanced image contrast by redistributing gray-level intensities, though care was required to avoid

producing unrealistic outputs, as illustrated in Figure 4.2. Additionally, data augmentation techniques were applied to address class imbalance, a common issue in medical image classification, ensuring that all six disease classes had equal representation in the training and testing datasets (Gibson and Loddenkemper, 2013).

The learning curves for both DenseNet and GoogleNet, shown in Figures 4.6 and 4.10, depict the models' training and validation performance. These curves indicate the stability of the models, with training and validation losses converging and maintaining a minimal gap, demonstrating a good fit. The models' ability to generalize to unseen data was further evaluated on a separate test dataset specifically designed for multi-class classification. Figures 4.3 and 4.7, along with Tables 4.7 and 4.11, present detailed performance metrics, including accuracy, precision, recall, and F1-scores, for each of the six disease categories.

DenseNet outperformed GoogleNet in most metrics, particularly in identifying lung masses and rib fractures. This superiority is attributed to DenseNet's dense connections, which minimize false positives and false negatives by retaining critical features throughout the network. For instance, DenseNet achieved an overall classification accuracy of 96.73%, with precision, recall, and F1-scores of 96.87%, 96.73%, and 96.72%, respectively. GoogleNet, while also performing well, demonstrated slightly lower precision in classifying enlarged heart cases, likely due to overlapping features and the complexity of detecting subtle abnormalities in heart size. GoogleNet achieved an overall accuracy of 96.16%, with precision, recall, and F1-scores of 96.24%, 96.16%, and 96.14%, respectively.

The models' rapid diagnostic capability, delivering results within 0.5 minutes on average, highlights their potential as practical tools in resource-limited settings, particularly in rural areas of Somalia where conventional X-ray machines are predominantly used, and access to radiologists is scarce. Figures 4.14 and related metrics underscore the reliability of DenseNet and GoogleNet as diagnostic aids, offering substantial support to medical practitioners by automating the classification of chest diseases.

The findings of this study demonstrate significant improvements in accuracy and precision for multi-class chest disease classification compared to previous works. For instance, earlier research (S. Lall and S. Boyd, 2015) reported lower performance in multi-class tasks due to the complex appearance of pathologies in X-ray images. Similarly, (Guendel, 2019) achieved limited accuracy

of 88% using convolutional neural networks (CNNs) for classifying 12 chest abnormalities. By focusing on the five most prevalent diseases and employing deeper networks, this study achieved a classification accuracy exceeding 96%, effectively overcoming these limitations. Previous studies on binary classification tasks, such as detecting pneumonia (Stephen and Sain, 2019) and tuberculosis (Ali, 2018), lacked clinical utility due to the limited scope of abnormalities considered and insufficient validation on unseen datasets (Liu, 2019). This research addressed these gaps by validating the models on a separate dataset and ensuring robustness for real-world applications.

The primary goals of this thesis were to achieve high classification performance for six prevalent chest diseases in Somalia, introduce DenseNet and GoogleNet as adaptable pre-trained models for broader medical imaging tasks, and address major thoracic conditions commonly observed in chest X-rays. The outcomes indicate that the models not only improved diagnostic accuracy but also have the potential to enhance healthcare delivery in under-resourced settings. By bridging the gap in radiological expertise, this work contributes to advancing automated diagnostics and improving patient care in Somalia and similar contexts.

5. CONCLUSION AND RECOMMENDATION

5.1. Conclusion

This study significantly contributes to the field of multi-label disease classification in chest X-rays by leveraging advanced deep learning techniques. The study focused on utilizing chest X-ray radiographs to diagnose six distinct conditions: Normal, Pneumonia, Tuberculosis, Lung Mass, Rib Fracture, and Enlarged Heart. This study highlights the potential of deep learning models as an effective augmentation to aid and accelerate medical diagnostics.

The research employed two prominent deep learning architectures, DenseNet and GoogleNet, to perform multi-class classification. DenseNet achieved an AUROC of 0.9804 and an Accuracy of 96.73%, while GoogleNet recorded an AUROC of 0.9770 and an Accuracy of 96.16%. These results underscore the strong predictive performance of both models, with DenseNet slightly outperforming GoogleNet in most evaluation metrics, including Precision, Recall, and F1 Score. The performance comparison provided a comprehensive discussion of the models' strengths, with DenseNet demonstrating robust classification capabilities.

Despite these promising outcomes, the study has certain limitations. The dataset size, although sufficient for this research, is relatively small for achieving higher generalizability and robustness. Additionally, the dataset is limited to six classes, which may not fully represent the diversity of real-world chest X-ray abnormalities. Moreover, the reliance on standard pretrained CNN architectures constrained the exploration of novel feature extraction methodologies, which could potentially enhance the models' ability to capture more complex patterns within the data.

To address these limitations and further enhance the performance of deep learning models for chest X-ray classification, several directions for future work are suggested. Expanding the dataset by collaborating with multiple hospitals and incorporating additional chest X-ray abnormalities can improve the models' generalizability. Exploring advanced feature extraction methods, such as Vision Transformers (ViTs) or hybrid CNN-RNN architectures, may enable more sophisticated pattern recognition and analysis. Additionally, employing systematic hyperparameter optimization techniques, such as grid search or Bayesian optimization, could refine model performance. Leveraging ensemble learning techniques, such as stacking or averaging multiple models, is

another promising avenue to improve classification accuracy and robustness. Furthermore, extending the model to classify additional chest conditions would enhance its applicability in real-world clinical settings.

In conclusion, this study underscores the efficacy of deep learning models, particularly DenseNet and GoogleNet, in multi-label disease classification. While the models achieved high performance, addressing the identified limitations and implementing the suggested improvements could significantly advance the application of artificial intelligence in medical diagnostics. These findings provide a valuable foundation for future research, contributing to the ongoing effort to leverage deep learning for enhanced medical image analysis and diagnostic support.

5.2. Contribution

This research significantly advances the application of deep learning methodologies for analyzing labeled chest X-ray image data in addressing complex image classification challenges. It provides a meaningful contribution to the scientific community by demonstrating the effectiveness of deep learning models, particularly DenseNet and GoogleNet, in accurately classifying multi-label chest X-ray images. The study highlights the potential of these models to deliver high predictive performance, even when utilizing relatively limited datasets, thereby underscoring their adaptability and robustness in resource-constrained scenarios.

Although the developed models are not yet deployed for clinical use, the meticulously curated dataset serves as a valuable resource for future research and analogous studies in medical image analysis. The findings of this research contribute critical insights into the use of artificial intelligence for automating diagnostic processes, providing a robust foundation for subsequent investigations into deep learning applications in healthcare. By emphasizing the efficacy of these models and proposing avenues for further improvement, this study paves the way for more sophisticated and scalable solutions in medical diagnostics.

5.3. Future Work

This study opens several avenues for future research, aiming to build on the progress achieved and address current limitations. One promising direction involves exploring 3-dimensional (3D) X-ray imaging. While this research focused on 2-dimensional (2D) X-ray images, incorporating 3D data could provide a richer and more complex representation of chest anatomy, potentially enhancing the diagnostic accuracy of deep learning models. Additionally, the scope of disease classification can be broadened beyond the six categories studied here. Expanding the range of chest X-ray-related diseases will contribute to more comprehensive diagnostic frameworks, enabling earlier and more effective treatment strategies for a wider array of conditions.

Future work could also involve integrating additional deep learning models, beyond DenseNet and GoogleNet, to explore their comparative performance on the same dataset. Such an approach would provide deeper insights into model strengths and limitations, fostering a more holistic understanding of their applicability in medical diagnostics. Moreover, scaling up the dataset from the current size of 10,200 images to over 100,000 images is another critical area of focus. A significantly larger dataset would improve the models' robustness, enhance generalizability, and reduce the likelihood of overfitting.

The future trajectory of this research will also prioritize the adoption of state-of-the-art machine learning and deep learning algorithms to achieve more precise classifications of chest X-ray images. Incorporating advanced models such as Vision Transformers (ViT), hybrid architectures combining CNNs with RNNs or transformers, and explainable artificial intelligence (XAI) frameworks will further enhance the diagnostic capabilities of these systems. By integrating these contemporary methods, future work aims to improve not only the accuracy but also the interpretability and clinical usability of AI-driven diagnostic tools.

6. REFERENCES

Abubakar. (2020). Classification of Pneumonia and Tuberculosis from Chest X-rays. *https://arxiv.org/ftp/arxiv/papers/2103/2103.14562.pdf*, arXiv:2103.14562.

Ali, S. (2018). 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.

Almezhghwi. (2021). Convolutional neural networks for the classification of chest X-rays in the IoT era. *Multimedia Tools and Applications*, 1-15.

Andrew Ng. (2020). Why is Deep Learning taking off? deeplearning.ai [online]. Available: . *https://www.coursera.org/learn/neural-networks-deep-learning/lecture/praGm/why-is-deep-learning-taking-off.*

Benjamens. (2020). The state of artificial intelligence-based fda-approved medical devices and algorithms. *An Online Database. NPJ Digital Medicine*, 3(1):1–8.

Bernal. (2018). Deep convolutional neural networks for brain image analysis on magnetic resonance imaging. *A review," 11 Jun 2018.*

Beutel. (2020). Handbook of Medical Imaging: medical image processing and analysis. *Bellingham*, vol. 2.

CheXNet. (2020). Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. *CheXNet*, [Last Accessed: 14- Sep- 2020].

Chhikara. (2020). Deep convolutional neural network with transfer learning for detecting pneumonia on chest X-rays. In Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals. *Singapore: Springer, 2020, DOI: 10.1007/978-981-15-0339-9_13.*, vol. 1064.

Coursera. (2021). Computer Vision-deeplearning.ai . *[online]. Available: https://www.coursera.org/learn/ convolutional-neural-networks/lecture/Ob1nR/computer-vision.*, Last Accessed Nov, 2021.

D. A. Pitaloka. (2017). Enhancing CNN with Preprocessing Stage in Automatic Emotion Recognition. *Procedia Comput. Sci*, vol. 116, pp. 523–529.

Dai. (2017). SCAN: Structure Correcting Adversarial Network for Organ Segmentation in Chest X-rays. *Arxiv*.

Dai D. (2019a). Can AI Read Chest X-rays like Radiologists? *https://towardsdatascience.com/can-machine-learning-readchest-x-rays-like-radiologists-part-1-7182cf4b87ff*, Part 1.

Dai D. (2019b). Can AI Read Chest X-rays like Radiologists? *https://towardsdatascience.com/can-machine-learning-readchest-x-rays-like-radiologists-part-2-aa77dba219f0*, Part 2.

De Jong. (2010). Differences between tuberculosis cases infected with Mycobacterium africanum, West African type 2, relative to Euro-American Mycobacterium tuberculosis: an update. *FEMS immunology and medical microbiology, 58(1). https://doi.org/10.1111/j.1574-695X.2009.00628.x,* 102–105.

Francesco. (2019). A technical report on convolution arithmetic in the context of deep learning [online]. Available:. *https://github.com/vdumoulin/conv_arithmetic.*

Geert Litjens, T. K. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88.

Gibson and Loddenkemper. (2013). Respiratory health and disease in Europe: The new European Lung White Book. *Eur. Respir. J., vol. 42, no. 3,*, pp. 559–563, .

Gonzalez & Woods. (2018). *Digital Image Processing.* Pearson Education.

Guendel, S. (2019). Multi-task Learning for Chest X-ray Abnormality Classification on Noisy Labels. *IEEE Xplore* , pp. 1–10.

Hasan. (2021). DenseNet Convolutional Neural Networks Application for Predicting COVID-19 Using CT Image. *SN COMPUT. SCI. 2, 389. https://doi.org/10.1007/s42979-021-00782-7.*

Hooda. (2017). Deep-learning: A potential method for tuberculosis detection using chest radiography. *IEEE Int. Conf. Signal Image Process. Appl. (ICSIPA), Kuching, Malaysia*, 497-502.

Huang. (2017). Densely Connected Convolutional Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261-2269.

J. Saikia. (2016). Removal of Salt and Pepper Noise Using Selective Adaptive Median Filter. *IEEE Xplore*, vol. 1, no. 1.

Jeoung. (2019). An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare. *Journal of Healthcare Engineering*.

Kermany. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *https://www.sciencedirect.com/*, 1122-31.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436-444.

Lee Jacobson. (2014). Introduction to Artificial Neural Networks. *http: // www.theprojectspot.com/ tutorial-post/ introduction-to-artificial-neural-networks-part-1/7*., Part 1 [online]. Available: .

Lee, Y., & Nam, S. (2021). Performance Comparisons of AlexNet and GoogLeNet in Cell Growth Inhibition IC50 Prediction. . *International journal of molecular sciences, 22(14), 7721.* , https://doi.org/10.3390/ijms22147721.

Litjens & Van der Laak. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis. *https://doi.org/10.1016/j.media.2017.07.*, 42, 60-88.

Liu. (2017). Detecting tuberculosis in chest X-ray images using convolutional neural network. *IEEEE*.

Liu, X. (2019). 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.

Matthew and Dnuggets. (2017). Neural Network Foundations, Explained: Activation Function. *[online]. Available: https://www.kdnuggets.com/2017/09/neural-network-foundations-explained-activation-function.html.*

MayoClinic. (2022). *Rib Fractures.* Retrieved from https://www.mayoclinic.org.

MoH. (1980 - 1985). National Health Plan. *Mogadishu, Republic of Somalia*.

Namdar. (2021). A Modified AUC for Training Convolutional Neural Networks: Taking Confidence into Account. . *https://pypi.org/project/GenuineAI*.

National Heart, Lung, and Blood Institute. (2021). *Cardiomegaly.* . Retrieved from https://www.nhlbi.nih.gov.

P. T. K.koonsanit. (2019). 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*.

Quing. (2014). Medical image classification with convolutional neural network. *IEEE. 13th international conference on control automation robotics & vision (ICARCV).*, 844-8.

Rajpurkar. (2020). CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. *Arxiv*.

Russakovsky. (2015). Imagenet large scale visual recognition challenge. *Int J Comput Vision.*, 115(3):211–52.

S. Lall and S. Boyd. (2015). Stabilization of Networked Control Systems with Sparse Observer Controller Networks. *IEEE transactions on automatic control,* , vol. 60, no. 6.

Save the Children. (2017). Pneumonia, the forgotten killer disease in Somalia. *https://reliefweb.int/sites/reliefweb.int/files/resources/Pneumonia%20press%20release%20final.pdf*.

Schiaffonati and Verdicchio. (2014). Computing and Experiments: A Methodological View on the Debate on the Scientific Nature of Computing. *Philosophy and Technology*, 27(3): 359–76.

Shears, P. (2014). Tuberculosis control in Somali refugee camps. *Tubercle; 65:1 11-6*.

Simonyan, K., & Zisserman, A. . (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint,* arXiv:1409.1556.

Skadiņš. (2010). Improving SMT for Baltic languages with factored models. Frontiers in Artificial Intelligence and Applications, 219. *https://doi.org/10.3233/978-1-60750-641-6-125*, 125–132.

Society, A. C. (2021). *Lung Cancer Diagnosis.* . Retrieved from, https://www.cancer.org.

Stephen and Sain. (2019). An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare. *J. Healthc. Eng.,* , vol. 2019.

Szegedy. (2016). Rethinking the Inception Architecture for Computer Vision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 27-30 June*, 2818-2826.

Tahir. (2020). Coronavirus: Comparing COVID-19, SARS and MERS in the eyes of AI. *http://arxiv.org/abs/2005.11524*, 2005.11524.

Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. . *Nature Medicine, https://doi.org/10.1038/s41591-018-0300-7*, 25(1), 44-56. .

Vianna VP. (2018). Study and development of a computer-aided diagnosis system for classification of chest X-ray images using convolutional neural networks pre-trained for imagenet and data augmentation. . *http://www.arXiv.org*, arXiv preprint arXiv :1806.00839.

Vom Brocke. (2020). Introduction to Design Science Research. *Springer International Publishing. https://doi.org/10.1007/978-3-030-46781-4_1*.

W. Gaul. (2019). Studies in Classification , Data Analysis , and Knowledge Organization. *Managing Editors*.

Wang. (2017). Chestx-ray8: hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *In: Proceedings of the IEEE conference on computer vision an*.

Ward, Lukowicz, and Gellersen. (2011). Performance Metrics for Activity Recognition. *ACM Transactions on Intelligent Systems and Technology*, 2(1): 111–32.

WHO. (2001). *Standardization of interpretation of chest radiographs for the diagnosis of pneumonia and tuberculosis in children. No. WHO/V&B/01.35.*, Geneva: World Health Organization.

WHO. (2014). The common causes of morbidity and mortality. *https://www.who.int/hac/about/donorinfo/somalia.pdf*.

WHO. (2021a). Pneumonia. *https://www.who.int/news-room/fact-sheets/detail/pneumonia*.

WHO. (2021b). Tuberculosis. *https://www.who.int/news-room/fact-sheets/detail/tuberculosis*.

Xray4Al. (2020). Trained Chex Pert dtaset provided by Stanford University. *https://xray4all.com*.

Yang, S. X. (2018). 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.

Yann. (2015). Deep Learning. *Nature*, 521(7553):436 444,.

APPENDICES

Appendix A: Approval Letter from Hargeisa Group Hospital

HARGEISA GROUP HOSPITAL

Ref: M015553 Date: Jan 4th, 2024

Approval Letter from HGH

I am writing on behalf of the Radiology Department at Hargeisa Group Hospital (HGH) to officially approve the utilization of the chest X-Ray image dataset collected from our department. This dataset is intended for the research project titled "*A Deep Learning Approach for Automated Analysis and Classification for Six Cases of Chest X-rays*", conducted by Mr. Suleiman Mohamed Abdi.

Mr. Suleiman has diligently collaborated with Dr. Mustafe Kowdan, a Radiologist in our department, to collect and label the dataset into six categories. Dr. Kowdan has played a crucial role in providing clinical insights, advice, and expertise during various stages of the research. Their collaboration extends beyond labeling the data, as they have worked together on multiple aspects of the research, including testing and verifying the deep learning model developed by Mr. Suleiman.

We appreciate the commitment and professionalism demonstrated by Mr. Suleiman and Dr. Kowdan in advancing this research, and we believe that their work will contribute significantly to the field of automated analysis and classification of chest X-rays.

Thank you for your attention to this request. We look forward to the successful completion of the research project and its potential positive impact on the medical community.

Sincerely,

Dr. Idiris Ali Hassan

Medical Coordinator, Hargeisa Group Hospital

Web: https://hgh.govsomaliland.org Email: info.hgh@sldgov.org

Appendix B: Implementation Code for the Models

```
import os
import numpy as np
from PIL import Image
import tensorflow as tf
import matplotlib.pyplot as plt
from keras.models import Model
import cv2
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import roc_auc_score
from tensorflow.keras import layers, models
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.models import Sequential
from keras.applications.densenet import DenseNet121
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.inception_v3 import preprocess_input
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, accuracy_score, balanced_accuracy_score
from tensorflow.keras.layers import Dense, Conv2D, MaxPooling2D, Flatten, Dropout, BatchNormalization, Activation
```

```
# Dataset path
dataset_path = '/Final Dataset'
train_dir = '/Final Dataset/train'
val_dir = '/Final Dataset/val'
test_dir = '/Final Dataset/test'
```

```
def print_label_image_counts(dataset_paths):
    total_images = 0.0

    for dataset_path in dataset_paths:
        print("\n=======================================")
        print(f"Dataset path: {os.path.basename(dataset_path)}\n=======================================")

        for label in os.listdir(dataset_path):
            label_path = os.path.join(dataset_path, label)
            if os.path.isdir(label_path):
                num_images = len(os.listdir(label_path))
                total_images += num_images
                print(f"{label}={num_images}")

    print("\n=======================================")
    print(f'Total Images: {total_images}')
```

```
========================================
Dataset path: train
========================================
Pneumonia=816
Enlarged Heart=816
Mass=816
Tuberculosis=816
Fractured Ribs=816
Normal=4080

========================================
Dataset path: test
========================================
Pneumonia=102
Enlarged Heart=102
Mass=102
Tuberculosis=102
Fractured Ribs=102
Normal=510

========================================
Dataset path: val
========================================
Pneumonia=102
Enlarged Heart=102
Mass=102
Tuberculosis=102
Fractured Ribs=102
Normal=510

========================================
Total Images: 10200.0
```

```python
# Define parameters
img_size = (90, 90)
batch_size = 32
nb_train_samples = 8160
nb_validation_samples = 1020
```

```python
# Preprocess images Median Filter, Histogram Equalization
def preprocess_image_test(img):
    # Check if the image is grayscale
    if len(img.shape) == 2 or (len(img.shape) == 3 and img.shape[2] == 1):
        img = cv2.cvtColor(img, cv2.COLOR_GRAY2RGB)

    # Noise Removal using Median Filter
    img = cv2.medianBlur(img, 3)

    # Image Enhancement using Histogram Equalization
    if len(img.shape) == 3:
        img_gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

        # Ensure img_gray is 8-bit single-channel
        if img_gray.dtype != np.uint8:
            img_gray = (img_gray ).astype(np.uint8)

        equalized = cv2.equalizeHist(img_gray)
        img = cv2.cvtColor(equalized, cv2.COLOR_GRAY2RGB)

    # Normalization
    img = img.astype('float32') / 255.0


    return img
```

```python
# Preprocess images Median Filter, Histogram Equalization and Rotation
def preprocess_image_train(img):
    # Check if the image is grayscale
    if len(img.shape) == 2 or (len(img.shape) == 3 and img.shape[2] == 1):
        img = cv2.cvtColor(img, cv2.COLOR_GRAY2RGB)

    # Noise Removal using Median Filter
    img = cv2.medianBlur(img, 3)

    # Image Enhancement using Histogram Equalization
    if len(img.shape) == 3:
        img_gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

        # Ensure img_gray is 8-bit single-channel
        if img_gray.dtype != np.uint8:
            img_gray = (img_gray ).astype(np.uint8)

        equalized = cv2.equalizeHist(img_gray)
        img = cv2.cvtColor(equalized, cv2.COLOR_GRAY2RGB)

    # Normalization
    img = img.astype('float32') / 255.0

    # Random Rotation
    rotation_angle = np.random.choice([45, 180, 270])
    img = np.rot90(img, k=rotation_angle // 90)

    return img
```

```python
# Plot Loss and Accuracy
def plot_training_history(history):
    plt.figure(figsize=(12, 8))

    plt.subplot(2, 2, 1)
    plt.plot(history.history['loss'], label='Loss')
    plt.plot(history.history['val_loss'], label='Val_Loss')
    plt.legend()
    plt.title('Loss Evaluation')

    plt.subplot(2, 2, 2)
    plt.plot(history.history['accuracy'], label='Accuracy')
    plt.plot(history.history['val_accuracy'], label='Val_Accuracy')
    plt.legend()
    plt.title('Accuracy Evaluation')

    plt.show()
```

```python
# Evaluate Model Confusion Matrix
def evaluate_model(validation_generator, model):
    y_true = validation_generator.classes
    y_pred_probs = model.predict(validation_generator)
    y_pred = np.argmax(y_pred_probs, axis=1)

    # Print accuracy score and AUROC
    accuracy = accuracy_score(y_true, y_pred)
    roc_auc = roc_auc_score(y_true, y_pred_probs,multi_class='ovr')
    print("AUROC:", roc_auc)
    print(f"Accuracy Score: {accuracy}")
    class_labels = list(validation_generator.class_indices.keys())
    # Create a confusion matrix
    conf_mat = confusion_matrix(y_true, y_pred)

    # Display the confusion matrix
    ConfusionMatrixDisplay(conf_mat, display_labels=class_labels).plot(values_format="d", xticks_rotation='vertical')
```

```python
# Plot Images with Predicted and True Labels
def plot_predicted_batch(model, data_generator):
    class_names = list(data_generator.class_indices.keys())

    images, true_labels = data_generator.next()

    predictions = model.predict(images)

    num_images = images.shape[0]

    plt.figure(figsize=(30, 30))
    num_rows, num_cols = 5, 5


    num_subplots = num_rows * num_cols
    num_images_to_display = min(num_images, num_subplots)

    for i in range(1, num_images_to_display + 1):
        plt.subplot(num_rows, num_cols, i)
        plt.axis("off")

        true_label = np.argmax(true_labels[i - 1])
        predicted_label = np.argmax(predictions[i - 1])

        color = "darkgreen"
        if true_label != predicted_label:
            color = "red"

        plt.title(f"True: {class_names[true_label]}\nPredicted: {class_names[predicted_label]}", color=color)
        plt.imshow(images[i - 1])
```

Raw X-ray Images

```python
def plot_image_per_folder(main_directory):
    '''Plot image from each directory of labels'''
    # Get a list of subdirectories
    subdirectories = [d for d in os.listdir(main_directory) if os.path.isdir(os.path.join(main_directory, d))]

    # Set up the plot
    fig, axs = plt.subplots(2, 3, figsize=(10, 7))

    # Iterate over subdirectories and plot one image from each
    for i, subdirectory in enumerate(subdirectories):
        # Get the list of image files in the subdirectory
        image_files = [f for f in os.listdir(os.path.join(main_directory, subdirectory)) if f.lower().endswith(('.png', '.jpg

        if image_files:
            img_path = os.path.join(main_directory, subdirectory, image_files[1])
            img = plt.imread(img_path)
            axs[i // 3, i % 3].imshow(img, cmap='gray')
            axs[i // 3, i % 3].axis('off')
            axs[i // 3, i % 3].set_title(subdirectory)

    plt.show()
# Raw Chest X ray Images
plot_image_per_folder(train_dir)
```

```
# Create ImageDataGenerator
train_image_generator = ImageDataGenerator(
    preprocessing_function=preprocess_image_train,
)

valid_image_generator = ImageDataGenerator(
    preprocessing_function=preprocess_image_train,
)

test_image_generator = ImageDataGenerator(
        preprocessing_function=preprocess_image_test
)


# Create data generators for training, validation and testing
train = train_image_generator.flow_from_directory(train_dir,
                                          target_size=img_size,
                                          batch_size=batch_size,
                                          class_mode='categorical',
                                          shuffle=True)


validation = valid_image_generator.flow_from_directory(val_dir,
                                               target_size=img_size,
                                               batch_size=batch_size,
                                               class_mode='categorical',
                                               shuffle=False)


test = test_image_generator.flow_from_directory(test_dir,
                                               target_size=img_size,
                                               batch_size=batch_size,
                                               class_mode='categorical',
                                               shuffle=False)
```

```
Found 8160 images belonging to 6 classes.
Found 1020 images belonging to 6 classes.
Found 1020 images belonging to 6 classes.
```

Plot Batch of Processed Images

```
def plot_processed_image(generator):
    class_names = list(generator.class_indices.keys())
    images, true_labels = generator.next()
    num_images = images.shape[0]

    plt.figure(figsize=(30, 30))
    num_rows, num_cols = 5, 5

    num_subplots = num_rows * num_cols
    num_images_to_display = min(num_images, num_subplots)

    for i in range(1, num_images_to_display + 1):
        plt.subplot(num_rows, num_cols, i)
        plt.axis("off")
        plt.title(f"True: {class_names[np.argmax(true_labels[i - 1])]}")
        plt.imshow(images[i - 1])

    plt.tight_layout()
    plt.show()
plot_processed_image(train)
```

```
def plot_image(img, title):
    plt.imshow(img)
    plt.title(title)
    plt.axis('off')
    plt.show()

# Meidan Filter
def median_filter(path):
    img = cv2.imread(path)
#       Resize Image (90, 90)
#     img = cv2.resize(img,img_size)
    if len(img.shape) == 2 or (len(img.shape) == 3 and img.shape[2] == 1):
        img = cv2.cvtColor(img, cv2.COLOR_GRAY2RGB)

    # Noise Removal using Median Filter
    img = cv2.medianBlur(img, 3)

    plot_image(img, 'Median Filter')
```

```
# Histogram Equalization
def histogram_equalization(path):
    img = cv2.imread(path)
#       Resize Image (90, 90)
#     img = cv2.resize(img,img_size)

    if len(img.shape) == 2 or (len(img.shape) == 3 and img.shape[2] == 1):
        img = cv2.cvtColor(img, cv2.COLOR_GRAY2RGB)

    if len(img.shape) == 3:
        img_gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

        # Ensure img_gray is 8-bit single-channel
        if img_gray.dtype != np.uint8:
            img_gray = (img_gray).astype(np.uint8)

        equalized = cv2.equalizeHist(img_gray)
        img = cv2.cvtColor(equalized, cv2.COLOR_GRAY2RGB)

    plot_image(img, 'Histogram Equalization')
```

```
# Rotation 45%, 180%, 270%
def rotation(path):
    img = cv2.imread(path)
#       Resize Image (90, 90)
#     img = cv2.resize(img,img_size)

    if len(img.shape) == 2 or (len(img.shape) == 3 and img.shape[2] == 1):
        img = cv2.cvtColor(img, cv2.COLOR_GRAY2RGB)
    # Choose Randomly
    rotation_angle = np.random.choice([45, 180, 270])
    img = np.rot90(img, k=rotation_angle // 90)

    plot_image(img, 'Rotation')
```

Plot Origianl Image, Median Filter, Rotation and Histogram Equalization

```
random_image = 'Final Dataset/train/Pneumonia/person101_virus_187.jpeg'
img = cv2.imread(random_image)
plot_image(img, 'Original')
median_filter(random_image)
rotation(random_image)
histogram_equalization(random_image)
```

Inception (GoogleNet)

```
# Create a pre-trained Inception (Google Net) model
google_net = InceptionV3(weights='imagenet', include_top=False, input_shape=(90, 90, 3))

for layer in google_net.layers:
  layer.trainable = False

x = layers.Flatten()(google_net.output)
x = layers.Dense(1024, activation='relu')(x)
x = layers.Dropout(0.2)(x)
x = layers.Dense(len(train.class_indices), activation='softmax')(x)

model_google_net = Model(google_net.inputs, x)

# Compile the model
model_google_net.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])

#Early Stopping
early_stopping = EarlyStopping(
    monitor='accuracy',  # Metric to monitor (e.g., validation loss)
    patience=3,           # Number of epochs with no improvement after which training will be stopped
    restore_best_weights=True  # Restore model weights from the epoch with the best value of the monitored metric
)
```

```
#Train Model
history = model_google_net.fit(
    train,
    steps_per_epoch = nb_train_samples // batch_size,
    validation_steps = nb_validation_samples // batch_size,
    validation_data = validation,
    epochs=20,
    callbacks=[early_stopping]
)
```

```
# Save the Model
model_google_net.save('GoogleNet_Xray_Images.h5')
```

```
/opt/conda/lib/python3.10/site-packages/keras/src/engine/training.py:3000: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```

```
# Plot Loss and Accuracy
plot_training_history(history)
```

```
# Predicted and True Lable Images
plot_predicted_batch(model_google_net, test)
```

Densenet

```
# Create a pre-trained Densenet model
dense_net = DenseNet121(weights='imagenet', include_top=False, input_shape=(90, 90, 3))

for layer in dense_net.layers:
  layer.trainable = False

x = layers.Flatten()(google_net.output)
x = layers.Dense(1024, activation='relu')(x)
x = layers.Dropout(0.2)(x)
x = layers.Dense(len(train.class_indices), activation='softmax')(x)

model_dense_net = Model(google_net.inputs, x)

# Compile the model
model_dense_net.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])

#Early Stopping
early_stopping = EarlyStopping(
    monitor='accuracy',
    patience=3,
    restore_best_weights=True
)
```

```
# Train the Model
history = model_dense_net.fit(
    train,
    steps_per_epoch = nb_train_samples // batch_size,
    validation_steps = nb_validation_samples // batch_size,
    validation_data = validation,
    epochs=20,
    callbacks=[early_stopping]
)
```

```
# Save the Model
model_dense_net.save('DenseNet_Xray_Images.h5')
```

```
/opt/conda/lib/python3.10/site-packages/keras/src/engine/training.py:3000: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```

```
# Plot Loss and Accuracy
plot_training_history(history)
```

```
# Predicted and True Lable Images
plot_predicted_batch(model_dense_net, test)
```

```
1/1 [==============================] - 0s 25ms/step
```