Chest X-Ray Image Classification using Pipeline System

BY

PINYADA RAJADANURAKS

SARAPORN SURANUNTCHAI

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING IN BIOMEDICAL ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2020

FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
PROJECT CERTIFICATE

Project Title	Chest X-Ray Image Classification using Pipeline system
Student Name	Miss Pinyada Rajadanuraks Student ID. 60011256 Miss Saraporn Suranuntchai Student ID. 60011267
Degree	Bachelor of Engineering in Biomedical Engineering

Project Advisor Signed: _____
(Asst. Prof. Dr. Treesukon Treebupachatsakul)

Project Co-Advisor Signed: _____
(Assoc. Prof. Dr. Suejit Pechprasarn)

Committee Signed: _____
(Asst. Prof. Dr. Suradej Tretriluxana)

Committee Signed: _____
(Dr. Wibool Piyawattanametha)

Committee Signed: _____
(Dr. Kasama Srirussamee)

Head of Department Signed: _____
(Asst. Prof. Dr. Sarinporn Visitsattapongse)

Project Title	Chest X-Ray Image Classification using Pipeline system
Student Name	Miss Pinyada Rajadanuraks Miss Saraporn Suranuntchai
Degree	Bachelor of Engineering in Biomedical Engineering
Project Advisor	Asst. Prof. Dr. Treesukon Treebupachatsakul,
Project Co-Advisor	Assoc. Prof. Dr. Suejit Pechprasarn
Academic Years	2020

ABSTRACT

Nowadays, to diagnose the disease or abnormally clinical conditions from chest X-Ray images, it can be done by reading chest X-Ray images by radiologists. The diagnosis from radiologists is considered as qualitative analysis which means that it is a process of collecting and analyzing data that is a nonnumerical. Moreover, it is a process that related to the experience to an individual. As a result, there may be some errors occurred while diagnosing such as misunderstanding about the basic of anatomy or some limitations which lead to a wrong diagnosis. Therefore, this research has a purpose for developing the software program which gathers Medical Image Processing and Artificial Intelligence, which are focusing on Deep Learning Neural Network, implementing on MATLAB program, in order to solve the problems by training and classifying on labelled and non-labelled chest X-Ray images between normal condition and 13 abnormal cases. In addition, it is stated that the results that got from Deep Learning Neural Network are considered as quantitative analysis which means that we can confirm or reject the hypothesis by testing causal relationship between variables. Furthermore, we applied several pre-trained architectures to our networks which, finally, we ended up with GoogLeNet architecture training with 3 classes in a time since it showed the highest percentage of accuracy when comparing with other architectures and, therefore, it is the network that appropriate with our image dataset. After ranked all possible networks in descending order, there were 38 networks could perform over

95 percent which is the threshold that we set to continue on pipeline system, the technique that used to validate our data and networks. We inputted 50 images for validating each class as well as created confusion matrix chart for gathering probability of each class. To sum up, the average of probability getting from validating the diseases by pipeline system is 93.02 percent.

ACKNOWLEDGEMENTS

Throughout the writing of this dissertation and for the success in our project, we have received a great deal of supports and assistance.

Firstly, we would like to express our deep and sincere gratitude to our supervisors which are Assistant Professor Dr. Treesukon Treebupachatsakul, Department of Biomedical Engineering, King Mongkut's Institute of Technology Ladkrabang and Association Professor Dr. Suejit Pechprasarn, College of Biomedical Engineering, Rangsit University, whose expertise were invaluable in formulating the research questions, methodology and especially for all the opportunities that we were given to further our research. Their insightful feedbacks pushed us to sharpen our thinking and brought our work to a higher level. Moreover, they also provided us with the tools that we needed to lead us to the right direction and the completion of our dissertation.

Secondly, we felt appreciate to all coding societies on the Internet especially Mathworks and Stackoverflow for sharing their implementing expedient information and some advice from comments to learn.

Last but not least, we would like to acknowledge our colleagues from King Mongkut's Institute of Technology Ladkrabang for their patient supports, practical suggestions, and wonderful collaboration throughout the duration of this research project. Finally, we would like to thank to our family for their encouragement, understanding and huge support as well as happy distractions to rest our mind outside of our research.

Pinyada Rajadanuraks

Saraporn Suranuntchai

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ ABBREVIATIONS

Symbols/Abbreviations	Terms
CNN/ ConvNet	Convolutional Neural Network
DNN	Deep Learning Neural Network
Conv layer	Convolutional layer
AI	Artificial Intelligence
ML	Machine Learning
ResNet	Residual Neural Network
IMDS	Image Data Store
NIH	National Institute of Health
imdsTraining	Trained images that storing in Image Data Store
imdsValidation	Validated images that storing in Image Data Store
GPU	Graphics Processing Unit
CXR	Chest X-Ray
FC layer	Fully Connected layer
ReLU	Rectified Linear Unit
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

CHAPTER 1
INTRODUCTION

1.1 Background and Motivation

Many diseases may not be visible to us directly through our eyes. Some diseases can be a silent threat that lurks into our body asymptomatic, or that we have pain within our body. Anywise, if we want to check those symptoms, X-Ray examination is one of the ways which can help us figure it out.

The X-Ray examination is where X-Rays are radiated to the chest or lungs, where images from the radiation are recorded on a film. The doctors will use to diagnose the integrity of the internal organs including the chest - heart abnormalities in size or shape or disorders of veins and arteries, lungs - pulmonary disorders, tuberculosis, pneumonia, tumors including lung cancer, and neighboring bone structures - fractures of the ribs and spine or natural distortion of the bone including the collarbone. Thus, making the X-Ray examination is a one of the popular examinations for the diagnosis of lung diseases since its procedure is quite simple and can be screened initially with a low amount of radiation used. On each examination, the patient will receive X-Ray film and results on the day of examination. In addition, the X-Ray film usually results in a large film in which the process of developing the X-Ray films is quite time consuming and expensive due to the specialized equipment and personnel. Nevertheless, these X-Ray films, nowadays, can be stored as a photo data on the computer or can be recorded on CD or external hard disk. As a result, tons of X-Ray films had been gathered and presented in terms of hospital-scale chest X-Ray databases. In addition, X-Ray examination is not only used at the laboratory in hospital, but also used in the vehicle such as a truck which known as a mobile X-Ray and check-up unit. It is a vehicle that containing an X-Ray machine and a photographic darkroom equipment inside to do X-Ray examination. This device, which also known as a “little Curie”, has been created by Marie Curie since World War I to help and save wounded soldiers in the war (Halliday, 2017). Furthermore, this machine is popularly used in Thailand, which are mostly used at places where are far from hospitals since it could be driven to any places to check-up lung’s health of patients easily and quickly. According to its benefits, it is

also used at school in the annual health screening program. This event is known as a health's check-up program which is checking about basic issues or diseases that may occur in children. This program is a program that covers measuring height, weight, blood pressure, eyesight and chest X-Ray which uses X-Ray machine to check whether it is a normal condition case or not. There are no problems from regular measurements but in chest X-Ray examination. One of the problems that we found from examined chest X-Ray is its lateness, which occurs because the radiologists must take some times to read film and diagnose each patient whether their lungs are at normal condition case or not.

In these recent years, Deep Learning Neural Network knowledge is being used and significantly developed in many fields especially in medical field since it can perform tasks that normally require human intelligence and most of it can demonstrate the results correctly. Therefore, we decided to combine DNN knowledge with image processing and image classification to our research project. According to the advantages of DNN and issues that we found from the examination of chest X-Ray, we foresee the possibility to implement this accumulated database together with DNN knowledge for the chest X-Ray image classification based on chest X-Ray images using convolutional neural networks development and training networks by various kinds of network architecture which are simple convolutional neural network, AlexNet architecture, ResNet architecture and GoogLeNet architecture, then, validating networks with the use of pipeline system technique. In result, we assure this work can screen and accurately identify the anomaly of the X-Ray images. Furthermore, this study is also provided to reduce the need for patients to rely on other medical examinations for urgent diagnosis.

1.2 Project Objectives

1.2.1 Developing a software program which gathering image processing, image classification and Artificial Intelligence via MATLAB program

1.2.2 Learning to apply Deep Learning Neural Network toolbox from MATLAB program into modified networks to predict the class of chest X-Ray images.

1.2.3 Training and classifying labelled chest X-Ray datasets between normal condition and abnormal conditions.

1.2.4 Developing a Deep Convolutional Neural Network capable of handling and increasing robustness when working with the nonlabelled dataset.

1.3 Scope of Project

This project focuses on the application of the Deep Learning Neural Network method to train network with a Convolutional Neural Network: CNN or ConvNet, a class of Deep Learning Neural Networks, of both the simple and applied ones. There are various applied architectures that combined with our network, which are AlexNet, ResNet-18, ResNet-50 and GoogLeNet in order to compare and observe its trends and results which one would be the desirable architecture that performs the highest accuracy. After getting the most appropriate architecture to use with our network and image dataset, focusing on training, and improving its performance to the best is what we aim for. Only one architecture efficiently works until the validation accuracy percentages of predicting the classes of chest X-Ray image reach to our set threshold which is over 95 percent. There are 13 diseases and normal condition that used in this research project which are:

1.3.1 Atelectasis contains 4,215 images
1.3.2 Cardiomegaly contains 1,093 images
1.3.3 Consolidation contains 1,310 images
1.3.4 Edema contains 628 images
1.3.5 Effusion contains 3,955 images
1.3.6 Emphysema contains 892 images
1.3.7 Fibrosis contains 782 images
1.3.8 Hernia contains 110 images
1.3.9 Mass contains 2,139 images
1.3.10 Nodule contains 2,705 images
1.3.11 Pleural thickening contains 1,126 images
1.3.12 Pneumonia contains 3,883 images
1.3.13 Pneumothorax contains 2,194 images
1.3.14 Normal contains 1,349 images

1.4 Expected Benefits

1.4.1 The project researchers get to study the Artificial Intelligence process of Deep Learning Neural Network part in image classification process.

1.4.2 The project researchers acquire practical skills of using Deep Learning Neural Network simulation on MATLAB to help in the image classification.

1.4.3 The classification simulation using the Convolutional Neural Network methodology can be applied to other complex workpieces.

1.4.4 The trained networks can predict untrained data of chest X-Ray images correctly.

1.4.5 The trained network to be acceptable and can be applied for the use in medical field as a differential diagnosis.

1.5 Project Schedule

No.	Plans	Months								
		Aug.	Sept.	Oct.	Nov.	Dec.	Jan.	Feb.	Mar.	Apr.
1	Project Plan									
1.1	Decide on a project topic									
1.2	Literature review									
1.3	Review examples from a main software and related sources.									
2	Project Proposal									
2.1	Propose purposes and a scope of study									
2.2	Design a workflow flowchart									
3	Experiment									
3.1	Material preparation									
3.2	Train network with a simple convolutional neural network									
3.3	Train network with AlexNet architecture									
3.4	Train network with ResNet-18 architecture									
3.5	Train network with ResNet-50 architecture									
3.6	Train network with GoogLeNet architecture									
4	Finalization									
4.1	Validate networks in pipeline system									
4.2	Summarise results									
5	Report									
5.1	Final Project Report Preparation				*First Half*			*Second Half*		

Table 1.1 Project Schedule

CHAPTER 2
RELATED THEORIES AND LITERATURE REVIEWS

2.1 Lungs

The lungs are a pair of spongy air-filled organs located on either side of the chest. The right lung is divided into 3 lobes; the upper, middle and lower lobes while the left lung is divided into 2 lobes; the upper and lower lobes. Because the heart takes up more room on the left side of the body, the left lung becomes smaller than the right one. A thin lining layer of pleura - the same thin tissue lines the inside of the chest cavity, surrounds the lungs as a lubricant to protect and help lungs smoothly slide back and forth against the chest wall from expansion and contraction during breathing. Below the lungs, a thin dome-shaped muscle of the diaphragm separates the chest from the abdomen.

These sponge-liked organs play a significant role in respiration gas exchange. The lungs have a system of tubes responsible for taking in oxygen during inhalation and releasing carbon dioxide during exhalation, lungs' main functions.

The windpipe or trachea conducts inhaled air from mouth or nose into lungs through its tubular branches of bronchi - branches into a left and right bronchus. The bronchi enter the lungs then divide into smaller branches of bronchioles. Finally, the bronchioles eventually end in clusters of microscopic air sacs known as alveoli. During gas exchange in the lungs between the alveoli and capillaries - a network of tiny blood vessels located in the walls of the alveoli, oxygen from the inhaled air is absorbed into the blood and passes into bloodstream where oxygen is carried to all the cells in our body. At the same time, carbon dioxide, a waste product of metabolism, travels across bloodstream into the alveoli where it can be exhaled and removed.

2.2 Lungs diseases

During a normal day, oxygen is needed for the cells in the human body to work and grow efficiently. Normally, it is necessary for lungs to take in oxygen from the air and properly deliver it through the bloodstream to all the cells in our body without any

defects. Nevertheless, people with lung disease have faced difficulty in breathing due to many factors causing lungs abnormality in both respiratory and pulmonary function.

Lung diseases are some of the most common medical conditions in the world. The term lung disease refers to disorders that affect the lungs and prevent them from functioning properly. As the majority duty of lungs is to exchange gas, breathing problems caused by lung disease may prevent the body from getting enough oxygen and containing too much carbon dioxide. Many disorders affecting the lungs commonly comprise of asthma, chronic obstructive pulmonary disease, influenza, pneumonia, tuberculosis, lung cancer, and many other breathing problems. As there are many kinds of lung diseases, numerous factors associated with the cause of effect also vary from bacteria, viral, or fungal infections to environmental factors. Anyhow, a lung disease can be hard to diagnose as the causes of all types of lung disease are not confirmed, and early signs of lung disease are also easy to overlook. Further, an early sign and symptoms can differ by the type of lung disease. Even when lung diseases do cause signs and symptoms, many people may often mistake them for other problems, thus delaying the diagnosis.

To confirm the diagnosis, some additional imaging tests and/or biopsies may be required to screen or rule out some of the potential causes of symptoms apart from general health and medical history information. Imaging tests are the tests commonly used both before and after a rapid diagnosis of lung diseases. This is due to several reasons including to seek for suspicious areas, to learn how far the disease might have spread, to help determine if treatment is properly working, to look for possible signs if the disease will be coming back after treatments, and many more. In terms of the procedure, imaging tests use X-Rays, magnetic fields, sound waves, or radioactive substances in creating pictures of the body inside known as radiograms. The radiogram and any abnormal findings will be further considered by the radiologist for screening. The main benefit of screening is to lower the chance of dying from diseases as it is meant to find abnormality in people who do not have symptoms of the disease. Notably, some lung diseases can be found by screening; however, screening may not find all, and not all the found diseases will be found early.

2.2.1 Atelectasis

Atelectasis (Staff C. H., 2020) is a condition in which the lungs are not fully dilated, caused by limited movement, shallow breathing, a small bronchial obstruction, narrowing of the alveoli in the lungs, a pressure from the pleural cavity, inability to cough up mucus efficiently and inadequate ventilation. Later, the blood that flows through the lungs is unable to exchange oxygen with the lungs. The oxygen level in the arterial blood thus decreases, leading to hypoxia and possibly unconsciousness.

Figure 2.1 Example of Atelectasis chest X-Ray image (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018)

2.2.2 Cardiomegaly

Cardiomegaly (Charunwikon, n.d.) is a condition in which the heart is enlarged than normal which can be diagnosed from lung x-ray images. There are many causes that can cause an enlarged heart, both from natural conditions such as pregnancy, and from heart disease itself, for example, an abnormal heart valves, abnormal heart muscle, irregular heartbeat including abnormal pericardium.

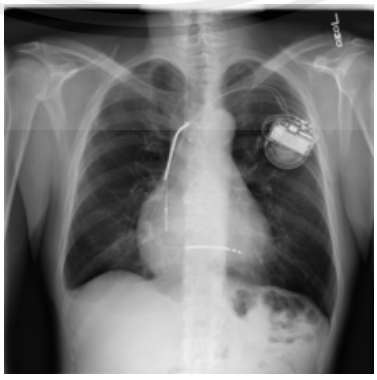

Figure 2.2 Example of Cardiomegaly chest X-Ray image (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018)

2.2.3 Effusion

Effusion (Staff P. , Pleural Effusion, n.d.) occurs when there is an excess of fluid in the area between the pleura and the membrane, with an increased amount of fluid pressing on the lungs. As a result, the lungs cannot fully expand. Normally, within the pleural space and the membrane, there is a small amount of fluid to prevent the lungs from rubbing against the chest cavity during expansion in the respiratory process. However, excessive amounts of this fluid can cause pleural effusion. This condition can further be categorized into two main types based on the reasons for this increase in fluid volume which are transudate, caused by increased intravascular pressure or low blood protein pressure causing fluid to leak into the pleural cavity, and exudate, caused by inflammation, cancer, clogged blood vessels or lymph vessels.

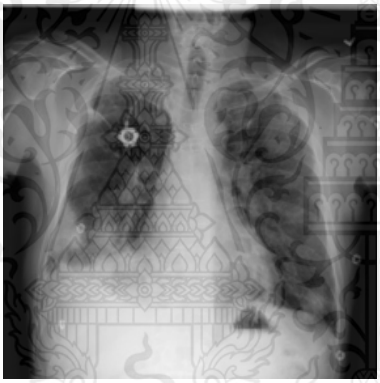

Figure 2.3 Example of Effusion chest X-Ray image (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018)

2.2.4 Mass

Mass (Eldridge, 2021) is defined as an abnormal spot or area in the lungs larger than 3 centimeters in size. Around 4-5% of masses found in the lungs further turn out to be lung cancer. Even if lung cancer is a mass or growth in the lung made up of cancer cells, not all masses in the lung are caused by cancer. Moreover, there are many different types of growths that can form in the lungs, and how they are diagnosed and treated depends on patient symptoms.

Figure 2.4 Example of Mass chest X-Ray image (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018)

2.2.5 Nodule

Nodule (MaryAnn De Pietro & Cameron White, 2019) is defined as a small growth on the lung in which the growth must be smaller than 3 centimeters to qualify as a nodule. Although the diagnosis of any growth in the lung can be frightening, a lung nodule does not always indicate lung cancer. Fewer than 5% of lung nodules end up being cancer meaning that it is not that severe.

Figure 2.5 Example of Nodule chest X-Ray image (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018)

2.2.6 Pneumonia

Pneumonia (Staff M. C., Pneumonia, 2020) is a disease caused by inflammation of the lungs in the terminal and respiratory bronchioles, alveoli, and interstitial tissues, which are caused by the infection. Common causes of pneumonia vary by age with the cause of viruses, bacteria, and atypical pathogen groups.

Figure 2.6 Example of Pneumonia chest X-Ray image (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018)

2.2.7 Pneumothorax

Pneumothorax (Staff W. , n.d.) is a condition where there is air in the pleural space. This can be divided into spontaneous pneumothorax which is a spontaneous pleural air leakage in patients without primary spontaneous pneumothorax or in patients with existing pulmonary pathology or secondary spontaneous pneumothorax, iatrogenic pneumothorax which refers to pleural air leakage occurs after a medical procedure such as lung biopsy, etc., and traumatic pneumothorax which refers to pleural air leakage occurs in an accidental patient.

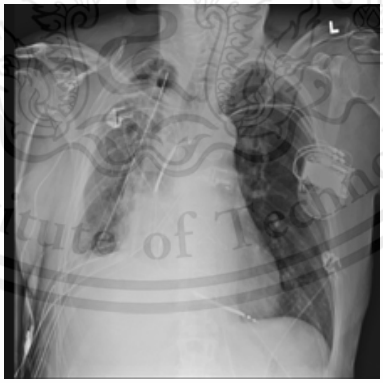

Figure 2.7 Example of Pneumothorax chest X-Ray image (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018)

2.2.8 Consolidation

Consolidation (Marcin & Nancy Moyer, 2018) occurs when the air that usually fills the small airways in the lungs is replaced with something else. Depending on the cause, the air may be replaced with a fluid, such as pus, blood, or water, or a solid, such

as stomach contents or cells. Consolidation almost always makes it difficult to breathe as air cannot get through the consolidation, so the lung cannot bring in fresh air and remove the air the body has used. Like lung consolidation, a pleural effusion, a collection of fluid in the space between chest wall and lungs, also looks like white areas against the darker air-filled lungs on chest X-ray. Since an effusion is a fluid in a relatively open space, it will usually move due to gravity when the patient changes position. Nevertheless, a lung consolidation may also be fluid, but since it is inside the lung, it cannot move when the patient changes position. This is one way to tell the difference between the two. Moreover, some of the causes of pleural effusions, such as congestive heart failure, pneumonia, and lung cancer, also cause lung consolidation. So, it is possible to have both at the same time.

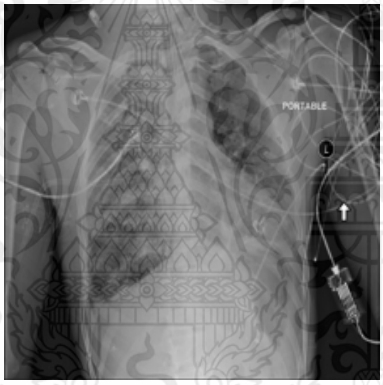

Figure 2.8 Example of Consolidation chest X-Ray image (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018)

2.2.9 Edema

Edema (Staff P. , Edema, n.d.) is a condition where large amounts of fluid or lymph build up within the body's connective tissue. This is caused by fluid leaking from the tiny blood vessels in the body causing the surrounding tissues to swell. However, it is not an inherited condition, and it can be the result of various illnesses, each of which may result in severe symptoms varying.

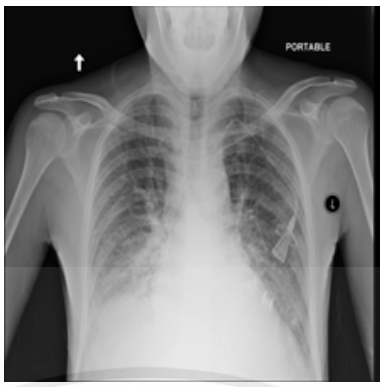

Figure 2.9 Example of Edema chest X-Ray image (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018)

2.2.10 Emphysema

Emphysema (Staff P. , Emphysema, n.d.) is a disease that belongs to the Chronic Obstructive Pulmonary Disease, COPD, category, caused by inflammation and rupture of the lung tissue in the alveoli, causing the lungs to contain many small alveoli, resembling a bunch of grapes, and combined with the air sacs that are adjacent to each other to form large alveolar alveoli, resulting in reduced oxygen exchange surface in the lungs or more residual air in the lungs.

Figure 2.10 Example of Emphysema chest X-Ray image (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018)

2.2.11 Fibrosis

Fibrosis (Staff M. C., Pulmonary fibrosis, 2018) is a lung disease that occurs when lung tissue becomes damaged and scarred. This thickened, stiff tissue makes it more difficult for the lungs to work properly. Pulmonary fibrosis scars and thickens the tissue around and between the air sacs, alveoli in the lungs which makes it more difficult

for oxygen to pass into the bloodstream. In most cases, doctors cannot pinpoint what is the cause of the problem. Nevertheless, when a cause cannot be found, the condition is termed idiopathic pulmonary fibrosis.

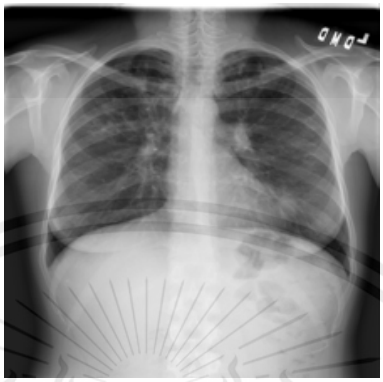

Figure 2.11 Example of Fibrosis chest X-Ray image (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018)

2.2.12 Pleural Thickening

Pleural thickening (Molinari & Charlevois, 2021) develops when scar tissue thickens the delicate membrane lining the lungs, the pleura. It can develop following asbestos exposure or other conditions, such as infection and may be a symptom of a more severe diagnosis such as malignant pleural mesothelioma. Depending on the cause, pleural thickening may form in different parts of the pleura. To distinguish between types, imaging scan is used to examine the patient's lungs and pleura: visceral pleura, which is the membrane directly covering the lung tissue, parietal pleura, which is the outer membrane of the lung attached to the chest wall, and pleural space, which is the space between the visceral and parietal pleura. The location of the impacted pleura and amount of thickening can help determine the type: apical pleural thickening which is a thickening of the top-most portions of the pleura and is benign unless the pleura has thickened more than 2 centimeters, focal pleural thickening which is a thickening confined to one or more specific areas of the pleura, diffuse pleural thickening which is a thickening of 50% or more of either the left or right pleura and may also be diagnosed in a patient with thickening of 25% or more of both pleura, mesothelioma-specific diffuse pleural thickening which is a thickening and scarring of the visceral pleura which can lead to the collapse of the pleural space, and nodular pleural thickening which is a thickening that creates raised bump-like nodules from scar tissue.

Figure 2.12 Example of Pleural thickening chest X-Ray image (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018)

2.2.13 Hernia

Hernia (Staff U. o., n.d.) refers to part of a lung pushing through a tear, or bulging through a weak spot, in the chest wall, neck passageway or diaphragm, in which the condition as a natural congenital occurrence is rare. Normally, as a person inhales, the lungs expand, thus this in turn expands the chest. If there is an opening or a soft spot in the chest wall, neck opening or diaphragm, it is possible that the lung as it expands will push through or cause a bulge at that point. Two out of three lung hernias involve the thoracic wall, and most of the rest involve the cervical area.

Figure 2.13 Example of Hernia chest X-Ray image (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018)

2.3 Chest X-Ray (CXR)

A chest X-Ray is usually the first and foremost common test used to diagnose diseases. It is a rapid and painless imaging test that uses electromagnetic waves to create pictures of the structures of the chest, lungs, heart, large arteries, ribs, and diaphragm.

This test can help diagnose and monitor various lung conditions, such as with the help of a radiologist to analyze and report to a doctor. Generally, an X-Ray is a fast, safe and painless test with no special preparation required. Nevertheless, similarly to all medical tests, it still provides some possible risks due to high energy rays despite being monitored and regulated to provide the minimum amount of radiation exposure needed to produce the image. Further, it cannot give a definitive diagnosis as it lacks the ability to distinguish between certain diseases and other conditions. Hence, most experts, including doctors and radiographers, reassure the benefits of having the test outweigh these risks.

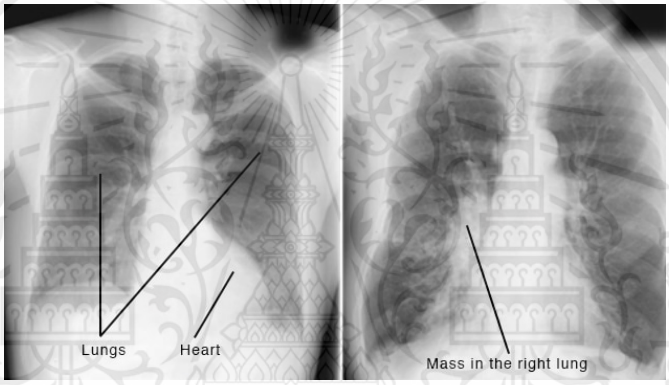

Figure 2.14 Example of chest X-Ray image

Left: Normal condition. Right: Mass in the right lung. (Staff M. C., Mayo Clinic, 2020)

2.4 Deep Learning Neural Network (DNN)

Deep Learning Neural Network (Litjens, et al., 2017) (LeCun, Yoshua Bengio, & Geoffrey Hinton, Deep learning, 2015) is a class of a broader family of machine learning, which can be abbreviated as ML, methods that uses multiple layers to progressively extract higher-level features from the raw input based on artificial neural networks with representation learning - can be supervised, semi-supervised or unsupervised. Deep Learning Neural Network mimics the working of human brain similar to the way information is processed in the human brain with the use of multiple layers of transformations of a deep architecture. Hence, it has become a widely used tool in various research domains where a large amount of data needs to be analyzed and human-like intelligence is required.

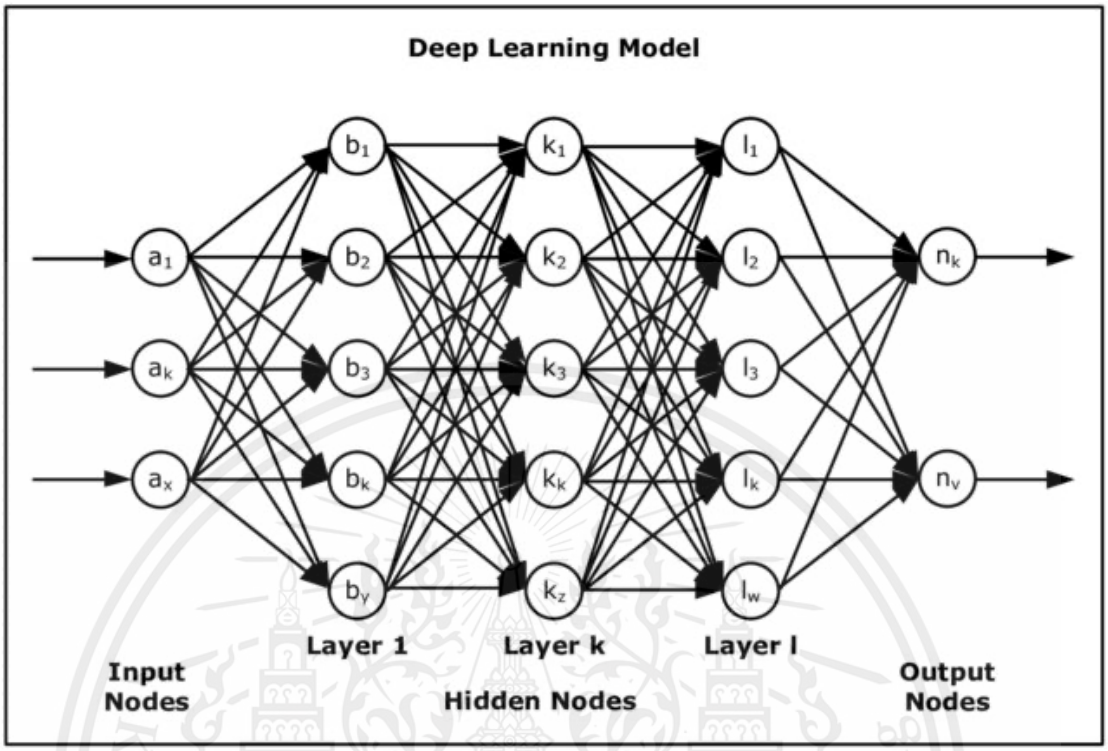

Figure 2.15 Diagram of Deep Learning Neural Network (Serrano, 2017)

In modern times, Deep Learning Neural Network models are mostly based on Artificial Intelligence Neural Networks, specifically, Convolutional Neural Networks. Further among all Deep Learning Neural Network techniques, Convolutional Neural Networks are trusted and actively used for the purpose of medical image analysis (Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, & Andrew H. Beck, 2016) (Gulshan, et al., 2016) (Esteva, et al., Dermatologist-level classification of skin cancer with deep neural networks, 2017).

2.5 Convolutional Neural Network (CNN or ConvNet)

Convolutional Neural Network (LeCun, et al., 1990) is a multi-layer neural network in the class of Deep Neural Networks in Deep Learning Neural Network. These convolutional networks are a specialized type of neural networks that used convolution in the place of general matrix multiplication within the layers. A typical Convolutional Neural Network mainly consists of an input and an output layer, as well as multiple hidden convolutional layers typically comprise of ReLu layer as an activation function and pooling layers, fully connected layers and batch normalization layers as additional convolutions.

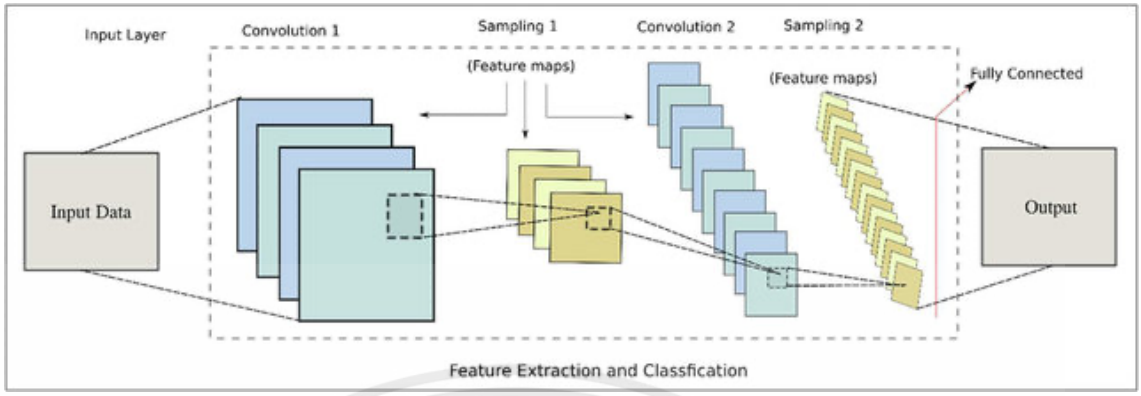

Figure 2.16 Diagram of CNN architecture (Jan, et al., 2019)

As a major advantage of automatic feature extraction ability and relatively little pre-processing, they have been applied for the use in various research fields including computer vision, bioinformatics, image and video recognition, recommender systems, image classification, medical image analysis, and many more, therefore, they have produced a better classification result compared to its predecessors and comparable to and, in some cases, even surpassed human performance. Nevertheless, the results can vary with the number of images used, the number of classes, and the choices of model.

The success of Convolutional Neural Network has captured attention in various fields and industries beyond academia. Nowadays, a Convolutional Neural Network is considered as one of the most widely used Deep Learning Neural Network techniques and has shown performance improvement in various Machine Learning applications. Although the concept of Convolutional Neural Network has existed for decades, training with multiple stacked layers is still quite challenging in many ways and was achieved only recently.

2.6 AlexNet architecture

AlexNet (Krizhevsky, Ilya Sutskever, & Geoffrey E. Hinton, 2017), the winner of the ILSVC 2012 competition, was proposed by Alex Krizhevsky and is based on convolutional neural networks. The architecture is comprised of eight layers in total, out of which the first 5 layers are convolutional layers and the last 3 layers are fully connected layers. The first two convolutional layers are connected to overlapping max-pooling layers to extract a maximum number of features. The third, fourth, and fifth convolutional layers are directly connected to the FC layers. All the outputs of the

convolutional and FC layers are connected to ReLu non-linear activation function. The final output layer is connected to a Softmax activation layer, which produces a distribution of 1,000 class labels.

The input dimensions of the network are 256*256*3 meaning that the input to AlexNet is an RGB, 3 channels, image of 256*256 pixels. To reduce overfitting during the training process, the network uses both data augmentation and dropout layers. The neurons that are dropped out do not contribute to the forward pass and do not participate in backpropagation. These layers are present in the first two FC layers.

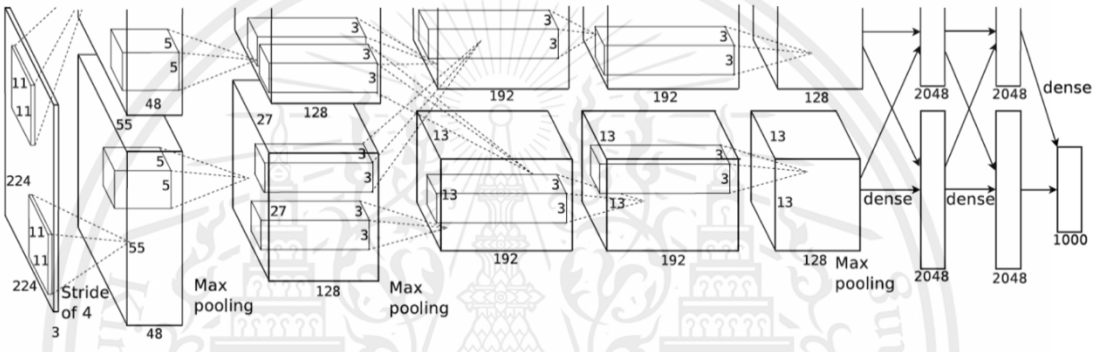

Figure 2.17 Diagram of AlexNet architecture (Krizhevsky, Ilya Sutskever, & Geoffrey E. Hinton, 2017)

2.7 ResNet architecture

Residual Neural Network (He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun, Deep Residual Learning for Image Recognition, 2015), the winner of the ILSVC 2015 competition, is a residual learning framework developed by Kaiming He et al. It was introduced as a novel architecture with skip connections. Such skip connections are also known as gated units or gated recurrent units. The key feature of skip connections allows training very deep neural networks by skipping some layers in between instead of waiting for the gradient to propagate back one layer at a time enables the gradient to reach those beginning nodes with greater magnitudes.

ResNet-18 or 18-layer-deep ResNet is the residual network with the baseline architectures the same as the plain nets, expect that a shortcut connection is added to each pair of 3*3 filters. ResNet reduces the top-1 error by 3.5%, resulting from the successfully reduced training error which verifies the effectiveness of residual learning on extremely deep systems.

ResNet-50 or 50-layer-deep ResNet is an improved version of the baseline 34-layer-deep ResNet. The network replaces each 2-layer block in the 34-layer net with a 3-layer bottleneck block, resulting in a 50-layer ResNet which He et al. empirically showed that this is more accurate than the 34-layer ones by considerable margins. Good performance of ResNet on image recognition and localization tasks showed that representational depth is of central importance for many visual recognition tasks.

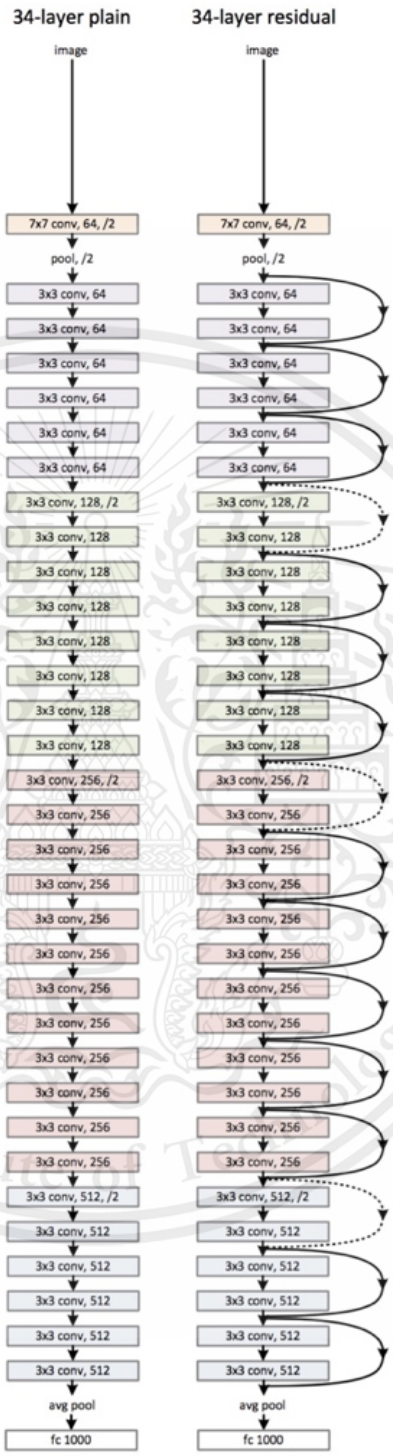

Figure 2.18 Diagram of ResNet architecture

Left: A plain network with 34 parameter layers. Right: A residual network with 34 parameter layers. (He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun, Deep Residual Learning for Image Recognition, 2015)

2.8 GoogLeNet architecture

GoogLeNet or InceptionV1 (Huang, Zhuang Liu, Laurens van der Maaten, & Kilian Q. Weinberger, 2016), the winner of the ILSVRC 2014 competition, was proposed by research at Google with the collaboration of various universities in 2014. So far there are three versions of Inception Networks, which are named Inception Version 1, 2, and 3. With the techniques of 1*1 convolutions in the middle of the architecture and global average pooling which enable it to create deeper architecture, this has provided a significant decrease in error rate as compared to previous winners AlexNet, the winner of ILSVRC 2012 and ZF-Net, the winner of ILSVRC 2013.

The GoogLeNet architecture is 22 layers deep, with 27 pooling layers included. There are 9 inception modules stacked linearly in total. The ends of the inception modules are connected to the global average pooling layer. As a network built with many deep layers might face the problem of overfitting, the GoogLeNet architecture was proposed with the idea of having filters with multiple sizes that can operate on the same level with Naive Inception Module. Since neural networks are time-consuming and expensive to train, the GoogLeNet architecture limits the number of input channels by adding an extra 1*1 convolution before the 3*3 and 5*5 convolutions to reduce the dimensions of the network and perform faster computations. With this idea, the GoogLeNet actually becomes wider rather than deeper.

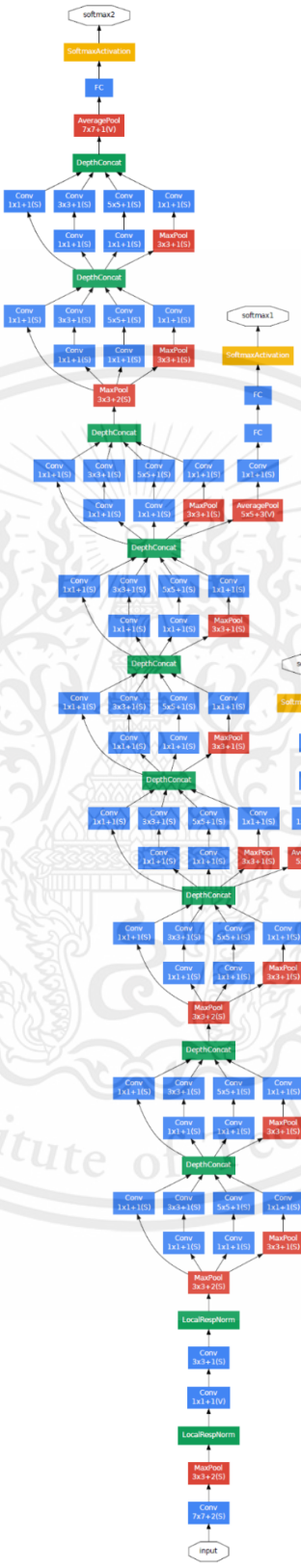

Figure 2.19 Diagram of GoogLeNet architecture (Huang, Zhuang Liu, Laurens van der Maaten, & Kilian Q. Weinberger, 2016)

2.9 Confusion Matrix

A confusion matrix (Narkhede, 2018) is a table used to evaluate the performance of a classification model for machine learning classification problem which is extremely useful for measuring recall, precision, specificity, accuracy and most importantly AUC-ROC curve. The table consists of 4 different combinations of predicted and actual values - each row represents an actual class while each column represents a predicted class, with 2 possible predicted classes of yes and no. In general, the most basic terms generated from this confusion matrix include true positives or TP - cases in which we predicted they have the disease and they actually do have one, true negatives or TN - cases in which we predicted they do not have the disease and they actually do not have one, false positives or FP - cases in which we predicted they have the disease but they actually do not have one, and false negatives or FN - cases in which we predicted they do not have the disease but they actually do have one.

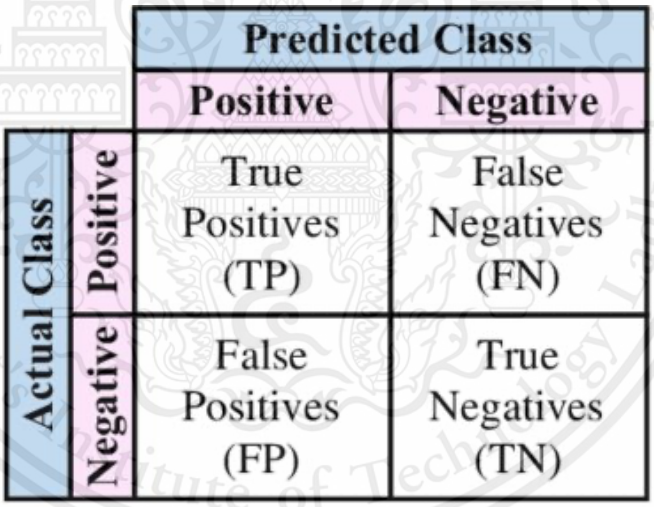

Figure 2.20 Diagram of Confusion matrix for binary classification (Sheng, Oscar Moroni Moosman, Borja del Pozo Cruz, Jesús Del Pozo-Cruz, & Rosa M. Alfonso-Rosa, 2020)

2.10 Literature Review

Research into the early detection of lung diseases within chest X-Ray is being done in many medical centers worldwide. The advancement in Deep Learning Neural Network methods has inspired medical imaging researchers to incorporate Deep Learning Neural Network in medical image analysis. Recent advances and studies have

shown that Deep Learning Neural Network algorithms are successfully used for medical image segmentation, computer aided diagnosis, disease detection and classification and medical image retrieval, where it has proven to give a high and robust performance in medical image analysis domain compared with other techniques applied in similar application areas. In medical image analysis, multiple convolutional neural network architectures reported in literature dealing with a variety of imaging modalities and tasks and has demonstrated superior performance in solving many difficult image classification problems even surpassed human capability in benchmarking tests (He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, 2015) (Esteva, et al., Dermatologist-level classification of skin cancer with deep neural networks, 2017) (Ting, Carol Yim-Lui Cheung, Gilbert Lim, & et al, 2017) (Lindsey, et al., 2018) (Peng, et al., 2019). The success of convolutional neural network in medical image analysis has strong biological plausible evidence support from a wide spectrum of literature recently available.

Looking at these successes of convolutional neural network in medical domain, Convolutional Network-based methods have exponentially gained popularity in vision systems as well as medical image analysis domain and seems to play a crucial role in the future medical image analysis development systems in recent years.

Traditionally, clinician experts or radiologists detect these abnormalities; however, this requires a lot of human effort and is time consuming. Therefore, the development of automated systems to help assist the radiologist and clinical practitioners in interpreting the medical images for the detection of abnormalities would be advantageous and thus is gaining importance (Yates, L.C. Yates, & H. Harvey, 2018) (Dunnmon, et al., 2018). Regular chest X-Rays have been studied for lung disease screening in order to find a disease in people whose symptoms have not shown, reducing the risk of dying. Generally, the Convolutional Neural Network-based architectures have been found wider success in dealing with medical image data compared to other Deep Learning Neural Network frameworks. As the architecture of convolutional neural network has been proved very successful in solving image classification problems. Further research-based on convolutional neural network have significantly shown the best performance improvement for many medical image

databases, the network has been adapted with a wide variety of medical imaging modalities used for the purpose of clinical prognosis and diagnosis serves as a second reader assists with radiologists in making decisions to increase diagnostic accuracy in clinical practice. In most cases the images look similar, making difficulty in differentiating between a healthy and non-healthy image. Therefore, underlying data with the convolutional neural network features, the learning mechanism of data is driven and learnt in an end to end causing the error signal obtained by the loss function is propagated back to improve the feature extraction part, hence, results in better representation. However, in some cases, the diagnosis cannot be confirmed radiologically and is fairly difficult even for experienced physicians due to the lack of clinical guidelines as well as the large quantity of radiological data.

In this project, we proposed a convolutional neural network with the pipeline system model for the classification of chest X-Ray images. The proposed method has been trained with the chest X-Ray dataset from the National Institute of Health Clinical Center and Mendeley Data, and further evaluate on Medical Imaging Databank of the Valencia Region chest X-Ray dataset. Furthermore, the results are confirmed for their superiority compared to previous works. To the best of our knowledge, this is the first time of the Deep Convolutional Neural Network that has been designed and validated in permutation using the pipeline technique for the characterization of chest X-Ray images for the diagnosis of lung diseases.

CHAPTER 3
METHODOLOGY

There are several methodologies that are used to find out which methods can bring out the best results from the Chest X-Ray image classification. These methodologies use common procedures but the different point is its network architecture inside each methodology. It can be divided into 5 main parts which are:

3.1 Overview workflow of chest X-Ray image classification via Deep Learning Neural Network (DNN)

3.2 Dataset and Data preparation

3.3 Convolutional Neural Networks (CNN, ConvNet)

3.4 Other optional applied commands

3.5 Pipeline system

3.1 Overview workflow of chest X-Ray image classification via DNN

According to the workflow shown in Figure 3.1, there are 8 main steps to do the image classification through Deep Learning Neural Network. Starting the process with importing image dataset, choosing an algorithm or network architectures to train networks, then, do the pre-processing on the image dataset which images will be stored in Image Datastore (IMDS) through a command in MATLAB program. To know the number of networks that are possible and needed to be trained, it can be computed from the binomial coefficient formula showing in Equation 3.1 which will return all possible combinations of choosing k out from n possibilities (Weisstein, 2021). After finished inputting a network architecture for training image dataset, there are 2 main tasks that needed to be done which are, firstly, do model training on each possible network and, secondly, do the image validation checking on labelled images that have already split and stored in imdsValidation. If the percentage of accuracy is not over the set threshold which is 95 percent, it means that the chosen algorithm may not be appropriate with this dataset which needed to change the algorithm and redo all of the processes since the beginning. The reason why we set the threshold at 95 percent is it is an acceptable number that is high enough to give us the confidence when do the prediction and validation. When getting the most appropriate trained models for the dataset which also

known as the predictive models, we are allowed to input some new data which have not been trained or validated before to each predictive model in order to get output.

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!} \tag{3.1}$$

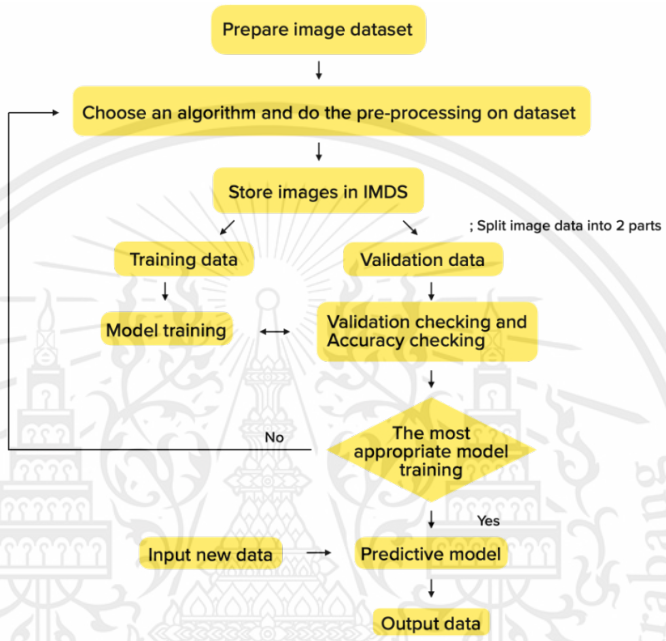

Figure 3.1 Overview workflow of image classification via DNN

3.2 Dataset and Data preparation

The image dataset used in this research were came from 3 sources which are National Institute of Health Clinical Center (Wang, Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers, 2018) abbreviated as NIH, Mendeley Data (Roberts, Surabhi Datta , & Kirk, 2020) and Medical Imaging Databank of the Valencia Region (Bustos, Antonio Pertusa, Jose-MariaSalinas, & Maria de la Iglesia-Vayá, 2020), abbreviated as BIMCV, which first two sources were for training and the last one was for validation.

The NIH is one of the clinical centers that provides one of the largest publicly available chest X-Ray image datasets, which also include advanced lung diseases and all common thoracic diseases, to scientific community. The reason why there are more than 100,000 of chest X-Ray image scans in the collection of NIH Clinical Center is they have foreseen that the bigger dataset that they provided for image classification, the better results they will get from the artificial intelligence implement since it can be improve its performance from the number of images in dataset.

The chest X-Ray image dataset provided by NIH Clinical Center consists of 14 main folders which composes of 13 diseases and 1 no finding condition which, in this research, we decided to classify the classes based on lungs diseases containing in this dataset. This dataset comes together with a data sheet that contains the information of each image within the dataset, such as the number of pictures and the name of the diseases that have already named it to relate to images in each folder. Since every disease is blended in the same folder, we have to write a simple program to sort image files and, also do the labelling on each disease before do the image classification via Deep Learning toolbox in MATLAB program.

For the other two datasets, there are also containing similar diseases to NIH dataset which, as mentioned above, the diseases containing in these two datasets will be sorted and categorized the classes based on diseases contained in NIH to make these 3 datasets sync and perform well on our chosen training models as much as possible.

To continue DNN process further, not only categorizing image files is required but also resizing image dimension to be equaled to each other represented in the form of matrix before training the networks is a must. However, the sizes of image are not the same depending on trained models. Furthermore, if there are different images size containing in the same folder or the sizes are not appropriate with chosen trained model, the process could not be done. The example of a simple program for sorting and resizing images is shown in Figure 3.2.

```
for ii=1:112120
    disp(ii);
    filename=char(table2array(DataEntry2017v2020(ii,1)));
    A=imresize((imread(['C:\Users\minnie\Documents\MATLAB\summer training\dataset\evt\' filename])),[512 512]);
if ~contains(['C:\Users\minnie\Documents\MATLAB\summer training\sorting\' char(table2array(DataEntry2017v2020(ii,1))) '\' filename],'<undefined>')
if isequal(char(table2array(DataEntry2017v2020(ii,2))),'Consolidation')

    imwrite(A,['C:\Users\minnie\Documents\MATLAB\summer training\sorting\classes\Consolidation\' filename]);
else
end
end
end
```

Figure 3.2 A simple program for sorting and resizing images

All of the diseases are thoracic diseases and lung diseases which can be enumerated as Atelectasis, Cardiomegaly, Consolidation, Edema, Effusion, Emphysema, Fibrosis, Hernia, Mass, Nodule, Pleural thickening, Pneumonia and Pneumothorax, plus one no finding condition.

Data augmentation process was also done on these 3 datasets, to be fair, which is a process that can prevent the network from overfitting and can memorize the details

of training images leading to gain higher accuracy on both training and validation trends after finished training networks. To clarify some more details about this process, there are 2 commands used in this process which are augmentedImageDatastore command and imageDataAugmenter command. For the augmentedImageDatastore, it is a command that used to avoid mistakenly overwrite images during the process as referenced above, on the other hand, the imageDataAugmenter is a command that used to configure a set of preprocessing options for generating batches of image from original images such as resizing, rotation and reflection. This process can be done by a software program as shown in Figure 3.3.

```
augimdsTrain = augmentedImageDatastore(imageSize, imdsTrain, 'ColorPreprocessing', 'gray2rgb');
augimdsValidation = augmentedImageDatastore(imageSize, imdsValidation, 'ColorPreprocessing', 'gray2rgb');
pixelRange = [-30 30];
scaleRange = [0.9 1.1];
imageAugmenter = imageDataAugmenter(...
    'RandXReflection', true, ...
    'RandXTranslation', pixelRange, ...
    'RandYTranslation', pixelRange, ...
    'RandXScale', scaleRange, ...
    'RandYScale', scaleRange);
```

Figure 3.3 A software program of Data augmentation process

3.3 Convolutional neural network (CNN or ConvNet)

The convolutional neural network is a network that fully connected feed forward neural networks and it is one of the networks that is appropriate to do image classification works because of its high accuracy when doing the prediction. Furthermore, it is also very effective in reducing the number of parameters without losing the quality of models which the number of parameters direct variation with the number of layers in the neural networks.

3.3.1 Simple convolutional neural network

The first network that used for training chest X-Ray image dataset is a simple convolution neural network. To have a fully connected layers which means that all of the neurons in the network are connected and made the output being generated, the main layers that network should contain are an input layer, hidden layers, which, as claimed, the larger the number of hidden layers that network contains, the higher performance that it can analyze and solve complex problems, and an output layer.

After finished the process of data preparation, there are several values that needed to be configured before training the network which are the dimension of input size, the ratio of images that will be split in IMDS, the size and number of filters of both convolutional and pooling layer, initial learning rate, validation frequency and number of epochs.

In this research, the dimension of input size that is appropriate with the network is 512*512*1, which 512 referred to the width and height of image and 1 referred to the depth of image or the number of channels which is a grayscale image.

The hidden layers can be added or removed to this network which the number of layers depend on how complex of that dataset and networks which the size of hidden layers, also known as kernel, should relate to the size of input matrix since there are multiple kernels stacked on top of each other. Each pixel on kernel will be multiplied with the corresponding pixels on input matrix by moving along the image in every pixel from left to right and top to bottom, then, sum up for the value in the corresponding position in output matrix. In addition, the number of parameters produced during the process can be reduced by pooling layers, however, the same as hidden layers, the number of layers may vary depending on their associated weights and biases. Therefore, we can know these values from trial-and-error method, training until get the most appropriate values for the network. Finally, the last two variables that must also be concerned are initial learning rate and validation frequency which initial learning rate is the values that used to control how quickly the model responded to the problems and also estimate the errors regarding to the amount of updated weight in each epoch and the validation frequency is a value that defines the number of iterations between evaluation of validation metrics. The example of configuring layers and training options is shown in Figure 3.4.

```
inputSize = [512 512 1];
numClasses = 2;

layers = [
         imageInputLayer(inputSize)

         convolution2dLayer(5,8, 'Padding', 'same')
         batchNormalizationLayer
         reluLayer

         maxPooling2dLayer(5,'Stride',2)

         convolution2dLayer(5,16,'Padding','same')
         batchNormalizationLayer
         reluLayer

         maxPooling2dLayer(5,'Stride',2)

         convolution2dLayer(5,32,'Padding','same')
         batchNormalizationLayer
         reluLayer

         maxPooling2dLayer(5,'Stride',2)

         convolution2dLayer(5,64,'Padding','same')
         batchNormalizationLayer
         reluLayer

         maxPooling2dLayer(5,'Stride',2)

         convolution2dLayer(5,128,'Padding','same')
         batchNormalizationLayer
         reluLayer

         fullyConnectedLayer(numClasses)
         softmaxLayer
         classificationLayer()];

options = trainingOptions('sgdm', ...
    'InitialLearnRate',0.001, ...
    'MaxEpochs',20, ...
    'Shuffle','every-epoch', ...
    'ValidationData',imdsValidation, ...
    'ValidationFrequency',30, ...
    'Verbose',false, ...
    'Plots','training-progress');
```

Figure 3.4 The example of configuring layers and training options in simple CNN

3.3.2 Applied AlexNet architecture to convolutional neural network

The second network architecture that is used with chest X-Ray image dataset is AlexNet architecture which is one of architectures that used GPU to boost its performance. There are various layers containing in this AlexNet architecture which are convolutional layers used together with ReLu, one of non-linear activation functions, max pooling layers, normalization layers, fully connected layers and Softmax layer.

Furthermore, one of the different points between simple convolutional neural network and AlexNet architecture is its input size which, the dimension of input size that suitable for training this model is 227*227*3. Since the image size that we used in previous method is 512*512*1, we have to downsize the input size by a simple program via MATLAB program as shown in Figure 3.5.

```
digitDatasetPath = fullfile('/Users/minnie/Documents/MATLAB/summer training/227folder/Atelectasis');
imds = imageDatastore(digitDatasetPath, ...
    'IncludeSubfolders',true,'LabelSource','foldernames');

img2=zeros(227,227,3); %
for i=1:length(imds.Labels)
    img=readimage(imds,i);
    % disp(i)
    img1 = imresize(img,[227 227]);
    if size(img1,3)==3
        img1 = rgb2gray(img1);
    end
    img2 = cat(3,img1,img1,img1);
    imwrite(img2,cell2mat(imds.Files(i)))
end
```

Figure 3.5 A simple program for resizing images for AlexNet architecture

The way to run the process is similar to simple convolutional neural network, however, another uncommon point is we do not have to customize the number of hidden layers as we did in previous network as the AlexNet architecture is a pre-trained network. This model has already combined every essential layer into the network meaning that the only things that needed to be set are variables in training options such as mini batch size, epochs, and initial learning rate. We can define one variable for storing AlexNet architecture inside, then, starting the process of training, predicting the diseases and calculating the validation accuracy by trainNetwork, classify and sum commands as shown respectively in Figure 3.6.

```
net = alexnet;
layersTransfer = net.Layers(1:end-3);
% numClasses = numel(categories(imdsTrain.Labels));
numClasses = 3;
layers = [
    layersTransfer
    fullyConnectedLayer(numClasses,'WeightLearnRateFactor',20,'BiasLearnRateFactor',20)
    softmaxLayer
    classificationLayer];
netTransfer = trainNetwork(imdsTrain,layers,options);
[YPred,scores] = classify(netTransfer,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
```

Figure 3.6 AlexNet architecture software design

3.3.3 Applied ResNet-18 architecture to convolutional neural network

The third network architecture that is applied to convolutional neural network is ResNet architecture which is one of the architectures that is created to improve the performance of AlexNet architecture. There are several types of ResNet architecture, however, this research will mainly focus on ResNet-18 architecture and ResNet-50 architecture.

We, firstly, trained networks with ResNet-18 architecture which is an architecture that contains 18 layer-deep and also one of the pre-trained networks as same as AlexNet architecture. Nevertheless, the dimension of input size that is suitable to run the process with ResNet architecture is 224*224*3. Hence, the first task that has to be done is downsizing images from the dimension of 227*227*3 to 224*224*3 by a simple program as shown in Figure 3.7.

```
digitDatasetPath = fullfile('/Users/minnie/Documents/MATLAB/summer training/224folder/Atelectasis');
imds = imageDatastore(digitDatasetPath, ...
    'IncludeSubfolders',true,'LabelSource','foldernames');

img2=zeros(224,224,3);
for i=1:length(imds.Labels)
    img=readimage(imds,i);
    % disp(i)
    img1 = imresize(img,[224 224]);
    if size(img1,3)==3
        img1 = rgb2gray(img1);
    end
    img2 = cat(3,img1,img1,img1);
    imwrite(img2,cell2mat(imds.Files(i)))
end
```

Figure 3.7 A simple program for resizing images for ResNet architecture

Not only the procedures of ResNet architecture are similar to AlexNet architecture, but also its network types which is also one of the pre-trained networks, thereby, configure one variable to recall this architecture, then, apply to the input matrix and start training the networks via MATLAB program as shown in Figure 3.8. With this architecture, the variables that needed to be defined are also the same as AlexNet architecture which are mini batch size, number of epochs, and initial learning rate. However, these values might not be the same depending on how complex of the network.

```
net = resnet18;
%analyzeNetwork(net)
numClasses = numel(categories(imdsTrain.Labels));
lgraph = layerGraph(net);

newFCLayer = fullyConnectedLayer(numClasses,'Name','new_fc','WeightLearnRateFactor',10,'BiasLearnRateFactor',10);
lgraph = replaceLayer(lgraph,'fc1000',newFCLayer);
newClassLayer = classificationLayer('Name','new_classoutput');
lgraph = replaceLayer(lgraph,'ClassificationLayer_predictions',newClassLayer);


inputSize = net.Layers(1).InputSize;
inputSize= [224 224];
augimdsTrain = augmentedImageDatastore(inputSize,imdsTrain);
augimdsValidation = augmentedImageDatastore(inputSize,imdsValidation);
```

```
trainedNet = trainNetwork(imdsTrain,lgraph,options);

[YPred,probs] = classify(trainedNet,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
%accuracy = mean(YPred == imdsValidation.Labels)
```

Figure 3.8 ResNet-18 architecture software design

3.3.4 Applied ResNet-50 architecture to convolutional neural network

The fourth network architecture that is used with chest X-Ray image dataset is also one of the kinds of ResNet architecture which is ResNet-50 architecture. This architecture can perform higher performance than ResNet-18 architecture in terms of both percentage of accuracy and operations due to its larger number of layers containing inside. According to the same kind of network architecture, the dimension of input size is the same which is 224*224*3. In addition, the overall of the process is the same as the previous method which the only thing that needed to be changed is the kind of pre-trained network as shown in Figure 3.9.

```
net = resnet50;
lgraph = layerGraph(net);
resnet50
  newLearnableLayer = fullyConnectedLayer(numClasses, ...
      'Name','new_fc', ...
      'WeightLearnRateFactor',20, ...
      'BiasLearnRateFactor',20);
  lgraph = replaceLayer(lgraph,'fc1000',newLearnableLayer);
  newsoftmaxLayer = softmaxLayer('Name','new_softmax');
  lgraph = replaceLayer(lgraph,'fc1000_softmax',newsoftmaxLayer);
  newClassLayer = classificationLayer('Name','new_classoutput');
  lgraph = replaceLayer(lgraph,'ClassificationLayer_fc1000',newClassLayer);
augimdsTrain = augmentedImageDatastore(inputSize,imdsTrain);
augimdsValidation = augmentedImageDatastore(inputSize,imdsValidation);
trainedNet = trainNetwork(imdsTrain,lgraph,options);
[YPred,probs] = classify(trainedNet,imdsValidation);

[label,score] = classify(trainedNet,imdsValidation);

YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)

testLabels = imdsValidation.Labels; % actual label
```

Figure 3.9 ResNet-50 architecture software design

3.3.5 Applied GoogLeNet architecture to convolutional neural network

The last network architecture that is used with chest X-Ray image dataset is GoogLeNet architecture which this architecture contains 27-layer deep including pooling layers. Even it contains fewer layers than ResNet-50 architecture, it can perform better than previous architecture in this case as this architecture is known as one of the architectures that suitable for image classification tasks. To apply GoogLeNet architecture into the network, it can be done by loading pre-trained network architecture to the software program as shown in Figure 3.10. Then, start training the

architecture, predicting and validating the images that inputted for the classification. Simply like this since the overall of the process is alike to the previous methods that have done before.

```
net = googlenet;
inputSize = net.Layers(1).InputSize
lgraph = layerGraph(net);
numClasses = numel(categories);
disp(categories(randperm(numClasses, 3)))
net = trainNetwork(augimdsTrain, lgraph, options);
[YPred, PredScores] = classify(net, augimdsValidation);

YValidation = imdsValidation.Labels;
accuracy = mean(YPred == YValidation)
```

Figure 3.10 GoogLeNet architecture software design

3.4 Other optional applied commands

3.4.1 Confusion matrix command

The only values that we got after finished training process is the percentage of validation accuracy which counted as one of the obstacles that we encountered. The reason why it is counted as an obstacle is we could not know other gained values such as the probabilities of each class in that network in order to know which class that the image is likely to belong to. To solve this kind of problem, applying confusion matrix to the networks is a key, shown in Figure 3.11, which its advantages are not only helping us visualize the values of the probability easier but also helping us to observe and evaluate the performance of classifiers.

There are 2 types of result merged and brought up in the same matrix which are predicted class and actual class computed from the probabilities gained in each network. Normally, the results that we got directly from confusion matrix are shown in command window in the form of matrix as shown in Figure 3.12, however, to visualize those values easier which come together with its labelled classes, it can be done by showing the values in the form of chart shown in Figure 3.13. Noted that, the values shown in Figure 3.12 and Figure 3.13 are the example of results which came from one of the networks that we trained which the values from the predicted class indicated in X-axis and, on the other hand, the values from actual class aligned in Y-axis.

```
confMat = confusionmat(YValidation, YPred);
confMat = bsxfun(@rdivide, confMat, sum(confMat, 2))
mean(diag(confMat));
confusionchart(imdsValidation.Labels, YPred, ...
    'Normalization', "row-normalized");
```

Figure 3.11 Applying confusion matrix command to the network

```
confMat =

    0.4839         0    0.5161
         0    1.0000         0
    0.0914         0    0.9086
```

Figure 3.12 The example of the matrix generated from confusion matrix

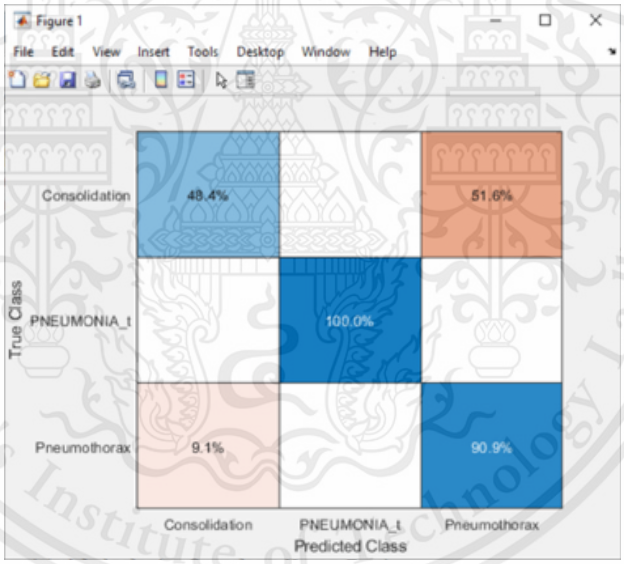

Figure 3.13 The example of chart generated from confusion matrix

3.4.2 Sort and scores command

With these commands, they are literal meaning as its name which, firstly, sort command is used for sorting the classes containing in trained network and, secondly, scores command is used for doing the scoring.

Since there are various classes trained in one network, some errors may occur during the prediction and validation such as the prediction mismatches with the labelled data. Due to this reason, applying these 2 commands are one of the ways to track and inspect on each class contained in the network.

First of all, we classify the untrained image from trained network, then afterward, keep the results of prediction and probability values in variables which are label and scores, respectively. Next, as mentioned above about the usability of sort command, we used this command together with scores, variable that we kept results from the prediction, which it can sort the probabilities values in either in ascending or descending order depending on what we desire. After finished sorting, we will rank and match those scores with its class name, then, show in horizontal bar graph along with the vertical line at 0.90 to set the minimum threshold that would be acceptable for doing further steps. Furthermore, we also plot the image that chosen to make a prediction on the left side of the graph and also show the percentage of predictions of each disease computed from scores above the image. The software design and the example of results are shown in Figure 3.14 and Figure 3.15, respectively.

```
[~, idx] = sort(scores, 'descend');
idx = idx(3:-1:1);
classes = net.Layers(end).Classes;
classNamesTop = string(categories(idx));
scoreTop = scores(idx);

h = figure(2);
h.Position(3) = 2*h.Position(3);
ax1 = subplot(121); %row col #pic
ax2 = subplot(122);

image(ax1, randimg);

title(ax1, {['Predicted disease: ' char(predictedLabel)] ...
    ['Actual disease: ' char(actualLabel)] ...
    ['Percent of each prediction: ' num2str(100*scores)]});

b = barh(ax2, scoreTop);
title(ax2, 'Top 3 Predictions')
xlim(ax2, [0 1])
xticks(ax2, 'auto')
xticklabels(ax2, 'auto')
xlabel(ax2, 'Probability')

yticks(ax2, 'manual')
yticklabels(ax2, classNamesTop)
ylabel(ax2,'Diseases')
ax2.YAxisLocation = 'left';

threshold = xline(ax2, 0.90, 'Color','red','LineStyle','--');
```

Figure 3.14 A software design to generate bar chart including a threshold marker

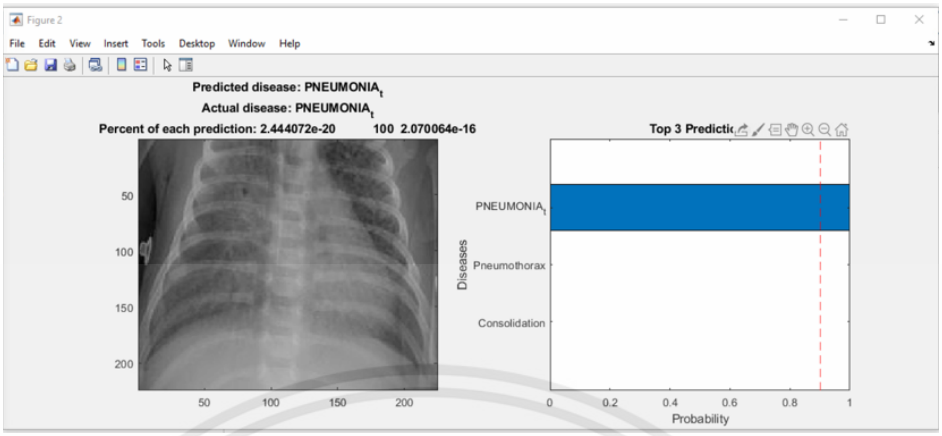

Figure 3.15 Bar graph representing the top predictions including its percent predictions

3.5 Pipeline system

Pipeline system is a method that will be done after finished training all of the possible networks that calculated from binomial theorem in data preparation step. Firstly, rank all of the saved networks in descending order in order to observe its validation trends easily and select only networks that gave the validation accuracy higher than set threshold which, moreover, these networks must contain all of 13 diseases, plus normal case. Then, count number of times of each disease in chosen networks which, to be fair, the number of times should be equal or be the nearest to each other as much as possible. Inputting untrained images to the software program and applying activation command pointing to the Softmax layer of each network are the next steps to do to get probabilities of each disease. The probabilities of each disease will be added together and divided by the number of times to calculate the average of each disease. The overall of the process for each chosen network is shown in Figure 3.16.

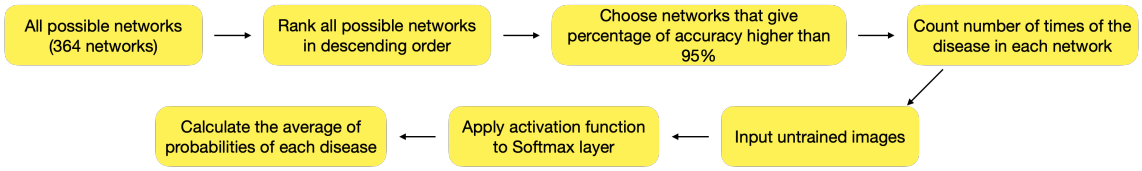

Figure 3.16 The overall of pipeline system for each chosen network

To give more details how the pipeline system works, let's assume the list of classes and the list of networks to be disease A, disease B until reached to disease N,

the last trained disease, and network 1, network 2 until reached to network 38, the last saved network, to visualize the procedure easily.

As we mentioned that the chosen networks must contain all of 14 classes, for instant, firstly, disease A, disease B and disease C were contained in network 1, secondly, disease D, disease E and disease F were contained in network 2 and, lastly, disease L, disease M and disease N were contained in network 38, there are 14 folders which each folder contained only 1 disease in order to validate the pure illness and get the most accurate validation results.

The validation process will be done in for loop which, in this case, there are 38 loops to be run due to 38 networks contained in the system. Furthermore, there are 14 times to input folder of diseases to validate in the system. Start running the first loop with inputting disease A to the system, then, inputting disease B when the first loop ended which we will keep continue inputting other diseases until reached to the disease N which is the last disease to be validated as shown in Figure 3.17. Furthermore, one of the values that will be gotten from every time of inputting a new folder, disease A to disease N, is the probability of 14 classes which will be gathered in 14*1 table. Each cell shows the probability of each disease which the predicted class corresponds to rows, whereas the actual class corresponds to a column. In addition, the total must be equal to or be nearest to 1.00 as much as possible. After finished validated networks with 14 folders, we, therefore, will get the 14*14 table containing the probability of 14 classes which also known as a confusion matrix.

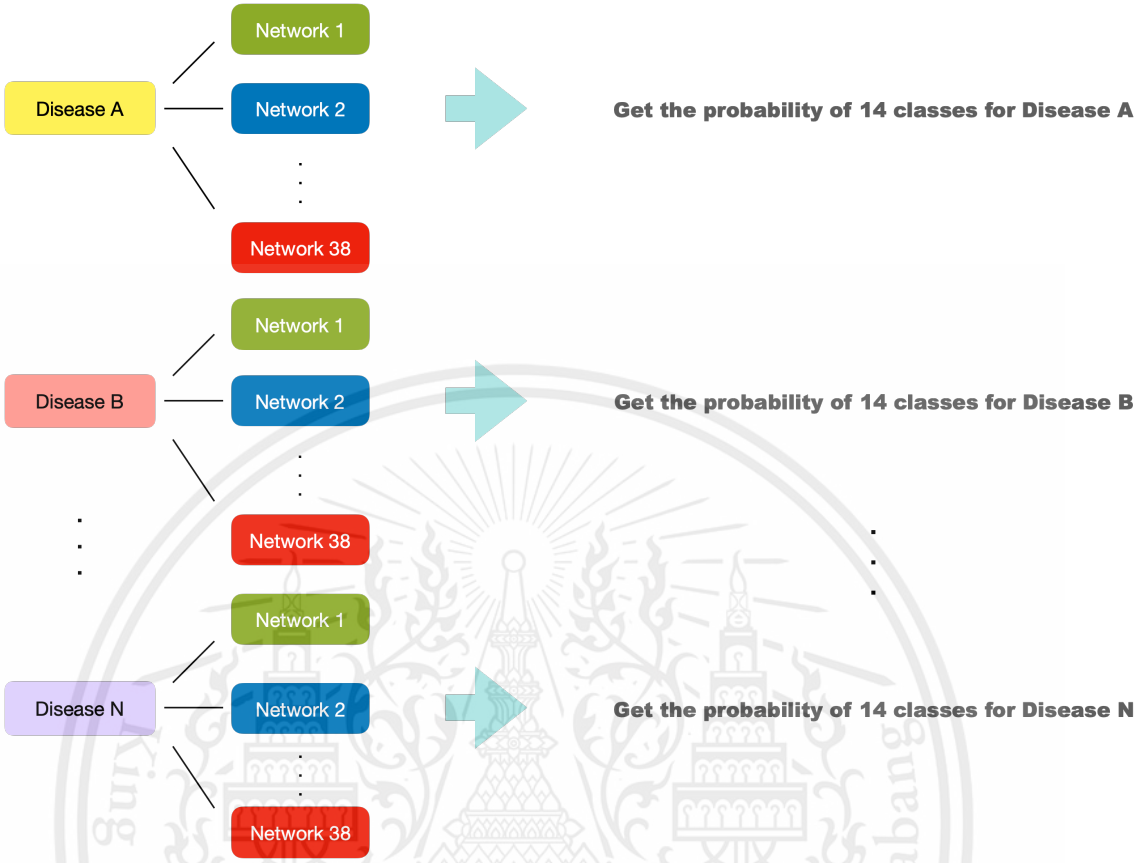

Figure 3.17 Diagram of pipeline system procedure

CHAPTER 4
EXPERIMENTAL RESULTS AND DISCUSSION

In chapter 3, we described the software design and the implementations that are being used to figure out the results from each software system and architecture. All of the collected testing results will be provided in this chapter. The results containing in this chapter can be split and summarized into 6 main parts which are from simple convolutional neural network, AlexNet architecture, ResNet-18 architecture, ResNet-50 architecture, GoogLeNet architecture and pipeline system in order to be discussed and illustrated the details easily. These subtopics are organized as follows:

4.1 Results from simple convolutional neural network
4.2 Results from AlexNet architecture
4.3 Results from ResNet-18 architecture
4.4 Results from ResNet-50 architecture
4.5 Results from GoogLeNet architecture
4.6 Results from pipeline system

4.1 Results from simple convolutional neural network

The first results that will be mentioned is the results from simple convolutional neural network. In this neural network, the input images were resized to 512*512 pixels. Since the simple convolutional neural network is not a pre-trained network, it can be easily customized by adding or removing any filtering layers and creating more feature maps. We started training a network with its default settings, which contain 1 input layer, 2 hidden layers, 2 max pooling layers, 2 fully connected layers and 1 output layer to observe its accuracy and loss trends on both training and validation.

For the hidden layers, also known as a convolutional layer, containing in this network were started training a network from 2 layers and adding a layer up in every single training in order to observe the maximum number of layers that make the percentage of accuracy of network becomes stable.

Before training a network, according to data preparation step, we split the images into 2 parts which were storing in imdsTrain and imdsValidation in the ratio of 0.75 meaning that there are 75 percent of images storing in IMDS put into imdsTrain

part and the rest put into another. To reduce the number of parameters of the network and bring out the best output, it can be done by max pooling layer, a layer that used to summarize the most activated presence of features in the patches of feature map, which the size that used in this research is 5*5 in both convolutional layer and max pooling layer, yet, in max pooling layer, it comes together with a stride of 2. In addition, not only the ratio of number of images and the filter size were set but also the initial learning rate and validation frequency which these values are set as 0.001 and 30, respectively.

At the beginning, there are 13 classes that chosen to be trained in a time which are Cardiomegaly, Consolidation, Edema, Effusion, Emphysema, Fibrosis, Hernia, Mass, Nodule, Pleural thickening, Pneumonia, Pneumothorax and no finding condition which the reason why we decided to train the network with these bunch of classes is we would like to validate the model whether it could perform on complex data well or not. The validation accuracy getting from training networks started the number of layers at 2 and ended at 7 are shown in Table 4.1.

Table 4.1 Percentage of validation accuracy training with 13 classes

Number of classes	Number of hidden layers	Validation accuracy (%)
13	2	21.08
13	3	29.34
13	4	29.63
13	5	30.48
13	6	31.34
13	7	31.62

According to Table 4.1, it showed that the validation accuracy got from training network with 2 hidden layers was 21.08 at the beginning which was too low to be used in further steps. Thereby, we added hidden layers up in every training process, however, the percentage of validation accuracy tended to be fluctuated since the third layer,

seemed to become stable after reached to the fifth layer and gave the highest validation accuracy at the seventh layer. According to these results, we summarized that the maximum number of hidden layers appropriate with this dataset was at the seventh layer. The average of validation accuracy trends was approximate to 28.92 percent which can be predicated that training too many diseases with a network that contains few layers might not work as, in accordance with research, most of them stated that the larger number of hidden layers it contains, the more it can handle with complex network.

Moreover, we reduced number of classes in the range from 5 classes to 3 classes which the chosen diseases are the diseases that we thought the key findings are clearly showed on images which are Cardiomegaly, Consolidation, Edema, Pneumonia and normal condition, so, the percentage of accuracy might gain higher during training and validation process. The number of hidden layers that will be used were 7 since it was the maximum number that could bring about the highest percentage of validation accuracy as referenced earlier. In addition, due to randomly splitting images in IMDS in the ratio of 0.7, it brought about shuffling images of each disease during the process which means that some trained and tested images will not be the same in each round of training and validation. Thus, we averaged the percentage of validation accuracy getting from networks that have the same number of layers in order to observe its accuracy and loss trends easily shown in Table 4.2.

Table 4.2 Percentage of validation accuracy training with 7 hidden layers

Number of classes	**Number of hidden layers**	**Averaged validation accuracy (%)**
5	7	69.93
4	7	77.26
3	7	86.04

With reference to Table 4.2, owing to using number of layers that could give us the highest percentage of accuracy and decreasing number of diseases to be trained and

classified by simple CNN easily, it led to the rising of the percentage of validation accuracy which gave higher values than the previous method in significant way. Moreover, it can be noticed that, in this time, we used normal condition folder instead of no finding condition which the different point between these 2 folders is, in no finding condition folder, it contains both normal lungs images and other diseases that were undefined from the dataset, on the contrary, normal condition folder contains only pure normal lungs images. However, these results were still lower than set threshold, thereby, we decided to train networks with the binary class.

When training networks with the binary class, we split the results into 2 parts which are from training with one chosen disease and no finding case, and another is one chosen disease and normal condition in order to compare the results whether training the binary class with normal condition could give higher percentage of accuracy than training with no finding or not. The results from training networks with the 2 cases; no finding condition and normal condition, are shown in Table 4.3 and Table 4.4, respectively.

The chosen classes that, firstly, used to train with the binary class were one of the diseases in the dataset and no finding condition which most of them are the combination of no finding condition and Hernia. The number of layers that applied to the networks were in the range from 2 to 7 which we also shuffled images several times during training each network. Similarly, images in IMDS were also split in the ratio of 0.7 meaning that the percentage of validation accuracy of trained networks containing the same number of layers will be averaged in order to observe its trends easily as shown in Table 4.3.

Table 4.3 Results from simple CNN training with no finding condition

Number of classes	Number of hidden layers	Averaged validation accuracy (%)
2	2	65.00
2	3	60.20

2	4	55.01
2	5	61.26
2	6	88.80
2	7	79.66

Referring to Table 4.3, one of the obviously points is the percentage of validation accuracy were fluctuated in the range from 50 percent to 90 percent and did not seem to be higher than this. The average of percentage of accuracy was approximated to 68.32 percent which was too low and unacceptable even adding more hidden layers and training with the binary class.

Then, we, secondly, trained networks which, in this case, Pneumonia is chosen to train with normal condition instead of no finding. The number of hidden layers were 7 layers as we knew from the previous method that it is the biggest number for hidden layers before the accuracy becomes stable in order to use the same conditions when comparing the results. The validation accuracy was around 97 percent shown in Table 4.4.

Table 4.4 Results from simple CNN training with normal condition

Number of classes	**Number of hidden layers**	**Validation accuracy (%)**
2	7	97.41

As stated in Table 4.3 and Table 4.4, it can be confirmed that training the networks with images that contained pure normal condition could perform better and gave higher percentage of validation accuracy than no finding condition on both training and validation trends. Even the validation accuracy was over the set threshold, it was only over with this combination of diseases which might because Pneumonia and normal images are 2 diseases that obviously showed the key findings to diagnosis.

Moreover, the percentage of accuracy became stable when reached to the seventh layer, consequently, we did not further train the network with this model and

decided to change and apply other training network architectures that have ability to work with more complex networks in order to enhance the network performance and gain more percentage of accuracy of both training and validation trends.

4.2 Results from AlexNet architecture

The second architecture applied to convolutional neural network is AlexNet architecture which is one kind of the pre-trained networks. As this architecture consists of 5 convolutional layers and 3 fully connected layers, it can perform more powerful than simple convolutional neural network. However, one of the drawbacks of pre-trained network is that it cannot be added or removed any layers from this architecture.

With this model, we also started trained networks with many diseases in a time which we, firstly, started with 13 diseases which are Atelectasis, Cardiomegaly, Consolidation, Edema, Effusion, Emphysema, Fibrosis, Hernia, Mass, Nodule, Pleural thickening, Pneumonia and Pneumothorax. Secondly, we decreased diseases to be 12 which cut Atelectasis and Hernia off and added normal condition instead showing the results in Table 4.5. The reason why we cut Hernia off during training process was, in our software program, we applied min command to count the minimum images containing in each folder, then, makes the number of images became equal at those values. All of images were, consequently, cut off at 110 images since Hernia is a disease that contains the fewest number of images. Owing to this action, it might lead to the low validation accuracy because there were too small number of images to be trained in the network.

Table 4.5 Results from AlexNet architecture training with multiclass works

Number of classes	Validation accuracy (%)
13	26.97
12	28.56

Similar to simple CNN case that training network with too many diseases in a time with a model that might not be able to support complex works would be one of the reasons leading to low accuracy.

As a result, we decreased number of the diseases to be, firstly, 7 diseases, and, secondly, 3 diseases showing the validation accuracy together with graphs of both accuracy and loss during training network with 7 diseases in Table 4.6 and Figure 4.1, respectively.

Table 4.6 Results from AlexNet architecture training with a normal condition

Number of classes	Average validation accuracy (%)
7	53.95
3	58.59

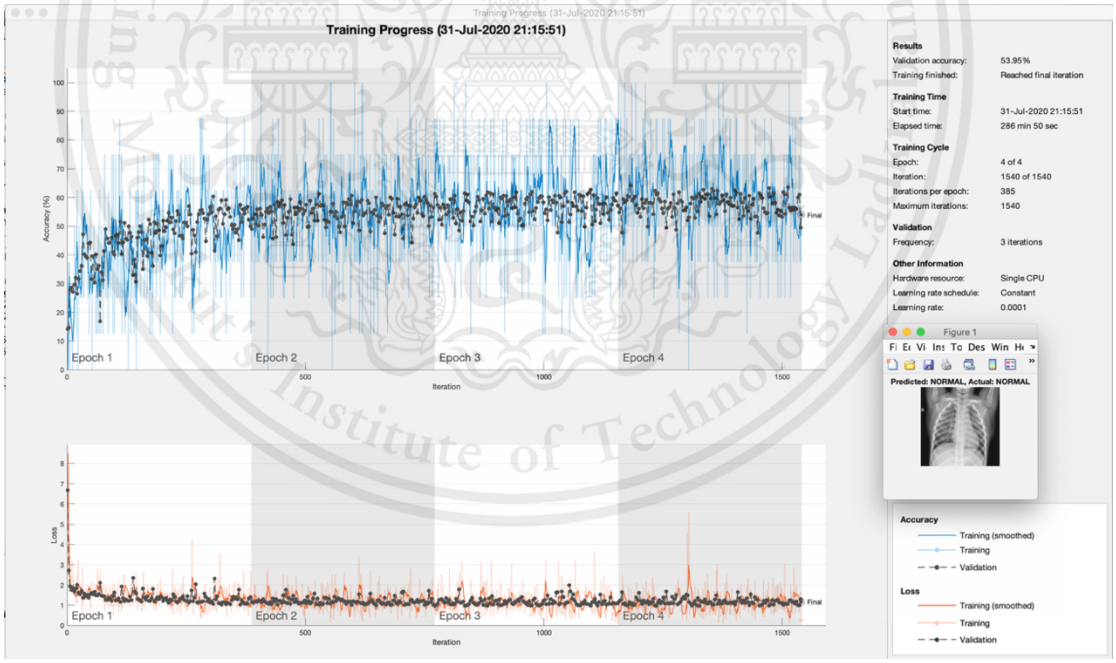

Figure 4.1 Accuracy and loss trends during training a network with 7 classes

According to Table 4.6 and Figure 4.1, it inferred that even number of classes were decreased to 3, the percentage of accuracy fluctuated no higher than 60 percent which was not over the set threshold. Moreover, not only resulting in low percentage of accuracy but also consuming substantial computing resources because of huge

parameters that occurred during training a network which caused a long duration during training each process. Due to these reasons, it can be indicated that AlexNet architecture may also not be suitable for this CRX dataset.

Therefore, we decided to change network architecture to ResNet-18 architecture which is more powerful than AlexNet architecture and might lead to higher percentage of accuracy and be able to predict the diseases as accurate as much as possible.

4.3 Results from ResNet-18 architecture

The third architecture applied to convolutional neural network is ResNet-18 architecture which is also another kind of the pre-trained network. Since, there are 18 layers containing in this architecture, it means that this architecture might be able to perform better than 2 previous models leading to giving higher percentage of accuracy when do the predictions.

Correspondingly method in previous models, we, firstly, trained with multiclass work several networks to keep track of the range of validation accuracy which the chosen diseases are Cardiomegaly, Consolidation, Edema, Pneumonia, and normal condition for training 5 diseases and Cardiomegaly, Consolidation and Edema for training 3 diseases. Since there are several networks be trained, we, again, averaged the validation accuracy that got from networks that contained the same diseases shown in Table 4.7.

Table 4.7 Results from ResNet-18 architecture training with multiclass works

Number of classes	Average validation accuracy (%)
5	75.23
3	61.80

Secondly, we trained networks with the binary class which the chosen diseases are Cardiomegaly and Consolidation. The average of validation accuracy was shown in Table 4.8. Noted that, to be fair and get the most accurate results, all of the values that

needed to be set such as number of trained images, number of epochs and initial learning rate must be the same which are 572, 10 and 0.001, respectively.

Table 4.8 Results from ResNet-18 architecture training with the binary class

Number of classes	Average validation accuracy (%)
2	62.70

In addition, Figure 4.2, Figure 4.3 and Figure 4.4 are the figures that showed the results from training networks with 5 diseases, 3 diseases and 2 diseases, respectively in order to observe and compare the trends easily.

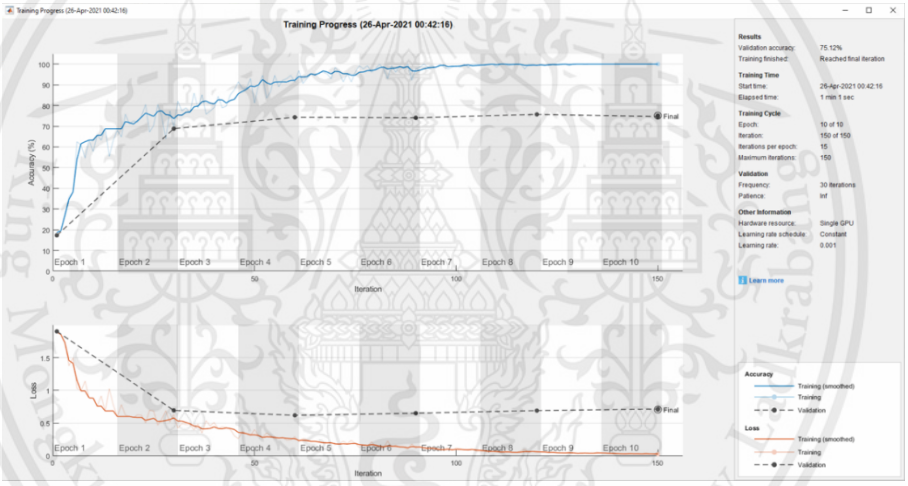

Figure 4.2 Accuracy and loss trends during training a network with 5 classes

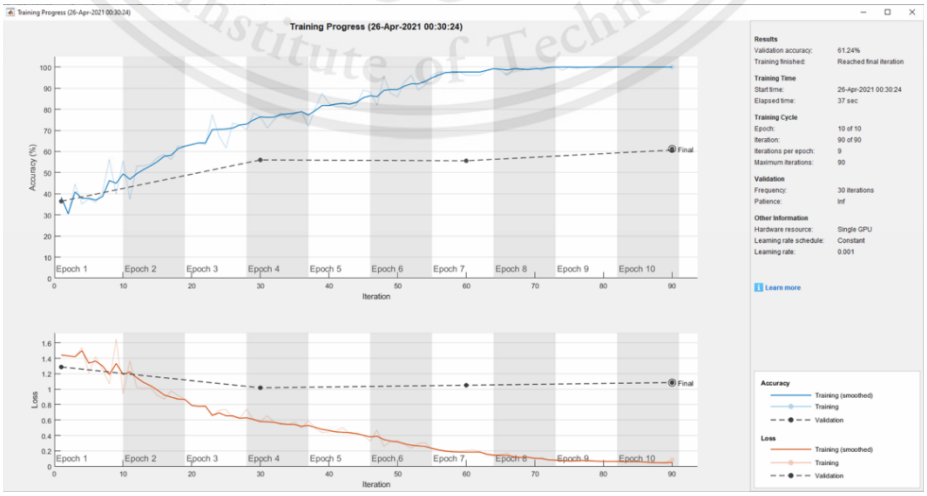

Figure 4.3 Accuracy and loss trends during training a network with 3 classes

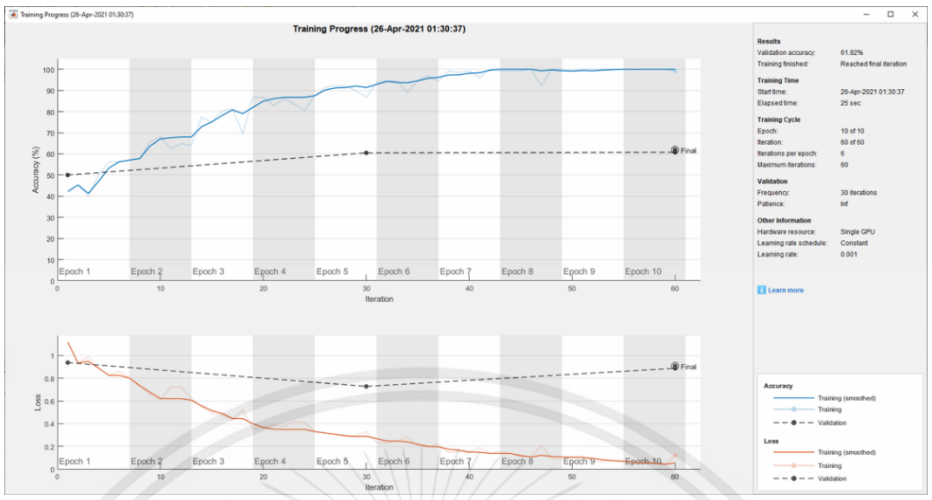

Figure 4.4 Accuracy and loss trends during training a network with 2 classes

In accordance with Table 4.7, Table 4.8, Figure 4.2, Figure 4.3 and Figure 4.4, it can be noticed that whether training networks with the binary class or multiclass works, one of the obvious points shown that this dataset may not appropriate with this architecture were the trends of both training and validation which tended to move apart since the beginning of the process. The training graph gradually rose and fell in the big range on both accuracy and loss trends, however, the validation graph seemed to be stable on both accuracy and lose trends.

It can be concluded that even the number of parameters can be reduced by running the process on ReNet-18 architecture, the validation accuracy and its training and validation trends were still not in the range that acceptable and not over the set threshold whether training networks with binary class or multiclass works which the average of the validation accuracy on these 2 tasks approximated to 65.58 percent. Hence, we had to change the training architecture to be others that contain more layers and be able to support complex tasks than this model.

4.4 Results from ResNet-50 architecture

The fourth architecture applied to convolutional neural network is ResNet-50 architecture which is also one of the pre-trained networks. Owing to some problems encountered while running the process through simple convolutional neural network, AlexNet and ResNet-18 architecture, we decided to use ResNet-50 architecture as a

trained model instead. The number of layer-depth containing in this architecture are 50 leading to getting higher training and validation of accuracy.

We, firstly, trained the networks with 13 classes similar to things that we did in previous networks which the reason why we still started with 13 classes is we would like to reinspect and make a confirmation that training a network with too many diseases in a time works with this architecture or not. There were 13 classes chosen to be trained which are Cardiomegaly, Consolidation, Edema, Effusion, Emphysema, Fibrosis, Hernia, Mass, Nodule, Pleural thickening, Pneumonia and Pneumothorax, plus one normal condition. We shuffled the images training in the networks for 3 times to observe its trends easily showing the results in Table 4.9.

Table 4.9 Results from ResNet-50 architecture training with 13 classes

Number of classes	Validation accuracy (%)
13	27.74
13	28.67
13	32.40

According to Table 4.9, it can be confirmed that training the networks with 13 classes in a time was not literally success even trained on ResNet-50 architecture, an architecture that contains the highest number of layers in this time. The average of validation accuracy that got from this training was 29.60 percent which was too far from the set threshold.

Thus, we, secondly, decreased the number of classes to 12 classes which cut Hernia off with the same reason as we did in previous method. The training process was also done for 3 times to use the same condition as training with 13 classes which the results were shown in Table 4.10.

Table 4.10 Results from ResNet-50 architecture training with 12 classes

Number of classes	Validation accuracy (%)

12	30.04
12	40.16
12	42.02

As shown on Table 4.10, it indicated that the validation accuracy was not vary from training networks with 13 classes much which the average of the validation accuracy in this case was approximate to 37.41 percent.

As a result, we, thirdly, decreased the number of classes once again to be 5 classes. We split the combination of diseases into 2 groups which the first group is Cardiomegaly, Consolidation, Edema, Pneumonia and normal condition and the second group is Consolidation, Mass, Nodule, Pneumonia and normal condition. Noted that, to be fair, both 2 groups were used the same conditions such as the number of images, number of training times and number of epochs. The results from first group of diseases and second group of disease are shown in Table 4.11 and Table 4.12, respectively.

Table 4.11 Results from ResNet-50 architecture training with the first group of diseases

Number of classes	**Validation accuracy (%)**
5	73.51
5	75.48
5	76.06

Table 4.12 Results from ResNet-50 architecture training with the second group of diseases

Number of classes	**Validation accuracy (%)**
5	54.48

5	55.90
5	57.28

Referring to Table 4.11 and Table 4.12, it can be presumed that even training the networks with the first group could give higher percentage of validation accuracy than another, it still could not gain accuracy over the set threshold which the average of the validation accuracy was around 75.02 percent. Moreover, not only the first group could not over the set threshold, but also the second group which resulted lower percent than the first one with the average of accuracy at 55.88 percent.

Thereby, we decided to train the network with the binary class in order to clearly observe and compare its performance and trends of training and validation accuracy. There were many combinations that we tried to classify and predict such as combining normal condition with Atelectasis, Cardiomegaly, Consolidation, Edema, Effusion, Pneumonia and Pneumothorax, plus combining Atelectasis with its compliment which composed of other diseases except Atelectasis. Since each network was trained in several times, the validation, shown in Table 4.13, would also be shown in the averaged values.

Table 4.13 Results from ResNet-50 architecture training with the binary class

Disease	**Disease**	**Averaged validation accuracy (%)**
Atelectasis	Atelectasis compliment	70.53
Normal	Pneumonia	94.21
Normal	Effusion	99.75
Normal	Pneumothorax	99.83
Normal	Atelectasis	100
Normal	Cardiomegaly	100

Normal	Consolidation	100
Normal	Edema	100

According to Table 4.13 and its layer-depth, most of the percentage of validation accuracy getting from this architecture reached approximately over 90 percent, except from training network with the compliment combination, which the average of validation accuracy was around 95.54 percent. After we found that the average of validation accuracy could perform over the set threshold, we validated the networks whether it could do the classification and prediction correctly or not by inputting the untrained images to several saved networks, then observed its predictions. However, most of the predictions getting from these networks were wrong, shown in Figure 4.5, even the validation accuracy was very high. As a result, it can be assumed that there might be some errors occurred during the training process. Since the results getting from the binary classification were only true or false, meaning that the predictions would be either class A or class B. Due to this result, the probability of miscalculation may be higher when comparing with the predictions from multiclass works because of fewer choices for the predictions. On the other hand, it can also be assumed that the process would only done by a single check. Hence, we decided to add more classes in order to make it has more choices to predict the disease, then average the probability values getting from Softmax layer.

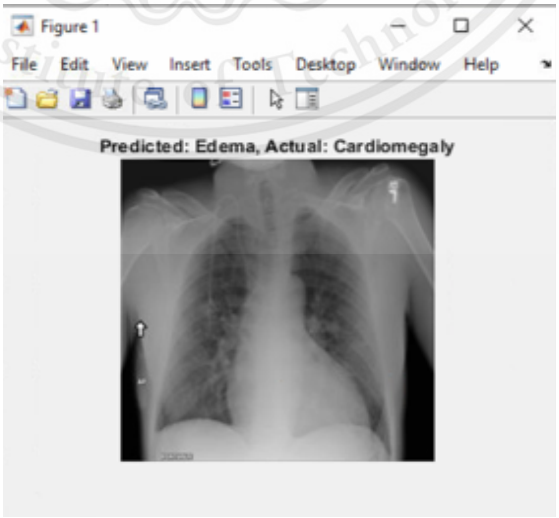

Figure 4.5 Result after inputted untrained image to a network

Consequently, we applied more diseases into the network to make it be able to support multiclass works which, referring to the results from every previous architecture, we found that the number of classes that seemed to be suitable with our dataset and could give the highest percentage of accuracy, comparing with other multiclass works, is 3. Accordingly, we chose the number of 3 for training with the rest of our works, both in ResNet-50 architecture and GoogLeNet architecture.

With this architecture, the number of epochs were changed to be 200 in order to optimize the training process for DNN and produce maximum results as much as possible. The results showing in Table 4.14 were some of the results that we got from training with 3 classes, just to show the fluctuated range of accuracy trends.

Table 4.14 Results from ResNet-50 architecture training with 3 classes

Disease	**Disease**	**Disease**	**Percentage of accuracy**
Edema	Nodule	Mass	43.88
Mass	Hernia	Pleural thickening	48.56
Emphysema	Nodule	Pleural thickening	54.36
Edema	Emphysema	Nodule	56.41
Hernia	Fibrosis	Emphysema	58.41
Edema	Effusion	Mass	65.94
Edema	Effusion	Nodule	67.23
Edema	Hernia	Pleural thickening	75.21
Pneumonia	Nodule	Atelectasis	80.31
Emphysema	Nodule	Pneumonia	82.99

Normal	Pneumonia	Edema	96.51
Normal	Pneumonia	Emphysema	97.31
Normal	Pneumonia	Pneumothorax	98.20
Normal	Pneumonia	Atelectasis	98.30
Normal	Pneumonia	Consolidation	98.84

Regarding Table 4.14, after training the network and validating the data, the percentage of accuracy was significantly higher than three previous methods even though it was trained with multiclass. The accuracy varied from 40 to 98 percent depending on the chosen disease pair which somewhat fluctuated.

When observing the trends on both the accuracy and loss trends, the percentage of validation accuracy was high when matching the normal condition and Pneumonia with any other diseases especially Edema, Emphysema, Pneumothorax, Atelectasis and Consolidation which these networks were the networks that could achieve percentage of accuracy over the set threshold. The combination of illness that gave us the highest accuracy was from the normal condition, Pneumonia and Consolidation which was at 98.84 percent. However, the combination of the other cases, such as matching Nodule with Atelectasis and Pneumonia, the percentage of accuracy was approximate to 80 percent. Ultimately, the average percentage of accuracy from this table was at 74.83 percent.

Further, we also tried to train the network with 2 chosen diseases and its compliment class, in which, as mentioned earlier that the compliment class is a class that collects all the images that are not labelled as 2 chosen diseases, to have more results to compare and analyze which paths are suitable to our work most. There were 14 networks trained with its compliment class which the results getting from these networks are shown in Table 4.15. In addition, the example of the accuracy and loss trends, the bar graph representing the top predictions including its percent predictions and its numeric confusion matrix are provided in Figure 4.6, Figure 4.7 and Figure 4.8.

Table 4.15 Results from ResNet-50 architecture training a network with its compliment class

Disease	**Disease**	**Disease**	**Percentage of Validation accuracy**
Nodule	Pleural thickening	Compliment	55.69
Consolidation	Edema	Compliment	56.01
Mass	Nodule	Compliment	56.86
Fibrosis	Hernia	Compliment	58.97
Emphysema	Fibrosis	Compliment	61.38
Hernia	Mass	Compliment	62.82
Effusion	Emphysema	Compliment	63.72
Edema	Effusion	Compliment	64.53
Cardiomegaly	Consolidation	Compliment	65.25
Atelectasis	Cardiomegaly	Compliment	67.54
Pleural thickening	Pneumonia	Compliment	73.01
Normal	Atelectasis	Compliment	79.39
Pneumonia	Pneumothorax	Compliment	79.50
Pneumothorax	Normal	Compliment	84.65

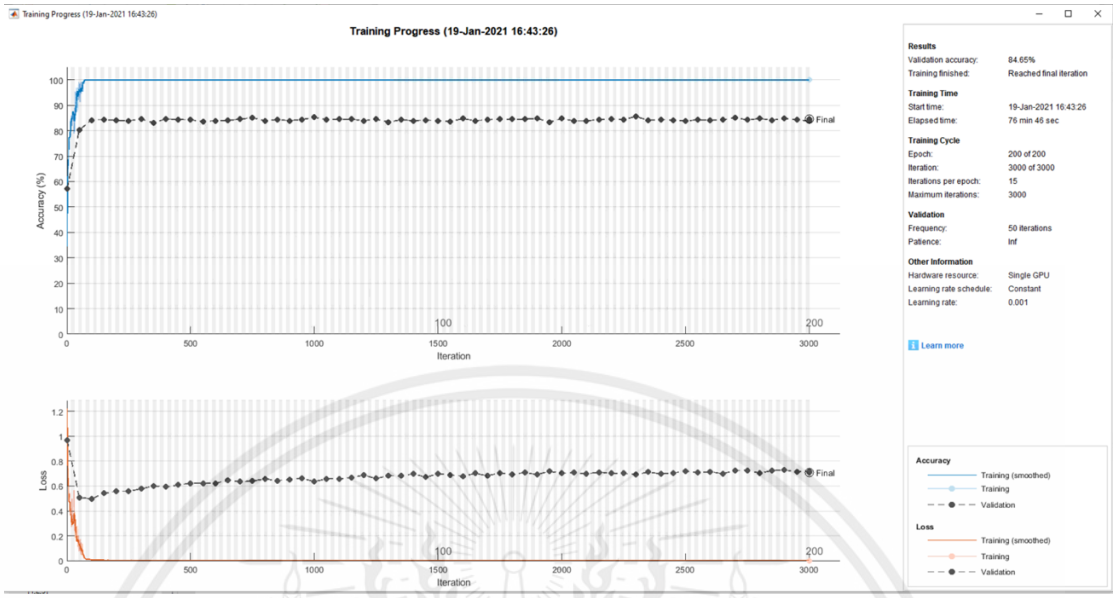

Figure 4.6 Accuracy and loss trends during training Pneumothorax, Normal condition and its compliment

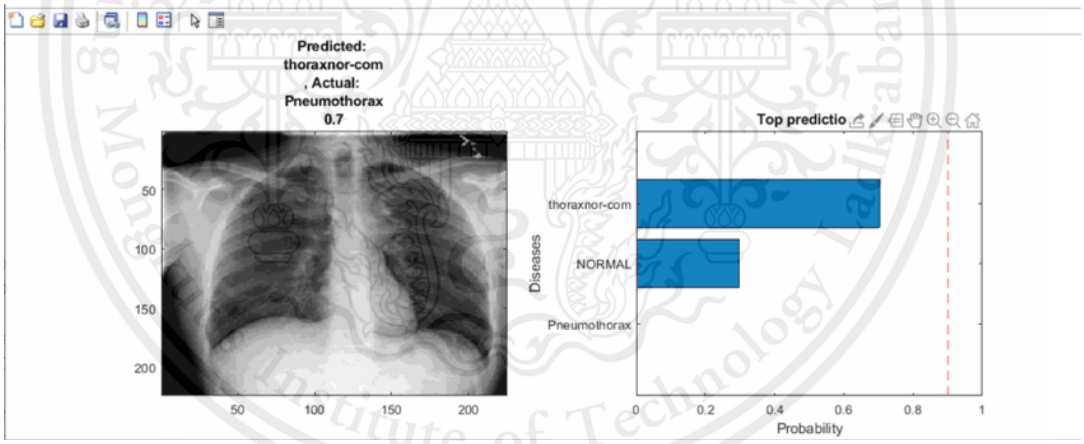

Figure 4.7 Bar graph representing the top predictions of a network trained with Pneumothorax, Normal condition and its compliment

```
confMat =

    0.9964         0    0.0036
         0    0.7734    0.2266
    0.0180    0.2122    0.7698
```

Figure 4.8 Numeric confusion matrix of Pneumothorax, Normal condition and its compliment

According to results containing in Table 4.15, it indicated that the percentage of accuracy was gradually increased from 55 percent, however, the highest percentage of accuracy was only around 84 percent. Therefore, it can be summarized that there was no network that gained percentage of validation accuracy over the set threshold.

Moreover, in accordance with Figure 4.6, Figure 4.7 and Figure 4.8, it can be another evidence that even picking up the network that could perform the highest percentage of validation accuracy, the training and validating trends still tended to move apart and give the value lower than the set threshold. After analyzed more details about the accuracy and loss trends, we found that the accuracy trend seemed to be stable since the first stage of training process, whereas the loss trend fluctuated and rose up to 0.7 percent at the end of process.

Although this architecture could perform in a better performance than the other previously trained architectures, the percentage of validation accuracy was also unacceptable when trained with 2 chosen diseases and its compliment class. Therefore, it can be assumed that this architecture may also not be suitable for these chest X-Ray datasets. Since we need the higher performance architecture to do the prediction, we believe there may be any other architectures that can perform and give us a better result. Hence, we researched more of which architecture can lead us to gain more percentage of accuracy.

4.5 Results from GoogLeNet architecture

The last architecture that applied to convolutional neural network is GoogLeNet architecture. Even it contains only 22 layer-dept, it can perform so well with computer vision tasks especially image classification. Moreover, as we decreased number of parameters creating while training the model, it leads to a faster training process and lessen a chance of network overfitting.

Since we found that training networks with 3 classes at a time gave the most satisfy results in term of both validation accuracy and process time consuming, we kept continue training with this combination.

Firstly, we trained the network with the same classes as we did with ResNet-50 which were 2 chosen diseases and its compliment class which the reason is to compare the results whether this GoogLeNet architecture could actually give a higher percentage

of accuracy than ResNet-50 architecture as we expected or not. There were 14 networks to be trained, same as previous architecture which the results from training networks with these combinations are as shown in Table 4.16. Moreover, the example of the trends and its numeric confusion matrix and chart of some networks are shown in Figure 4.9, Figure 4.10 and Figure 4.11.

Table 4.16 Results from GoogLeNet architecture training a network with its compliment class

Disease	**Disease**	**Disease**	**Percentage of accuracy**
Effusion	Emphysema	Compliment	84.82
Mass	Nodule	Compliment	85.28
Atelectasis	Cardiomegaly	Compliment	86.32
Normal	Atelectasis	Compliment	86.32
Edema	Effusion	Compliment	86.45
Nodule	Pleural thickening	Compliment	86.92
Cardiomegaly	Consolidation	Compliment	88.63
Consolidation	Edema	Compliment	90.08
Hernia	Mass	Compliment	91.09
Pneumonia	Pneumothorax	Compliment	91.92
Pneumothorax	Normal	Compliment	92.09
Emphysema	Fibrosis	Compliment	92.47
Pleural thickening	Pneumonia	Compliment	93.39

Fibrosis	Hernia	Compliment	96.53

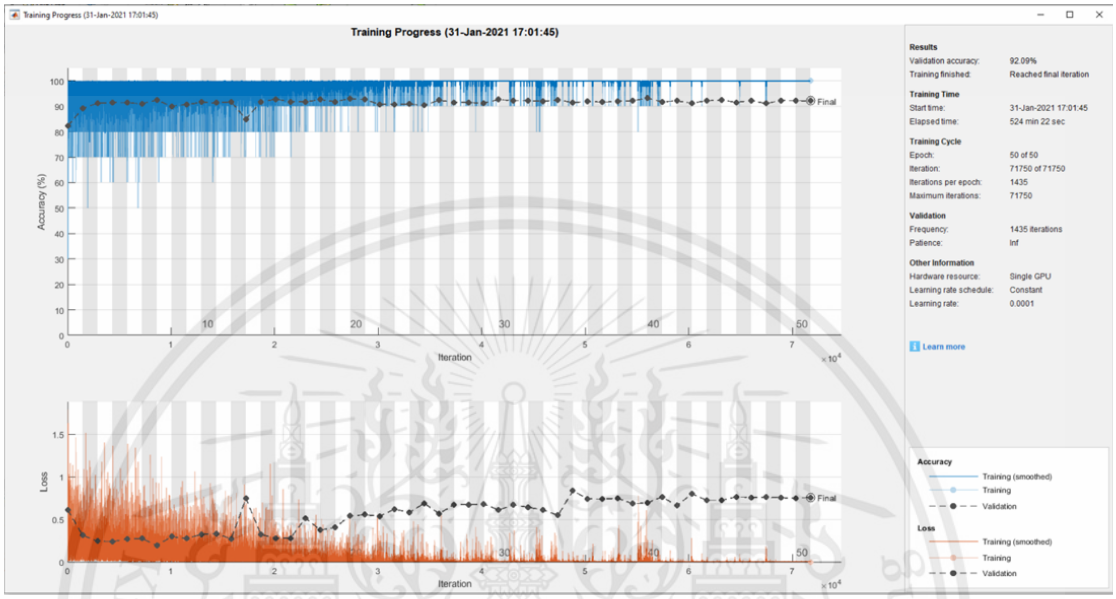

Figure 4.9 Accuracy and loss trends during training Pneumothorax, Normal condition and its compliment

```
confMat =

    0.9084         0    0.0916
         0    0.0860    0.9140
    0.0458         0    0.9542
```

Figure 4.10 Numeric confusion matrix of Pneumothorax, Normal condition and its compliment

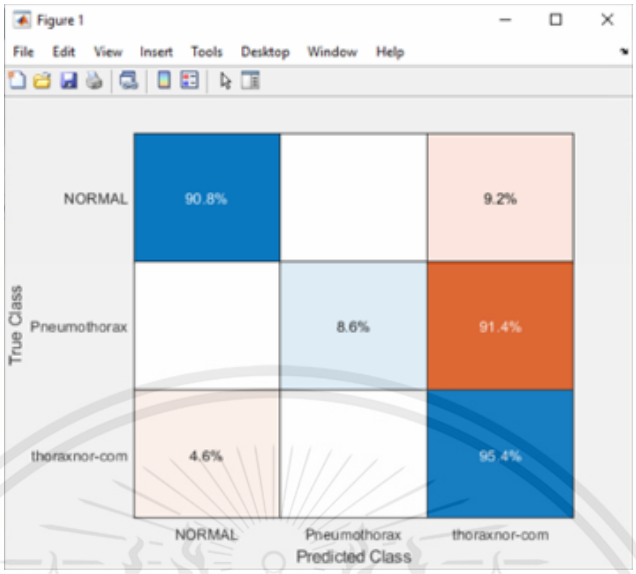

Figure 4.11 Confusion matrix chart of Pneumothorax, Normal condition and its compliment

In accordance with Table 4.16, it indicated that the percentage of accuracy was in range from 80 to 96 in which the trend was much smoother and gave better percentage of accuracy compared with the previous architectures. On top of that, the prediction that made from randomly picking 1 image up from imdsValidation during the training process was also high as described in Figure 4.10 and Figure 4.11 which indicated the percent of correctly and incorrectly classified observations for each corresponding true class. However, there were only 1 network that gained percentage of accuracy over the set threshold which was from the network of Fibrosis, Hernia and its compliment.

Consequently, we changed the work path training the networks with 3 classes of disease instead of training with its compliment with the thought that it could give wider range of trained networks leading to have more probabilities to get the networks that could gain the accuracy over the set threshold. Then, we calculated all the possible networks with the Equation 3.1 as mentioned above via MATLAB program to know the exact number of networks which there were 364 networks to be trained further. Noted again that, to be fair, every essential variable for the training process was configured to be the same as other previous architectures except the number of epochs which, with this architecture, we decreased the values back to 50 to reduce training

process time consuming after proved that the percentage of accuracy from training networks with 50 epochs was not vary from training networks with 200 epochs much. The results shown in Table 4.17 were completely ordered in descending order to keep track of results easily. Moreover, we also provided the examples of the bar graph representing the top predictions including its percent predictions, the accuracy and loss trends and its numeric confusion matrix and chart of some networks of both gained validation accuracy over the set threshold and lower the set threshold in Figure 4.12, Figure 4.13, Figure 4.14, Figure 4.15, Figure 4.16, Figure 4.17, Figure 4.18 and Figure 4.19, respectively.

Table 4.17 Results from GoogLeNet architecture training with 3 classes, 364 networks

Disease	**Disease**	**Disease**	**Percentage of accuracy**
Effusion	Hernia	Pneumonia	98.82
Mass	Normal	Pneumonia	98.80
Atelectasis	Hernia	Pneumonia	98.77
Edema	Hernia	Pneumonia	98.76
Consolidation	Hernia	Pneumonia	98.56
Normal	Pleural Thickening	Pneumonia	98.55
Normal	Pneumonia	Pneumothorax	98.51
Hernia	Pneumonia	Pneumothorax	98.38
Nodule	Normal	Pneumonia	98.31
Effusion	Normal	Pneumonia	98.22
Emphysema	Hernia	Pneumonia	98.21
Fibrosis	Hernia	Pneumonia	98.17
Cardiomegaly	Hernia	Pneumonia	98.09
Hernia	Mass	Pneumonia	98.06
Atelectasis	Hernia	Normal	97.97
Hernia	Pleural Thickening	Pneumonia	97.95

Effusion	Hernia	Normal	97.94
Atelectasis	Normal	Pneumonia	97.93
Consolidation	Normal	Pneumonia	97.91
Hernia	Nodule	Pneumonia	97.84
Cardiomegaly	Normal	Pneumonia	97.83
Hernia	Normal	Pneumothorax	97.54
Emphysema	Normal	Pneumonia	97.39
Edema	Fibrosis	Pneumonia	97.21
Hernia	Normal	Pneumonia	97.19
Hernia	Nodule	Normal	97.12
Edema	Normal	Pneumonia	96.92
Fibrosis	Normal	Pneumonia	96.81
Edema	Hernia	Normal	96.69
Cardiomegaly	Fibrosis	Pneumonia	96.65
Consolidation	Hernia	Normal	96.62
Hernia	Mass	Normal	96.42
Hernia	Normal	Pleural Thickening	96.19
Emphysema	Hernia	Normal	96.03
Effusion	Fibrosis	Pneumonia	95.49
Cardiomegaly	Emphysema	Pneumonia	95.40
Cardiomegaly	Hernia	Normal	95.29
Fibrosis	Pneumonia	Pneumothorax	95.00
Cardiomegaly	Pneumonia	Pneumothorax	94.93
Edema	Fibrosis	Normal	94.92
Cardiomegaly	Edema	Pneumonia	94.69
Atelectasis	Edema	Pneumonia	94.40
Edema	Emphysema	Pneumonia	94.36
Emphysema	Fibrosis	Pneumonia	94.07

Fibrosis	Hernia	Normal	94.06
Cardiomegaly	Effusion	Pneumonia	93.97
Edema	Pleural Thickening	Pneumonia	93.96
Atelectasis	Fibrosis	Pneumonia	93.68
Edema	Pneumonia	Pneumothorax	93.58
Cardiomegaly	Pleural Thickening	Pneumonia	93.24
Cardiomegaly	Consolidation	Pneumonia	93.00
Cardiomegaly	Mass	Pneumonia	93.00
Consolidation	Fibrosis	Pneumonia	92.94
Atelectasis	Cardiomegaly	Pneumonia	92.64
Cardiomegaly	Nodule	Pneumonia	92.62
Edema	Effusion	Pneumonia	92.57
Edema	Nodule	Pneumonia	92.47
Edema	Mass	Pneumonia	92.46
Cardiomegaly	Fibrosis	Normal	91.89
Effusion	Fibrosis	Normal	91.83
Effusion	Emphysema	Pneumonia	91.27
Fibrosis	Pleural Thickening	Pneumonia	90.79
Emphysema	Nodule	Pneumonia	90.71
Consolidation	Emphysema	Pneumonia	90.56
Atelectasis	Cardiomegaly	Normal	90.52
Atelectasis	Edema	Normal	90.42
Cardiomegaly	Effusion	Normal	90.41
Consolidation	Edema	Pneumonia	90.24
Cardiomegaly	Emphysema	Normal	90.08
Emphysema	Pleural Thickening	Pneumonia	90.08
Emphysema	Mass	Pneumonia	90.02
Cardiomegaly	Mass	Normal	89.97

Atelectasis	Emphysema	Pneumonia	89.82
Atelectasis	Fibrosis	Normal	89.76
Edema	Emphysema	Normal	89.64
Cardiomegaly	Normal	Pneumothorax	89.50
Atelectasis	Pneumonia	Pneumothorax	89.36
Fibrosis	Nodule	Pneumonia	89.14
Effusion	Nodule	Pneumonia	88.97
Edema	Normal	Pleural Thickening	88.93
Fibrosis	Mass	Pneumonia	88.90
Edema	Normal	Pneumothorax	88.79
Pleural Thickening	Pneumonia	Pneumothorax	88.68
Cardiomegaly	Consolidation	Normal	88.60
Cardiomegaly	Edema	Normal	88.55
Consolidation	Pleural Thickening	Pneumonia	88.50
Fibrosis	Normal	Pneumothorax	88.42
Edema	Effusion	Normal	88.23
Consolidation	Pneumonia	Pneumothorax	88.01
Cardiomegaly	Normal	Pleural Thickening	87.84
Atelectasis	Edema	Hernia	87.79
Emphysema	Pneumonia	Pneumothorax	87.73
Effusion	Mass	Pneumonia	87.67
Effusion	Pneumonia	Pneumothorax	87.47
Mass	Pneumonia	Pneumothorax	87.24
Emphysema	Fibrosis	Normal	87.22
Consolidation	Nodule	Pneumonia	87.06
Consolidation	Fibrosis	Normal	86.93
Effusion	Fibrosis	Hernia	86.93
Atelectasis	Pleural Thickening	Pneumonia	86.87

Atelectasis	Mass	Pneumonia	86.83
Edema	Mass	Normal	86.82
Effusion	Pleural Thickening	Pneumonia	86.67
Edema	Nodule	Normal	86.65
Atelectasis	Nodule	Pneumonia	86.16
Cardiomegaly	Effusion	Hernia	86.09
Consolidation	Mass	Pneumonia	86.09
Atelectasis	Consolidation	Pneumonia	85.98
Cardiomegaly	Nodule	Normal	85.68
Mass	Pleural Thickening	Pneumonia	85.40
Effusion	Emphysema	Normal	85.13
Atelectasis	Emphysema	Normal	85.01
Fibrosis	Mass	Normal	84.96
Atelectasis	Effusion	Pneumonia	84.82
Atelectasis	Fibrosis	Hernia	84.14
Emphysema	Mass	Normal	84.13
Emphysema	Nodule	Normal	84.13
Effusion	Nodule	Normal	83.81
Atelectasis	Cardiomegaly	Edema	83.70
Nodule	Pleural Thickening	Pneumonia	83.68
Fibrosis	Nodule	Normal	83.15
Fibrosis	Normal	Pleural Thickening	83.04
Nodule	Pneumonia	Pneumothorax	82.95
Consolidation	Effusion	Pneumonia	82.79
Emphysema	Normal	Pleural Thickening	82.47
Atelectasis	Cardiomegaly	Hernia	82.37
Consolidation	Emphysema	Normal	82.31
Consolidation	Edema	Normal	82.06

Cardiomegaly	Hernia	Pneumothorax	82.02
Edema	Fibrosis	Hernia	81.94
Edema	Effusion	Hernia	81.90
Normal	Pleural Thickening	Pneumothorax	81.74
Cardiomegaly	Hernia	Mass	81.59
Mass	Normal	Pneumothorax	81.55
Cardiomegaly	Emphysema	Hernia	81.47
Atelectasis	Mass	Normal	81.20
Atelectasis	Normal	Pneumothorax	80.71
Effusion	Mass	Normal	80.71
Emphysema	Normal	Pneumothorax	80.38
Edema	Emphysema	Hernia	80.33
Fibrosis	Hernia	Pneumothorax	80.23
Consolidation	Normal	Pneumothorax	80.06
Mass	Nodule	Pneumonia	79.85
Nodule	Normal	Pneumothorax	79.23
Effusion	Normal	Pleural Thickening	79.20
Consolidation	Nodule	Normal	79.10
Effusion	Normal	Pneumothorax	78.88
Effusion	Hernia	Nodule	78.72
Atelectasis	Normal	Pleural Thickening	78.65
Effusion	Emphysema	Hernia	78.26
Atelectasis	Normal	Normal	78.03
Cardiomegaly	Effusion	Fibrosis	77.94
Atelectasis	Effusion	Normal	77.83
Cardiomegaly	Consolidation	Hernia	77.78
Consolidation	Mass	Normal	77.76
Cardiomegaly	Edema	Fibrosis	77.72

Atelectasis	Edema	Fibrosis	77.62
Consolidation	Normal	Pleural Thickening	77.58
Edema	Effusion	Fibrosis	77.01
Cardiomegaly	Fibrosis	Hernia	76.90
Atelectasis	Hernia	Pneumothorax	76.55
Cardiomegaly	Hernia	Nodule	76.54
Cardiomegaly	Hernia	Pleural Thickening	76.36
Cardiomegaly	Edema	Effusion	76.23
Edema	Hernia	Pleural Thickening	76.07
Atelectasis	Consolidation	Normal	75.86
Atelectasis	Emphysema	Hernia	75.86
Edema	Hernia	Nodule	75.85
Effusion	Hernia	Mass	75.46
Cardiomegaly	Edema	Hernia	75.45
Atelectasis	Cardiomegaly	Fibrosis	75.12
Edema	Hernia	Pneumothorax	75.08
Mass	Normal	Pleural Thickening	74.68
Consolidation	Effusion	Normal	74.58
Atelectasis	Hernia	Mass	73.46
Consolidation	Fibrosis	Hernia	73.41
Cardiomegaly	Fibrosis	Pneumothorax	73.40
Cardiomegaly	Emphysema	Fibrosis	73.36
Cardiomegaly	Effusion	Emphysema	73.17
Cardiomegaly	Edema	Emphysema	73.12
Hernia	Nodule	Pneumothorax	72.85
Edema	Effusion	Emphysema	72.55
Cardiomegaly	Effusion	Nodule	72.49
Cardiomegaly	Effusion	Mass	71.85

Emphysema	Fibrosis	Hernia	71.78
Effusion	Emphysema	Fibrosis	71.52
Hernia	Pleural Thickening	Pneumothorax	71.28
Edema	Hernia	Mass	71.23
Nodule	Normal	Pleural Thickening	71.20
Atelectasis	Edema	Emphysema	71.12
Consolidation	Emphysema	Hernia	70.84
Cardiomegaly	Effusion	Pneumothorax	70.72
Atelectasis	Cardiomegaly	Mass	70.67
Edema	Emphysema	Fibrosis	70.57
Hernia	Mass	Pneumothorax	70.53
Effusion	Fibrosis	Nodule	70.50
Atelectasis	Cardiomegaly	Consolidation	70.44
Cardiomegaly	Edema	Nodule	70.33
Atelectasis	Cardiomegaly	Emphysema	70.29
Atelectasis	Cardiomegaly	Pneumothorax	70.25
Cardiomegaly	Consolidation	Fibrosis	70.20
Cardiomegaly	Edema	Pleural Thickening	70.20
Effusion	Fibrosis	Mass	70.07
Edema	Fibrosis	Pneumothorax	69.95
Atelectasis	Emphysema	Fibrosis	69.94
Atelectasis	Effusion	Hernia	69.93
Atelectasis	Cardiomegaly	Pleural Thickening	69.88
Atelectasis	Edema	Mass	69.86
Effusion	Hernia	Pleural Thickening	69.84
Edema	Effusion	Nodule	69.53
Atelectasis	Fibrosis	Pneumothorax	69.38
Cardiomegaly	Fibrosis	Mass	69.31

Cardiomegaly	Mass	Pneumothorax	69.25
Cardiomegaly	Edema	Mass	68.91
Cardiomegaly	Edema	Pneumothorax	68.79
Effusion	Fibrosis	Pneumothorax	68.76
Atelectasis	Hernia	Pleural Thickening	68.73
Fibrosis	Hernia	Mass	68.57
Effusion	Hernia	Pneumothorax	68.52
Atelectasis	Cardiomegaly	Nodule	68.48
Cardiomegaly	Effusion	Pleural Thickening	68.22
Cardiomegaly	Emphysema	Mass	68.13
Hernia	Nodule	Pleural Thickening	68.02
Fibrosis	Hernia	Nodule	67.93
Mass	Nodule	Normal	67.47
Atelectasis	Consolidation	Hernia	67.40
Effusion	Emphysema	Nodule	67.36
Atelectasis	Fibrosis	Mass	67.29
Atelectasis	Edema	Pleural Thickening	67.22
Atelectasis	Edema	Pneumothorax	66.96
Atelectasis	Cardiomegaly	Effusion	66.89
Cardiomegaly	Consolidation	Emphysema	66.88
Edema	Effusion	Mass	66.74
Atelectasis	Edema	Nodule	66.30
Atelectasis	Effusion	Emphysema	66.06
Atelectasis	Effusion	Fibrosis	65.73
Cardiomegaly	Fibrosis	Pleural Thickening	65.51
Emphysema	Hernia	Mass	65.46
Emphysema	Hernia	Nodule	65.19
Cardiomegaly	Emphysema	Nodule	65.15

Fibrosis	Hernia	Pleural Thickening	64.86
Consolidation	Hernia	Pleural Thickening	64.74
Cardiomegaly	Nodule	Pneumothorax	64.43
Atelectasis	Fibrosis	Nodule	64.37
Consolidation	Hernia	Mass	64.34
Edema	Nodule	Pneumothorax	64.32
Cardiomegaly	Consolidation	Edema	64.25
Cardiomegaly	Fibrosis	Nodule	64.25
Atelectasis	Consolidation	Fibrosis	64.24
Atelectasis	Edema	Effusion	64.13
Consolidation	Edema	Hernia	64.12
Consolidation	Hernia	Nodule	63.87
Atelectasis	Hernia	Nodule	63.85
Cardiomegaly	Consolidation	Effusion	63.79
Atelectasis	Fibrosis	Pleural Thickening	63.70
Atelectasis	Consolidation	Edema	63.59
Cardiomegaly	Pleural Thickening	Pneumothorax	63.55
Cardiomegaly	Emphysema	Pleural Thickening	63.48
Consolidation	Effusion	Fibrosis	63.46
Edema	Effusion	Pleural Thickening	63.27
Effusion	Emphysema	Mass	63.27
Edema	Emphysema	Nodule	63.22
Fibrosis	Mass	Pneumothorax	63.18
Atelectasis	Emphysema	Mass	63.08
Effusion	Fibrosis	Pleural Thickening	63.07
Atelectasis	Nodule	Pneumothorax	62.73
Edema	Fibrosis	Nodule	62.68
Cardiomegaly	Consolidation	Mass	62.66

Atelectasis	Emphysema	Pneumothorax	62.54
Edema	Emphysema	Mass	62.52
Cardiomegaly	Consolidation	Pleural Thickening	62.44
Edema	Fibrosis	Mass	62.28
Atelectasis	Mass	Pneumothorax	62.22
Cardiomegaly	Consolidation	Pneumothorax	62.17
Fibrosis	Nodule	Pneumothorax	62.12
Edema	Effusion	Pneumothorax	61.95
Effusion	Emphysema	Pneumothorax	61.93
Mass	Pleural Thickening	Pneumothorax	61.90
Consolidation	Effusion	Hernia	61.65
Consolidation	Fibrosis	Mass	61.44
Edema	Fibrosis	Pleural Thickening	61.40
Cardiomegaly	Mass	Pleural Thickening	61.23
Emphysema	Mass	Pneumothorax	60.96
Atelectasis	Effusion	Mass	60.87
Atelectasis	Consolidation	Emphysema	60.78
Cardiomegaly	Consolidation	Nodule	60.76
Cardiomegaly	Emphysema	Pneumothorax	60.75
Consolidation	Hernia	Pneumothorax	60.75
Atelectasis	Effusion	Nodule	60.61
Effusion	Nodule	Pneumothorax	60.43
Effusion	Mass	Nodule	60.42
Effusion	Nodule	Pleural Thickening	60.38
Atelectasis	Emphysema	Pleural Thickening	60.29
Emphysema	Nodule	Pneumothorax	60.28
Edema	Emphysema	Pleural Thickening	60.24
Consolidation	Emphysema	Fibrosis	60.23

Edema	Mass	Pneumothorax	60.13
Effusion	Mass	Pneumothorax	59.88
Edema	Pleural Thickening	Pneumothorax	59.81
Hernia	Mass	Nodule	59.54
Atelectasis	Pleural Thickening	Pneumothorax	59.28
Effusion	Emphysema	Pleural Thickening	59.01
Hernia	Mass	Pleural Thickening	59.01
Consolidation	Edema	Emphysema	58.52
Cardiomegaly	Nodule	Pleural Thickening	58.36
Consolidation	Effusion	Emphysema	58.31
Atelectasis	Effusion	Pleural Thickening	58.17
Atelectasis	Consolidation	Nodule	57.92
Atelectasis	Effusion	Pneumothorax	57.82
Emphysema	Nodule	Pleural Thickening	57.73
Consolidation	Edema	Effusion	57.72
Atelectasis	Mass	Pleural Thickening	57.66
Effusion	Pleural Thickening	Pneumothorax	57.57
Consolidation	Edema	Nodule	57.52
Emphysema	Hernia	Pleural Thickening	57.42
Effusion	Mass	Pleural Thickening	57.41
Consolidation	Effusion	Pneumothorax	57.34
Consolidation	Edema	Emphysema	57.09
Emphysema	Mass	Pleural Thickening	56.92
Emphysema	Pleural Thickening	Pneumothorax	56.89
Atelectasis	Nodule	Pleural Thickening	56.75
Edema	Mass	Pleural Thickening	56.01
Atelectasis	Consolidation	Effusion	55.98
Consolidation	Emphysema	Mass	55.97

Emphysema	Hernia	Pneumothorax	55.97
Mass	Nodule	Pneumothorax	55.97
Consolidation	Effusion	Nodule	55.92
Emphysema	Fibrosis	Mass	55.63
Consolidation	Effusion	Mass	55.50
Atelectasis	Consolidation	Pneumothorax	55.38
Atelectasis	Emphysema	Nodule	55.30
Atelectasis	Consolidation	Mass	55.19
Consolidation	Fibrosis	Nodule	55.02
Edema	Nodule	Pleural Thickening	54.94
Consolidation	Emphysema	Nodule	54.89
Emphysema	Fibrosis	Nodule	54.79
Atelectasis	Consolidation	Pleural Thickening	54.78
Cardiomegaly	Mass	Nodule	54.73
Atelectasis	Mass	Nodule	54.34
Consolidation	Edema	Pleural Thickening	54.30
Edema	Emphysema	Pneumothorax	54.22
Consolidation	Effusion	Pleural Thickening	54.19
Consolidation	Fibrosis	Pneumothorax	54.15
Consolidation	Edema	Mass	53.96
Consolidation	Fibrosis	Pleural Thickening	53.01
Consolidation	Edema	Pneumothorax	52.74
Emphysema	Fibrosis	Pneumothorax	52.63
Consolidation	Emphysema	Pleural Thickening	51.87
Emphysema	Fibrosis	Pleural Thickening	51.84
Fibrosis	Nodule	Pleural Thickening	51.59
Consolidation	Mass	Pneumothorax	51.55
Consolidation	Mass	Pleural Thickening	50.58

Nodule	Pleural Thickening	Pneumothorax	50.58
Consolidation	Nodule	Pneumothorax	50.49
Edema	Mass	Nodule	50.28
Fibrosis	Pleural Thickening	Pneumothorax	50.00
Consolidation	Nodule	Pleural Thickening	49.53
Mass	Nodule	Pleural Thickening	48.83
Consolidation	Emphysema	Pneumothorax	48.68
Fibrosis	Mass	Pleural Thickening	48.64
Consolidation	Pleural Thickening	Pneumothorax	48.01
Fibrosis	Mass	Nodule	47.52
Emphysema	Mass	Nodule	47.25
Consolidation	Mass	Nodule	42.96

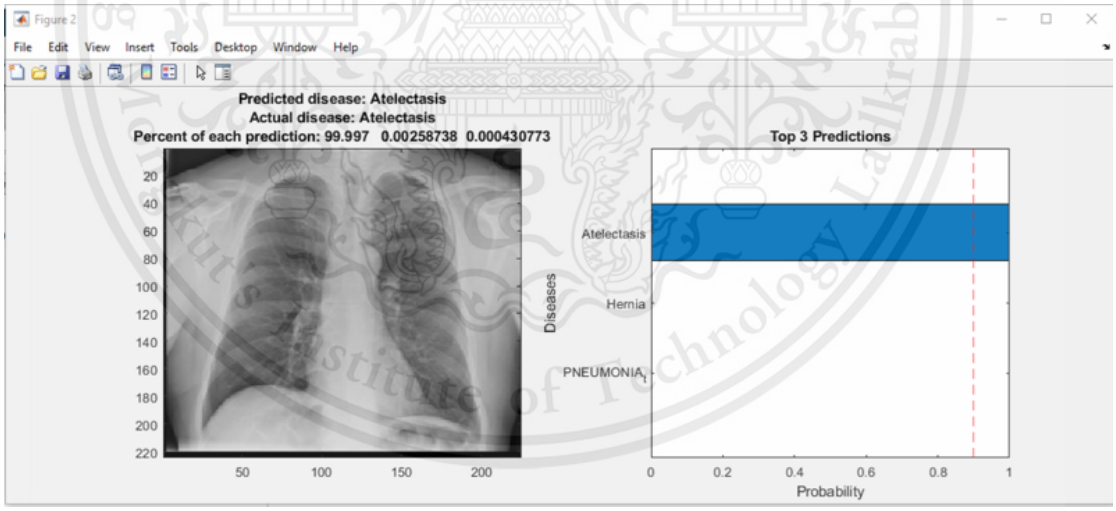

Figure 4.12 Bar graph representing the top predictions of a network that gained accuracy over the set threshold

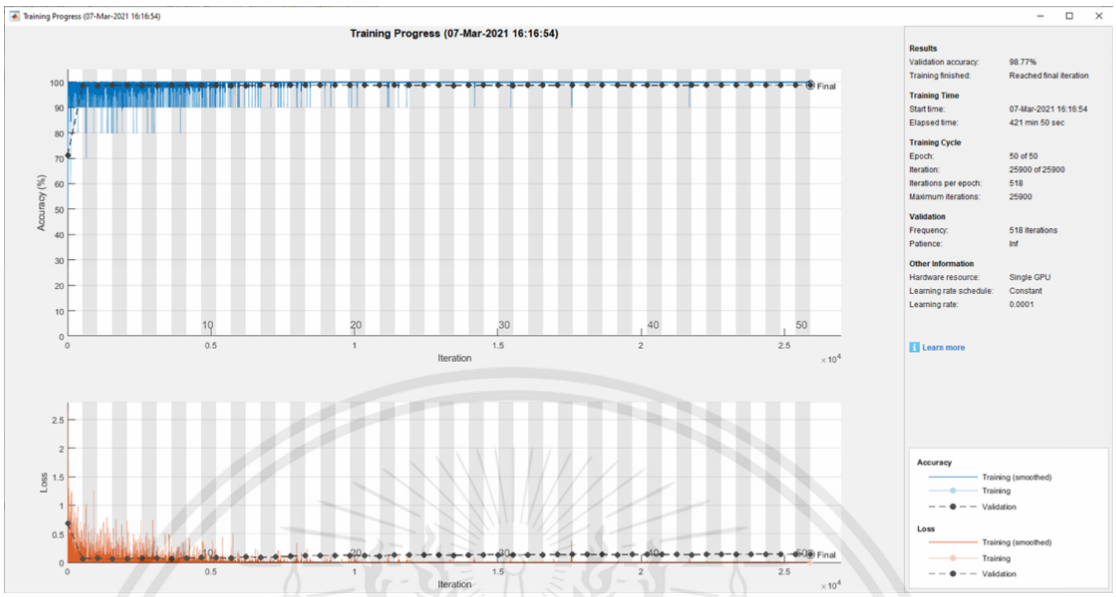

Figure 4.13 Accuracy and loss trends during training Atelectasis, Hernia and Pneumonia

```
confMat =

    0.9980    0.0020         0
    0.8824    0.1176         0
         0         0    1.0000
```

Figure 4.14 Numeric confusion matrix of Atelectasis, Hernia and Pneumonia

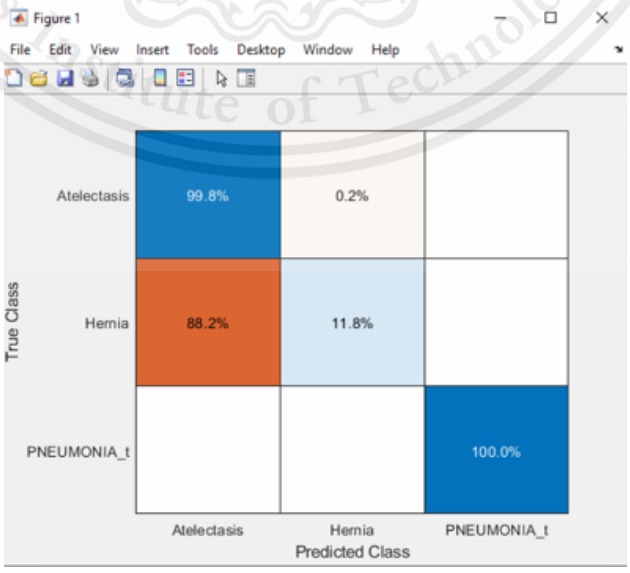

Figure 4.15 Confusion matrix chart of Atelectasis, Hernia and Pneumonia

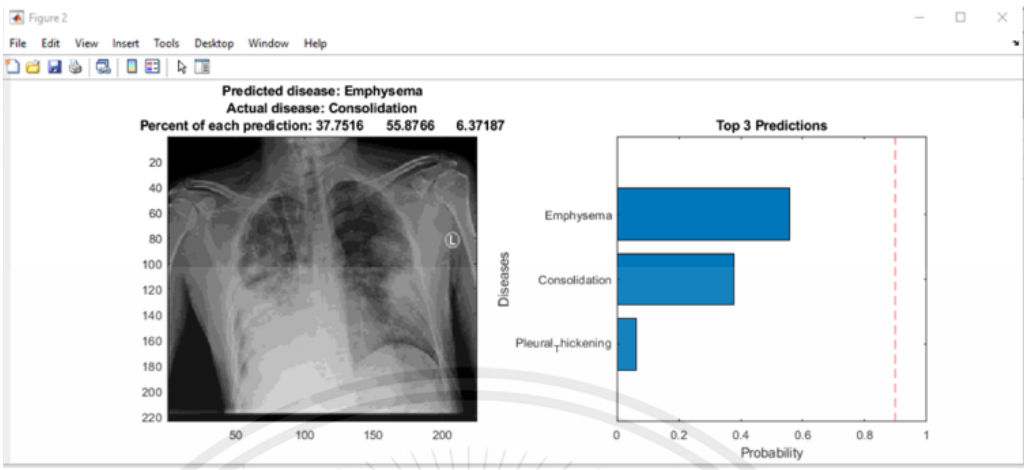

Figure 4.16 Bar graph representing the top predictions of network that gained accuracy lower the set threshold

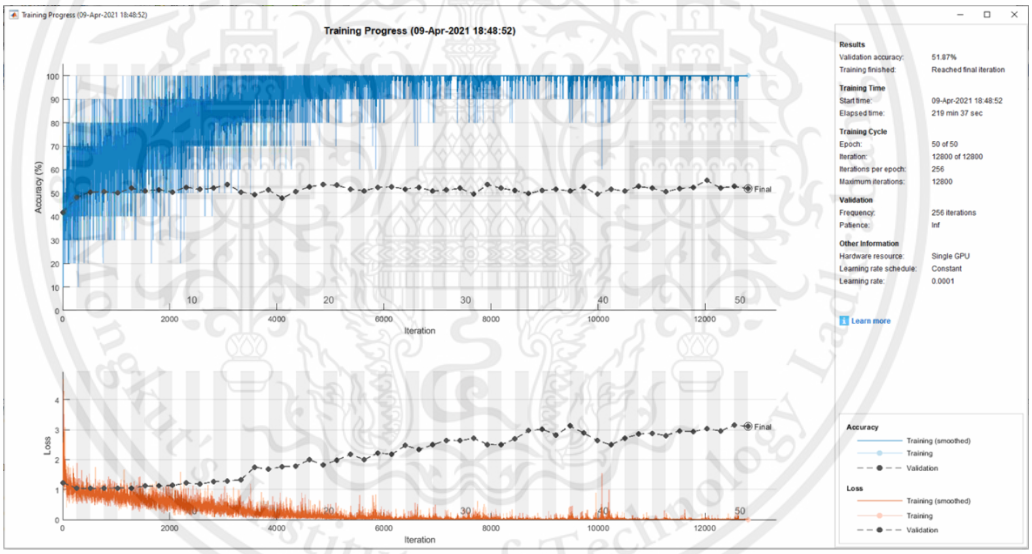

Figure 4.17 Accuracy and loss trends during training Consolidation, Emphysema and Pleural thickening

```
confMat =

    0.5766    0.1573    0.2661
    0.2586    0.4540    0.2874
    0.2773    0.2182    0.5045
```

Figure 4.18 Numeric confusion matrix of Consolidation, Emphysema and Pleural thickening

Figure 4.19 Confusion matrix chart of Consolidation, Emphysema and Pleural thickening

According to Table 4.17, it shown that after training the networks and validating the data, the percentage of accuracy was significantly higher than those previous methods even though it was trained with multiclass under the same circumstance. The accuracy achieved was around 40 to 98 percent depending on chosen disease pair.

Apart from the results containing in Table 4.17, another thing that also need to be discussed is the accuracy and loss trends of both gained the validation accuracy over the set threshold and lower the set threshold networks which we focused on the results from Atelectasis, Hernia and Pneumonia network versus Consolidation, Emphysema and Pleural thickening network.

Referring to Figure 4.13 and Figure 4.17, it indicated that the trends of accuracy and loss were completely different which, in Figure 4.13, the trends of training and validation tended to move along together since the beginning of the process, whereas another tended to move separately since the first stage of the training process leading to significantly lower validation accuracy and various values resulted in numeric confusion matrix as shown in Figure 4.17, Figure 4.18 and Figure 4.19, respectively.

Observing the trends on both the accuracy weight and the loss weight, the percentage of accuracy is high when matching Pneumonia with any other diseases. The combination of illness that gave us the highest accuracy was from Effusion, Hernia, and Pneumonia network which is around 98 percent. Moreover, the combination of the

other cases when matching with Pneumonia also constantly results the percentage of accuracy in the range of 80 to 90 percent.

Furthermore, after finished ranked all of the validation accuracy in descending order as shown in Table 4.17, it indicated that there were 38 networks that gave validation accuracy equal to and higher than the threshold that we set which was at 95 percent. Thereby, these networks will be further done on the pipeline system.

4.6 Results from pipeline system

After trained networks with every algorithm, GoogLeNet architecture could perform and give the most accurate predictions when comparing with the other architectures. Moreover, it could also give the highest percentage of validation accuracy which over the set threshold. Owing to this summation, the networks that were chosen to use with the pipeline system were only from GoogLeNet architecture.

We, firstly, started the process with saving and collecting every network that could over the set threshold of 95% in one folder. Secondly, sorted images that were split for the validation tasks which every disease contained 50 images except Consolidation, Effusion, Hernia and Mass which contained 14 images, 40 images and 9 images, respectively. Thirdly, run every saved network in a loop to count the number of times of each disease that occurred in the system as shown in Table 4.18. Then, calculated each disease probability of each network to know the percentage outputs and averaged the results by divided with the number of images of each disease which these results will be collected in one table. This table was also known as a confusion matrix chart which the actual class corresponded to the columns of confusion matrix, whereas the predicted class corresponded to the rows of confusion matrix. Moreover, the shaded diagonal cells and off-diagonal cells corresponded to the correctly and incorrectly classified predictions, respectively shown in Table 4.19.

Table 4.18 Number of times and images of each disease

List of diseases	Number of times	Number of validate images
Normal	6	50

Atelectasis	3	50
Cardiomegaly	5	50
Consolidation	3	14
Edema	4	50
Effusion	4	40
Emphysema	4	50
Fibrosis	6	50
Hernia	22	45
Mass	3	9
Nodule	3	50
Pleural thickening	3	50
Pneumonia	8	50
Pneumothorax	4	50

Table 4.19 Confusion matrix chart

Actual / Predicted	Normal	Atelectasis	Cardiomegaly	Consolidation	Edema	Effusion	Emphysema	Fibrosis	Hernia	Mass	Nodule	Pleural Thickening	Pneumonia	Pneumothorax
Normal	0.9567	0.0099	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Atelectasis	0.0000	0.9564	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1947	0.0000	0.0000	0.0000	0.0000	0.0000
Cardiomegaly	0.0000	0.0000	0.9858	0.0000	0.0000	0.0000	0.0000	0.1599	0.1446	0.0000	0.0000	0.0000	0.0000	0.0000
Consolidation	0.0000	0.0000	0.0000	0.9965	0.0000	0.0000	0.0000	0.0000	0.1444	0.0000	0.0000	0.0000	0.0000	0.0000
Edema	0.0000	0.0000	0.0000	0.0000	0.9670	0.0000	0.0000	0.0497	0.0249	0.0000	0.0000	0.0000	0.0000	0.0000
Effusion	0.0000	0.0000	0.0000	0.0000	0.0000	0.9797	0.0000	0.2793	0.1448	0.0000	0.0000	0.0000	0.0033	0.0000
Emphysema	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9972	0.0000	0.1548	0.0000	0.0000	0.0000	0.0000	0.0000
Fibrosis	0.0000	0.0000	0.0000	0.0000	0.0300	0.0398	0.0000	0.7328	0.1593	0.0000	0.0000	0.0000	0.0000	0.5241
Hernia	0.0000	0.0000	0.0000	0.0000	0.0050	0.0000	0.0000	0.0098	0.7631	0.0099	0.0000	0.0000	0.0000	0.0000
Mass	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1597	0.9195	0.0000	0.0000	0.0050	0.0000
Nodule	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0999	0.0000	0.9865	0.0000	0.0000	0.0000
Pleural Thickening	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1477	0.0000	0.0000	0.9999	0.0000	0.0000
Pneumonia	0.0232	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0100	0.0000	0.0000	0.9826	0.0000
Pneumothorax	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0299	0.0800	0.0000	0.0000	0.0000	0.0000	0.7997
Total	0.9799	0.9663	0.9858	0.9965	1.0020	1.0195	0.9972	1.2614	2.2179	0.9394	0.9865	0.9999	0.9909	1.3238

According to Table 4.18, it shown that the average of the number of times counted were approximate to 4 times per disease except Hernia disease that got the

highest value which was 22 times. The reason why it was counted the most may because it was one of the diseases that obviously showed the key findings on CXR images, therefore, this disease appeared in many networks.

In Table 4.19, to sum up, after validated the CXR images with the trained networks, it indicated that most of the validation accuracy were over the set threshold which the average of the validation accuracy was at 98.08 percent. Only 4 classes that could not over the threshold in validation process which are Fibrosis, Hernia, Mass and Pneumothorax leading to the decreasing of overall validation accuracy. The reason of the dropping of the validation accuracy might be either from mislabeling diseases or mixing of various diseases in one folder which the average of validation accuracy after including these diseases were 93.02 percent.

CHAPTER 5
CONCLUSION

5.1 Project Summary

This dissertation aimed to develop effective convolutional neural network-based software program with pipeline system technique for the rapid diagnosis in medical examinations. Based on a quantitative and qualitative analysis in response to campaign datasets, it can be concluded that the architecture as well as numbers of parameters are important factors to consider when designing to work efficiently with the dataset provided to be trained.

In conclusion, the results from training with various network architectures including simple convolutional neural network, AlexNet architecture, ResNet-18 architecture, ResNet-50 architecture and GoogLeNet architecture and testing can be summarized as follows:

5.1.1 In the dataset preparation process, techniques for reducing the data overlapping, for adding non-directional sampling data, and for adding image information by rotating the image are techniques that optimize datasets for training neural networks which help to increase accuracy.

5.1.2 From the analysis of the results trained with the neural network models, it was found that the convolutional neural network with increasing number of layers and/or classes resulted in a decrease in the accuracy of training. Moreover, when trained with other kinds of CNN architecture including AlexNet, ResNet-18, ResNet-50 and GoogleNet, it was found that the GoogleNet architecture yielded the most accurate results with the highest percentage of accuracy at 98.82.

5.1.3 In training and practicing anomaly prediction using chest X-ray dataset that contains 14 classes identified as 13 diseases and a normal condition with the use of the GoogleNet architecture for all classes, the accuracy result was found at an extremely low percentage. Therefore, the test was conducted separately by practice classes: Disease 1, Disease 2, and Disease 3, with a 14 choose 3 matchmaking technique. The training accuracy was greatly

increased over with 13 classes of practice. In which, there were 38 networks that could perform the percentage of accuracy over the set threshold at 95 percent. Hence, all 38 networks were included for further use in testing with pipeline tecnique.

5.1.4 Based on the pipeline system predictive characterization test, the results showed that predictive behavior was largely in line with the physician's diagnosis as most of the networks give validation accuracy over 95 percent; however, the average of overall true labels class is 93.02 percent.

5.2 Project Review

The neural network training with different architecture, when practiced with increasing data classes results in the same direction as of which the accuracy decreases due to the quality of the chest X-ray images from the databases as the dataset for training and validation used in practice have an inaccurate identification of all abnormalities displayed on X-ray images. This resulting in neural network training learning defects and degradation of accuracy. However, it is also possible to add or modify a data of X-ray images that clearly identify the anomaly to correct the cognitive defects of the neural network.

5.3 Recommendations

According to the result discussion and project review, the project can further be developed as follows:

5.3.1 In the preparation of data used in the practice of neural networks, the X-ray datasets should be obtained from additional sources in order to obtain a wide variety of X-ray datasets.

5.3.2 The X-ray images should be reviewed by a medical professional in order to clearly identify any abnormalities on the chest X-ray before being trained on the neural network.

5.3.3 In the practice of neural networks, the structure of the neural network should be suitable for the dataset.

REFERENCES

Halliday, A. (2017, September 15). Marie Curie Invented Mobile X-Ray Units to Help Save Wounded Soldiers in World War I. Retrieved November 20, 2020, from https://www.openculture.com/2017/09/marie-curie-invented-mobile-x-ray-units-to-help-save-wounded-soldiers-in-world-war-i.html

Staff, C. H. (2020, September 7). Atelectasis. (Chersery Home) Retrieved from Chersery Home: https://www.thesenizens.com/%E0%B8%A3%E0%B8%B2%E0%B8%A2%E0%B8%A5%E0%B8%B0%E0%B9%80%E0%B8%AD%E0%B8%B5%E0%B8%A2%E0%B8%94/%E0%B8%9B%E0%B8%AD%E0%B8%94%E0%B9%81%E0%B8%9F%E0%B8%9A_Und_(Atelectasis)_Und_-_Und_Chersery_Und_Home#:~:text=%E0%B9%80%E0%B8%9B%E0%B9%87

Wang, X., Yifan Peng, Le Lu, Zhiyong Lu, & Ronald M. Summers. (2018, January 12). TieNet: Text-Image Embedding Network for Common Thorax Disease Classification and Reporting in Chest X-rays. Retrieved from arXiv: https://arxiv.org/abs/1801.04334

Charunwikon, C. (n.d.). Cardiomegaly. (Chularat 3 International Hospital) Retrieved from Chularat 3 International Hospital: https://www.chularat3.com/knowledge_detail.php?lang=en&id=432

Staff, P. (n.d.). Pleural Effusion. (Pobpad) Retrieved from Pobpad: https://www.pobpad.com/pleural-effusion#:~:text=Pleural%20Effusion%20%E0%B8%AB%E0%B8%A3%E0%B8%B7%E0%B8%AD%20%E0%B8%A0%E0%B8%B2%E0%B8%A7%E0%B8%B0%E0%B8%99%E0%B9%89%E0%B8%B3,%E0%B8%AB%E0%B8%A3%E0%B8%B7%E0%B8%AD%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%AA%E0%B8%B6%E

Eldridge, L. (2021, April 15). Possible Causes of a Lung Mass. (Verywell Health) Retrieved from Very Well Health: https://www.verywellhealth.com/lung-mass-possible-causes-and-what-to-expect-

2249388#:~:text=Brianna%20Gilmartin%2C%20Verywell-,Definition,diameter%20are%20considered%20lung%20nodules.

MaryAnn De Pietro, & Cameron White. (2019, September 30). What to know about lung nodules. (Medical News Today) Retrieved from Medical News Today: https://www.medicalnewstoday.com/articles/317531#:~:text=A%20lung%20nodule%20is%20a,cancerous%20and%20can%20grow%20quickly.

Staff, M. C. (2020, June 13). Pneumonia. (Mayo Clinic) Retrieved from Mayo Clinic: https://www.mayoclinic.org/diseases-conditions/pneumonia/symptoms-causes/syc-20354204

Staff, W. (n.d.). Pneumothorax. (Wongkarnpat) Retrieved from Wongkarnpat Plus Media: http://www.wongkarnpat.com/viewpat.php?id=812#:~:text=Pneumothorax%20%E0%B8%AB%E0%B8%A1%E0%B8%B2%E0%B8%A2%E0%B8%96%E0%B8%B6%E0%B8%87%20%E0%B8%A0%E0%B8%B2%E0%B8%A7%E0%B8%B0%E0%B8%97%E0%B8%B5%E0%B9%88,%E0%B8%AD%E0%B8%A2%E0%B8%B9%E0%B9%88%E0%B9%80%E0%B8%94%E

Marcin, J., & Nancy Moyer. (2018, September 16). Lung Consolidation: What It Is and How It's Treated. (Healthline) Retrieved from Healthline: https://www.healthline.com/health/lung-consolidation#outlook

Staff, P. (n.d.). Edema. (Pobpad) Retrieved from Pobpad: https://www.pobpad.com/%E0%B8%AD%E0%B8%B2%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%9A%E0%B8%A7%E0%B8%A1%E0%B8%99%E0%B9%89%E0%B8%B3-edema#:~:text=%E0%B8%AD%E0%B8%B2%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%9A%E0%B8%A7%E0%B8%A1%E0%B8%99%E0%B9%89%E0%B8%B3%20(Edema)%20%E0%B9

Staff, P. (n.d.). Emphysema. (Pobpad) Retrieved from Pobpad: https://www.pobpad.com/%E0%B8%96%E0%B8%B8%E0%B8%87%E0%B8%A5%E0%B8%A1%E0%B9%82%E0%B8%9B%E0%B9%88%E0%B8%87%E0%B8%9E%E0%B8%AD%E0%B8%87#:~:text=%E0%B8%96%E0%B8%B8%E0%B8%87%E0%B8%A5%E0%B8%A1%E0%B9%82%E0%B8%9

B%E0%B9%88%E0%B8%87%E0%B8%9E%E0%B8%AD%E0%B8%87%20(Emp

Staff, M. C. (2018, March 6). Pulmonary fibrosis. (Mayo Clinic) Retrieved from Mayo Clinic: https://www.mayoclinic.org/diseases-conditions/pulmonary-fibrosis/symptoms-causes/syc-20353690

Molinari, L., & Charlevois, A. (2021, February 26). Pleural Thickening. (Mesothelioma.com) Retrieved from Mesothelioma: https://www.mesothelioma.com/asbestos-cancer/pleural-thickening/#:~:text=Is%20Pleural%20Thickening%3F-,What%20Is%20Pleural%20Thickening%3F,such%20as%20malignant%20pleural%20mesothelioma.

Staff, U. o. (n.d.). Lung Hernia. (University of Rochester Medical Center Rochester) Retrieved from University of Rochester Medical Center Rochester: https://www.urmc.rochester.edu/encyclopedia/content.aspx?contenttypeid=22&contentid=lunghernia

Staff, M. C. (2020, May 2). Mayo Clinic. Retrieved from Chest X-rays: https://www.mayoclinic.org/tests-procedures/chest-x-rays/about/pac-20393494

Litjens, G., Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, . . . Clara I. Sánchez. (2017, June 4). A Survey on Deep Learning in Medical Image Analysis. Retrieved from arXiv: https://arxiv.org/abs/1702.05747

LeCun, Y., Yoshua Bengio, & Geoffrey Hinton. (2015, May 27). Deep learning. Retrieved from Nature: https://www.nature.com/articles/nature14539

Serrano, W. (2017, February 6). Smart Internet Search with Random Neural Networks. Retrieved from Cambridge University Press: https://www.cambridge.org/core/journals/european-review/article/abs/smart-internet-search-with-random-neural-networks/188045D7749594B49BFA451E1366EEDC

Wang, D., Aditya Khosla, Rishab Gargeya, Humayun Irshad, & Andrew H. Beck. (2016, June 18). Deep Learning for Identifying Metastatic Breast Cancer. Retrieved from arXiv: https://arxiv.org/abs/1606.05718

Gulshan, V., Lily Peng, Marc Coram, Martin C. Stumpe, Derek Wu, Arunachalam Narayanaswamy, . . . Dale R. Webster. (2016, December 13). Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. Retrieved from JAMA Network: https://jamanetwork.com/journals/jama/fullarticle/2588763

Esteva, A., Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, & Sebastian Thrun. (2017, January 25). Dermatologist-level classification of skin cancer with deep neural networks. Retrieved from Nature: https://www.nature.com/articles/nature21056

LeCun, Y., Bernhard Boser, John Denker, Donnie Henderson, R. Howard, Wayne Hubbard, & Lawrence Jackel. (1990, June 2). Handwritten digit recognition with a back-propagation network. Retrieved from ACM Digital Library: https://dl.acm.org/doi/10.5555/109230.109279

Jan, B., Haleem Farman, Murad Khan, Muhammad Imran, Ihtesham Ul Islam, Awais Ahmad, . . . Gwanggil Jeon. (2019, May). Deep learning in big data Analytics: A comparative study. Retrieved from ScienceDirect: https://www.sciencedirect.com/science/article/abs/pii/S0045790617315835?via%3Dihub

Krizhevsky, A., Ilya Sutskever, & Geoffrey E. Hinton. (2017, May). ImageNet classification with deep convolutional neural networks. Retrieved from Communications of the ACM: https://dl.acm.org/doi/10.1145/3065386

He, K., Xiangyu Zhang, Shaoqing Ren, & Jian Sun. (2015, December 10). Deep Residual Learning for Image Recognition. Retrieved from arXiv: https://arxiv.org/abs/1512.03385

Huang, G., Zhuang Liu, Laurens van der Maaten, & Kilian Q. Weinberger. (2016, August 25). Densely Connected Convolutional Networks. Retrieved from arXiv: https://arxiv.org/abs/1608.06993

Narkhede, S. (2018, May 9). Understanding Confusion Matrix. (Towards Data Science) Retrieved from Towards Data Science: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Sheng, B., Oscar Moroni Moosman, Borja del Pozo Cruz, Jesús Del Pozo-Cruz, & Rosa M. Alfonso-Rosa. (2020, March 15). A comparison of different machine learning algorithms, types and placements of activity monitors for physical activity classification. Retrieved from ScienceDirect: https://www.sciencedirect.com/science/article/abs/pii/S0263224120300178?via%3Dihub

He, K., Xiangyu Zhang, Shaoqing Ren, & Jian Sun. (2015, February 6). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. Retrieved from arXiv: https://arxiv.org/abs/1502.01852

Esteva, A., Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, & Sebastian Thrun. (2017, January 25). Dermatologist-level classification of skin cancer with deep neural networks. Retrieved from Nature: https://www.nature.com/articles/nature21056

Ting, D. S., Carol Yim-Lui Cheung, Gilbert Lim, & et al. (2017, December 12). Development and Validation of a Deep Learning System for Diabetic Retinopathy and Related Eye Diseases Using Retinal Images From Multiethnic Populations With Diabetes. Retrieved from JAMA: https://jamanetwork.com/journals/jama/fullarticle/2665775

Lindsey, R., Aaron Daluiski, Sumit Chopra, Michael Mozer, Serge Sicular, Douglas Hanel, . . . Hollis Potter. (2018, November 6). Deep neural network improves fracture detection by clinicians. Retrieved from Proceedings of the National Academy of Sciences of the United States of America: https://www.pnas.org/content/115/45/11591

Peng, Y., Shazia Dharssi, Qingyu Chen, Tiarnan D. Keenan, Elvira Agrón, Wai T. Wong, . . . Zhiyong Lu. (2019, April 1). DeepSeeNet: A Deep Learning Model for Automated Classification of Patient-based Age-related Macular Degeneration Severity from Color Fundus Photographs. Retrieved from Elsevier Inc.: https://www.aaojournal.org/article/S0161-6420(18)32185-7/fulltext#%20

Yates, E., L.C. Yates, & H. Harvey. (2018, September 9). Machine learning “red dot”: open-source, cloud, deep convolutional neural networks in chest radiograph

binary normality classification. Retrieved from ScienceDirect: https://www.sciencedirect.com/science/article/abs/pii/S000992601830206X

Dunnmon, J. A., Darvin Yi, Curtis P. Langlotz, Christopher Ré, Daniel L. Rubin, & Matthew P. Lungren. (2018, November 13). Assessment of Convolutional Neural Networks for Automated Classification of Chest Radiographs. Retrieved from Radiological Society of North America: https://pubs.rsna.org/doi/10.1148/radiol.2018181422

Weisstein, E. W. (2021, April 29). Binomial Coefficient. Retrieved May 4, 2021, from https://mathworld.wolfram.com/BinomialCoefficient.html

Roberts, Surabhi Datta , & Kirk. (2020, June 9). A dataset of chest X-ray reports annotated with Spatial Role Labeling annotations. Retrieved from Mendeley Data: https://data.mendeley.com/datasets/yhb26hfz8n/1

Bustos, A., Antonio Pertusa, Jose-MariaSalinas, & Maria de la Iglesia-Vayá. (2020, August 20). PadChest: A large chest x-ray image dataset with multi-label annotated reports. Retrieved from ScienceDirect: https://www.sciencedirect.com/science/article/pii/S1361841520301614?via%3Dihub

APPENDICES

APPENDIX A
PROGRAM FOR RESIZING IMAGES

```
clc;
clear all;
close all;

digitDatasetPath = fullfile('/Users/minnie/Documents/MATLAB/summer
training/readyforval/PNEUMONIA_t');
imds = imageDatastore(digitDatasetPath, ...
   'IncludeSubfolders',true,'LabelSource','foldernames');

img2=zeros(224,224,3); %
for i=1:length(imds.Labels)
   img=readimage(imds,i);
   % disp(i)
   img1 = imresize(img,[224 224]);
   if size(img1,3)==3
      img1 = rgb2gray(img1);
   end
   img2 = cat(3,img1,img1,img1);
   imwrite(img2,cell2mat(imds.Files(i)))
end
```

APPENDIX B
PROGRAM FOR SORTING IMAGES

```
clc
close all
T = readtable('PADCHEST_labels.csv');

for ii= 45219:48233
   disp(ii);
   filename=char(table2array(T(ii,2)));
   A=imresize((imread(['/Users/minnie/Documents/MATLAB/summer training/14/'
filename])),[224 224]);
if ~contains(['/Users/minnie/Documents/MATLAB/summer training/'
char(table2array(T(ii,2))) '\' filename],'<undefined>')
if isequal(char(table2array(T(ii,32))),'Normal') %go to label col
   %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
   imwrite(A,['/Users/minnie/Documents/MATLAB/summer training/sort/Normal/'
filename]);

elseif isequal(char(table2array(T(ii,32))),'Atelectasis')
   %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
   imwrite(A,['/Users/minnie/Documents/MATLAB/summer training/sort/
Atelectasis/' filename]);

elseif isequal(char(table2array(T(ii,32))),'Cardiomegaly')
   %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
   imwrite(A,['/Users/minnie/Documents/MATLAB/summer
training/sort/Cardiomegaly/' filename]);

elseif isequal(char(table2array(T(ii,32))),'Consolidation')
   %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
```

```
    imwrite(A,['/Users/minnie/Documents/MATLAB/summer
training/sort/Consolidation/' filename]);

elseif isequal(char(table2array(T(ii,32))),'Edema')
    %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
        imwrite(A,['/Users/minnie/Documents/MATLAB/summer training/sort/ Edema/'
filename]);

elseif isequal(char(table2array(T(ii,32))),'Effusion')
    %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
    imwrite(A,['/Users/minnie/Documents/MATLAB/summer training/sort/ Effusion/'
filename]);

elseif isequal(char(table2array(T(ii,32))),'Emphysema')
    %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
        imwrite(A,['/Users/minnie/Documents/MATLAB/summer training/sort/
Emphysema/' filename]);

elseif isequal(char(table2array(T(ii,32))),'Fibrosis')
    %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
    imwrite(A,['/Users/minnie/Documents/MATLAB/summer training/sort/ Fibrosis/'
filename]);

elseif isequal(char(table2array(T(ii,32))),'Hernia')
    %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
    imwrite(A,['/Users/minnie/Documents/MATLAB/summer training/sort/Hernia/'
filename]);

elseif isequal(char(table2array(T(ii,32))),'Mass')
    %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
    imwrite(A,['/Users/minnie/Documents/MATLAB/summer training/sort/Mass/'
filename]);
```

```
elseif isequal(char(table2array(T(ii,32))),'Nodule')
   %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
   imwrite(A,['/Users/minnie/Documents/MATLAB/summer training/sort/Nodule/'
filename]);

elseif isequal(char(table2array(T(ii,32))),'Pleural thickening')
   %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
   imwrite(A,['/Users/minnie/Documents/MATLAB/summer training/sort/Pleural
thickening/' filename]);

elseif isequal(char(table2array(T(ii,32))),'Pneumothorax')
   %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
   imwrite(A,['/Users/minnie/Documents/MATLAB/summer
training/sort/Pneumothorax/' filename]);

elseif isequal(char(table2array(T(ii,32))),'Pneumonia')
   %char(table2array(DataEntry2017v2020(ii,2)))=='No Finding'
   imwrite(A,['/Users/minnie/Documents/MATLAB/summer
training/sort/Pneumonia/' filename]);

end
end
```

APPENDIX C
PROGRAM FOR RUNNING SIMPLE CNN

```
clc;
clear all;
close all;

outputFolder = fullfile('/Users/minnie/Documents/MATLAB/summer training');

rootFolder = fullfile(outputFolder, 'chest_xray');

categories = {'NORMAL', 'PNEUMONIA', 'Cardiomegaly', 'Consolidation', 'Edema',
'Effusion', 'Emphysema', ...
   'Fibrosis', 'Mass', 'Hernia', 'Nodule', 'Pleural_Thickening', 'Pneumothorax'};

imds = imageDatastore(fullfile(rootFolder,categories), ...
       'IncludeSubfolders',true,'LabelSource','foldernames');

tbl = countEachLabel(imds)

minSetCount = min(tbl{:,2})

imds = splitEachLabel(imds, minSetCount, 'randomize');
countEachLabel(imds)

[imdsTrain,imdsValidation] = splitEachLabel(imds, 0.75,'randomize');

inputSize = [512 512 1];

augmentedTraining = augmentedImageDatastore(inputSize,
imdsTrain,'ColorPreprocessing','rgb2gray');
```

```
augmentedValidation = augmentedImageDatastore(inputSize,
imdsValidation,'ColorPreprocessing','rgb2gray');

img = readimage(imds,1);
size(img)

inputSize = [512 512 1];
numClasses = 13;

layers = [
        imageInputLayer(inputSize)

        convolution2dLayer(5,8, 'Padding', 'same')
        batchNormalizationLayer
        reluLayer

        maxPooling2dLayer(5,'Stride',2)

        convolution2dLayer(5,16,'Padding','same')
        batchNormalizationLayer
        reluLayer

        maxPooling2dLayer(5,'Stride',2)

        convolution2dLayer(5,32,'Padding','same')
        batchNormalizationLayer
        reluLayer

        maxPooling2dLayer(5,'Stride',2)

        convolution2dLayer(5,64,'Padding','same')
        batchNormalizationLayer
```

```
    reluLayer

    maxPooling2dLayer(5,'Stride',2)

    convolution2dLayer(5,128,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(5,'Stride',2)

    convolution2dLayer(5,256,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(5,'Stride',2)

    convolution2dLayer(5,512,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(5,'Stride',2)

    convolution2dLayer(5,512,'Padding','same')
    batchNormalizationLayer
    reluLayer

    fullyConnectedLayer(numClasses)
    SoftmaxLayer
    classificationLayer()];

options = trainingOptions('sgdm', ...
    'InitialLearnRate',0.001, ...
```

```
    'MaxEpochs',10, ...
    'Shuffle','every-epoch', ...
    'ValidationData',augmentedValidation, ...
    'ValidationFrequency',30, ...
    'Verbose',false, ...
    'Plots','training-progress');

net = trainNetwork(augmentedTraining,layers,options);
net.Layers

YPred = classify(net,augmentedValidation);
YValidation = imdsValidation.Labels;

accuracy = sum(YPred == YValidation)/numel(YValidation)

img1 = readimage(imdsValidation,27);

ds = augmentedImageDatastore(inputSize, ...
     img1, 'ColorPreprocessing', 'rgb2gray');    %image set

actualLabel = imdsValidation.Labels(270);
predictedLabel = net.classify(ds);
imshow(img1);
title(['Predicted: ' char(predictedLabel) ', Actual: ' char(actualLabel)])

save(['train_13diseases_simpleCNN' '.mat'], 'net');
```

APPENDIX D
PROGRAM FOR RUNNING ALEXNET ARCHITECTURE

```
clc;
clear all;
close all;

outputFolder = fullfile('/Users/minnie/Documents/MATLAB/summer training');

rootFolder = fullfile(outputFolder, '227folder');

categories = {'NORMAL', 'PNEUMONIA', 'Cardiomegaly', 'Consolidation', 'Edema',
'Effusion', 'Emphysema'};

imds = imageDatastore(fullfile(rootFolder,categories), ...
    'IncludeSubfolders',true,'LabelSource','foldernames');

tbl = countEachLabel(imds)
minSetCount = min(tbl{:,2})

imds = splitEachLabel(imds, minSetCount, 'randomize');
countEachLabel(imds)

[imdsTrain,imdsValidation] = splitEachLabel(imds,0.7,'randomized');
net = alexnet;
layersTransfer = net.Layers(1:end-3);

numClasses = 7;
layers = [
  layersTransfer
```

```
fullyConnectedLayer(numClasses,'WeightLearnRateFactor',20,'BiasLearnRateFactor',
20)
   SoftmaxLayer
   classificationLayer];
options = trainingOptions('rmsprop', ...
   'MiniBatchSize',8, ...
   'MaxEpochs',4, ...
   'InitialLearnRate',1e-4, ...
   'Shuffle','every-epoch', ...
   'ValidationData',imdsValidation, ...
   'ValidationFrequency',3, ...
   'Verbose',false, ...
   'Plots','training-progress');

netTransfer = trainNetwork(imdsTrain,layers,options);
[YPred,scores] = classify(netTransfer,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)

img = readimage(imds,101);
actualLabel = imds.Labels(101);
predictedLabel = netTransfer.classify(img);
imshow(img);
title(['Predicted: ' char(predictedLabel) ', Actual: ' char(actualLabel)])

save(['train_7diseases_alexnet' '.mat'], 'trainedNet');
```

APPENDIX E
PROGRAM FOR RUNNING RESNET-18 ARCHITECTURE

```
clc;
clear all;
close all;

outputFolder = fullfile('C:\Users\rsu\Matlab\summer training');
rootFolder = fullfile(outputFolder, '224folder');

categories = {'Cardiomegaly', 'Consolidation'};

imds = imageDatastore(fullfile(rootFolder,categories), ...
    'IncludeSubfolders',true,'LabelSource','foldernames');

tbl = countEachLabel(imds)

minSetCount = min(tbl{:,2})

imds = splitEachLabel(imds, minSetCount, 'randomize');
countEachLabel(imds)

[imdsTrain,imdsValidation] = splitEachLabel(imds,0.7,'randomized');

net = resnet18();
numClasses = 2;

lgraph = layerGraph(net);
newFCLayer =
fullyConnectedLayer(numClasses,'Name','new_fc','WeightLearnRateFactor',10,'BiasL
earnRateFactor',10);
```

```
lgraph = replaceLayer(lgraph,'fc1000',newFCLayer);
newClassLayer = classificationLayer('Name','new_classoutput');
lgraph = replaceLayer(lgraph,'ClassificationLayer_predictions',newClassLayer);

inputSize = net.Layers(1).InputSize;
inputSize= [224 224 3];
augimdsTrain = augmentedImageDatastore(inputSize,imdsTrain);
augimdsValidation = augmentedImageDatastore(inputSize,imdsValidation);

options = trainingOptions('sgdm', ...
    'MaxEpochs',10, ...
    'InitialLearnRate',0.001, ...
    'Shuffle','every-epoch', ...
    'ValidationData',imdsValidation, ...
    'ValidationFrequency',30, ...
    'Verbose',false, ...
    'Plots','training-progress');
trainedNet = trainNetwork(imdsTrain,lgraph,options);

[YPred,probs] = classify(trainedNet,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
img = readimage(imdsValidation,101);
actualLabel = imdsValidation.Labels(101);
predictedLabel = trainedNet.classify(img);
imshow(img);
title(['Predicted: ' char(predictedLabel) ', Actual: ' char(actualLabel)])

save(['train_2diseases_resnet18' '.mat'], 'trainedNet');
```

APPENDIX F
PROGRAM FOR RUNNING RESNET-50 ARCHITECTURE

```
clc;
clear all;
close all;

outputFolder = fullfile('/Users/minnie/Documents/MATLAB/summer training');

rootFolder = fullfile(outputFolder, '224folder');

categories = {'NORMAL', 'PNEUMONIA', 'Cardiomegaly', 'Consolidation', 'Edema'};

imds = imageDatastore(fullfile(rootFolder,categories), ...
    'IncludeSubfolders',true,'LabelSource','foldernames');

tbl = countEachLabel(imds)

minSetCount = min(tbl{:,2})

imds = splitEachLabel(imds, minSetCount, 'randomize');

countEachLabel(imds)

[imdsTrain,imdsValidation] = splitEachLabel(imds,0.7,'randomized');

net = resnet50;
lgraph = layerGraph(net);
% {
resnet50
   newLearnableLayer = fullyConnectedLayer(numClasses, ...
```

```
        'Name','new_fc', ...
        'WeightLearnRateFactor',10, ...
        'BiasLearnRateFactor',10);
    lgraph = replaceLayer(lgraph,'fc1000',newLearnableLayer);
    newSoftmaxLayer = SoftmaxLayer('Name','new_Softmax');
    lgraph = replaceLayer(lgraph,'fc1000_Softmax',newSoftmaxLayer);
    newClassLayer = classificationLayer('Name','new_classoutput');
    lgraph = replaceLayer(lgraph,'ClassificationLayer_fc1000',newClassLayer);

inputSize = net.Layers(1).InputSize;
inputSize= [224 224 3];

augimdsTrain = augmentedImageDatastore(inputSize,imdsTrain);
augimdsValidation = augmentedImageDatastore(inputSize,imdsValidation);

options = trainingOptions('sgdm', ...
    'MaxEpochs',3, ...
    'InitialLearnRate',0.001, ...
    'Shuffle','every-epoch', ...
    'ValidationData',imdsValidation, ...
    'ValidationFrequency',17, ...
    'Verbose',false, ...
    'Plots','training-progress');
trainedNet = trainNetwork(imdsTrain,lgraph,options);

[YPred,probs] = classify(trainedNet,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)

save simpleDL.mat trainedNet lgraph

img = readimage(imdsValidation,101);
```

```
actualLabel = imdsValidation.Labels(101);
predictedLabel = trainedNet.classify(img);
imshow(img);
title(['Predicted: ' char(predictedLabel) ', Actual: ' char(actualLabel)])

save(['train_5diseases_resnet50-Epoch-5' '.mat'], 'trainedNet');
```

APPENDIX G
PROGRAM FOR RUNNING GOOGLENET ARCHITECTURE

```
clear
clc
close all

outputFolder = fullfile('C:\Users\rsu\Matlab\summer training');
rootFolder = fullfile(outputFolder, '224folder');
categories = {'Hernia', 'Effusion', 'PNEUMONIA'};

imds = imageDatastore(fullfile(rootFolder, categories), ...
    'IncludeSubfolder', true, ...
    'LabelSource', 'foldernames');

tbl = countEachLabel(imds)

[imdsTrain, imdsValidation] = splitEachLabel(imds, 0.8, 0.2);
imageSize = [224 224 3];

augimdsTrain = augmentedImageDatastore(imageSize, imdsTrain,
'ColorPreprocessing', 'gray2rgb');
augimdsValidation = augmentedImageDatastore(imageSize, imdsValidation,
'ColorPreprocessing', 'gray2rgb');

net = googlenet;
inputSize = net.Layers(1).InputSize
lgraph = layerGraph(net);
numClasses = numel(categories);
disp(categories(randperm(numClasses, 3)))
```

```
newfcLayer = fullyConnectedLayer(numClasses, ...
    'Name', 'new_fc', ...
    'WeightLearnRateFactor', 40, ...
    'BiasLearnRateFactor', 40);
lgraph = replaceLayer(lgraph, net.Layers(end-2).Name, newfcLayer);
newClassLayer = classificationLayer('Name', 'new_ClassificationOutput');
lgraph = replaceLayer(lgraph, net.Layers(end).Name, newClassLayer);
figure(1), plot(lgraph), title('Network Architecture of GoogleNet');

pixelRange = [-30 30];
scaleRange = [0.9 1.1];
imageAugmenter = imageDataAugmenter(...
    'RandXReflection', true, ...
    'RandXTranslation', pixelRange, ...
    'RandYTranslation', pixelRange, ...
    'RandXScale', scaleRange, ...
    'RandYScale', scaleRange);

miniBatchSize = 10;
valFrequency = floor(numel(augimdsTrain.Files)/miniBatchSize);
options = trainingOptions('sgdm', ...
    'MiniBatchSize',miniBatchSize, ...
    'MaxEpochs',50, ...
    'InitialLearnRate',0.0001, ...
    'Shuffle','every-epoch', ...
    'ValidationData',augimdsValidation, ...
    'ValidationFrequency',valFrequency, ...
    'Verbose',true, ...
    'Plots','training-progress');

net = trainNetwork(augimdsTrain, lgraph, options);
[YPred, PredScores] = classify(net, augimdsValidation);
```

```
YValidation = imdsValidation.Labels;
accuracy = mean(YPred == YValidation)

confMat = confusionmat(YValidation, YPred);
confMat = bsxfun(@rdivide, confMat, sum(confMat, 2))
mean(diag(confMat));
confusionchart(imdsValidation.Labels, YPred, 'Normalization', "row-normalized");

randValidate = randperm(numel(imdsValidation.Files), 1); %randomly choose 1 img
from val
randimg = readimage(imdsValidation, randValidate);

[label, scores] = classify(net, randimg);

predictedLabel = YPred(randValidate);
actualLabel = imdsValidation.Labels(randValidate);

[~, idx] = sort(scores, 'descend');
idx = idx(3:-1:1);
classes = net.Layers(end).Classes;
classNamesTop = string(categories(idx));
scoreTop = scores(idx);

h = figure(2);
h.Position(3) = 2*h.Position(3);
ax1 = subplot(121); %row col #pic
ax2 = subplot(122);

image(ax1, randimg);

title(ax1, {['Predicted disease: ' char(predictedLabel)] ...
```

```
    ['Actual disease: ' char(actualLabel)] ...
    ['Percent of each prediction: ' num2str(100*scores)]});

b = barh(ax2, scoreTop);
title(ax2, 'Top 3 Predictions')

xlim(ax2, [0 1])
xticks(ax2, 'auto')
xticklabels(ax2, 'auto')
xlabel(ax2, 'Probability')

yticks(ax2, 'manual')
yticklabels(ax2, classNamesTop)
ylabel(ax2,'Diseases')
ax2.YAxisLocation = 'left';

threshold = xline(ax2, 0.90, 'Color','red','LineStyle','--');

save train_EffusionHerniaPNEUMONIA_googlenet-Epoch-50_cat3.mat net lgraph
save(('train_3diseases_googlenet-Epoch-50_cat3' '.mat'), 'net');
```

APPENDIX H
PROGRAM FOR RUNNING PIPELINE SYSTEM

```
clear
clc
close all
table_to_sum=zeros(38,16);
A = categorical({'NORMAL'; 'NORMAL_t'; 'Atelectasis'; 'Cardiomegaly';
'Consolidation'; 'Edema'; 'Effusion'; 'Emphysema'; 'Fibrosis'; 'Hernia'; 'Mass';
'Nodule'; 'Pleural_Thickening'; 'PNEUMONIA'; 'PNEUMONIA_t'; 'Pneumothorax'});
valueset = 1:16;
for ii=1:38
   loadednet=[num2str(ii) '.mat'];
   load(loadednet);
   labels=net.Layers(144,1).Classes;
   % Table of array
   row_in_table = find(labels(1)==A);
   table_to_sum(ii,row_in_table)=table_to_sum(ii,row_in_table)+1;
   row_in_table = find(labels(2)==A);
   table_to_sum(ii,row_in_table)=table_to_sum(ii,row_in_table)+1;
   row_in_table = find(labels(3)==A);
   table_to_sum(ii,row_in_table)=table_to_sum(ii,row_in_table)+1;
end
%%
valFolder = 'D:\Nobel Minnie\trained images\Pneumothorax';
%create image data store to help managing data as it is operated only on image file
location
imds =
imageDatastore(valFolder,'IncludeSubfolders',true,'LabelSource','foldernames');
auimds = augmentedImageDatastore([224 224
3],imds,'ColorPreprocessing','gray2rgb');
```

```
[nc, nr] = size(imds.Labels);
table_percentage_data=zeros(16,16);

for kk=16:16%
   network_to_load=find(table_to_sum(:,kk)==1);
   [aa, bb]=size(network_to_load);
   table_percentage=zeros(16,1);
   number_of_times=zeros(16,1);
   for ii=1:aa
      loadednet=[num2str(network_to_load(ii)) '.mat'];
      load(loadednet);
      labels=net.Layers(144,1).Classes;
      % Table of array
      percentage_output=double(squeeze(activations(net,auimds,'prob')));
      aa=percentage_output<0.98;
      percentage_output(aa)=0;    % cutoff at 0.9
      if nc>1
        percentage_output = sum(percentage_output');
      end
      for jj=1:3
         row_in_table = find(labels(jj)==A);

table_percentage(row_in_table)=table_percentage(row_in_table)+percentage_output(j
j);
         number_of_times(row_in_table)=number_of_times(row_in_table)+1;
      end
   end

table_percentage_data(:,kk)=table_percentage_data(:,kk)+table_percentage./number_
of_times./nc;
end
```