ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCE

SCHOOL OF INFORMATION SCIENCE

DETECTION OF PNEUMONIA USING DEEP LEARNING APPROACH FROM X-RAY IMAGES

A Thesis Submitted to Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science and Information Systems

By: MELAT ALEMAYEHU
ID:NO/GSR/5550/15
Advisor: MILLION MESHESHA (PhD)

SEPTEMBER 27, 2024
ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCE

SCHOOL OF INFORMATION SCIENCE

DETECTION OF PNEUMONIA USING DEEP LEARNING APPROACH FROM X-RAY IMAGES

By: MELAT ALEMAYEHU

Name and Signature of Members of the Examining Board

Million Meshesha (PhD)	____________________	____________________
Advisor	Signature	Date
__________________________	____________________	____________________
Examiner	Signature	Date
__________________________	____________________	____________________
Examiner	Signature	Date

DEDICATION

This thesis has not been previously awarded any degree and is not currently under consideration for any other degree at any university.

I affirm that the thesis reflects my own research efforts, except where explicitly noted. While I received guidance and support from my research advisor, the study was conducted independently. All external sources used are properly cited, and a comprehensive list of references is provided.

Signature: ____

Melat Alemayehu

This thesis has been submitted for examination with my approval as university advisor.

Advisor Signature: ____

Million Meshesha (PhD)

ACKNOWLEDGEMENT

First and foremost, I am deeply grateful to God for unwavering support throughout every journey I undertake.

To my esteemed advisor, Dr. Million Meshesha, your kind support, guidance, insightful ideas, and corrections have been invaluable to me. I am profoundly thankful for your mentorship.

I extend my heartfelt appreciation to the Radiology Department and ICT Office of Tikur Anbessa Hospital for generously offering their time and assistance whenever needed.

My sincere gratitude goes to Hawassa University for granting me the scholarship opportunity, enabling me to advance and enhance my skills.

I wish to express my gratitude to Addis Ababa University, particularly the College of Natural and Computational Sciences, School of Information Science, for their facilitation and endorsement of my research project. Their support has been instrumental in its success.

I am grateful to my beloved family, especially my mother, ***Weyneshit Eshte***, for her unwavering support in caring for my little child and providing assistance whenever needed. To my husband, ***Robel Getachew*** and my little angel, ***Yoadan Robel***, your patience and understanding during times when my attention was consumed by this work mean the world to me.

Lastly, I extend my heartfelt thanks to all my friends who have stood by me, offering their solid support and encouragement, contributing to the success of this endeavor.

ABSTRACT

Pneumonia classification is developing an automated system capable of distinguishing between three distinct categories—normal, non-pneumonia lung diseases, and pneumonia—using chest X-ray images. Pneumonia is a severe lung infection that can lead to significant morbidity and mortality if not promptly diagnosed and treated. However, the challenge arises from the fact that other lung conditions, like chronic obstructive pulmonary disease (COPD) or tuberculosis, can present with symptoms and X-ray findings similar to pneumonia, making accurate differentiation crucial. The objective of this study is to design and evaluate a machine learning model that can accurately classify these categories, thereby aiding in the early and accurate diagnosis of pneumonia versus other conditions. This classification task is vital for improving clinical outcomes, reducing diagnostic errors, and ensuring that patients receive appropriate treatment as quickly as possible.

The primary objective of this research is to develop an effective model for pneumonia classification using deep learning approach. By categorizing chest X-ray images into three classes—non-pneumonia (other lung diseases), normal (healthy lungs with no signs of disease), and pneumonia—we aim to enhance diagnostic accuracy and improve radiologist performance. To achieve this, we utilize a data set collected from Tikur Anbessa Specialized Hospital, comprising 3000 chest X-ray images, with 1000 images per class to ensure balanced representation before augmentation; and also, we used a ratio of 80:10:10 (80% training, 10% validation, and 10% testing) splitting ratio.

We employed an experimental research approach, selecting three state-of-the-art pretrained models—InceptionV3, ResNet50, and VGG16—for transfer learning, alongside constructing a custom CNN-L5 model. Through a series of experiments, we investigated various image prepossessing techniques, including re-sizing, normalization, and image enhancement, to optimize model performance. We explored different combinations of epochs, batch size, and learning rates for all models, while also experimenting with fine-tuning the pretrained models. The most effective model for pneumonia classification was then selected based on these trials. The experimental results showed that the CNN-L5 model, trained on enhanced image data sets with 30 epochs, a batch size of 64, and the inclusion of dropout, achieved superior performance with a classification accuracy of 96.8%. This highlights the importance of using appropriate preprocessing techniques and optimizing model architecture to achieve high performance in pneumonia classification.

However, this study was limited to data from a single hospital, which may not represent the diversity of patient populations, imaging techniques, and conditions found in other healthcare settings. Consequently, the model's performance may not generalize well across different environments. To address this limitation, future work should incorporate data from multiple hospitals to improve the model's robustness and broader applicability.

Keywords: *Pneumonia classification, Deep Learning, Convolutional Neural Network, Digital Image processing*

TABLE OF CONTENTS

]

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI	Artificial Intelligence
Bi-LSTM	Bidirectional LSTM
CADx	Computer-Aided Diagnosis
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
CNN-L5	Convolutional Neural Network with 5 layers
CT	Computed Tomography
CXR	Chest X-Ray
DL	Deep Learning
GAN	Generative Adversarial Networks
GP	General Practitioner
HE	Histogram equalization
HRCT	High-Resolution Computed Tomography
ML	Machine Learning
MRI	Magnetic Resonance Imaging
PET	Positron Emission Tomography
RAD	Radiography
ResNet	Residual Network
RNN	Recurrent Neural Network
VGG:	Visual Geometry Group
WHO	World Health Organization

CHAPTER-ONE

INTRODUCTION

1.1. Background of the study

Pneumonia is an inflammatory lung illness mostly affecting the alveoli, which are small air sacs found in the lung. Asthma, food aspiration, chemical irritants, and other non-infectious factors can also cause it, although infections with viruses, bacteria, or other microorganisms are usually the cause [1].

Pneumonia is among the most common causes of death for both young people and the elderly worldwide. A variety of symptoms, such as cough, fever, difficulty breathing, and chest pain, can indicate pneumonia. Lung infection brought on by pneumonia can be fatal if not identified in a timely manner [2].

Medical imaging methods like computed tomography (CT) scans or chest x-rays are frequently used in the diagnosis of pneumonia in order to image the lungs and identify areas of inflammation or consolidation.

Digital image preprocessing refers to the manipulation and enhancement of digital images before they are analyzed or used for a specific application. This process involves a series of techniques and algorithms aimed at improving the quality, clarity, and interpretability of the images [3]

Digital image preprocessing plays a crucial role in various fields such as computer vision, medical imaging, remote sensing, and industrial inspection, where high-quality images are essential for accurate analysis and decision-making.

Medical imaging preprocessing and feature extraction uses a cutting-edge technology to analyze and interpret medical pictures for research and diagnostic applications for deep learning-based classification. A variety of modalities, including computed tomography (CT), magnetic resonance imaging (MRI), ultrasound, and positron emission tomography (PET), are included in the field of medical imaging. These modalities generate large volumes of complex image data, which can be effectively processed and analyzed using deep learning techniques for accurate diagnosis and classification of medical conditions [4].

A form of medical imaging procedure called a chest X-ray (CXR) involves briefly exposing the chest to generate an image of the chest and its major organs. It is an essential diagnostic tool for pneumonia anywhere in the world. It does, however, require specific knowledge and experience to interpret the X-ray pictures correctly. As a result, making an accurate and thorough diagnosis of pneumonia using X-ray images might be difficult. The reason for this is that, same opacities in images can also be found in a number of other medical diseases, such as lung cancer and excess fluid. Therefore, being able to accurately read images is highly desirable [5].

CXR analysis can help identify high-risk individuals during an epidemic so they can get priority care. As a result, several deep learning models have been used by researchers to help classify CXR [6].

The classification of chest X-ray images is a significant application of artificial intelligence in medical imaging. Using deep learning algorithms, chest X-ray pictures must be analyzed and interpreted in order to diagnose various lung diseases. The primary goal of categorizing chest X-ray images is to aid radiologists in making more accurate diagnoses of conditions including lung cancer, pneumonia, tuberculosis, and other pulmonary ailments [7].

Deep learning is a subset of machine learning, which is a branch of artificial intelligence (AI) that aims to imitate the way humans gain certain types of knowledge. Deep learning algorithms are designed to learn from data and automatically improve their performance. One of the most popular deep learning architectures for image classification is Convolutional Neural Networks (CNNs) [8].

A Convolutional Neural Network (CNN) is a specialized deep learning model that excels in processing and analyzing visual data, such as images, by learning hierarchical features through convolutional and pooling layers. Rather than training a CNN from scratch, which requires substantial data and computational power, transfer learning allows the use of pretrained models—such as InceptionV3 or ResNet—that have been trained on large datasets like ImageNet. These pretrained models serve as a starting point, with their learned features being fine-tuned for specific tasks, making them highly efficient for new applications with limited data, while improving accuracy and reducing training time [9].

Deep learning is crucial for the accurate classification of pneumonia and plays a significant role in improving radiologists' productivity and providing high-quality healthcare services.

1.2. Statement of the problem

The most popular way to diagnose pulmonary problems is by chest radiography, or X-rays [10]. But the differentiation and interpretation of pneumonia signals is complicated by the visual similarities among the pathogenic aspects of Chest X-Ray (CXR) images [5]. To interpret the results, a radiology specialist with an advanced training is required. However, even skilled radiologists occasionally misread the radio-graphs, took more time to give a service, professionals give service for few patients per a day when they use manually, and also sometimes there is scarcity of a professional radiologists [11]**.**

Rejected films are those radiographs that are thrown away because they are not useful for diagnosis. A rejected image is defined as one that must be retaken due to poor image quality, which prevents the image from providing diagnostic information to clinical inquiries. This issue is common in our nation [14].

Computer-aided detection and diagnosis are often required due to the lack of skilled radiologists [15], discrepancy reports between radiologists and physicians, low quality x-ray images, and the rapid increase in workload and complexity that increases x-ray images misinterpretation [2]. Therefore, the **purpose** of this study is to detect pneumonia using better image processing and deep learning techniques to solve the problem.

In previous studies [16], usually the machine is trained using regular x-ray images. However, since most of the x-ray images are poor in quality, there is a need to apply image enhancement for improving the quality of x-ray image and enhance the performance of pneumonia classification. This study seeks to address the existing gaps and provide answers to the research questions.

To this end, the study aims to address the following research questions in order to address the aforementioned problem.

1. Which CNN architectures are most effective for pneumonia detection in terms of accuracy and performance?
2. How do image enhancement techniques impact image quality and model performance in pneumonia classification?

1.3. Objective of the study

1.3.1. General Objective

The general objective of this research is to develop an effective model for the detection of pneumonia from x-ray image using deep learning approach.

1.3.2. Specific Objectives

- ✓ To review the literature to have a detailed understanding of concepts, principles, and domain areas of image processing, machine learning, and deep learning
- ✓ To acquire and prepare datasets of x-ray images.
- ✓ To select suitable image processing and deep learning algorithms for experimentation
- ✓ To train and construct the proposed model to detect the pneumonia disease
- ✓ To test and evaluate the performance of the proposed model

1.4. Significance of the study

This research has a great contribution towards facilitating the work of radiology professionals to increase the accuracy of the interpretation of a result, to identify a pneumonia disease from other similar disease, enhance the efficiency and effectiveness of radiologist. This study is also useful for patients by reducing their waiting time for a result, early known of a disease to treat it timely and to get accurate result, this saves more adults and children from death. In addition to this, the study is important for healthcare to facilitate work process and to give good service for their patients. Furthermore, A. The result of this research is also used as input for further research investigation in the area of image-based classification and it plays an important role in everyday life.

1.5. Scope and limitation of the study

1.5.1. Scope of the study

This research focuses on the analysis and classification of chest x-ray images for the presence or absence of pneumonia using deep learning techniques. The goal of the study is to build a model that helps radiologists identify pneumonia in patients with chest x-rays. The scope includes the development and evaluation of model that can identify normal, pneumonic and non-pneumonic chest x-ray images. To train and test the performance of the classification algorithms, a dataset of chest x-ray pictures collected from Addis Ababa's Tikur Anbesa Specialized Hospital, encompassing all normal, pneumonia-affected cases and non-pneumonic (other chest disease) is gathered. The research also entails investigating several methods of image enhancement and deep learning models, more specifically CNNs in order to efficiently extract features from the x-ray images and perform precise classification.

1.5.2. Limitation of the study

The research encountered several significant limitations that may affect the validity and comprehensiveness of the findings. One significant drawback is that data collection is limited to a single hospital, making it impossible to see variation of datasets and working conditions throughout several hospitals.

Collecting data from different hospitals would introduce variations in factors such as image quality, patient demographics, disease severity, and differences in imaging protocols. These variations would contribute to a more comprehensive dataset that reflects a broader range of real-world conditions, enhancing the model’s ability to generalize across diverse populations.

The other constraint is that gathering and organizing data takes longer time than we expected due to the length of time needed to process and convincing hospital staff members to obtain clearance letters for access the Med-Web system.

1.5.3. Ethical issues and data privacy methods

Ethical issues in data privacy involve moral concerns related to the handling of personal information in the digital age, where protecting individuals' rights is crucial. To maintain patient confidentiality, especially in radiology, it's essential to anonymize data using various de-identification techniques such as data masking, pseudonymization, aggregation, generalization, and suppression. This study focuses on the suppression technique, which involves completely removing specific data items from medical images (e.g., X-rays) using the Micro-Dicom tool. By eliminating irrelevant data, the study ensures that only essential information, such as patient IDs, is retained for analysis. The remaining data is then categorized into three classes—pneumonia suspected, non-pneumonia, and normal—based on image results, demonstrating a careful balance between data utility and privacy.

CHAPTER-TWO

LITERATURE REVIEW

2.1. Overview

This chapter primarily focuses on the background information and review of the literature of the domain of this thesis. It includes a detailed explanation about pneumonia infected and healthy, x-ray-based detection of pneumonia using image processing and deep learning algorithms. Finally, the chapter is concluded with a summary of related works and the main gaps which should be solved in this thesis.

2.2. Healthy and Pneumonia Infected lung

A lung infection is exactly what it sounds like: an infection of one or both of your lungs. There are a couple of common types of lung infections such as: Pneumonia, Bronchitis, Tuberculosis (TB), Pulmonary Abscess, Influenza (Flu), Respiratory Syncytial Virus (RSV) Infection. Fungal Infections and COVID-19. Since the focus of this study is Pneumonia, our discussion is also concentrates on this lung infection.

Pneumonia is among the most typical lung infections. A person’s lungs' tiny air sacs are impacted by pneumonia. Although viruses and fungi can also cause it, infectious bacteria are the main culprits. Inhaling pneumonia-causing germs or a virus from a sick person can cause pneumonia in yourself. This could happen if the pneumonia sufferer coughs or sneezes [24].

As pointed out in [25], Pneumonia and healthy lungs have the following differences

- ✓ Pneumonia is a chest infection that causes inflammation to the air sacs in the lungs, while healthy lungs do not have inflammation.
- ✓ Pneumonia can be caused by bacteria, viruses, or fungi, while healthy lungs do not have any infection.
- ✓ In pneumonia, the lungs become filled with fluid and inflamed, leading to breathing difficulties, while healthy lungs do not have fluid in them.

- ✓ Chest X-rays with pneumonia often show areas of abnormal white or hazy lung, while normal chest X-rays show clear lungs.

Recognizing the differences between healthy and pneumonia-infected lungs is crucial, and imaging plays a vital role in highlighting these distinctions. Chest X-rays, for example, can reveal key diagnostic indicators such as inflammation and fluid buildup associated with pneumonia. A clear understanding of what defines an image is essential for accurately interpreting these medical features.

2.2.1. What is an image?

An image is a two-dimensional function denoted by f (x, y) that contains significant image-processing events relevant to diagnosis in medicine [28].

The intensity or gray level of the image at each given pair of coordinates (x, y) is represented by the amplitude of F, where x and y are spatial (plane) coordinates. An image is considered digital when its amplitude, x, and y are all discrete, finite values. A two-dimensional array with rows and columns is another technique to define an image. A picture with 500 x 400 (width x height) dimensions, for instance, has 200000 total pixels. Each of the finite elements that make up a digital image has a unique position and set of values. These elements are referred to as pixels, picture elements, or image elements. The most common term for the components of a digital image is "pixel." The pixel carry value indicates how many intensities are present in the image. Value 0 represents the color black, while Value 1 represents the color white [28].

Digital signals that analyze and manipulate images are subjected to digital image processing. The images digital clarity is enhanced, and its intensity distribution is flawless. Digital image processing is a quick and less expensive method of storing and retrieving images. It makes use of effective picture compression techniques that minimize the quantity of data needed and yield high-quality photos. It makes use of an image segmentation technique to identify discontinuities brought on by broken connection paths [29].

The process of transferring an image to a digital format and carrying out specific operations on it to improve the image quality or extract more information is known as image processing. All images are typically treated as 2D signals by the image processing system when specific preset signal processing techniques are used. This kind of signal processing takes an image as input, such as a video frame or an image, and outputs either the picture itself or some attributes or features related to the images. It is one of the fastest-growing technologies today, with its use in various business sectors.

2.2.2. Types of images

Four fundamental categories of pictures exist: binary, grayscale, red-green-blue (RGB), and multispectral. The following are the descriptions of each of these images.

- ✓ **Binary image**: the most basic kind of image is called binary, and it only requires two values, usually black and white, or 0 &1. Because each pixel in a binary image is represented by a single binary number, binary images are also known as 1-bit images. These kinds of photos are commonly utilized in applications such as optical character recognition (OCR), where the only information needed is a general shape or outline. From grayscale images, binary images are commonly created by a threshold operation. Pixels over the threshold value are converted to white ('1), while those below it is converted to black ('0').

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Figure 2.1 Binary image

- ✓ **Grayscale image**: Monochrome (one color) pictures are also known as grayscale pictures. They are devoid of color information and solely include grayscale data. Every pixel determines the various gray levels that are available. Typically, they have 8 bits of data per pixel, or 256 (0–255) distinct brightness (gray) levels.

Figure 2.2 Grayscale image

- ✓ **Color images:** Color images are made up of three bands of monochrome image data, in which each band of image data corresponds to a different color. In each spectral band, there is gray-level information which is the actual information stored in the digital image. Color images are also known as RGB images because color images are represented as red, green, and blue. Color images would have 24 bits/pixels by using the 8-bit monochrome standard as a model and 8-bits for each of the three-color bands (red, green, and blue).

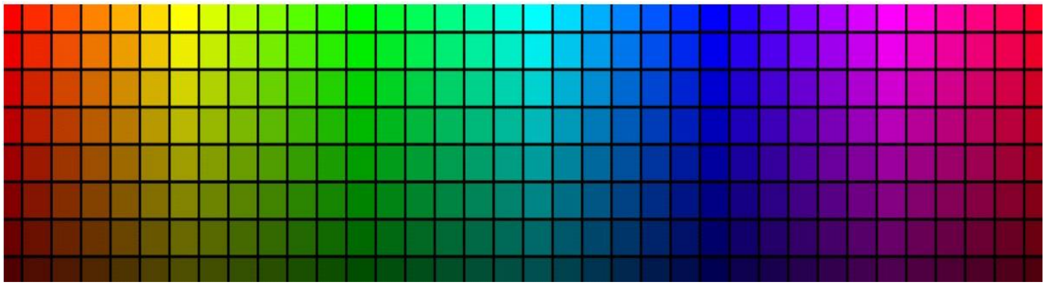


Figure 2.3 Color images

- ✓ **Multispectral images:** Multispectral images usually contain information outside the normal human perceptual range consisting of Infrared, ultraviolet, x-ray, acoustic, or radar data. They are not really images in the usual experience (not representing scenes of the physical world, but as alternative they provide information such as depth). However, by mapping the different spectral bands to RGB elements the data is represented in visual form.

Each type of image, whether binary, grayscale, RGB, or multispectral, provides unique data that can be processed and analyzed. Leveraging these image types, computer vision combines pattern recognition and image processing to extract meaningful information. This approach enables deeper understanding and interpretation of visual data for tasks like medical diagnosis.

2.3. Computer vision

Pattern recognition and image processing are combined to create computer vision. Understanding of images is the process's outcome in computer vision. While image processing focuses on applying computational image modifications, such as contrast and sharpening, its main goal is the creation of models, data extractions, and information from images [26].

Artificial intelligence's field of computer vision gives machines the ability to perceive, interpret, and comprehend visual stimuli. It entails imparting human-like vision skills to robots so they can learn to "see" and understand the content of digital photos or videos. This is achieved by applying a variety of techniques and algorithms to extract and analyze pertinent data from visual inputs.

Computer vision has many uses, and as technology advances, so do its applications computer vision algorithms are employed in self-driving cars to analyze their environment, identify objects, traffic signs, and other vehicles, and enable safe autonomous driving. It enables automated checkout systems in retail by tracking and recognizing the merchandise. With amazing accuracy, it analyzes medical photos to help with disease diagnosis in the healthcare industry. Social media networks use computer vision to apply filters in real time and recognize faces. The se are but a handful of the nearly infinite applications that computer vision can be used for [27].

2.4. Digital image processing

The editing and analysis of digital images are at the center of image processing, a crucial topic in computer science and digital technology. It comprises an extensive array of methods and formulas intended to improve, get data from, or modify digital images in order to accomplish particular objectives. From enhancing image quality and removing flaws like noise and distortion to deriving significant information from photos, such recognizing objects or patterns, these objectives can take many different forms. Applications in many different industries, such as medical imaging, remote sensing, entertainment, and more, depend heavily on image processing [18].

Digital image processing is the use of a digital computer to process digital images through an algorithm. It enables the augmentation of image aspects of importance while attenuating details irrelevant to a particular application, allowing one to extract relevant scene information from the enhanced image. Types of digital image processing

According to Faisal [30], the intermediate step between image processing and computer vision is image understanding. The continuum from image processing at one end to computer vision at the other lacks distinct borders. Three categories of computerized processes—low-, mid, and high-level processes—can be thought of as helpful paradigms within this field.

- ✓ Low-level processes: Primitive operations like image preprocessing to minimize noise, contrast enhancement, and sharpening are examples of low-level processes. The feature that both its inputs and outputs are images defines a low-level process.
- ✓ Mid-level processing: Image segmentation (splitting an image into areas or objects), object description (reducing an image to a format that can be processed by a computer), and object classification (identifying individual objects) are examples of mid-level image processing tasks. A mid-level process is defined by the fact that, although it typically receives images as inputs, its outputs are properties (such as edges, contours, and the identify of individual objects) that are taken from those images.
- ✓ Higher-level processing: Higher-level processing includes, at the extreme of the continuum, carrying out the cognitive processes often connected to vision as well as "making sense" of a group of recognized objects, as in image analysis.

2.5. Medical imaging modalities

For the purpose of clinical analysis and medical intervention, medical imaging modalities are methods and processes that produce visual representations of the inside of the body. These modalities are vital to the diagnosis of illnesses, the tracking of therapeutic outcomes, and the direction of numerous medical procedures. There are several types of medical imaging modalities [31].

- ✓ x-ray: One of the modalities in medicine that is most frequently utilized is x-ray imaging. It generates images of bones, tissues, and organs using ionizing radiation. When it comes to identifying fractures, tumors, infections, and other anomalies, x-rays are especially helpful.

- ✓ Computed Tomography (CT) Scan: A CT scan creates finely detailed cross-sectional images of the body by combining computer technology and X-rays. CT scans are useful for the diagnosis of vascular disorders, malignancies, and internal traumas.

- ✓ Magnetic Resonance Imaging (MRI): MRI creates finely detailed images of the body's soft tissues, organs, and structures by using radio waves in conjunction with a strong magnetic field. When imaging the brain, spinal cord, joints, and muscles, MRI is very useful.

- ✓ Ultrasound: Ultrasound imaging utilizes high-frequency sound waves to generate real-time images of internal organs and tissues. It is frequently employed to monitor fetal growth in pregnancy, as well as assess abdominal organs and blood vessels.

- ✓ Nuclear imaging: Nuclear imaging in medicine involves the use of radioactive substances (radiopharmaceuticals) administered to patients, along with specialized cameras used to detect the radiation emitted by these substances. This method is valuable for assessing organ functionality, identifying tumors, and diagnosing specific illnesses.

- ✓ Positron Emission Tomography Scan: PET scans require the insertion of a radioactive material into the body to observe metabolic activities at the cellular level. These scans are commonly utilized in oncology for identifying cancerous growths and evaluating how well treatments are working.

- ✓ Mammography: Mammography is a specialized form of X-ray imaging utilized for breast cancer screening and diagnosis. Advanced methods like digital mammography and 3D mammography (tomosynthesis) provide enhanced detection capabilities.

2.6. Steps in Digital Image Processing

As stated in [29], **digital image processing** is a field that involves the manipulation of digital images using computer algorithms. Figure 2.4 below presents the fundamental steps followed in digital image processing and analysis.

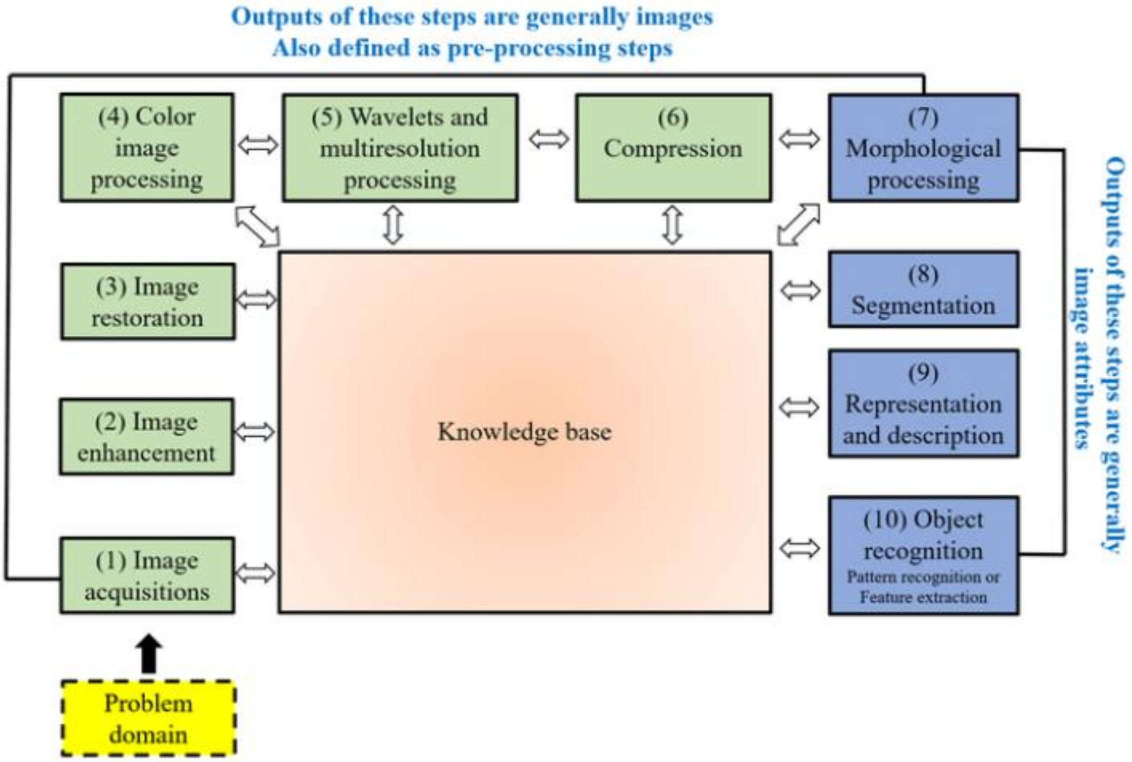


Figure 2.4 Fundamental steps in digital image [24]

As presented in the above figure 2.4, there are a sequence of steps followed in digital image processing and each of them are presented as follows.

2.6.1. Image Acquisition:

This is the first step in digital image processing, where an image is captured using devices such as cameras or scanners. The quality of the acquired image plays a crucial role in the subsequent processing steps.

For quality images, use 300 DPI for standard scanning and 600 DPI or higher for detailed work. In photography, 8-12 MP suffices for everyday use, 16-24 MP for larger prints, and 30+ MP for professional quality. DPI is crucial for scanning details, while MP determines overall image detail in photography. A camera captures real-world scenes with MP resolution, while a scanner captures flat objects like documents with DPI resolution for fine detail [32].

2.6.2. Image Enhancement and restoration:

Image enhancement techniques are used to improve the quality of an image by adjusting its contrast, brightness, and sharpness. This step aims to make the image more visually appealing and easier to analyze. Image restoration techniques are employed to remove noise and other imperfections from an image. This step helps in recovering the original information that might have been lost during the acquisition process.

Scholars propose methods for image enhancement and restoration, such as histogram equalization for contrast, Fourier and wavelet transforms for noise reduction, and spatial filtering for smoothing or sharpening. Deconvolution reverses blurring, morphological operations enhance shapes, and adaptive filtering adjusts to image details. Deep learning with CNNs provides advanced solutions like super-resolution and denoising. These methods are often combined for the best results [33].

2.6.3. Color Image Processing:

In this step, color images are processed to manipulate their color components, such as hue, saturation, and intensity. Color image processing is essential for various applications like medical imaging and remote sensing. Color components like hue, saturation, and brightness (or value/lightness) are processed in color models like HSV or HSL. Hue represents the type of color and can be shifted to change tones, saturation controls the intensity of colors, making them more vivid or muted, and brightness adjusts the lightness or darkness of the image. These components are often modified separately for color correction or enhancement and then converted back to the RGB model for display, allowing precise control over an image's color and overall appearance [34]

2.6.4. Compression and Decompression:

Compression techniques are used to reduce the size of digital images for efficient storage and transmission, while decompression reverses this process to reconstruct the original image from the compressed data. Scholars suggest various image compression methods to reduce file size while maintaining quality, which are classified into lossy and lossless compression. Lossy compression reduces file size by removing some of the original image data that are imperceptible to the human eye, whereas lossless compression reduces file size by eliminating redundant or unnecessary metadata. Lossless compression techniques like Huffman Coding and LZW (Lempel-Ziv-Welch) preserve all the original image data, enabling exact reconstruction of the image without any loss of quality, making them ideal for applications where data integrity is critical, such as medical imaging, technical drawings, or archival storage.

2.6.5. Morphological Processing:

Morphological processing involves operations like dilation, erosion, opening, and closing on binary images. These operations help in extracting useful information from images based on their shapes and structures. Dilation, erosion, opening, and closing are fundamental morphological operations used in image processing to modify the structure of objects within binary or grayscale images Dilation expands objects and fills small holes by setting pixels to the maximum value under a structuring element. Erosion shrinks objects and removes noise by setting pixels to the minimum value. Opening removes small objects and noise while preserving larger structures by applying erosion followed by dilation. Closing fills small gaps and holes within objects while preserving their overall shape by applying dilation followed by erosion. These operations modify the structure of objects in binary or grayscale images for various preprocessing tasks [35].

2.6.6. Segmentation:

Segmentation divides an image into meaningful regions or objects for further analysis. It plays a vital role in tasks like object recognition, image understanding, and medical imaging. Image segmentation methods include thresholding (based on pixel intensity), edge detection (finding boundaries), region growing (expanding from seed pixels), clustering (grouping similar pixels), watershed transformation (separating regions in a topographic view), graph cuts (partitioning an image as a graph problem), and deep learning (using CNNs for advanced segmentation). Each method is chosen based on the image's complexity and segmentation needs [36].

2.6.7. Feature Extraction:

Feature extraction involves identifying and extracting relevant features from an image, such as edges, textures, or shapes. These features are then used for tasks like pattern recognition and object detection. Scholars suggest various techniques for feature extraction, including edge detection methods like Sobel and Canny, texture analysis techniques such as GLCM and LBP, and shape descriptors like HOG and SIFT. Additionally, deep learning models like CNNs automatically extract complex features for tasks such as object detection and classification [37].

2.6.8. Image Recognition and Analysis:

Image recognition uses machine learning algorithms to classify images into predefined categories or recognize specific objects within an image. Deep learning (DL) enhances image recognition by using Convolutional Neural Networks (CNNs) to automatically learn and extract complex features from images. This improves classification and object detection, which is crucial for applications like facial recognition, disease classification, and autonomous driving, where accurate and efficient image interpretation is essential [37].

The final step in digital image processing involves interpreting the processed images and analyzing them to extract meaningful information using machine learning. This information can be used for decision-making in various fields like healthcare, agriculture, and security.

2.7. Machine learning and Deep learning

2.7.1. Machine learning (ML)

Machine Learning (ML) enables machines to mimic human behavior by learning from data and making predictions. It is a subset of artificial intelligence (AI) focused on using algorithms and models to analyze patterns in data. However, Deep Learning (DL), a subset of ML, is more advanced and uses neural networks with multiple layers to handle large, complex data sets. Unlike traditional ML, DL can automatically learn features from data, eliminating the need for manual feature extraction [38]. In this study, the focus is on DL algorithms, which are particularly effective in tasks like image classification, natural language processing, and other complex applications due to their ability to learn from vast amounts of data without much human intervention

According to Ray [39]**,** there are three important techniques used in Machine Learning which vary according to the type of ML problem and the input data. These techniques are supervised learning, unsupervised learning, and reinforcement learning.

- ✓ Supervised Learning

One of the most widely used machine learning techniques is supervised learning, which uses performance metrics and historical data sets to learn from experience. The foundation of supervised learning is the process of determining the optimal approach to map input data to target output by studying past data sets. Supervised Learning deals with labelled data and produce labelled data as well. The main methods used in Supervised Learning are Classification and Regression and also the most popular algorithms in each Supervised Learning techniques-Classification or Regression, are Linear Classifier, Logistic Regression, Support Vector Machines (SVM), Naive Bayesian (NB) Networks, K-Means, Decision Tree, Neural Network, and Multi-layer Perceptron.

- ✓ Unsupervised Learning

Unsupervised Learning is ML's second technique. When there are no previous data labels to train on, it is utilized to construct a prediction model. When handling unlabeled data, unsupervised learning relies on finding patterns in the data. After analyzing the unlabeled input, unsupervised learning algorithms create new groups with relevant data. Determining the hidden structure in unlabeled data or input is the algorithm's job in unsupervised learning. It accomplishes this by providing the learning algorithms with large number of input data, characteristics, and features so the algorithms can make necessary observation to find out the best grouping or relationships between data. The main methods used in unsupervised learning are clustering and association. The most popular algorithms for clustering are K-means, and Hierarchical clustering similarly algorithms for association are Apriopri Algorithm and FT Growth Algorithm.

- ✓ Reinforcement Learning:

The third technique of ML is Reinforcement Learning. RL uses past data with no predefined label to improve the future manipulation of dynamic system. It is a learning technique where an agent interacts with an environment by producing actions and discover errors or rewards. The agent is put in an unknown environment to behave and learn by receiving signals. These signals can be rewards or punishment depending on the agent action. it allows the agent to adjust action and learn better for future actions. The most popular algorithms and techniques for RL are Monte-Carlo, Temporal Difference Methods, SARSA (State-Action-Reward-State-Action), and Direct Policy Search, Q-learning.

2.7.2. Deep learning

Machine learning (ML) is the study of using machines to mimic human abilities and behavior. It is regarded as a technology that uses data to provide meaning and answers inquiries. One of the most well-liked and often applied subsets of machine learning is deep learning. An artificial neural network with multiple layers is necessary for deep learning (ANN). Human biological neural networks serve as the foundation for the idea. These networks contain nodes and multi-layers that communicate with each other to understand and analyze the data received. DL consists of several layers, while each layer may consist of many neurons which is called Processing Unit. The main role of neurons is to multiply the input value with the given weight and calculate the sum of the results. This is how Deep Learning models acquire higher accuracy [40].

Machine learning and deep learning have many things in common. Both ideas rely on data classification and model training. With DL, the machine can pick up on and recognize examples. A more complicated subset of machine learning is deep learning. Big data and powerful computer power are required for DL to handle the data. It deals with Image Processing, Natural Language Processing (NLP), or Voice Recognition. Deep Learning uses different approaches to train data depending on the type of data. The most popular approaches are Convolution Neural Network (CNN), Recurrent Neural Network (RNN), Generative Adversarial Networks (GAN**),** Transformers and Autoencoders. RNNs are suitable for sequential data, LSTMs improve on RNNs by handling long-term dependencies better through their gating mechanisms.[41].

✓ Convolutional Neural Networks (CNNs)

Convolutional neural network, also known as CNN or ConvNet, is commonly used for image recognition, object detection, and image classification tasks. CNNs are made up of fully connected layers for classification, pooling layers to lower dimensionality, and convolutional layers that extract features from input images. Among the well-known CNN architectures are ResNet, Alex Net, VGG, and GoogLeNet (Inception). These architectures have achieved state-of-the-art performance on benchmark datasets like ImageNet [42].

- ✓ **Recurrent Neural Networks (RNNs)**

RNNs are designed to handle sequential data by maintaining a hidden state that captures information from previous time steps. A type of RNN that can learn long-term dependencies and solve the vanishing gradient problem is called **Long Short-Term Memory (LSTM)**. Natural language processing activities including machine translation, sentiment analysis, speech recognition, and more are applications of RNNs and LSTMs [43].

- ✓ **Bi-LSTM (Bidirectional LSTM)**

Bi-LSTM (Bidirectional LSTM) is an extension of LSTM networks that processes sequences in both forward and backward directions. This allows it to capture context from both past and future data points, improving performance in tasks like language modeling and sentiment analysis where understanding both sides of a sequence is crucial [44].

- ✓ **Generative Adversarial Networks (GANs)**

In order to produce realistic synthetic data, GANs are made up of two neural networks—a discriminator and a generator—that have been trained adversarial. Super-resolution, image production, style transfer, and other generative challenges have all been tackled with GANs. Notable GAN architectures that have produced remarkable success in producing high-quality photographs include StyleGAN, Cycle GAN, and DCGAN (Deep Convolutional GAN) [45].

- ✓ **Transformer Architecture**

Transformers are becoming more and more common in jobs involving natural language processing because of their effectiveness in capturing long-range dependencies. The transformer architecture is based on self-attention mechanisms that allow the model to focus on different parts of the input sequence. Advanced models such as BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer), T5 (Text-to-Text Transfer Transformer), and so on have been built on top of the Transformer model [46].

- ✓ **Autoencoders:**

Neural networks called autoencoders are made specifically for unsupervised learning tasks including anomaly detection, feature learning, and dimensionality reduction. They are made up of a decoder network that reconstructs the input from the latent space and an encoder network that compresses the input data into a latent representation. Variants of autoencoders include denoising autoencoders, variational autoencoders (VAEs), and sparse autoencoders [47].

In this study an attempt is made to use Convolutional neural network architecture for image-based pneumonia classification.

2.7.3. CNN architecture

Convolutional neural network is a class of deep neural networks where the connectivity pattern between neurons is inspired by the structure of an animal visual cortex [39]38]. Though they have been effectively used for a variety of computer vision applications, particularly for the analysis of visual images, convolutional neural networks are quite similar to ordinary neural networks. The role of ConvNet is to compress the images into a form that is easier to process while preserving features that are important for obtaining a good prediction. This is critical for designing an architecture that is capable of learning features while also being scalable to large datasets.

There are a number of faults in regular neural networks, especially in the scope of image processing or computer vision applications. This is because, when some space information is dispersed, images, which frequently include a lot of information, may be lost. For example, if we take a grayscale image with a size of 1280 by 720, it will hold 921,600 pixels. If a fully connected network accepts the pixel size of this image as input, the neuron weight required is 921,600. In another example, 2,073,600 weights in 1920 × 1080 images are required. If the image is RGB (colored), the color scale is multiplied by three (3). Therefore, when the image size is increasing; the number of free parameters in the network increases. So, whenever the model is getting larger the performance of the network decreases and it leads to overfitting. Overfitting is a problem of machine learning algorithms that happens when the size of the network is larger and there is no data to fit the model. The problem decreases the generalization ability of the machine learning model. CNN solves this problem by using layers of neurons that are arranged in three dimensions

(3D): width, height, and depth. Each layer of CNN takes a 3D volume of input data (in this case, an image) and uses a differentiable function to generate another 3D volume of data as an output [48] [49] [50].

CNNs gather image data, train the model, and automatically classify characteristics into healthful categories, just like classical models. Each pixel's size is represented by these pixel values. The image may be RGB or grayscale. A grayscale image is a matrix of pixel values having a single plane, the pixel values are ranging from 0 to 255, and where 0 represents black and 255 represents white. The RGB image is the same but it has three planes; i.e. Red, blue, and green. Each of these metrics would have a value also ranging from 0 to 255 where each of these numbers represents the intensity of the pixels [48].

CNNs are composed of different layers, but those layers are not fully connected. Each layer has a different set of filters that are applied through the image which helps a CNN automatically learn different values (features). These filters introduce translation invariance and parameter sharing. For example, it may learn to identify images from raw pixels, using those edges to identify shapes, and using those shapes it may detect an object in the highest layer to make predictions[49]. Thus, the concept of building high-level features from the low level is exactly why CNN is powerful in pattern recognition. Since the late 1980s and in 1990 CNN has given interesting results in handwritten digit classification and face recognition. The following figure illustrates CNN architecture.

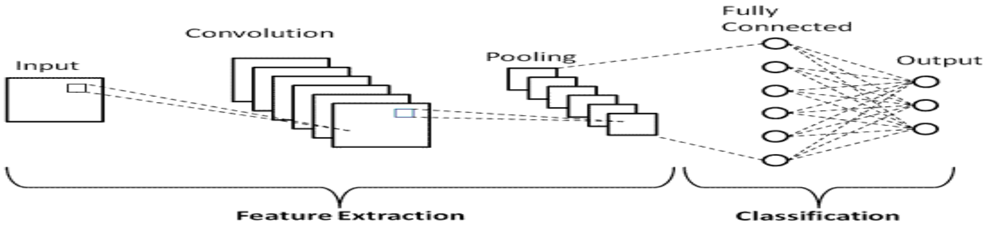


Figure 2.5 CNN basic architecture [48]

Convolutional neural network (CNN) is made to handle input that has a grid-like structure, like an image, which is made up of pixels. With a shared-weight architecture, the CNN architecture is specifically made to capitalize on the 2D structure of an input image and local spatial correlations [70].

According to Saxena, Aarush [70] there are several types of CNN architectures that have been developed over the years to address different challenges in image recognition, classification, and other computer vision tasks. The following is a list of some of the pre-trained models used in this investigation.

- ✓ **Resnet-50 (Residual Network-50)**

A popular convolutional neural network (CNN) design for computer vision applications, especially picture categorization, is called ResNet-50. It was introduced in a paper titled "Deep Residual Learning for Image Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in 2016 In order to overcome the difficulty of training extremely deep neural networks, ResNet-50 presents the idea of residual learning. As depth increases, traditional networks have trouble with vanishing gradients, which lowers performance. By using skip connections, residual learning helps to reduce this problem by enabling the network to learn residual functions in relation to the layer inputs. ResNet-50 gets its name from its 50 layers. It is constructed from fundamental building blocks known as residual blocks.

Multiple convolutional layers, batch normalization, ReLU activations, and shortcut connections are all included in each residual block. By allowing the network to skip over some levels, these shortcut connections help the gradients flow more easily during training. Big datasets like ImageNet are usually used to pretrained ResNet-50. Through pretraining, the network is able to extract useful elements from images and fine-tune them for certain purposes. Pretrained models are used in many computer vision applications as a feature extractor or backbone. ResNet-50 is frequently used for problems including object detection, picture segmentation, and image classification. It is a well-liked option for computer vision research as well as real-world applications due to its performance and adaptability [71]. Deep Residual Learning for Image Recognition. -Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)].

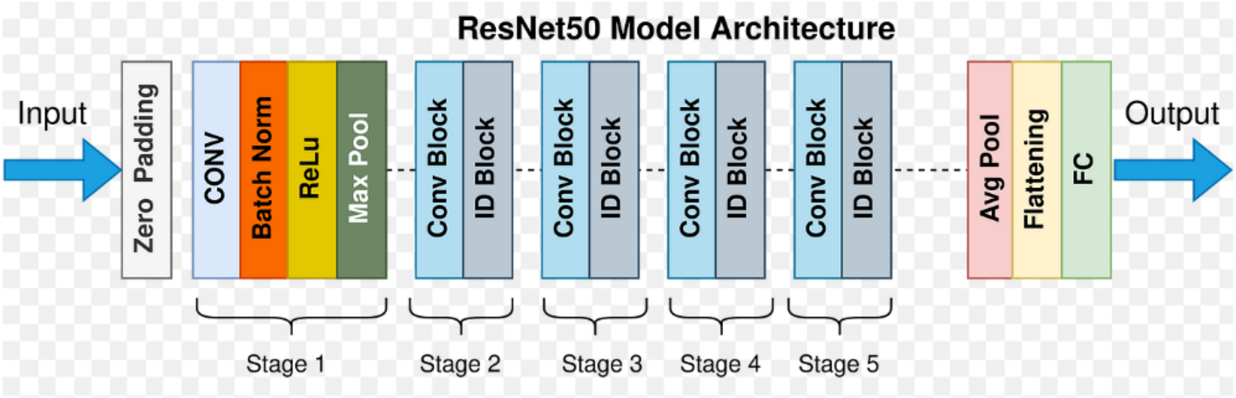


Figure 3.6 Resnet-50 Architecture

- ✓ **InceptionV3**

A deep convolution neural network InceptionV3 is intended for effective and high-performance picture classification. The network can collect features at numerous scales nice one to the Inception Modules, which perform multiple convolution operations (with varying filter sizes) in parallel and concatenate their outputs. By splitting up bigger convolutions into smaller ones—for example, by substituting two 3x3 convolutions for a 5x5 convolution—the network lowers computing costs. InceptionV3 has auxiliary classifiers that branch off the main network and direct learning in order to enhance training and prevent disappearing gradients. By down sampling feature maps, these modules preserve computational efficiency while boosting depth and decreasing spatial dimensions. Using batch normalization extensively speeds up training and enhances output. For tasks including object detection, image segmentation, and image classification, InceptionV3 strikes a balance between depth and efficiency [70].

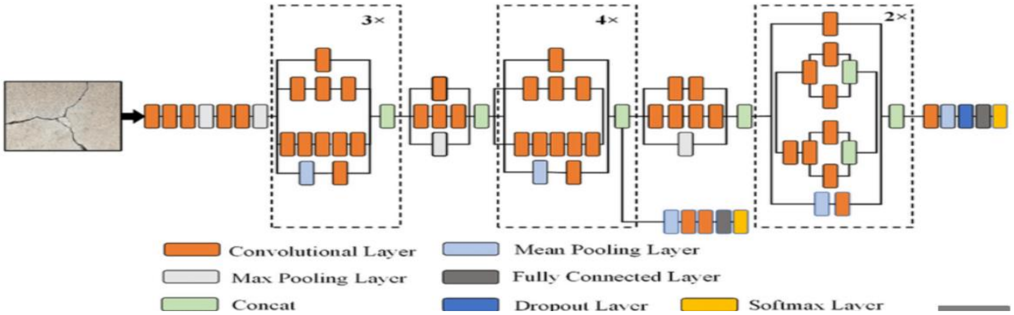


Figure 3.7 InceptionV3 Architecture

✓ VGG16

The University of Oxford's Visual Geometry Group created the convolutional neural network (CNN) known as VGG16, or "Visual Geometry Group 16-layer network," for the purpose of classifying images. The main characteristics of VGG16, which was introduced in 2014 by Karen Simonyan and Andrew Zisserman [73].

The architecture of VGG-16 contains total 16 layers, comprising 3 fully linked layers and 13 convolutional layers. Convolutional Layers: To preserve spatial resolution, tiny 3x3 filters with a stride of 1 are used. Pooling Layers: To minimize spatial dimensions, there are two by two max-pooling layers. Three fully connected layers make up the final layer, which is categorized into 1000 categories. Uniform Design: Makes use of 2x2 max-pooling and 3x3 filters consistently across the network.

In its Deep Architecture that enables learning of complex features, achieving high performance in image classification with Contains about 138 million parameters, making it computationally intensive but powerful. VGG16 is a popular choice for various computer vision tasks, including feature extraction and image classification.

Figure 3.8 VGG-16 Architecture

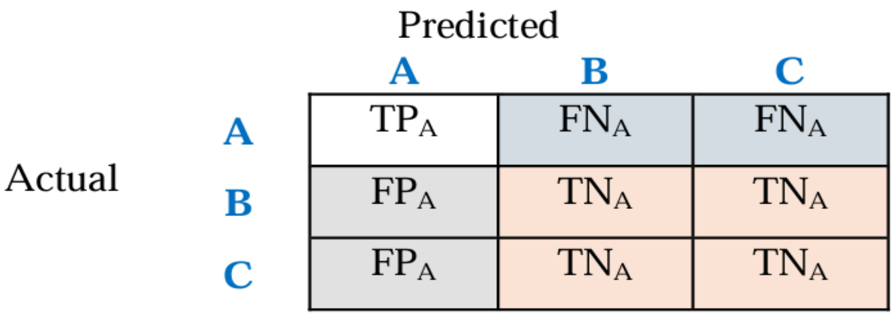
2.8. Evaluation Metrics

The dataset is split into training, validation, and testing datasets for the purposes of the classification accuracy metrics technique. We can feed the validation data to the model during training in order to obtain performance measures. The model returns the accuracy and loss of training data, and the accuracy and loss of validation data, which are training accuracy, validation accuracy, training loss, and validation loss. Then, we plotted the loss and accuracy learning Curve with respect to epochs by using these metrics to diagnose problems with learning, such as an underfit or overfit model. Finally, the testing data (images that have not been used in either the training or validation sets) is given to the trained model to test the performance of the model, then the model returns accuracy and loss of the testing data which is never seen during the training.

Several evaluation metrics, such as classification accuracy metrics, confusion matrix, precision, recall, F1-score, and AUC—all of which are suggested approaches for classification problems in machine and deep learning algorithms—are employed in this thesis to assess the effectiveness of the suggested approach.

2.8.1. Confusion Matrix

A table known as a confusion matrix is frequently used to illustrate how well a classification model performs when applied to a set of test data that contains four possible combinations of predicted and actual values. Compute assessment metrics such as error rate, recall, precision, specificity, accuracy, and AUCROC curves. The True Positive, True Negative, False Positive, and False Negative of the confusion matrix combinations for each class, as indicated in Table 3.1 below, must be known in order to calculate these evaluation metrics.



		Predicted		
		A	B	C
Actual	A	TP_A	FN_A	FN_A
	B	FP_A	TN_A	TN_A
	C	FP_A	TN_A	TN_A

Table 3.1 sample of confusion matrices

- ✓ **Accuracy**: It provides the overall accuracy of the model, and is determined by the total number of correctly classified CXR samples divided by the total number of CXR samples in the testing dataset. We can calculate it using the formula below:

$$\boldsymbol{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.1}$$

- ✓ **Error Rate**: It is calculated as the number of all incorrect predictions divided by the total amount of the dataset. It is often referred to as a classification error. We can calculate it using the formula below:

$$\boldsymbol{Error\_Rate} = \frac{FP+FN}{TP+TN+FP+FN} \tag{3.2}$$

- ✓ **Precision** (PREC). It is a measure of correctness that is achieved in the right prediction and is calculated as the number of correct positive predictions divided by the total number of positive predictions. It is often referred to as positive predictive value (PPV). We can calculate it using the formula below:

$$\boldsymbol{Precision} = \frac{TP}{TP+FP} \tag{3.3}$$

- ✓ **Recall (SN)**. It is a measure of **actual observations** which are predicted **correctly and** is calculated as the number of correct positive predictions divided by the total number of positives. It is often referred to as a true negative rate (TNR). We can calculate it using the formula below:

$$\boldsymbol{Recall} = \frac{TP}{TP+FN} \tag{3.4}$$

- ✓ **Specificity**: It indicates the percentage of all negative samples that were correctly classified as negative by the classifier. It is often referred to as the True Negative Rate (TNR). We can calculate it using the formula below:

$$\boldsymbol{Specificity} = \frac{TN}{TN+FP} \tag{3.5}$$

- ✓ **F1-score**: It combines recall and precision into a single measurement. It represents the harmonic mean of recall and precision. We can calculate it using the formula below:

$$\textbf{\textit{F1-score}} = 2 * \frac{precision * Recall}{precision + Recall} \tag{3.6}$$

These, all are evaluation metrics that used in this image-based classification study to measure model performance.

2.9. Related works

Various researchers studied the classification of pneumonia from x-ray image employing Deep learning with different dataset, preprocessing techniques and CNN architecture to modify the model and improve the performance of model detection.

Naralasetti et al. [51] aimed to develop efficient model to detect pneumonia and suggest CNN architecture. The success of the CNN model depends on different parameters tuning especially model regularization methods such as batch normalization, dynamic dropout, learning rate decay, kernel regularization weight decay avoids model over-fitting and enhances performance. In this study the authors proposed CNN model designed with 5 convolution blocks followed by 4 fully connected layers with specified hyper parameters and it allows the model to outperform several existing models by achieving accuracy of 97.73% and 91.17% respectively for binary and multi-class classification tasks of pneumonia disease.

Katoch et al., [6], focus on comparison of existing method of pneumonia detection and classification with their limitation. The authors review Various deep learning-based architectures with different framework such as InceptionV-4, CNN, ResNet50, VGG16, VGG19 as comparison techniques.in this study VGG-16 Net, Alex Net and Inception V3 Net frameworks are outperformed.

Amamra, and Amarouche, [11], reviewed and analyzed the usability, goodness factors, and computational complexity of the algorithms that were used for pneumonia identification in addition to this they Review and analyze the quality, usability, size, and class balance extent of the available CXR images datasets which limited on review of methods developed to extract valuable features from X-rays. The authors summarized the concepts as traditional ML methods can be employed when high computational resources are scarce but cannot be employed for industrial-scale intelligent detection of pneumonia. Most of the CXR datasets are highly imbalanced, and only some researchers have followed the proposed balancing techniques. In most cases, the authors have utilized under sampling and obtained their results with few images. To solve this the authors suggest generating high-quality synthesized images using GANs along with data augmentation techniques.

Irfan et al., [52], explored the use of deep transfer learning to classify pneumonia in chest X-ray images. In the time of authors experiments, three models, namely ResNet-50, Inception V3 and DensetNet121, were trained separately with 20 epochs using deep transfer learning and from scratch. The result of authors experiment summarized as Transfer learning achieved better results than training from scratch. And also support the effectiveness of transfer learning, providing a low-cost development option for systems based on deep learning for faster and more efficient clinical deployment.

Nafiiyah & Setyati, [18], Pointed to prove whether the chest X-ray image that was performed by contrast improvement had a significant effect in diagnosing Pneumonia. the study uses CLAHE enhance chest Xray image was trained with 8 CNN architectural models. The results of this experiments conducted in the eight CNN architectural models were 79.65%, 79.01%, 80.29%, 76.92%, 82.53%, 80.45%, 79.81%, 78.04%, respectively. The highest accuracy result when testing is 82.53% with the CNN 35 Layers architectural model, with a description of the input image is grayscale with a size of 224x224. According to result of experiment the authors proposes a process to improve chest X-ray images using CLAHE, the goal is to produce clear chest Xray's that can be used in diagnosing Pneumonia

Knok et al., [53] stated the purpose of this work was to create a model of an intelligent system that takes a lung x-ray as an input parameter and, after processing the image, outputs the likelihood of pneumonia. A transfer learning mechanism built on pre-defined convolution neural network topologies was used to accomplish the aforementioned capabilities. The authors used total of 5863 lung x ray images from Kaggle datasets and VGG16 architecture, consisting of a total of 16 layers which greatly contributes to the accuracy of the model using previously learned principles for the classification of image data. The result of the study conducted with this architecture were 94% of accuracy which is the indicator of high-performance model.

Khan et al., [54], aimed to summarized the literature on the topic of identifying pneumonia using chest x-rays, analyzed current algorithms in terms of their usability, goodness factor, and computing complexity, and provided a summary of the field. The study has shown that a range of datasets can be used to accomplish the stated purpose. The research focused on the comparison of datasets comprised of chest x-ray and comparison of pneumonia detection techniques.

Accordingly, the result of was summarized as the most famous and most significant dataset is MIMIC CXR and the Chest Xrays-14 dataset. Additionally, in a comparison of pneumonia detection techniques, Sousa et al. were able to achieve 96% accuracy using the NB algorithm and dimensionality reduction techniques.

Sharma et al., [55], the aim of this research is proposing different deep convolution neural network (CNN) architectures to extract features from images of chest X-ray and classify the images to detect if a person has pneumonia. Accordingly, research introduces two practice CNN designs for identifying pneumonia in chest X-ray images which is CNN with dropout layer and without dropout layer. Data augmentation techniques are employed to prevent overfitting. The experimental results demonstrate that the CNN with dropout, trained on augmented data, performs better than other mode

Antin et al., [56] stated: the research uses both supervised learning, logistic regression and neural network techniques for binary classification of x-ray images into pneumonia infected and health images. Based on the study deep learning uses DenseNet, and *CheXNet models* for classification and these have better results than logistic regression. And also because of imbalance data both evaluation metrics are based on AUC.

According to Saraiva et al., [57] Neural Network-Based Models for Learning to Classify X-ray Images for Pneumonia Detection. The study showed how two classification methods for pneumonia diagnosis might be compared. Cross-validation was used to validate the models, allowing the generalization potential to be confirmed. CNN and MLP were the artificial neural networks that were employed. The suggested classification models demonstrated their effectiveness, with CNN achieving an accuracy of 94.40% and MLP achieving 92%

Hereunder summary of related works done towards Pneumonia detection and Classification are presented in table 2.1 below.

Table 2.2 Summary of related works

Authors (year)	Problem	Approach	Result	Gap
Naralasetti et al., (2021) [51]	Deep Learning Models for Pneumonia Identification and Classification Based on X-Ray Images	CNN model with 5 convolution blocks followed by 4 fully connected layers, Model regularization	The proposed model performs 97.73% and 97.17% accuracy respectively for binary and multi-class classification	-They use online adults' datasets; also, contextual information is missing
Katoch et al., (2021) [6]	Pneumonia Disease Detection Using Deep Learning Methods from Chest X-Ray Images	Compare DL pre-trained architectures	VGG-16 Net, Alex Net and Inception V3 Net techniques outperformed others pretrained models	The experiment is not well designed; as a result, equal comparison criteria were not taken into consideration
Irfan et al., (2020) [52]	Classifying Pneumonia among Chest X-Rays Using Transfer Learning	ResNet-50, Inception V3 and DensetNet121 using Transfer Learning (TL) and without TL (from scratch)	Transfer learning achieved better results than training from scratch	the dataset may suffer from class imbalance and limited diversity, affecting the model's real-world applicability.
Nafiiyah & Setyati, (2021) [18]	Lung X-Ray Image Enhancement to Identify Pneumonia with CNN	Applied Noise reduction, contrast modification, and normalization to enhance the X-ray images' visual quality	-The augmented and improved images resulted in greater pneumonia detection accuracy, sensitivity, and specificity.	It faces limitation in image quality and model generalization. The enhancement techniques may not perform equally wel
Knok et al., (2019) [53]	Intelligent Pneumonia Identification from Chest X-Rays	used Convolutional Neural Networks (CNNs) for feature extraction and classification. Preprocessing techniques were also used to improve image quality.	The CNN-based model outperformed conventional techniques in detecting pneumonia from chest X-rays with good accuracy and resilience.	Need more study to solve the difficulty in differentiating pneumonia from other lung illnesses that present similarly on radiographs.

Khan et al., (2021) [54]	Intelligent Pneumonia Identification from Chest X-Rays: A Systematic Literature Review	utilized machine learning and deep learning models such as ResNet, Inception, and DenseNet	**-High-performing models** like ResNet, Inception, and DenseNet were emphasized.	Imbalanced dataset used for experiment
H. Sharma et al., (2020) [55]	Feature extraction and classification of chest X-ray images using CNN to detect pneumonia	Applied Preprocessing methods and used transfer learning to extract important features from the photos using a pre-trained CNN model.	-High accuracy was attained, showing a discernible increase above conventional machine learning techniques.	The need for Interpretability to comprehend CNN decision-making procedures, which is essential for clinical application.
Antin et al., (2017) [56]	Detecting Pneumonia in Chest X-Rays with Supervised Learning	Preprocessing the collected data, and also Fine-tuning pretrained model (such as VGG16 or ResNet)	-The study showed that pretrained CNNs can be fine-tuned to get good results in medical image classification tasks.	limited attention to interpretability, and absence of a clear performance comparison with radiologists' work
Saraiva et al., (2019) [57]	Models of learning to classify X-ray images for the detection of pneumonia using neural networks,	Used CNNs and pretrained models like ResNet and DenseNet.	Models for transfer learning performed better than model created from scratch	absence of clinical validation from real-world settings, the scant attention paid to model interpretability, and the lack of direct comparisons with radiologists.

The literature review highlights various approaches to pneumonia detection using deep learning and image processing techniques, demonstrating a trend toward using convolutional neural networks (CNNs) and transfer learning with pre-trained models such as ResNet, Inception, and DenseNet. Several studies, including those by Naralasetti et al. (2021) and Irfan et al. (2020), show that deep learning models can achieve high accuracy, with transfer learning outperforming models trained from scratch. However, common gaps persist, such as the use of imbalanced datasets, lack of generalization, and limited interpretability, as noted by researchers like Department & Lamongan (2021) and Sharma et al. (2020). While image enhancement techniques (e.g., CLAHE) have improved performance, studies often lack consistent experimental design and clinical validation. These gaps in the literature, including the need for better model comparison and real-world applicability, motivated this study to focus on developing a robust, high-performing CNN model with effective preprocessing techniques and ensuring generalizability by addressing dataset limitation.

CHAPTER THREE

DESIGN OF PROPOSED SYSTEM

3.1. Overview

This chapter describes the methods and techniques used to develop the proposed system. It begins with an outline of the system architecture, followed by a discussion of the algorithms and methods employed in the implementation. Additionally, the software and hardware tools utilized are highlighted, and the evaluation techniques used to assess model performance are presented.

3.2. Proposed architecture.

The main aim of this study is to construct an effective model for the classification of pneumonia from x-ray image using deep learning approach. Figure 3.1 below shows the proposed architecture of the study.

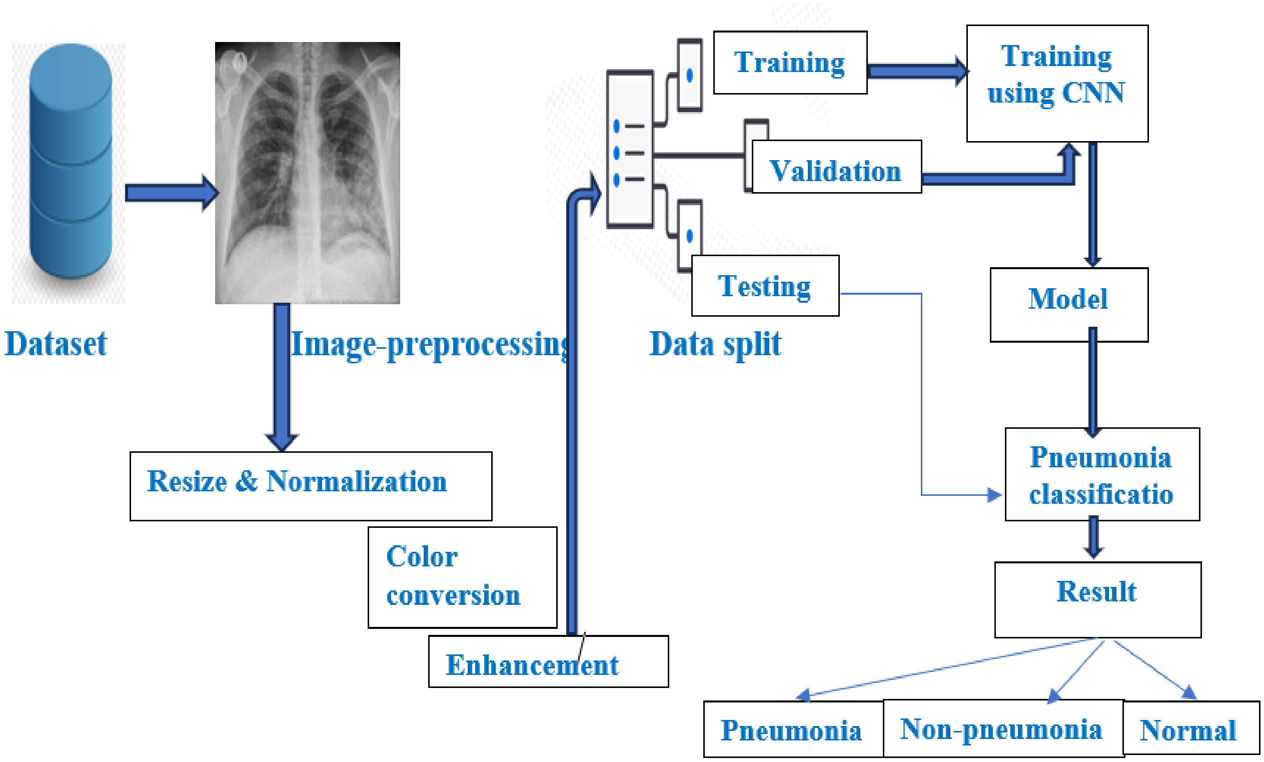


Figure 3.1 The proposed architecture

The proposed architecture for image-based pneumonia classification begins with a dataset of chest x-ray images. These images undergo preprocessing, including enhancement, normalization, and color conversion, to improve the quality and consistency of the input data. The processed images are then split into training, validation, and testing sets. The training data is fed into a Convolutional Neural Network (CNN) specifically designed for this task. The CNN is trained to differentiate between three categories: non-pneumonia, normal/healthy, and pneumonia. After training, the model is validated and tested to ensure its accuracy and effectiveness in classifying the images into the correct categories. This pipeline aims to automate the detection of pneumonia from X-ray images with high accuracy

3.3. Methodology of the study

The study follows experimental research to generate new knowledge and to investigate and solve the mentioned problem.

Experimental research is crucial for systematically investigating and evaluating the performance of different deep learning architectures, and image processing methods in classifying x-ray images. To conduct experimental research, the following main steps are undertaken: image acquisition and preparation, implementation and evaluation [17].

Various researchers study the classification of pneumonia from x ray image on Deep learning with different dataset, preprocess techniques and CNN architecture to modify the model and improve the performance of model detection. To gain a deeper comprehension of the topic understudy, a number of relevant works of literature on the topic of classifying pneumonia from x-ray images using deep learning and image processing techniques as well as other techniques have been reviewed from a variety of sources like articles, books, journals, and the Internet.

3.3.1. Image acquisition and preparation

In an experimental approach, gathering a dataset of images representative of the classes to be identified is the first stage. Both the size of the dataset and its annotation with the appropriate class labels should be sufficient to yield trustworthy findings. The study uses image resizing and cropping for image preprocessing. It downloads a collection of x-ray images in di-com format from Tikur Anbsa specialized hospital health care systems.

The images must be preprocessed after the dataset is gathered in order to get rid of any noise or artifacts and correct data imbalance that could impair the classification performance. Techniques for preprocessing images include augmentation, normalization, and scaling. In order to expand the dataset and lessen overfitting, image augmentation entails adding random modifications including rotation, scaling, and flipping.

A pre-processed input data is used by spatial domain image enhancement techniques to improve image quality. The most popular and suitable method for improving x-ray images is histogram equalization and CLAHE [18]. Thus, this histogram equalization and CLAHE approaches are used in the study.

3.3.2. Implementation

Choosing an appropriate model for images classification is the next stage. Nowadays, convolutional neural networks (CNNs) are a popular models used in image-based classification. CNNs are very well-liked because of their capacity to extract hierarchical information from pictures. Random forests work well when there are many characteristics, but SVMs are best suited for smaller datasets.

This work used improved or enhanced image dataset for classification. The required model is constructed by employing the CNN architecture. Anaconda is installed for authoring Python code. Python is the preferred language for the research due to its ease of use, readability, large standard library, robust community, and cross-platform compatibility [19]

As software tools to develop a CNN-based model in this thesis, different machine learning libraries (TensorFlow, Keras), data science libraries (Pandas, NumPy), visualization libraries (Matplotlib), Python programming language, and Jupyter notebook are used. These tools meet all the criteria for consideration, plus they are written in Python, which we are familiar with. Below is a detailed illustration of the fundamental software tools used in this investigation.

- ✓ **Python:** The entire work of the Thesis is coded in Python programming language. **Python** is one of the most popular programming languages that offer open-source, interpreted, interactive, powerful, fast, high-level, dynamic language and provides a great approach for object-oriented scripting language. Because of its simple syntax and ease of use, it is an easy language for non-engineering people to learn, and programmers can express concepts in fewer lines of code than with other high-level languages. Python typically supports a number of concurrent programming paradigms, including procedural, object-oriented, functional, and efficient programs. It is a wonderful tool to utilize for managing enormous volumes of data for training the deep learning model, inserting input, or even making sense of its output because its primary uses are data management, manipulation, and forecasting. Because of this Python feature, we were able to successfully implement the algorithm in smaller code lines [20].

- ✓ **Tensor Flow:** is an end-to-end open-source library or platform that is mostly used for creating large-scale machine learning and deep learning applications. TensorFlow is the free representative mathematical library, which was used for ML functions, mainly neural networks. All the computations in TensorFlow involve tensors (n-dimensional array) that represent all kinds of data. TensorFlow also uses a graph framework for the graphical representation of the series of computations during the training. It can run on many different platforms once training is completed [21]
- ✓ **Keras:** it is a high-level deep-learning API for building neural networks. It is used to simplify the implementation of neural networks and is written in Python**.** It also supports multiple backend neural network computations, running on top of machine learning platforms like TensorFlow. It is very simple to develop a model, user-friendly, easily extensible with python, and most importantly it contains pre-trained CNN models such as VGG16 and Inception [21].
- ✓ **Anaconda** is used for the implementation of the model. It is a free and open-source distribution of the Python and R programming languages for data science and machine learning-related applications that aims to simplify package management and deployment. It has many pre-installed packages used in data science such as NumPy, Pandas, Scipy, etc., and is one of the most popular python distributions in the world with over 11 million users. Anaconda also contains various IDEs such as Jupyter Notebook and Spyder, etc. for large data processing, data analysis [20][21]
- ✓ **Jupyter** Notebook is an open-source, web-based interactive environment, widely used in data science to create and share documents that include live code, mathematical equations, graphics, maps, plots, visuals, and narrative texts. It is a very fast web application to execute code, allows debugging and execution in the browser, and displays computing results [20]

- ✓ **NumPy** or numerical python, is one of the most widely used Python packages for data analysis, numerical data, and scientific computing. NumPy is a very popular Python library that includes a large multidimensional array object, various derived objects (such as masked arrays and matrices), and a collection of routines for performing fast operations on arrays [21].
- ✓ **Matplotlib** provides object-oriented APIs for integrating plots into applications. It is a cross-platform, data visualization, and graphical plotting library for Python. It can create a variety of visualization reports, including line plots, scatter plots, histograms, bar charts, pie charts, box plots, and many more [21]
- ✓ **Pandas:** is a Python library that provides high-performance and easy-to-use data structures and data analysis tools for the Python programming language. It is open-source, fast, powerful, and flexible. It's most commonly used for data science, data analysis, and machine learning tasks [21].
- ✓ **Google Colab:** is a product of Google Research that allows anyone to write and execute arbitrary python code through the browser and is particularly suited for machine learning, deep learning, data analysis, and education. From a technical perspective, Colab is a Jupyter notebook service that requires no configuration to use while providing free computing resources, including GPUs [22]
- ✓ **Micro-Dicom:** A cross-platform, free, open-source medical imaging tools and library is called Micro-DICOM. Developers who wish to include DICOM (Digital Imaging and Communications in Medicine) capability into their applications are the main target audience for this design. DICOM is a standard for managing, archiving, producing, and sending patient data associated with medical imaging [23]

3.3.3. Evaluation method

The model is evaluated on the test set, which is a different subset of the dataset, following training and validation. Images from the test set should be reflective of real-world situations and should be separate from the training and validation sets. Deep learning models for classifying chest X-ray images frequently employ a number of well-liked evaluation techniques. In this study the performance of the model is evaluated using techniques such as accuracy, precision, recall and F1 score. Accuracy is a general measure of correctness, Precision focuses on the accuracy of positive predictions, Recall emphasizes the ability to find all positive instances, F1 Score balances precision and recall.

The following section provides a detailed explanation of the study's methodology for better understanding.

3.4. Data preparation

We obtain the privilege of accessing the hospital's **Med-web** as radiologist role from the hospital system admin in order to gather X-ray datasets from the radiology department of Black Lion Hospital. After the download of every x-ray image available between 2019 and 2024, we were able to obtain 10,000 all type of x-ray datasets from which we selected 3000 relevant chest x-rays. (i.e. there is different type of x-ray image rather than chest x-ray like abdomen, kneels etc.) These included suspected cases of pneumonia, normal chest x-rays, suspected cases of non-pneumonia or cases of other diseases, such as TB, metastatic disease, and bronchitis. Then, we separate and categorize this 3000-chest x-ray as pneumonia, non-pneumonia and normal x-ray with 1000 of x-ray data for each category.

The last stage of data preparation is anonymization, which involves hiding patient private information using Micro-Dicom anonymizer software to address ethical and data privacy concerns. The data is now prepared for image pre-processing.

3.4.1. Image pre-processing

Image pre-processing is the next phase after acquiring an image, it is required since raw data are prone to noise, corruption, missing, and inconsistent data. And also, it is the set of operations carried out to format images before they are used in model training and inference. It also expedites model inference and reduces the time required for model training. Preparing the input images for the entire given image analysis activity is the responsibility of the pre-processing step. The training and validation chest x-ray images are fed into the pre-processing component [58]. In this subject study, loading, resizing, normalizing image, and data augmentation are the image pre-processing techniques used. In order to show how image enhancement approaches affect model performance, we applied image enhancement techniques.

According to [59] the following are the common image preprocessing tasks that we used in this study.

✓ **Read or Load Images**

We have an image of pneumonia, non-pneumonia and normal chest x-ray. When we look really closely at the sample images. We will notice that it is made up of small square boxes. These are called pixels. In this phase, we simply store the path to our image dataset in a variable and then build a function to load image-containing folders into arrays so that computers can handle it. We displayed the loaded images, using a 2D array matrix as follows.

Pneumonia suspected image

non-pneumonia image

normal x-ray image

Figure 3.2 Read or Load image

✓ Image Resizing

Image Resizing is one of the pre-processing techniques which brings the whole image to the same size, this is altering the image's dimensions, which entails adjusting both its width and height. All images must be resized to a fixed size before being fed to the CNN because neural networks only accept inputs of the same size. Therefore, size normalization was performed in our dataset in order to get a similar size of all images and to decrease the computational time of training. We have resized our image datasets to 224 X 224 pixels. There is no standard set to resize our images but we resized them based on the computational resources we have. Because, as we increase the image size, it needs more computational resources to be processed [60].

Image resizing is the process of changing the dimensions of an image, either by increasing or decreasing its width and height. This is a common preprocessing step in various image processing tasks, including machine learning, where input images need to have a uniform size [61]. Our original X-ray images are large in size, so they need to be resized and standardized to ensure uniform dimensions.

Sample code for re sizing images into 224X224 dimensions:

```
Size = 224

Image = image. Resize (size, size)
```

Sample result of image resizing is shown in figure 3.3 below.

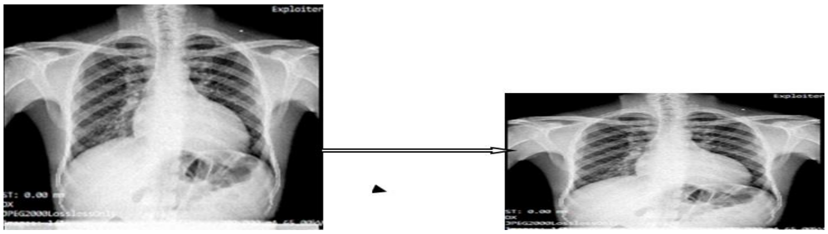
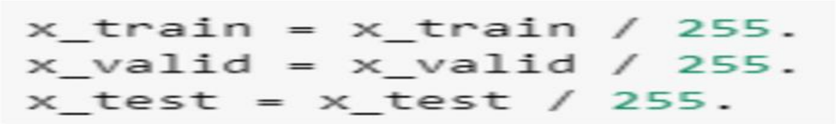


Figure 3.3 Re sizing image

✓ Normalization

Normalization is used to pre-process an input image before further processing begins. It is used to scale down the feature values in the range between 0 and 1. Gray scale image pixel values are an integer between the ranges of 0 to 255. Although these pixel values can be presented directly to the model, they can result in slower training time and overflow. Overflow happens when numbers get too big and the machine fails to compute correctly [62]. We normalized our data values down to a decimal between 0 and 1 by dividing the pixel values by 255. By normalizing the data, we ensure that the model processes inputs more efficiently, prevents overflow issues, and achieves faster convergence during training. Common methods include Min-Max Normalization, which scales values directly to the 0-1 range, and Z-Score Normalization, which standardizes pixel values to have a mean of 0 and a standard deviation of 1 [63].

Sample code for normalizing images into decimal values between 0 & 1 is given below.



```
x_train = x_train / 255.
x_valid = x_valid / 255.
x_test = x_test / 255.
```

✓ Data Augmentation

Data augmentation is a technique used to increase the size of the training dataset by generating new synthetic samples. This is especially helpful for image-based classification jobs, where it can be costly and challenging to gather a significant volume of labeled data. For image data, data augmentation methods include cropping, rotating, scaling, and flipping. By using these methods, overfitting can be lessened and the model's capacity to generalize to new images can be enhanced.

We use several data augmentation techniques to enhance the training dataset, including random rotations, width and height shifts, shear transformations, zooming, and horizontal flipping. These methods introduce variations in the orientation, position, scale, and perspective of the images, helping the model become more robust and better generalize to new, unseen data by learning to recognize objects under different conditions. These augmentations create a more diverse dataset, which improves the model's performance and adaptability.

After we applying data augmentation, the training dataset expanded from 2,400 to 9,600 images, significantly enhancing the model's performance. This increase in data diversity helps the model generalize better, reducing overfitting and improving accuracy on new, unseen data.

3.4.2. Image enhancement techniques

Image enhancement is a technique used to increase a digital image's quality by increasing its visual appeal or making it more suitable for a specific application. Numerous industries, including satellite imaging, photography, and medical imaging, heavily rely on this technology. The primary goal of image enhancement is to bring out the relevant features in an image that might be lost due to poor lighting conditions, noise, or low resolution [64].

In general, contrast enhancement is performed first for most of the image enhancement methods. Contrast enhancement techniques are categorized to spatial domain and frequency domain methods based on the operations on the pixels. In spatial domain methods, operations are directly applied on the image by means of algorithms that are usually based on gray-level content. Researchers have commonly used histogram equalization, log transformation and gamma correction for contrast enhancement [65].

✓ Histogram equalization (HE)

Histogram image shows how the pixel intensity levels are distributed. It provides insight into how an image's pixels might be altered to visualize a more natural-looking quality. Histogram equalization (HE) is one of the popular time domain techniques due to its easy implementation and performance.

Histogram Equalization (HE) is effective for real-world applications as it uses a Cumulative Distribution Function (CDF) from the input image, transforming it into a uniform distribution to cover all gray levels. While simple and reversible, it can cause over-enhancement if there are peaks in the histogram [66]. The CDF shows the probability that a random variable is less than or equal to a specific value, "x." For discrete variables, it's the sum of probabilities up to "x," while for continuous variables, it's the integral of the probability density function (PDF) up to "x." [65]. The CDF ranges from 0 to 1, providing a complete description of the variable's probability distribution [67].

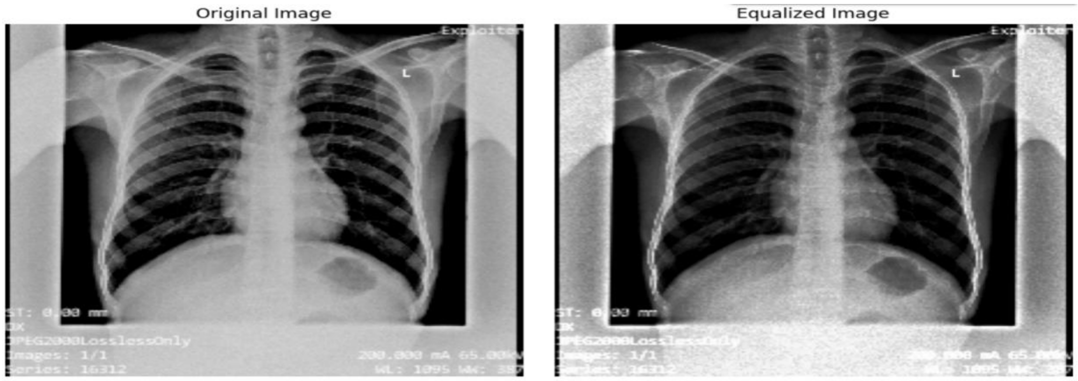


Figure 3.4 image after applying HE

✓ Contrast Limited Adaptive Histogram Equalization (CLAHE)

Contrast Limited Adaptive Histogram Equalization (CLAHE) is advanced image enhancement technique designed to improve local contrast in images. Traditional histogram equalization modifies the intensity values across the board, whereas CLAHE only affects smaller, non-overlapping areas or image tiles. By enhancing local contrast, this technique reduces noise amplification and improves the visibility of features [68].

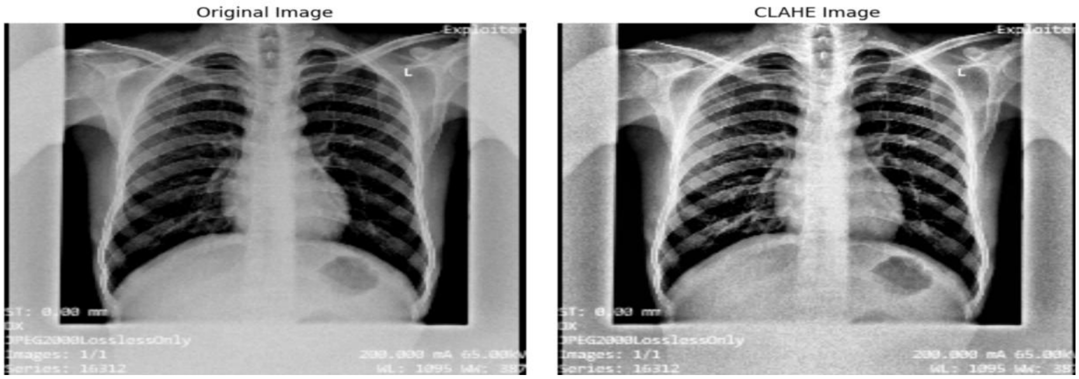


Figure 3.5 image after applying CLAHE

CLAHE (Contrast Limited Adaptive Histogram Equalization) differs from standard Histogram Equalization (HE) by enhancing contrast in localized regions of an image rather than globally. While HE redistributes pixel intensities across the entire image, potentially leading to artifacts and noise amplification, CLAHE divides the image into small tiles and performs histogram equalization on each tile separately. This local approach allows CLAHE to enhance contrast more effectively in varying regions while including a contrast-limiting step to prevent excessive enhancement and reduce noise. The tiles are then seamlessly blended to maintain a natural appearance [54]. In our study, we used both CLAHE (Contrast Limited Adaptive Histogram Equalization) and HE (Histogram Equalization) methods for image enhancement. Combining both methods can leverage the advantages of each: HE can provide a broad, initial contrast boost, making it useful for overall enhancement, while CLAHE can be applied subsequently to refine and improve local contrast and details, ensuring both global and local enhancements are achieved with reduced artifacts. The following image shows enhanced image after applying both HE and CLAHE techniques.

Figure 3.6 sample enhanced x-ray image

3.4.3. Data splitting

First, before any real training takes occur, the dataset is split into three sections: training, validation, and testing. The training split is used to train the model, where loss values are calculated via forward propagation and learnable parameters are updated via backpropagation, the validation split is used to assess the performance of the model which is built during the training and it is also used to fine-tune model parameters in order to select the best performing model. Finally, the test split is used only once at the end of the project in order to evaluate the performance of the final model that was fine-tuned and selected on the training process with training and validation sets. 80:10:10 is a commonly suggested data splitting ratio for test, validation, and training sets in deep learning pneumonia classification. This ratio is widely used in medical image analysis research to ensure a fair procedure for model evaluation and training [69]

Three thousand (3000) chest x-ray images were chosen for this thesis. The experiments are done with a ratio of 08:01:01 (80% training, 10% validation, and 10% testing). Thus, 2400 images, or

80% of the dataset, are used to train the model, 300 or 10% of images are used to verify the model, and 300 images, or 10% of the dataset, are used to test the trained model.

The dataset splitting task is done using the split folders package into train, validation, and test according to the ratio stated above. Using an equal number of images in each class for training and validation helps to avoid the problem of overfitting because during the training updating weights would not be biased in one of the categories.

After a period of trial and error during the training phase the datasets must be augmented to boost the performance of particular models. In order to improve each model's performance and artificially increase the amount of the training dataset by creating modified versions of images, we employed the data augmentation technique with the ImageDataGenerator class from Keras. To avoid process bias, only the training data is increased three times beyond the initial training set throughout the augmentation phase and the test and validation data stay the same. As a result, the training data set was expanded from 2400 to 9600 images (2400 original data plus 7200 augmented data).

3.4.4. Training and Testing phase

The goal of the training phase is to develop a model that can accurately classify pneumonia, non-pneumonia and normal x ray images. The classifier model builds by the classification algorithm by analyzing or “learning” from the training dataset made up of data tuples and their associated class labels also called labeled datasets. In the training phase, there are four components namely: image pre-processing, feature extraction, classification, and model training (building) components.

The goal of the testing phase is to check the accuracy of the model. It is used only once at the end of the project in order to evaluate the performance of the final trained model that was finetuned and selected on the training process with training and validation sets. The testing phase includes only the image pre-processing components, identical to those used in training.

3.5. Constructing a Model

One of the most important steps in creating precise and dependable algorithms for interpreting medical images is model selection in medical image categorization. In order to attain the best results in medical image classification, it entails selecting the most suitable machine learning algorithm and adjusting its parameters. The idea is to choose a model that can distinguish between various classes of medical images—for example, different kinds of tumors or abnormalities—effectively [55].

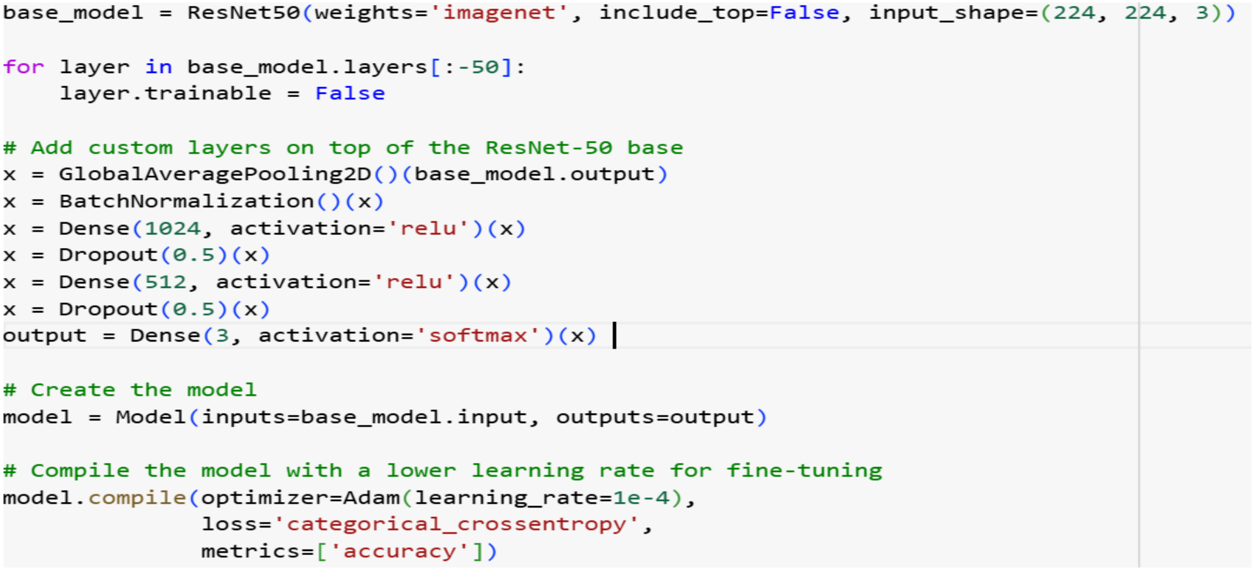
In this study, we used deep learning algorithms to develop models that are made up of several layers of neurons in a neural network. A deep learning algorithm which is CNN is chosen for this study based on different literature that was conducted in computer vision, especially in image classification. CNN is designed by emulating human's understanding of vision, so, for image-related tasks, CNN is better than other deep learning models. While simple neural   networks have

some success in classifying basic binary images, they cannot handle complex images with pixel dependencies. They don't have the computing power needed to handle images with large pixels, which is exactly where CNNs come in. CNN can help identify even the most complex images with high accuracy. CNN algorithms can apply correlation filters to identify spatial and temporal dependencies in images. The main advantage of using the CNN algorithm for pneumonia classification is that it is more robust and automated than classical machine learning algorithms For this investigation, we chose better convolutional neural network pre-trained models, such as ResNet-50, InceptionV3 and VGG-16 based on recommendations from many literatures for compression of models [53]and also, we used custom CNN-L5 architecture to construct optimal models.

3.5.1. Steps to apply Pretrained models

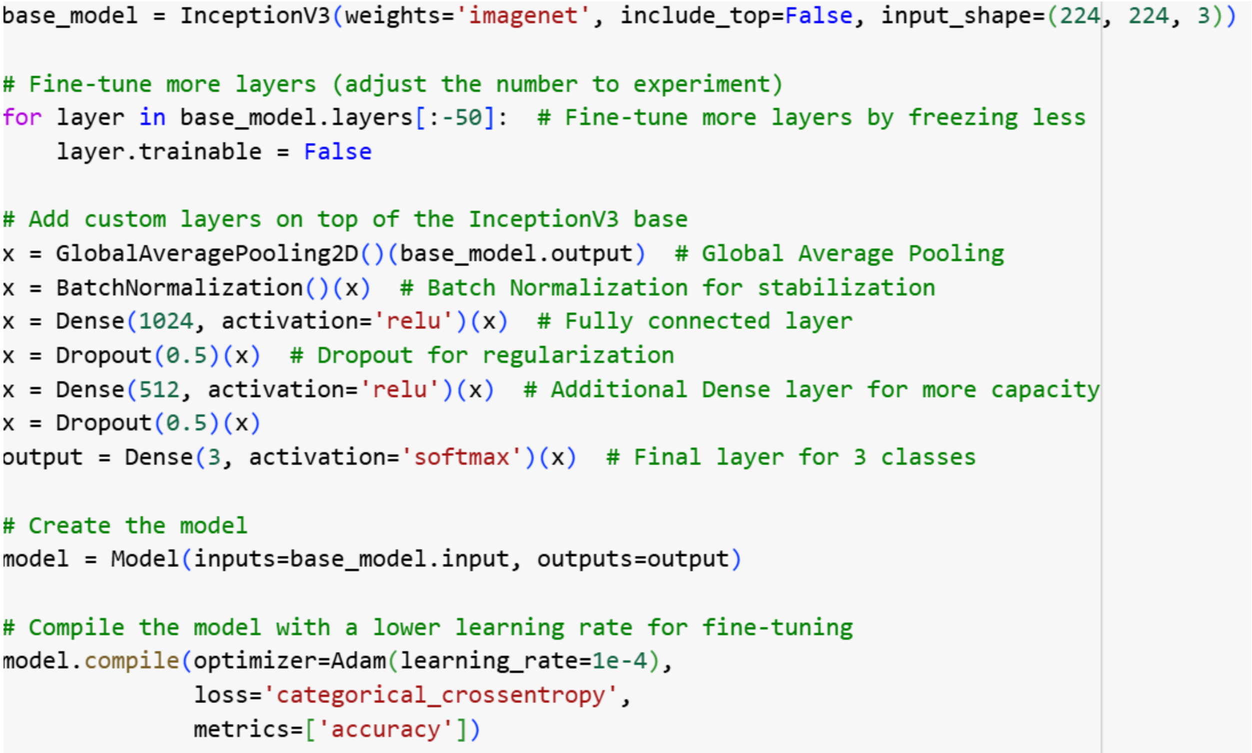
In order to improve training results, we adjust and optimize the Resnet-50 model in this work. The primary procedures that we followed in order to put this model into practice are outlined below.
To remove the default classification layer, we first load the pre-trained ResNet 50 model on ImageNet using include_top=False. Next, we freeze the layers up to the final few residual blocks. This avoids the need to retrain the preceding layers by maintaining ResNet-50's generic feature extraction capability. After the pre-trained ResNet 50 layers, add a Global Average Pooling (GAP) layer. Next, add a Dense layer, dropout (0.5) for pneumonia-classification. Fine-tune the last few layers, leaving the previous ones frozen. To make sure that learning goes smoothly, use early stopping and a reduced learning rate.

Sample code for Resnet-50 finetune layers:



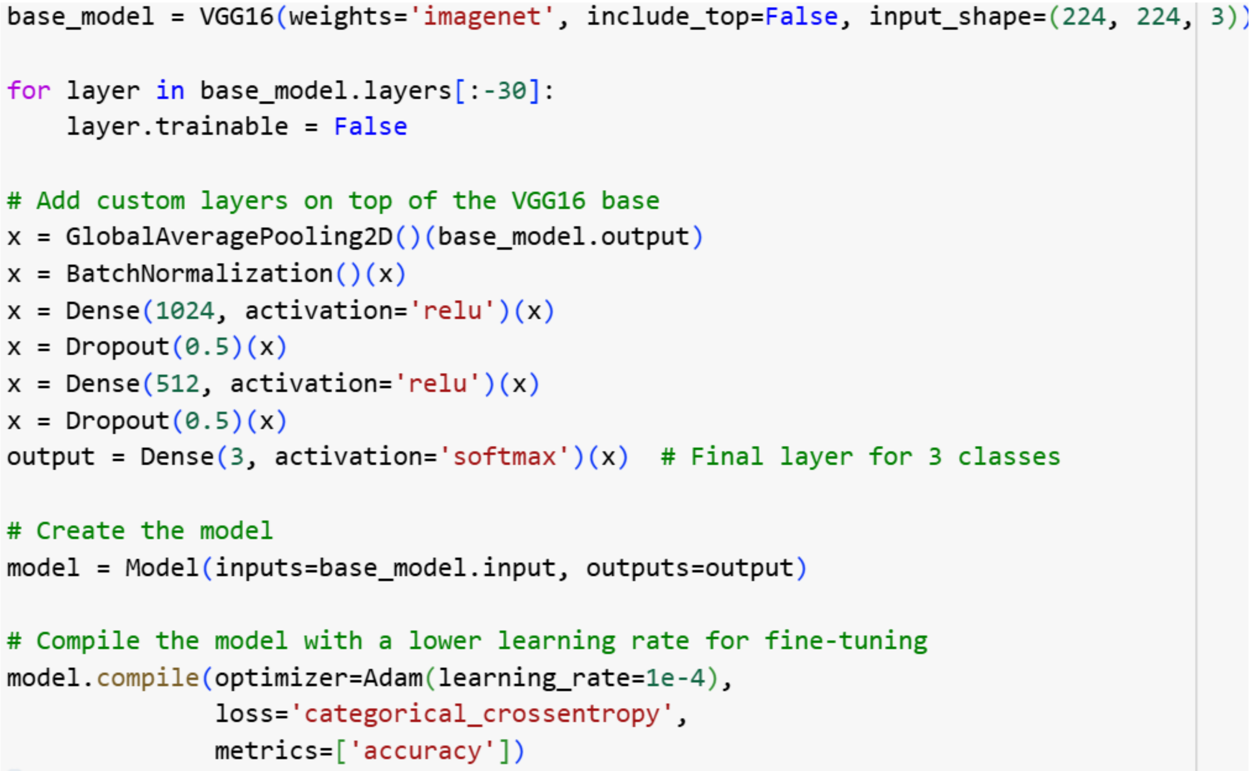
```
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

for layer in base_model.layers[:-50]:
    layer.trainable = False

# Add custom layers on top of the ResNet-50 base
x = GlobalAveragePooling2D()(base_model.output)
x = BatchNormalization()(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(3, activation='softmax')(x)

# Create the model
model = Model(inputs=base_model.input, outputs=output)

# Compile the model with a lower learning rate for fine-tuning
model.compile(optimizer=Adam(learning_rate=1e-4),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Here is the sample code for the Inception V3 model, which follows the same procedures as Resnet 50 mode with 50 fine-tuned layers.



```
base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Fine-tune more layers (adjust the number to experiment)
for layer in base_model.layers[:-50]:  # Fine-tune more layers by freezing less
    layer.trainable = False

# Add custom layers on top of the InceptionV3 base
x = GlobalAveragePooling2D()(base_model.output)  # Global Average Pooling
x = BatchNormalization()(x)  # Batch Normalization for stabilization
x = Dense(1024, activation='relu')(x)  # Fully connected layer
x = Dropout(0.5)(x)  # Dropout for regularization
x = Dense(512, activation='relu')(x)  # Additional Dense layer for more capacity
x = Dropout(0.5)(x)
output = Dense(3, activation='softmax')(x)  # Final layer for 3 classes

# Create the model
model = Model(inputs=base_model.input, outputs=output)

# Compile the model with a lower learning rate for fine-tuning
model.compile(optimizer=Adam(learning_rate=1e-4),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

The following code shows applying of VGG 16 in our study; it uses the same steps as the model mentioned above but with more refined layers.



```
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

for layer in base_model.layers[:-30]:
    layer.trainable = False

# Add custom layers on top of the VGG16 base
x = GlobalAveragePooling2D()(base_model.output)
x = BatchNormalization()(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(3, activation='softmax')(x)  # Final layer for 3 classes

# Create the model
model = Model(inputs=base_model.input, outputs=output)

# Compile the model with a lower learning rate for fine-tuning
model.compile(optimizer=Adam(learning_rate=1e-4),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

3.6. Implementation tools

3.6.1. Software tools

In order to select the best tool for implementing the CNN algorithm for image-based classification of pneumonia, an investigation of various software tools and their libraries is conducted.

During our research, we discovered that there are tools that are generic for both deep learning and machine learning algorithms, as well as tools that are particular to only one of them.

To enable select the right software tools and libraries, we took into account a number of factors before deciding on the tools. The programming language that will be utilized to implement the algorithm is the first crucial factor to consider. Other criteria include selecting tools with sufficient learning materials, such as free video tutorials and prior expertise, as well as the tools' ability to be utilized in computers with minimal resources (like CPU or GPU).

As software tools to develop a CNN-based model in this thesis, different machine learning libraries (TensorFlow, Keras), data science libraries (Pandas, NumPy), visualization libraries (Matplotlib), Python programming language, and Jupyter notebook were used. These tools meet all the criteria for consideration, plus they are written in Python, which we are familiar with. Additionally, a few little programs were created and used for data splitting and data preprocessing (image resizing). In the other side Web Browser, Google Golab, Micro-Dicom and other software's are implemented in this study.

3.6.2. Hardware tools

We use tools, instruments, or materials for data collection and for all this study. for instance, pen and papers, desktop and Pc computer, mobile phone, USB electronic tools etc.

CHAPTER FOUR

EXPERIMENT, RESULT AND DISCUSSION

4.1. Overview

This chapter describes the implementation of the image-based pneumonia classification process by using deep learning algorithm, which is specified in detail in the previous chapter. In this chapter, all the experimentation details such as the results of each experiment and a discussion of these results are presented briefly. The results of the experiments are shown in different graphs and tables.

4.2. Experimental setup

A model of a convolutional neural network is created and examined for several appropriate learning parameters in order to complete the pneumonia classification task, maximize the accuracy of the classification, and minimize the loss values. Resnet-50, InceptionV3, and VGG16 are the more effective and widely utilized CNN pre-trained models for pneumonia classification that we have chosen for this investigation based on previous research [39]. In order to determine which CNN architecture was best, we also attempted to compare these pre-trained CNN models with a newly built CNN-L5 sequential model.

Adding more dense layers with different units and activations on top of the base model, along with regularization techniques like dropout rate and batch normalization, number of batch-size, epochs, learning rate and using fine-tuning the pre-trained model are the main consideration and modification to improve their model performance. All of these architectures have different results with different consideration and modification of architectures which determine model performance.

When building our sequential model, we took into account the CNN-L5 architectures, which are often utilized and yield better results when models are trained with Dropout against a similar dataset. Numerous experiments have been examined in CNN-L5; the number of epochs, batch size, and learning rate with dropout vary between the studies.

As with most deep-learning classification algorithms, our experiments have **two main phases**. The first one is the training phase and the second one is the testing phase. In the training phase, as the data is repeatedly presented to the classifier, the weights with different listed considerations are updated to obtain the desired response. In the testing phase, the trained algorithm is applied to data (test data) that the classifier has never seen before to test the performance of the classification algorithm. In the following, we will look at the experimental results in detail.

4.3. Developing the proposed model

Selecting the number of layers and parameters (units) for each layer is a necessary step in designing a CNN model. We also do extensive empirical searches as part of our model construction process. We constructed many models with five layers, each including different values of padding, filters, and strids in order to get the optimal CNN model with dropout that performs the best.

We start with a minimal epoch number, batch size, and learning rate when building the model, and we adjust each value until we reach five layers with higher representational power. We increased the model's capacity (the size of its convolutional layers) and the number of filters during practice until the model produced accurate and useful results. The results show that the most practical final model has five convolution layers with good model performance. It includes epochs of 50, batch size of 64, learning rate of 0.0001, and dropout rate. The model's output is presented in the section under "Result and Discussion."

As we mentioned on the above, the convolutional neural network architecture called CNN consists of three main parts: Conv Layer, Max-pool Layer, and Fully Connected Layer. A convolution layer, called the first layer, is used to scan the incoming inputs and determine the weight, height, and depth of each image. In other words, this layer is the core of neural networks, which has hyperparameters that include the main process, stride, padding, the number of filters, the size of each filter, and the function to activate the ReLU layer. The second component is max-pooling, which is a layer between the Conv layers and minimizes the number of network parameters, especially the height, and width while maintaining the maximum input values. The fully connected layer (FC) is the third element, and this layer is the last layer of the CNN architecture to identify the class probability, and as the name of the layer indicates, each neuron in this layer is connected to all the neurons in the previous layer.

✓ CNN-L5 Architecture

The proposed CNN model consists of 18 layers, including 5 convolutions (Conv2D), 5 activation function layers, 5 max-pooling layers, 1 flatten layer, and 2 fully connected layers with input image shape (224, 224, 3). A kernel of size 5 × 5 with a stride and padding value of 2 was used in the first two Conv2D layers and a kernel of size 3 × 3 with a stride and padding value of 1 was used in the remaining three Conv2D layers. The filter size increased after every Conv2D layer, in the 1st of Conv2D, 32 filters are used to learn from the input, and the 2nd, 3rd, 4th, and 5th layers of Conv2D use 64, 96,128, and 192 filters respectively. After each Conv2D layer, a max pooling layer with a pool size of 2 × 2 was used and an activation layer with the ReLU function was used. Then, a flattened layer is used to transform the output of the 3D matrix generated by the final convolution & max pooling into a vector, which becomes the input for the 2 dense layers. The proposed CNN model is shown in Figure 3.11 and the implementation of the model is shown in Appendix A: Experiment of CNN5L Model.

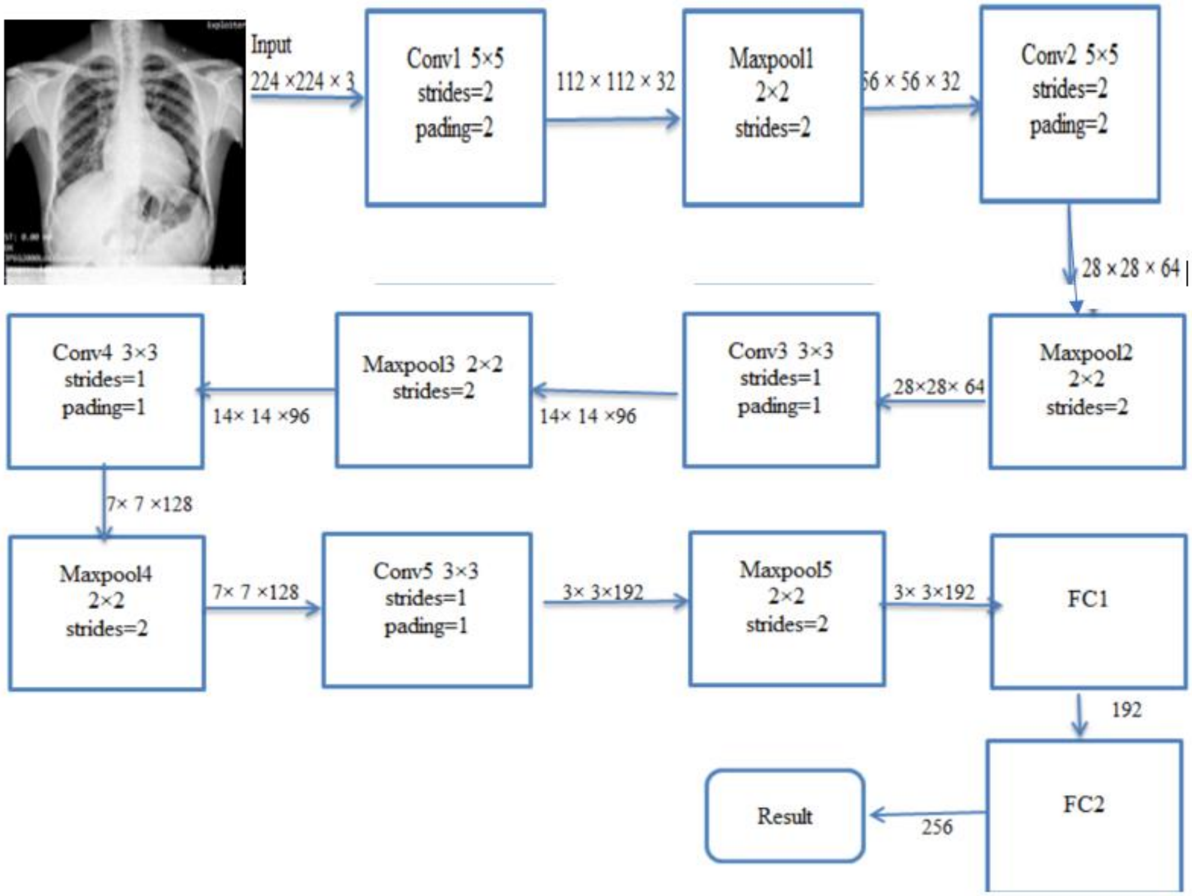


Figure 3.1 Proposed CNN5L Model

- ✓ **CNN-L5 Model Description**

The CNN-L5 model is a convolutional neural network designed for image classification tasks. It consists of several convolutional layers followed by max pooling layers, which help in reducing the spatial dimensions while retaining important features. The network employs ReLU (Rectified Linear Unit) activations to introduce non-linearity, enhancing the model's ability to learn complex patterns. After several convolutional and pooling operations, the feature maps are flattened and passed through fully connected (FC) layers, concluding with a SoftMax activation for multi-class classification. This architecture allows the model to efficiently learn and classify input images.

- ✓ **Proposed model parameters:** this table summarizes the key parameters for each layer in the CNN-L5 model, providing insights into the architecture and complexity of the model.

Table 3.1 Summary of the proposed model

Layer	**Filter/Kernel**	**Depth/Channels**	**Stride**	**Padding**	**Parameters**	**Output Size**
Input	Image	-	-	-	0	224×224×3
1. Conv2D + ReLU	5×5	32	2	Same	2,432	112×112×32
MaxPool2D	2×2	-	2	-	0	56×56×32
2. Conv2D + ReLU	5×5	64	2	Same	51,264	28×28×64
MaxPool2D	2×2	-	2	-	0	14×14×64
3. Conv2D + ReLU	3×3	96	1	Same	55,392	14×14×96
MaxPool2D	2×2	-	2	-	0	7×7×96
4. Conv2D + ReLU	3×3	128	1	Same	110,720	7×7×128
MaxPool2D	2×2	-	2	-	0	3×3×128
5. Conv2D + ReLU	3×3	192	1	Same	221,376	3×3×192
MaxPool2D	2×2	-	2	-	0	1×1×192
Flatten	-	-	-	-	0	192
6. FC + ReLU	-	256	-	-	49,408	256
Output FC + SoftMax	-	3 (classes)	-	-	771	3(class scores)

- ✓ **Hyperparameter Settings of the proposed model**

Hyperparameters are parameters whose values are set by the deep learning algorithm before the training process begins. There is no standard rule for choosing the best hyperparameters for a given problem. Therefore, many attempts are made to select hyperparameters. In the following, the hyperparameters selected for the proposed model are described.

Optimization Algorithms: The proposed model is trained using a gradient descent optimization algorithm to reduce the error rate and the error propagation algorithm is used to update the weights. Gradient descent is by far the most popular and widely used optimization algorithm in deep learning research. At the same time, every modern deep learning library contains implementations of gradual seeding algorithms such as Keras (used in this thesis). It improves the weight of the model and adjusts the parameters, therefore, reducing the loss function. To optimize the gradient descent, the Adaptive Moment Estimation (ADAM) optimizer is used. Adam calculates the adaptive learning rate at each parameter and uses squared gradients to measure the learning rate, as well as the moving average of the gradients.

Learning rate: Since back-propagation is used to train the proposed model, the learning rate is used during weight optimization. It controls the updated weights during backpropagation. The challenging part was choosing the appropriate learning rate during our experiment. In our experiments, we saw that a very small value of learning rate takes longer to train than a large value. But when we give a small value, the model is better than a model with a large education level. The experiment was performed using learning rates of 0.0001, 0.001, 0.01, and 0.1. Then, although it takes more time to train, a learning rate of 0.0001 is optimal for all experiments.

Loss function: The activation function used in the model's output layer—the final fully connected layer—directly influences the selection of the loss function, depending on whether we are solving a classification or regression problem. In the last fully connected layer of the proposed architecture, SoftMax is used as the activation function. Since we are addressing a classification problem, specifically a three-class classification task, the loss function employed for our model is categorical cross-entropy (CCE**)**.

Activation function: SoftMax activation functions are used in the experiments, and the suggested model performs better. The SoftMax activation function is employed in the model's output layer since it is the most effective option for the multiclassification problem.

The number of epochs: The number of iterations through the model or network that the entire data set is prepared and passed back. In our experiment, the model was trained using different epochs starting from 5 to 150, and we saw that when we use very small or large epochs during the training, the model creates a large gap between the training error and the validation error. After many tests, the model will be fine at epoch 30.

Batch Size: The amount of input data we transmit to the network at one time. It is very difficult to feed the whole dataset into the neural network at once, so we have to divide the input into many small batches. It is preferred in model training to reduce the calculation time of the machine. A batch size of 32 is used in our experiment during model training.

Table 3.2 Summary of best hyper parameter

Parameter	**epoch**	**Batch size**	**Activation function**	**Loss Function**	**Optimization Algorithm**	**Learning rat**
Value	30	32	SoftMax	CCE	Adam	0.0001

4.3. Experimental result

For comparison and the selection of the best performing model in both pre-trained and sequential model with enhanced datasets, the following parameters are taken into account in each trial and modified accordingly. The parameters include epochs number, batch size, dropout rate and learning rate for both pre-trained and sequential model and also presence and absence of fine-tune for pre-trained models only.

4.3.1. Experimenting InceptionV3

Hereunder table 4.1 presents summary of experiments conducted using InceptionV3 pretrained model. In the experiment we attempt to consider different epochs, batch size, learning rate and drop rate.

Table 4.3 Inception V3 result summery

Experiment	**Hyperparameter**					**Without enhanced data**	**With enhanced data**	
	Epoch	Batch size	Dropout	Learning rate	Finetune	Accuracy	Accuracy	Loss
1	20	32	0.2	0.1	frozen	78%	79.9%	0.20
2	20	32	0.1	0.01	frozen	79%	80.9%	0.19
3	30	32	0.5	0.01	frozen	80%	83.2%	0.17
4	40	64	0.5	0.001	unfrozen	87.2%	88%	0.12
5	40	32	0.5	0.001	unfrozen	89%	90.5%	0.09
6	50	64	0.5	0.0001	unfrozen	90.5%	92.8%	0.07

As shown in the above table 4.1, The experiments show how model performance improves with adjustments in hyperparameters and data enhancement. Across six different configurations, accuracy increases as parameters such as epoch count, batch size, and dropout are optimized, and learning rates are fine-tuned. Initially, with lower epoch counts, batch sizes, and higher learning rates, the model achieves accuracy around 78-80% without enhanced data and slightly higher with enhanced data. As the parameters are adjusted—particularly with increased epochs, lower learning rates, and the use of fine-tuning (enabled from Experiment 4 onwards)—the accuracy significantly improves. In the final setup (Experiment 6), the model reaches an accuracy of 90.5% without data enhancement and 92.8% with enhanced data, while loss decreases to 0.07, indicating that data enhancement and fine-tuning together effectively boost model performance.

4.3.2. Experimenting Resnet 50

Hereunder table 4.2 presents summary of experiments conducted using Resnet 50 pretrained model. In the experiment we attempt to consider different epochs, batch size, learning rate and drop rate.

Table 4.4 Resnet 50 result summery

Experiment	**Hyperparameter**					**Without enhanced data**	**With enhanced data**	
	Epoch	Batch size	Dropout	Learning rate	Finetune		Accuracy	Loss
1	20	32	0.2	0.1	frozen	50%	51%	0.49
2	20	32	0.1	0.01	frozen	51%	52%	0.48
3	30	32	0.5	0.01	frozen	53%	54%	0.46
4	40	64	0.5	0.001	unfrozen	62%	64%	0.36
5	40	32	0.5	0.001	unfrozen	69%	73%	0.27
6	50	64	0.5	0.0001	unfrozen	88.1	89.5%	0.10

The experiments demonstrate the impact of hyperparameter tuning, data enhancement, and fine-tuning on model accuracy and loss. Initially, with the pre-trained layers frozen and basic hyperparameter settings (20 epochs, high learning rates), the model shows moderate performance, achieving an accuracy of around 50-54%. As parameters are refined—particularly by lowering the dropout rate and learning rate, and increasing the number of epochs—performance improves gradually. From Experiment 4 onward, with fine-tuning enabled (unfreezing pre-trained layers), accuracy rises more significantly, especially when paired with data enhancement. The model achieves its best results in Experiment 6, reaching 88.1% accuracy without enhanced data and 89.5% with enhanced data, along with a notably reduced loss of 0.10, underscoring the combined effectiveness of fine-tuning and data enhancement in improving model performance.

4.3.3. Experimenting VGG16

Hereunder table 4.3 presents summary of experiments conducted using VGG16 pretrained model. In the experiment we try to consider different epochs, batch size, learning rate and drop rate.

Table 4.5 VGG16 result summery

Experiment	**Hyperparameter**					**Without enhanced data**	**With enhanced data**	
	Epoch	Batch size	Dropout	Learning rate	Finetune	Accuracy	Accuracy	Loss
1	20	32	0.2	0.1	frozen	45%	48.%	0.52
2	20	32	0.1	0.01	frozen	48.5%	49.2%	0.50
3	30	32	0.5	0.01	frozen	50%	51.8%	0.48
4	40	64	0.5	0.001	unfrozen	53.9%	54%	0.46
5	40	32	0.5	0.001	unfrozen	55%	57%	0.43
6	50	64	0.5	0.0001	unfrozen	77.2%	85.7%	0.14

The table summarizes the impact of various hyperparameters, data enhancement, and fine-tuning on model accuracy and loss. Initially, with frozen layers and basic configurations (20–30 epochs, higher dropout and learning rates), accuracy remains around 45-51% without enhanced data and slightly higher with it. Improvements begin as the learning rate decreases and layers are unfrozen from Experiment 4 onward, allowing the model to better adapt to the data. The final configuration (Experiment 6) with 50 epochs, fine-tuning enabled, and a very low learning rate yields the best results, achieving 77.2% accuracy without data enhancement and 85.7% with it, along with a significantly reduced loss of 0.14. This progression highlights the importance of fine-tuning and data enhancement for improved model performance.

4.3.4. Experimenting CNN-L5

Hereunder table 4.4 presents summary of experiments conducted using CNN-L5 pretrained model. In the experiment we investigate different epochs, batch size, learning rate and drop rate.

Table 4.6 CNN-L5 result summery

Experiment	**Hyperparameters**			**Without enhanced data**	**With enhanced data**	
	Epoch	Batch size	Learning rate	Accuracy	Accuracy	Loss
1	5	64	0.1	60%	62.%	0.38
2	10	64	0.01	73%	75.5%	0.24
3	15	32	0.001	87%	89.8%	0.10
4	30	32	0.0001	94.5%	96.8%	0.03

The table illustrates the effect of increasing epochs and decreasing learning rates on model accuracy and loss, both with and without data enhancement. Starting with 5 epochs and a higher learning rate of 0.1, the model achieves an initial accuracy of 60% without data enhancement and 62% with it. As epochs increase and the learning rate decreases (Experiment 2–4), the model's accuracy improves significantly, reaching 94.5% without enhancement and 96.8% with enhancement in Experiment 4. Additionally, loss decreases to a minimal 0.03, showing the benefits of longer training and lower learning rates combined with data enhancement for peak performance. Based on the result achieved by CNN-L5 we show its training and validation accuracy and loss in figure 4.1 and figure 4.2 as follows:

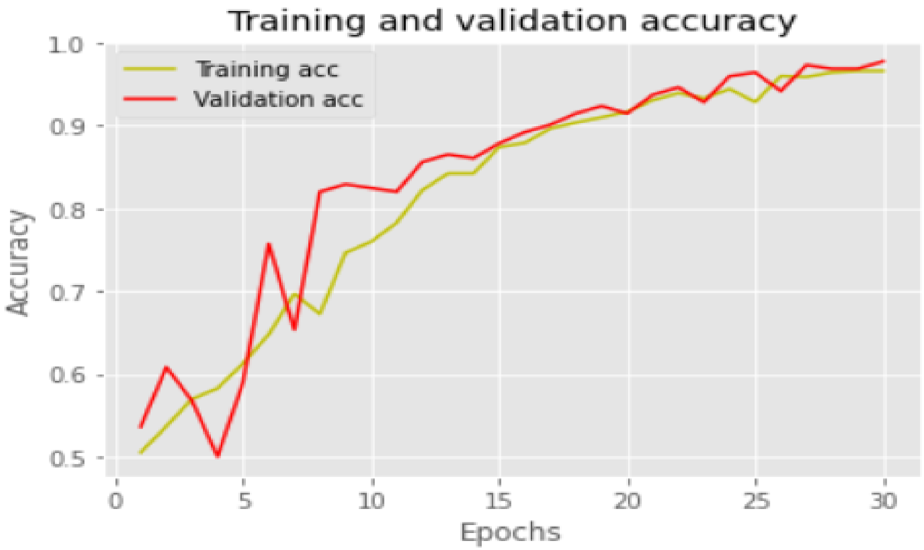


Figure 4.2 Training & validation accuracy

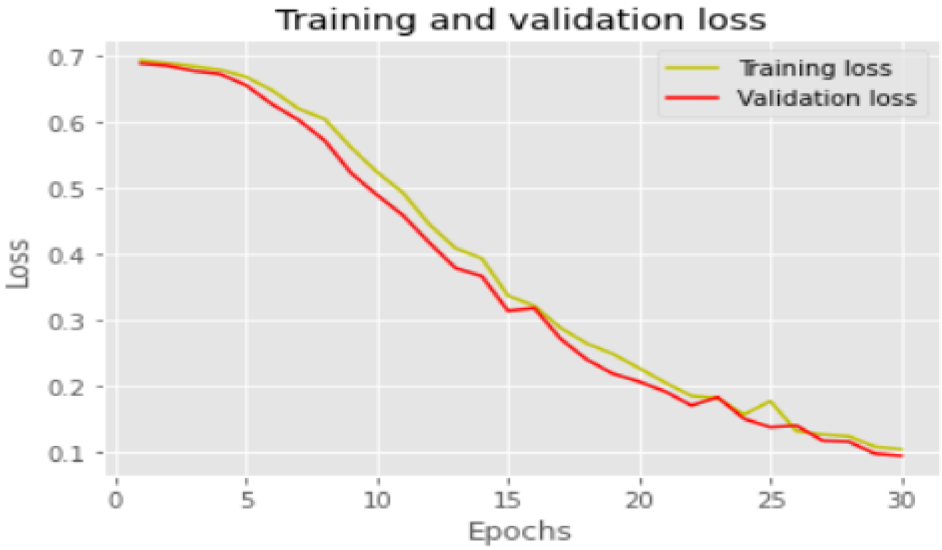


Figure 4.3 Training & validation loss

4.4. Models Comparison

The experiments were designed to explore the influence of different architectural features and to develop a high-performing model for pneumonia classification. Using an enhanced dataset, we built a new CNN-L5 sequential model for comparison alongside three powerful pre-trained models. We evaluated the performance of each model by selecting the one with the highest accuracy and lowest loss during each trial. This approach ensured we identified the model with the best balance between accuracy and loss for optimal performance across experiments.

Our trials, using hyperparameters of 50 epochs, a batch size of 64, a dropout rate of 0.5, and a learning rate of 0.0001 with fine-tuning across the pre-trained models, resulted in Inception V3 achieving 92.8% accuracy. In contrast, the CNN-L5 sequential model, with 30 epochs, a batch size of 32, a learning rate of 0.0001, and dropout, achieved an accuracy of 96.8%. When comparing the best results, the CNN-L5 model outperformed Inception V3.

Summary of the evaluation results achieved by the models experimented in this study is presented as follows in table 4.5, based on key evaluation metrics, including accuracy, precision, recall, and F1 score:

Table 4.7 The overall results of metrics

Models	**Accuracy**	**Precision**	**Recall**	**F1 score**
CNN-L5	96.8%	97.0%	95.6%	96.3%
Inception V3	92.8%	94.8%	91.3%	92.9%
Resnet 50	89.5%	90.1%	81.3%	85.4%
VGG16	85.7%	88.5%	77.3%	82.5%

As shown in Table 4.5, the CNN-L5 model achieved the highest overall performance with an accuracy of 96.8%, precision of 97.0%, recall of 95.6%, and an F1 score of 96.3%, indicating its superior ability to classify images correctly and consistently. Inception V3 follows with an accuracy of 92.8%, precision of 94.8%, recall of 91.3%, and an F1 score of 92.9%, demonstrating strong performance but with slightly lower metrics compared to CNN-L5. The ResNet 50 model recorded an accuracy of 89.5%, precision of 90.1%, recall of 81.3%, and an F1 score of 85.4%, showing good performance but with notable gaps in recall. VGG16, while achieving an accuracy of 85.7%, precision of 88.5%, recall of 77.3%, and an F1 score of 82.5%, exhibited the lowest performance among the models, particularly in recall. Overall, the CNN-L5 model outperforms the others in all evaluation metrics, highlighting its effectiveness in the classification task.

4.5. Discussion of result

This section provides interpretation of results, answering research questions, implications of findings, comparison with previous research and future research.

4.5.1. Interpretation of Results

The research results demonstrate that using various pre-trained CNN models (InceptionV3, ResNet-50, and VGG-16) alongside a custom-built CNN-L5 model, with adjustments to hyperparameters such as epochs, batch size, learning rate, and dropout rate, leads to effective classification of pneumonia in chest X-ray images. Each model was trained and tested on equal-sized datasets of 9,600 enhanced, augmented images, ensuring consistency in evaluation.

The findings reveal that image enhancement techniques notably improve image quality, thereby positively impacting model accuracy. Optimal performance was observed when increasing epochs to 50 and batch size to 64, which contributed to stability in training. Adjusting and adding layers to the models further optimized performance, particularly in the CNN-L5 model. Fine-tuning pre-trained models also proved beneficial in enhancing accuracy, and applying dropout helped reduce overfitting.

In comparison, the CNN-L5 model outperformed the pre-trained models when the epochs were reduced to 30, batch size lowered, and learning rate decreased, achieving superior classification results. These hyperparameters consistently boosted performance across all models, with CNN-L5 emerging as the most effective for pneumonia classification. Based on these outcomes, the CNN-L5 model is recommended for future applications in pneumonia detection from X-ray images.

4.5.2. Answering Research Questions

The main goal of this study is constructing effective model for pneumonia classification from x-ray images using deep learning approach. This research endeavor aims to investigate and address the following questions:

1. **Which CNN architectures are most effective for pneumonia detection in terms of accuracy and performance? (question 1)**

The study aimed to identify the most effective CNN architectures for pneumonia detection in terms of accuracy and performance. Various preprocessing techniques—including image resizing, normalization, image enhancement (like histogram equalization and CLAHE), and data augmentation—were applied to boost model performance. Among the models tested, the constructed CNN-L5 sequential model demonstrated superior accuracy and performance compared to InceptionV3, ResNet50, and VGG16 for pneumonia classification on the given dataset. This indicates that CNN-L5 is the most effective architecture for this task.

2. **How do image enhancement techniques impact image quality and model performance in pneumonia classification? (question 2)**

The study demonstrates that enhanced data consistently leads to better model performance compared to non-enhanced data. By applying image enhancement techniques, specifically Histogram Equalization (HE) and Contrast Limited Adaptive Histogram Equalization (CLAHE), image quality is notably improved, allowing for clearer feature extraction essential for accurate pneumonia classification. These techniques effectively enhance the contrast and detail within chest X-ray images, making critical features more distinguishable. This improvement in image quality directly contributes to higher model accuracy and reliability, as evidenced by the superior performance of models trained on enhanced data. Thus, the use of HE and CLAHE has a measurable positive impact on both image quality and the overall efficacy of the pneumonia detection model.

4.5.3. Implications of Findings

The suggested approach performs very well in diagnosing pneumonia from X-ray images. The model is a useful tool in medical diagnostics because of its high degree of precision and reliability, as seen by its 96.8% accuracy rate. It can significantly aid radiologists by providing quick and accurate assessments of X-ray images, potentially reducing the workload and improving the speed of diagnosis. Additionally, the efficiency of the CNN-L5 model ensures it can be deployed effectively in settings with limited access to skilled radiologists, extending its benefits to a wider range of healthcare environments.

4.5.4. Comparison with Previous works

As mentioned in the literature review section 2.9, this study attempts to close the previous research gap. Real local datasets and the Ethiopian context are used in the research. The medical sector's quality of x-ray diagnosis for pneumonia detection in our country differs from that of other countries due to a number of factors, including the volume of cases, lack of training and expertise, healthcare infrastructure, low-resolution X-ray machines, and digital imaging technologies. For this reason, the research is conducted in an Ethiopian environment in order to be useful to the health care system in our nation.

Previous studies have primarily employed usual preprocessing methods and regular data to train their models. To increase model performance, this study included histogram equalization and CLAHE image enhancement in addition to the usual preprocessing methods. Also, different hyperparameters are experimented to see their effect on the performance of the model.

4.5.5. Research contribution

This research makes several significant contributions to the field of digital image processing in healthcare.

- ✓ In the beginning, it entails the **release of a dataset** that was obtained from database systems in various formats. Furthermore, the datasets are arranged so that an equal amount of adult and pediatric x-ray images are taken into account. This closes the anatomical and physiological differences between the two groups, which had an impact on the results' generalizability. Researchers concentrating on image-based classification using deep learning techniques can access and benefit from the x-ray picture generated from this dataset, which is utilized in classification algorithms. This gift offers an invaluable resource to the academic community, facilitating additional research and progress in the area.
- ✓ The research's second contribution focuses on **optimizing the CNN-L5** model's performance through the construction of various options, or hyperparameters, and appropriate image enhancing approaches. It produces more accurate findings; the classifier has a 96.8% test accuracy with a lower loss value.

- ✓ The third contribution of our research is to **enhance the working environment and efficiency** of Ethiopian radiologists. Due to the research's focus on Ethiopia, it aims to close the gap between our nation and other developed countries in terms of modality quality, low x-ray machine resolution, workflow load, and a lack of highly qualified specialists. [72]

The research conducted here can **serve as a guide** for enhancing further the performance of image-based pneumonia classification and related image processing researches.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1. Overview

The main conclusions and insights from the study are summarized in this section. The conclusion highlights the important findings and demonstrates how they support the initial goals. It frequently considers the advantages or disadvantages of the chosen course of action. The recommendations section offers strategies to enhance future efforts, implement best practices, or go deeper into specific areas, all while providing practical measures based on the findings.

5.2. Experimental Workflow

The system is going to pass the major phases such as image acquisition, image preprocessing, data splitting, training phase, and testing phase. The image acquisition part is collecting image data sets from the problem domain; then we split the image datasets into the training phase (training and validation images) and the testing phase (testing image) which is used to train the model and is used for evaluating the trained model respectively. Image pre-processing, which includes operations like reading or loading an image, resizing an image, and feature scaling, is done during the training phase. Next, the pre-processed images from this step are sent into the CNN algorithm, which trains it by taking useful characteristics out of each image in the CNN stacked layers and classifying the images according to those features.

During the training of the model, we can access the performance of the model by using the validation dataset which is basically used to measure the performance of the model. After accessing the performance of the model, the model which is best performed is saved and used as a predictive model. Then, the testing phase is performed, which takes the output of the first phase framework (which is the trained model) and uses it to predict the model using unseen test image datasets. Finally, it gives class prediction which is the probability of the image belonging to one of the given classes during the training (in our case the classes are Pneumonia, Non-pneumonia or Normal).

5.3. Conclusions

Pneumonia is one of the leading causes of death worldwide, affecting both the young and the elderly. It can manifest through various symptoms, including fever, coughing, dyspnea, and chest pain. If a pneumonia-related lung infection is not detected promptly, it can be fatal.

Early detection and treatment of pneumonia require the expertise of skilled and experienced radiologists. However, in our country, several challenges impede effective diagnosis and treatment, such as poor machine resolution or low-quality imaging modalities, a shortage of qualified professionals, insufficient resources to handle a large number of cases quickly, inconsistency in results among radiologists and physicians, and delays in treating cases.

this study evaluated the effectiveness of various CNN architectures, including InceptionV3, ResNet50, VGG16, and the custom CNN-L5 model, to determine the most suitable model for pneumonia classification. Through extensive experimentation with both enhanced and non-enhanced datasets, the CNN-L5 model demonstrated superior accuracy and performance, achieving 96.8% accuracy with minimal loss on enhanced data. This result indicates that CNN-L5 is the most effective architecture among those tested for accurate pneumonia detection.

The study also highlights the positive impact of image enhancement techniques, such as histogram equalization and CLAHE, on both image quality and model performance. By improving contrast and feature clarity, these techniques contributed significantly to the model’s ability to accurately classify chest X-rays, particularly in differentiating between normal, non-pneumonia, and pneumonia cases.

Despite achieving promising results, the study faced limitations due to restricted access to diverse datasets from multiple hospitals, which may affect the generalizability of the findings. Further research is recommended to validate the model across a broader dataset to confirm its effectiveness in varied healthcare settings. Overall, the CNN-L5 model, combined with enhanced data, offers a reliable tool for pneumonia detection, especially in environments with limited access to skilled radiologists.

5.4. Recommendation

Based on the findings of our research, which demonstrated the high accuracy of the CNN-L5 model in detecting pneumonia from X-ray images, we propose the following recommendations to enhance the integration and effectiveness of this technology in clinical settings.

5.4.1. Integration with Hospital Diagnostic Systems:

- ✓ **Implementation**: Integrate the CNN-L5 model into hospital diagnostic systems to streamline the pneumonia detection process. This integration should allow for seamless processing and analysis of X-ray images within the existing workflow of healthcare providers.
- ✓ **Interoperability**: Ensure that the AI system is interoperable with various hospital information systems (HIS) and electronic health records (EHR) to facilitate easy access and utilization of diagnostic results.

5.4.2. Data Organization and Management:

- ✓ **Database System**: Develop a comprehensive hospital database system that organizes and categorizes X-ray datasets efficiently. This system should support easy retrieval and management of patient data, ensuring that relevant datasets are readily available for training and validation purposes.
- ✓ **Data Standards**: Establish standardized protocols for data collection, storage, and annotation to maintain the quality and consistency of the datasets used for training the AI models.

5.4.3. Training for Radiologists:

- ✓ **AI Education**: Provide training programs for radiologists to educate them about the fundamentals of artificial intelligence and its applications in medical imaging. This will help them understand the capabilities and limitations of AI tools.
- ✓ **Practical Training**: Conduct hands-on training sessions on how to use AI-powered diagnostic systems, in their daily workflow. This will improve their confidence and efficiency in utilizing AI tools.
- ✓ **Continuous Learning**: Implement continuous professional development programs to keep radiologists updated on the latest advancements in AI technology and medical imaging.

By implementing these recommendations, hospitals and healthcare providers can significantly enhance their pneumonia diagnosis capabilities, leading to faster, more accurate, and more efficient patient care. This will ultimately contribute to better health outcomes and reduced mortality rates associated with pneumonia.

5.5. Future research Direction

✓ **Expansion of Data Collection**:

Inclusion of Multiple Hospitals: Expanding the dataset to include data from multiple hospitals will increase the diversity and representativeness of the data, improving the generalizability of the findings.

✓ **Model Comparison and Enhancement**:

Exploration of Additional Models: Researchers should explore and compare various machine learning models beyond those used in this study. This could include deep learning architectures, ensemble methods, or newer models that may provide better performance.

✓ **Technical Improvements**:

- Utilization of Advanced Computational Resources: Leveraging platforms like Google Colab Pro or pro+ can address computational limitations. Future research could also explore other cloud-based solutions or local high-performance computing resources.
- Ensuring Robust Data Connection: A stable and high-speed internet connection is crucial for efficient data collection and system deployment. Researchers should invest in reliable network infrastructure or use offline data collection

REFERENCE

[1] M. A. Sahbal, M. O. Alghoneimy, S. S. Eldine, A. E. Elhadidy, and M. M. Kenawy, "Assessment of lung parenchyma recovery after antibiotic administration using lung ultrasound in critically ill patients with pneumonia," *Open Access Maced. J. Med. Sci.*, vol. 9, no. B, pp. 480–485, 2021, doi: 10.3889/oamjms.2021.6153.

[2] T. Agrawal and P. Choudhary, "Segmentation and classification on chest radiography: a systematic survey," *Vis. Comput.*, vol. 39, no. 3, pp. 875–913, 2023, doi: 10.1007/s00371-021-02352-7.

[3] T. Wójcicki and M. Nowicki, "Heuristic analysis model of nitride layers' formation consisting of the image processing and analysis and elements of artificial intelligence," *Materials (Basel).*, vol. 9, no. 4, 2016, doi: 10.3390/ma9040265.

[4] C. L. Chowdhary and D. P. Acharjya, "Segmentation and Feature Extraction in Medical Imaging: A Systematic Review," *Procedia Compute. Sci.*, vol. 167, no. 2019, pp. 26–36, 2020, doi: 10.1016/j.procs.2020.03.179.

[5] A. Kumar, P. Tiwari, S. Kumar, D. Gupta, and A. Khanna, "Identifying pneumonia in chest X-rays: A deep learning approach," Procedia Computer Science., vol. 145, pp. 511–518, 2019.

[6] S. Katoch, S. Sharma, and R. P. P. Singh, "Pneumonia Disease Detection Using Deep Learning Methods from Chest X-Ray Images: Review," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 10, no. 4, pp. 2734–2740, 2021, doi: 10.30534/ijatcse/2021/131042021.

[7] V. Naralasetti, R. K. Shaik, G. Katepalli, and J. D. Bodapati, "Deep learning models for Pneumonia identification and classification based on X-ray images," *Trait. du Signal*, vol. 38, no. 3, pp. 903–909, 2021, doi: 10.18280/ts.380337.

[8] X. Wang, Y. Zhao, and F. Pourpanah, “Recent advances in deep learning,” *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 4, pp. 747–750, 2020, doi: 10.1007/s13042-020-01096-5.

[9] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A Survey on Deep Transfer Learning," IEEE Transactions on Artificial Intelligence, vol. 1, no. 1, pp. 1–10, 2018.

[10] A. Jacobi, M. Chung, A. Bernheim, and C. Eber, “Portable chest X-ray in coronavirus disease-19 (COVID-19): A pictorial review,” *Clin. Imaging*, vol. 64, no. April, pp. 35–42, 2020, doi: 10.1016/j.clinimag.2020.04.001.

[11] S. N. B, A. Amamra, and I. A. Amarouche, *A Comparative Study of Chest X-Ray Image Enhancement Techniques for Pneumonia Recognition A Comparative Study of Chest X-Ray Image Enhancement Techniques for*, no. December. Springer International Publishing, 2021. doi: 10.1007/978-3-030-69418-0.

[12] S. Teferi, D. Zewdneh, D. Admassie, B. Nigatu, and K. Kebeta, “Original article X-ray film reject rate analysis at eight selected government hospitals in Addis Ababa, Ethiopia, 2010,” *Orig. Artic. X-ray*, no. 6, 2018.

[13] D. A. Daniel Zewdeneh, Seife Tefire, “x ray reject analysis tikur anbsa.pdf.” 2022.

[14] M. Zewdu, E. Kadir, and M. Berhane, “Analysis and Economic Implication of X-Ray Film Reject in Diagnostic Radiology Department of Jimma University Specialized Hospital , Southwest Ethiopia,” *Ethiop J Heal. Sci.*, 2017.

[15] D. X. Morgan, “Deep Learning in Radiology,” *Acad. Radiol.*, vol. 25, no. 11, pp. 1472–1480, 1990, doi: 10.1016/j.acra.2018.02.018.

[16] D. Abin, S. D. Thepade, and S. Dhore, “An Empirical Study of Dehazing Techniques for Chest X-Ray in Early Detection of Pneumonia,” pp. 2–6, 2021.

[17] P. Rajadanuraks, S. Suranuntchai, S. Pechprasam, and T. Treebupachatsakul, “Performance Comparison for Different Neural Network Architectures for chest X-Ray Image Classification,” *2021 7th Int. Conf. Eng. Appl. Sci. Technol. ICEAST 2021 - Proc.*, pp. 49–53, 2021, doi: 10.1109/ICEAST52143.2021.9426289.

[18] N. Nafiiyah and E. Setyati, “Lung X-Ray Image Enhancement to Identify Pneumonia with CNN,” *3rd 2021 East Indones. Conf. Comput. Inf. Technol. EIConCIT 2021*, pp. 421–426, 2021, doi: 10.1109/EIConCIT50028.2021.9431856.

[19] A. Nagpal and G. Gabrani, “Python for Data Analytics, Scientific and Technical Applications,” *Proc. - 2019 Amity Int. Conf. Artif. Intell. AICAI 2019*, no. February 2019, pp. 140–145, 2019, doi: 10.1109/AICAI.2019.8701341.

[20] D. Rolon-m and M. Ross, “Introduction to Anaconda and Python: Installation and setup,” vol. 16, no. 5, pp. 3–11, 2020.

[21] S. Gholizadeh, “Top Popular Python Libraries in Research,” pp. 1–8.

[22] R. L. Miller and R. L. Miller, “An Introduction to Google Scholar,” 2019.

[23] J. Chabriais and B. Gibaud, “DICOM, le standard pour l’ imagerie médicale,” pp. 1–24, 2010.

[24] “Pneumonia: Causes, Symptoms, Diagnosis & Treatment.” https://my.clevelandclinic.org/health/diseases/4471-pneumonia (accessed Sep. 13, 2024).

[25] “American Lung Association - Search.” https://www.bing.com/search?q=American+Lung+Association&cvid=44642d80d1bf4d16a35dffbfb17785ce&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIGCAEQABhAMgYIAhAAGEAyBggDEAAYQDIGCAQQABhAMgYIBRAAGEAyBggGEAAYQDIGCAcQABhAMgYICBAAGEDSAQgxNDgxajBqOagCCLACAQ&FOR

M=ANAB01&PC=EDGEDB (accessed Sep. 15, 2024).

[26] V. Wiley and T. Lucas, “Computer Vision and Image Processing: A Paper Review,” vol. 2, no. 1, pp. 28–36, 2018, doi: 10.29099/ijair. v2i1.42.

[27] “] ‘Difference Between Computer Vision and Image Processing - Eastgate Software,’ 2024. https://eastgate-software.com/difference-between-computer-vision-and-image-processing/ (accessed Feb. 19, 2024). - Search.” https://www.bing.com/search?q=%5D+“Difference+Between+Computer+Vision nd+Image+Processing+-+Eastgate+Software%2C”+2024.+https%3A%2F%2Feastgate-software.com%2Fdifference-between-computer-vision-and-image-processing%2F+(accessed+Feb.+19%2C+2024).&cvid=3f439d7d93d74b258056028954841710&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBBzg0MGowajmoAgSwAgE&FORM=ANAB01&PC=EDGEDB (accessed Sep. 15, 2024).

[28] R. C. Gonzalez, R. E. Woods, and B. R. Masters, “Digital Image Processing, Third Edition,” *J. Biomed. Opt.*, vol. 14, no. 2, p. 029901, 2009, doi: 10.1117/1.3115362.

[29] Rafael_Gonzalez_Richard_Woods, *Digital image processing*. 2002.

[30] Z. Faisal, *Second Edition*, no. June. 2019.

[31] J. Stevens and R. Wasserman, “Bio medical imaging modalities,” vol. 19, no. I, pp. 3–25, 1995.

[32] B. Practices and W. Group, “Best Practices Working Group BCR’ s CDP,” no. June, 2008.

[33] Y. Wang, W. Song, G. Fortino, L. Z. Qi, W. Zhang, and A. Liotta, “An Experimental-Based Review of Image Enhancement and Image Restoration Methods for Underwater Imaging,” *IEEE Access*, vol. 7, pp. 140233–140251, 2019, doi:

10.1109/ACCESS.2019.2932130.

[34] A. Journal, “Understanding Color Models: A Review,” vol. 2, no. 3, pp. 265–275, 2012.

[35] D. P. D. Amato, “Using morphology in document image processing,” vol. 1606.

[36] K. Parvati, B. S. P. Rao, and M. M. Das, “Image Segmentation Using Gray-Scale Morphology and Marker-Controlled Watershed Transformation,” vol. 2008, 2008, doi: 10.1155/2008/384346.

[37] K. He, “Deep Residual Learning for Image Recognition,” *{kahe, v-xiangz, v-shren, jiansun}@microsoft.com Abstr.*, 2015.

[38] J. Kokorian, F. Buja, and W. M. Van Spengen, “In-Plane Displacement Detection with Picometer Accuracy on a Conventional Microscope,” *J. Microelectromechanical Syst.*, vol. 24, no. 3, pp. 618–625, 2015, doi: 10.1109/JMEMS.2014.2335153.

[39] I. H. Sarker, “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions,” *SN Comput. Sci.*, vol. 2, no. 6, pp. 1–20, 2021, doi: 10.1007/s42979-021-00815-1.

[40] A. Shrestha, “Review of Deep Learning Algorithms and Architectures,” *IEEE Access*, vol. 7, pp. 53040–53065, 2019, doi: 10.1109/ACCESS.2019.2912200.

[41] M. Shyu, S. Chen, and S. S. Iyengar, “A Survey on Deep Learning: Algorithms, Techniques,” vol. 51, no. 5, 2018, doi: 10.1145/3234150.

[42] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[43] J. Bradbury, S. Merity, C. Xiong, and R. Socher, “Quasi-recurrent neural networks,” *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, pp. 1–11, 2017.

[44] S. Shabanian, D. Arpit, A. Trischler, and Y. Bengio, “Variational Bi-LSTMs,” pp. 1–12, 2017, [Online]. Available: http://arxiv.org/abs/1711.05717

[45] Z. Wang, Q. She, and T. E. Ward, “Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy,” *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–41, 2021, doi: 10.1145/3439723.

[46] X. Amatriain, A. Sankar, J. Bing, P. K. Bodigutla, T. J. Hazen, and M. Kazi, “Transformer models: an introduction and catalog,” 2023, [Online]. Available: http://arxiv.org/abs/2302.07730

[47] D. Bank, N. Koenigstein, and R. Giryes, “Autoencoders,” *Mach. Learn. Data Sci. Handb. Data Min. Knowl. Discov. Handbook, Third Ed.*, pp. 353–374, 2023, doi: 10.1007/978-3-031-24628-9_16.

[48] W. Rawat and Z. Wang, “Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review,” no. June, 2017, doi: 10.1162/NECO.

[49] “Information and Communication Technology for Intelligent Systems.pdf.crdownload.”

[50] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, “Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach,” *Procedia Comput. Sci.*, vol. 132, pp. 679–688, 2018, doi: 10.1016/j.procs.2018.05.069.

[51] V. Naralasetti, R. K. Shaik, G. Katepalli, and J. D. Bodapati, “Treatment du Signal Deep Learning Models for Pneumonia Identification and Classification Based on X-

Ray Images," vol. 38, no. 3, pp. 903–909, 2021.

[52] A. Irfan, A. L. Adivishnu, A. Sze-to, T. Dehkharghanian, S. Rahnamayan, and H. R. Tizhoosh, "Classifying Pneumonia among Chest X-Rays Using Transfer Learning *," pp. 2186–2189, 2020.

[53] Ž. Knok, K. Pap, and M. Hrnčić, "Implementation of intelligent model for pneumonia detection," *Teh. Glas.*, vol. 13, no. 4, pp. 315–322, 2019, doi: 10.31803/tg-20191023102807.

[54] W. Khan, N. Zaki, and L. Ali, "Intelligent Pneumonia Identification From Chest X-Rays: A Systematic Literature Review," pp. 51747–51771, 2021, doi: 10.1109/ACCESS.2021.3069937.

[55] H. Sharma, J. S. Jain, P. Bansal, and S. Gupta, "Feature extraction and classification of chest X-ray images using CNN to detect pneumonia," *Proc. Conflu. 2020 - 10th Int. Conf. Cloud Comput. Data Sci. Eng.*, pp. 227–231, 2020, doi: 10.1109/Confluence47617.2020.9057809.

[56] B. Antin, J. Kravitz, and E. Martayan, "Detecting Pneumonia in Chest X-Rays with Supervised Learning," *Semant. org*, no. 46632050, pp. 1–5, 2017, [Online]. Available: http://cs229.stanford.edu/proj2017/final-reports/5231221.pdf

[57] A. A. Saraiva *et al.*, "Models of learning to classify X-ray images for the detection of pneumonia using neural networks," *BIOIMAGING 2019 - 6th Int. Conf. Bioimaging, Proceedings; Part 12th Int. Jt. Conf. Biomed. Eng. Syst. Technol. BIOSTEC 2019*, no. Biostec, pp. 76–83, 2019, doi: 10.5220/0007346600760083.

[58] K. Maharana, S. Mondal, and B. Nemade, "A review : Data pre-processing and data augmentation techniques," vol. 3, no. April, pp. 91–99, 2022, doi: 10.1016/j.gltp.2022.04.020.

[59] N. D. Jyotismita] Chaki, *A Beginner's Guide to Image Preprocessing Techniques*. 2018.

[60] C. Szegedy, "Deep Neural Networks for Object Detection," pp. 1–9, 2019.

[61] S. Jadhav and P. Dighe, "Image resizing for thumbnail images by using seam carving," *Int. Conf. Energy Syst. Appl. ICESA 2015*, no. Icesa, pp. 636–640, 2016, doi: 10.1109/ICESA.2015.7503427.

[62] S. K. A. Domini, "Computer Vision Metrics," 2014.

[63] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015.

[64] S. Rahman, "Image Enhancement in Spatial Domain: A Comprehensive Study," no. December, pp. 22–23, 2014.

[65] D. V. Malaya, K. Nath, and O. Prakash, *A Comprehensive Survey on Image Contrast Enhancement Techniques in Spatial Domain*, vol. 21, no. 1. Springer US, 2020. doi: 10.1007/s11220-020-00305-3.

[66] G. Srivastava, "Histogram Equalization: A Comparative Analysis & a Segmented Approach to Process Digital Images," no. 1, pp. 81–85, 2013.

[67] A. Assenza, M. Valle, and M. Verleysen, "A comparative study of various probability density estimation methods for data analysis," *Int. J. Comput. Intell. Syst.*, vol. 1, no. 2, pp. 188–201, 2008, doi: 10.1080/18756891.2008.9727616.

[68] "(Contrast Limited Adaptive Histogram Equalization - MATLAB & Simulink (mathworks.com) - Search," 2024. https://www.bing.com/search?pglt=41&q=(Contrast+Limited+Adaptive+Histogram+Equalization+-+MATLAB+%26+Simulink+(mathworks.com)&cvid=337b85041a4b401b972d20f

332964817&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCDExMjNqMGoxqAIAsAIA&FORM=ANNTA1&PC=EDGEDB (accessed Sep. 15, 2024).

[69] R. Kundu, R. Das, Z. W. Geem, G. T. Han, and R. Sarkar, "Pneumonia detection in chest X-ray images using an ensemble of deep learning models," *PLoS One*, vol. 16, no. 9 September, 2021, doi: 10.1371/journal.pone.0256630.

[70] A. Saxena, "An Introduction to Convolutional Neural Networks," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 12, pp. 943–947, 2022, doi: 10.22214/ijraset.2022.47789.

[71] T. Hattiya, K. Dittakan, and S. Musikasuwan, "Diabetic Retinopathy Detection using Convolutional Neural Network: A Comparative Study on Different Architectures," *Eng. Access*, vol. 7, no. 1, pp. 50–60, 2021, doi: 10.14456/mijet.2021.8.

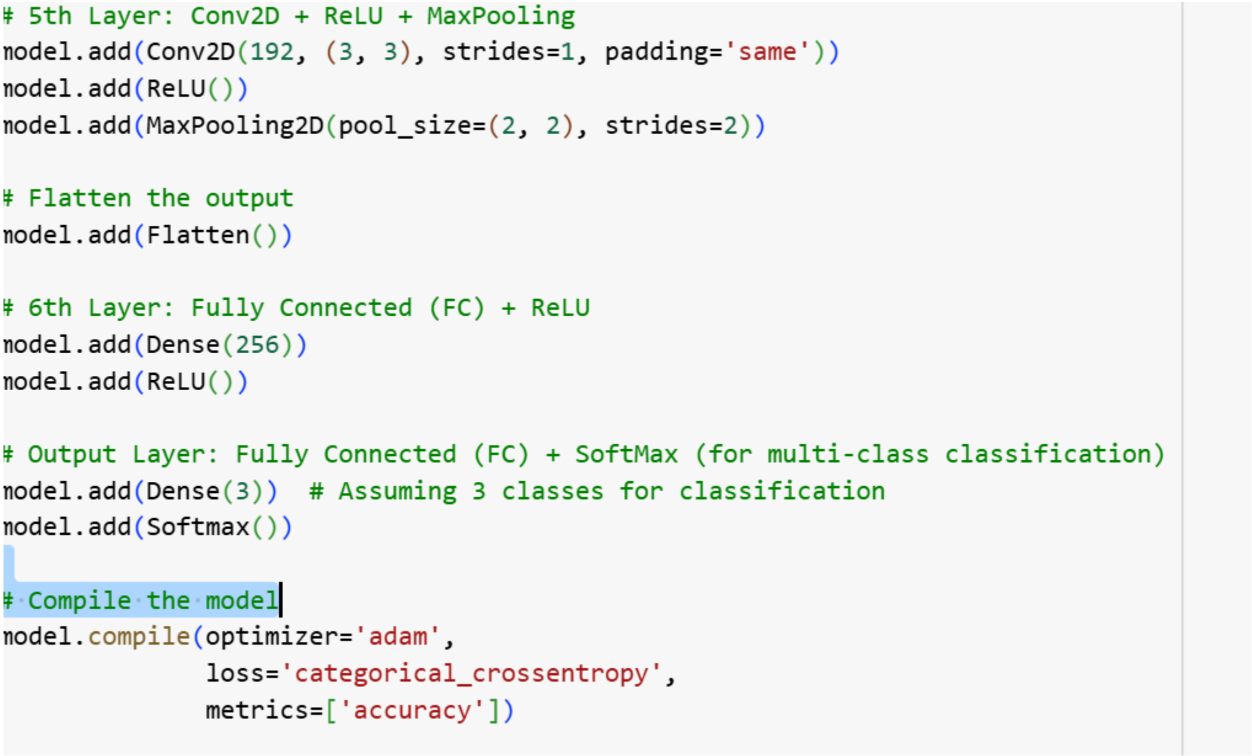
[72] B. Antin, J. Kravitz, and E. Martayan, "Detecting Pneumonia in Chest X-Rays with Supervised Learning," pp. 1–5.

[73] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.

APPENDICES

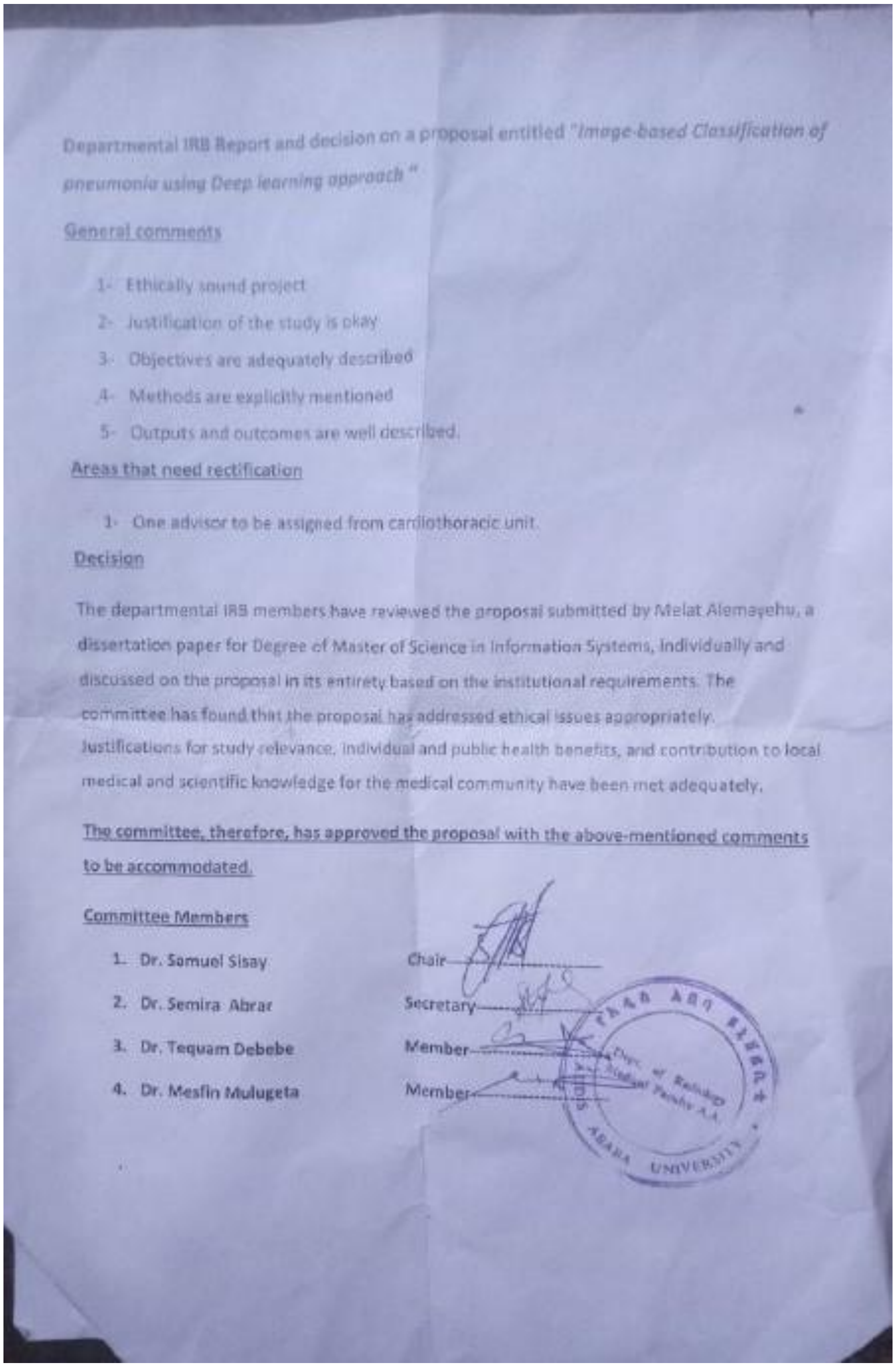
Appendices 1 Sample code for Powerful CNN-L5 model

```
input_shape = (224, 224, 3)

# 1st Layer: Conv2D + ReLU + MaxPooling
model.add(Conv2D(32, (5, 5), strides=2, padding='same', input_shape=input_shape))
model.add(ReLU())
model.add(MaxPooling2D(pool_size=(2, 2), strides=2))

# 2nd Layer: Conv2D + ReLU + MaxPooling
model.add(Conv2D(64, (5, 5), strides=2, padding='same'))
model.add(ReLU())
model.add(MaxPooling2D(pool_size=(2, 2), strides=2))

# 3rd Layer: Conv2D + ReLU + MaxPooling
model.add(Conv2D(96, (3, 3), strides=1, padding='same'))
model.add(ReLU())
model.add(MaxPooling2D(pool_size=(2, 2), strides=2))

# 4th Layer: Conv2D + ReLU + MaxPooling
model.add(Conv2D(128, (3, 3), strides=1, padding='same'))
model.add(ReLU())
model.add(MaxPooling2D(pool_size=(2, 2), strides=2))
```



```
# 5th Layer: Conv2D + ReLU + MaxPooling
model.add(Conv2D(192, (3, 3), strides=1, padding='same'))
model.add(ReLU())
model.add(MaxPooling2D(pool_size=(2, 2), strides=2))

# Flatten the output
model.add(Flatten())

# 6th Layer: Fully Connected (FC) + ReLU
model.add(Dense(256))
model.add(ReLU())

# Output Layer: Fully Connected (FC) + SoftMax (for multi-class classification)
model.add(Dense(3))  # Assuming 3 classes for classification
model.add(Softmax())

# Compile the model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Appendices 2 Ethical clearance for data collection



Departmental IRB Report and decision on a proposal entitled "*Image-based Classification of pneumonia using Deep learning approach* "

General comments

1- Ethically sound project
2- Justification of the study is okay
3- Objectives are adequately described
4- Methods are explicitly mentioned
5- Outputs and outcomes are well described.

Areas that need rectification

1- One advisor to be assigned from cardiothoracic unit.

Decision

The departmental IRB members have reviewed the proposal submitted by Melat Alemayehu, a dissertation paper for Degree of Master of Science in Information Systems, individually and discussed on the proposal in its entirety based on the institutional requirements. The committee has found that the proposal has addressed ethical issues appropriately. Justifications for study relevance, individual and public health benefits, and contribution to local medical and scientific knowledge for the medical community have been met adequately.

The committee, therefore, has approved the proposal with the above-mentioned comments to be accommodated.

Committee Members

1. **Dr. Samuel Sisay** Chair
2. **Dr. Semira Abrar** Secretary
3. **Dr. Tequam Debebe** Member
4. **Dr. Mesfin Mulugeta** Member

Dept. of Radiology

ADDIS ABABA UNIVERSITY