Explainable Artificial Intelligence for Medical Image Classification Using Deep Learning Methods

BY

Awika Ariyametkul

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 2023

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

PROJECT CERTIFICATE

Project Title	Explainable Artificial Intelligence for Medical Image Classification Using Deep Learning Methods
Student Name	Miss Awika Ariyametkul Student ID. 63011114
Degree	Bachelor of Engineering in Biomedical Engineering

Project Advisor Signed: _____

(Dr. May Phu Paing)

Committee Signed: _____

(Prof. Dr. Chuchart Pintavirooj)

Committee Signed: _____

(Assoc. Prof. Dr. Wibool Piyawattanamatha)

Committee Signed: _____

(Asst. Prof. Dr. Treesukon Treebupachatsakul)

Committee Signed: _____

(Asst. Prof. Dr. Kasama Srirussamee)

Head of Department Signed: _____

(Assoc. Prof. Dr. Sarinporn Visitsattapongse)

Project Title	Explainable Artificial Intelligence for Medical Image Classification Using Deep Learning Methods
Student Name	Miss Awika Ariyametkul
Degree	Bachelor of Engineering in Biomedical Engineering
Project Advisor	Dr. May Phu Paing
Academic Years	2023

ABSTRACT

Artificial intelligence, or AI, is a powerful tool for humans to work with enormous amounts of data. AI is widely used these days, and several attempts are being made to apply it across numerous sectors. The classification of medical images is one of them. However, the lack of transparency and explanation in AI is one of its limitations. The "black box"—the absence of knowledge of AI's thought process—is the source of the vulnerability. The way to solve this problem is through XAI, or explainable artificial intelligence. XAI is the method used to explain the process behind AI's decision-making. The main purpose of this thesis is to compare three XAI methodologies, namely LIME, Grad-CAM, and Grad-CAM++. Their explainability was analyzed and compared focusing on the DenseNet201 based breast cancer classification. The experimental findings pointed out all of these XAIs are able to highlight salient features or part of the images contributed the most to the classification decisions. However, in order to get a precise assessment, this study also applied ground truth bounding boxes made by medical experts and compared the heatmaps of XAI with the bounding box lesion area. Based on the comparative results, Grad-CAM++ provides better explanation compared to its counterparts.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Dr. May Phu Paing, my adviser, for her encouragement and feedback. Dr. May is always a valuable resource for advice on this research. She was a great help to me. I really appreciate each and every recommendation she makes for me. Without her, I could never finish this study. I am also appreciative of all the professors in my department who have encouraged my knowledge throughout the years at the university. A special mention here should go to my colleagues at the university for their support. Finally, I would like to thank my wonderful family for their unwavering support in every aspect of my life. They constantly motivate me to work. Their confidence in me serves as my driving force to continue working.

Awika Ariyametkul

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

Symbols/Abbreviations	Terms
ANN	Artificial Neuron Network
CAM	Class Activation Mapping
CNN	Convolution Neuron Network
DL	Deep learning
Grad-CAM	Gradient-weighted Class Activation Mapping
Grad-CAM++	Grad-CAM Plus Plus
IT	Information Technology
LIME	Local Interpretable Model-Agnostic Explanations
LSTM	Long-Short Term Memory
ML	Machine learning
RNN	Recurrent Neural Networks
XAI	Explainable Artificial Intelligence

CHAPTER 1
INTRODUCTION

This chapter introduces the overarching themes of this report and places the motivation for the work into context. Thereafter, the rationale and goals defined for the investigation of the project are discussed, followed by a summary of the overall project in each chapter.

XAI, or explainable artificial intelligence, is an important tool to prove the reliability of AI, especially in medical fields. Numerous projects in the healthcare sector attempt to apply AI in various ways. There is an attempt to use AI for medical image analysis. This is a prediction of where the abnormal is located in the medical image, such as x-ray film, ultrasound film, and mammography. Another example of AI usage in the healthcare industry is predictive analytics. This one is used to predict a patient's outcome from the patient's previous stage. There are so many works out of my example, but all of them still have a week. Because AI is a "black box," no one can explain what goes on within the AI process to generate the prediction, which is one of its limitations. This is the lack of faith people have in AI. There are also other benefits of AI, but the lack of trust will be more significant when it comes to medical work. Trust is very important for doctors and patients to use any tools related to their security. That's why XAI is very important here. This chapter is going to introduce you to the background of the study, the objective, the scope of the study, and the report outline.

1.1 Background and significance of the study

Artificial intelligence, or AI, is now widely used in many industries, for example, financial services, insurance, telecommunications, life sciences, and healthcare. For financial services, there is AI for rebalancing portfolios. The AI will calculate the most effective way to invest. Like a fraud, the fraud manager will collect investors' money and organize them to invest in appropriate stocks, but AI will organize investors' money individually. For insurance, there is AI-powered underwriting.

Nowadays, insurance companies utilize AI to evaluate risks based on enormous amounts of data that include information on things like prescription medicine usage and pet ownership. For telecommunication, they used AI to maintain flawless operation; networks must adapt to accommodate shifting traffic and react swiftly to abnormalities. In life science, AI can help in drug discovery. The way molecules interact has a lot of possibilities. Here, AI can predict the interactions between different substances and even the way a medicine would work against its intended target.

The healthcare industry also uses AI in many ways, such as predictive analytics, AI-assisted diagnosis and treatment, wearable devices and sensors, AI-driven genomics, AI-assisted telemedicine, etc. One of them is image classification. A doctor may need to take a photograph and examine it as part of a procedure, like an x-ray film. To identify the disease, the physician needs to look for abnormal things in the image. To do so, the physician needs more than 10 years to be an expert. This work can be accomplished with AI. The most popular way to do image classification to Convolution Neuron Network, or CNN. This is one of the deep learning methods that try to imitate human neuron network. Nowadays, AI can help give the doctor more information for the diagnosis. When AI classifies an x-ray film as having cancer or not having cancer and just provides the result to the doctor, the doctor will reflect on the reasoning behind the classification. One type of x-ray film is mammography, which is a special x-ray image of a woman's breast. The specialty of this x-ray picture is how it can be taken. Mammography was taken by a mammogram, which is a special machine designed to take a picture of a woman's breast. This mammography is the medical image that this study focus on.

Among these is the popularity of AI. It still has some challenges. One of them is the lack of transparency and explanation. An explanation is very important. Considering yourself when you decide to believe someone’s decision The first thing you will do is asking them why they made that decision. Once you have the answer, you can choose whether or not to follow them. AI is also facing this same difficulty. AI is still not widely trusted, and its accuracy cannot be guaranteed. Knowing the rationale behind an AI decision will make it easier for you to choose whether to accept its

response. This will be more important in the medical industry because it deals with people's lives. A single mistake could result in serious issues. The doctor and other medical professionals cannot just believe what AI gives them because they have the knowledge and expertise to diagnose and treat a lot of patients. Additionally, no AI can make medical decisions in the modern era. The AI is merely a tool to give the doctor access to a different viewpoint. Because of this, an explainable AI is coming. This weakness of AI can be called the black box, which means the lack of understanding of the AI internal processes. The user can provide input and receive output that has been determined by the computer, but they are unaware of the reasoning. In breast cancer image classification, the input is mammography. The output has two classes: benign and malignant. The black box that we don't know is which area in the breast cancer image is the decision's area of the AI.

Fig 1.1: Concepts of black box [1]

The solution to this problem of AI is XAI, or explainable artificial intelligence. XAI is a tool to explain the reason behind AI's decision. It has many types of AI. XAI also has many types to deal with different kinds of data. For example, LIME can work with many types of data, including words, tubular data, and images. Since the subject of this work is image classification, XAI for image classification will be of interest. The most straightforward way to explain the decision-making process is to visualize the activation map. This is how Grad-CAM and Grad-CAM++ work.

1.2 Objectives

This study focuses on explainable artificial intelligence, or XAI, that visualizes the decision's area of image classification using deep learning methods (Convolutional Neuron Network, or CNN). The objective of this project is to compare the different types of XAI, including LIME, Grad-CAM, and Grad-CAM++, and find the best XAI for image classification.

1.3 Scope of the study

The scope of the study covered everything from the Convolution Neuron Network, or CNN, to the comparison of three XAI methodologies (LIME, Grad-CAM, and Grad-CAM++). The CNN model that was used in this project is DenseNet201. This DenseNet201 was applied to learn the breast cancer image dataset. This dataset provided the ground truth bounding box of the lesion in the image, which would be used to compare the results of all the XAIs later. Then, three XAI methodologies, including LIME, Grad-CAM, and Grad-CAM++, were used to explain the internal process of DenseNet201. The results of these XAIs were compared with the bounding box from the dataset.

1.4 Report outline

The rest of this report is organized as follows:

Chapter 2 reviews theory-related
This chapter will explain the related theory used in this study. Most of them are about the concepts of AI, the CNN model, and the XAI methodologies.

Chapter 3 methodology
This chapter is about how to do the CNN model (DenseNet201) and how to do the three XAI methodologies. (LIME, Grad-CAM, Grad-CAM++) The interesting problem arising from the experiment is also mentioned in this section.

Chapter 4 experimental results and discussion

This chapter is a comparison of the results of three XAI methodologies. (LIME, Grad-CAM, and GradCAM++) and discuss which is the most suitable XAI for a medical image.

Chapter 5 conclusion

This chapter is the summary of all the chapters above and the key message of this study. Then follow up with future work.

CHAPTER 2
REVIEW OF THEORY RELATED

2.1 Introduction

This chapter will provide some context related to deep learning, CNN models, XAI, LIME, Grad-Cam, Grad-Cam++, and also the medical image that I will use in the experiment. I will discuss how we will use this theory in the experiment and how these theories improve the project. Deep learning is a subset of machine learning. I will explain what the difference is between these two. The CNN model is from a convolutional neural network. I will explain what it is and how to create the model. XAI is from Explainable AI. I will tell you what it is and why it is important, especially for biomedical AI. LIME, Grad-Cam and Grad-Cam++ are all methods of the XAI. I will explain the concept and the differences between them.

2.2 Artificial Intelligence

Artificial intelligence (AI) is a machine's ability to perform the cognitive functions we usually associate with human minds, such as perceiving, reasoning, learning, interacting with an environment, problem solving, and even exercising creativity [2]. AI was started in 1950, which is a long time ago, by computer scientist and mathematician Alan Turing. Alan Turing was known as the man who created the first computer in the world, which was used during World War II. AI and computers are related to each other because they share the same idea of wanting to create a machine that can think by itself. Since the day of Alan Turing, computers have been developed by many computer scientists. Finally, the first personal computer was released in the 1970s. Until now, personal computers have become other common tools that everyone needs. Now, it is one of our lives, and we cannot work without it. Computers have had the same goal as AI since the beginning. Until now, scientists around the world have still developed this technology, and it is improving fast.

AI is very popular nowadays and will continue to increase in everyone's daily lives. AI makes the business more efficient and profitable [2]. AI has been used for a while in various industries such as, for example, financial services, insurance, telecommunications, life science, food technology, agriculture, automotive, logistics, etc. For financial services, there is AI for rebalancing portfolios. The AI will calculate the most effective way to invest. Like a fraud, the fraud manager will collect investors' money and organize them to invest in appropriate stocks, but AI will organize investors' money individually. For insurance, there is AI-powered underwriting. Nowadays, insurance companies utilize AI to evaluate risks based on enormous amounts of data that include information on things like prescription medicine usage and pet ownership. For telecommunication, they used AI to maintain flawless operation; networks must adapt to accommodate shifting traffic and react swiftly to abnormalities. In life science, AI can help in drug discovery. The way molecules interact has a lot of possibilities. Here, AI can predict the interactions between different substances and even the way a medicine would work against its intended target. As an example of food technology, there is a robot that can create a new tea recipe. This robot uses AI and IOT technology as a base to create the tea recipe from a web interface or mobile app. For agriculture, they used AI to analyze the variables of the plant: the sunlight, the humidity in the soil, and the heat. Automotive is also another topic that is of interest right now. It is about the self-driving car. Aphabet is now providing a self-driving taxi that can actually send passengers across California. The next topic is logistics and transportation. The use of machine learning has already transformed supply chain management, making it a seamless process. Many warehouses use AI-powered robots for sorting and packaging products [3]. For the most part, AI systems function by consuming enormous quantities of labeled training data, searching the data for correlations and patterns, and then utilizing these patterns to forecast future states.

In the medical industry, there is a lot of technology that uses AI, such as in radiology. They used AI to find the abnormality in the medical image, such as mammography, CT scans, x-rays, and ultrasounds. This technology is called the CAD system. There are two parts to the CAD system. The first part is computer-aided detection (CADe). This is used to detect the abnormal part in the medical image, for

example, the cancer in mamograhy. (Fig2.1) The second part is computer-aided diagnosis (CADx). CADx helps to evaluate the structures identified in CADe [4]. The CAD system helps a lot in the early detection of breast cancer. One of the leading causes of cancer-related deaths is breast cancer. Mammograms should be performed on women around the age of 40 as an early detection method to stop breast cancer. CAD can also be used with other kinds of medical images, such as ultrasounds and x-rays. The use of AI in the healthcare sector is a lot more than CAD systems. IBM Watson (an AI tool) uses AI to analyze the patient's medical record to predict the potential treatment for the doctor. Robotic surgery is another topic that is interesting nowadays. Even though the robot may now do the surgery on its own, doctors must still oversee any AI use. AI is utilized as a helper to provide the physician access to a second approach. They still have the last decision on the matter.

Fig 2.1: Computer aid detection image

2.3 Machine learning and Deep learning

Deep learning (DL) and machine learning (ML) are both methods of artificial intelligence (AI). They both share the same idea of learning the data to generate new data. They are both in the field of intelligence, but they are not the same. The difference between machine learning (ML) and deep learning (DL) is down below.

2.2.1 Machine learning

An algorithm that can recognize patterns in data, learn from them, and gradually increase efficiency is called machine learning. In the years since its widespread deployment, which began in the 1970s, machine learning has had an impact in a number of industries, including medical-imaging analysis and high-resolution weather forecasting [5].

2.2.2 Deep learning

Deep learning (DL) is a subset of machine learning. DL can learn a variety of information, such as images, text, and signals. DL has higher accuracy compared to ML and can also learn more complex information. Deep learning is trying to imitate how the human brain thinks in reality, which is through the neuron cell. In the human brain, there are a lot of neurons that used to receive and send information to each other. In the same way, DL transmits the input data both forward and backward so that the computer can learn the information and apply it to forecast new data. There are three types of deep learning: feed-forward neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs).

- Feed-forward neural networks is the one way information move. It only gose forward without traveling backward. It is the most simple one. It was first proposed in 1958. The feed-forward neural networks was used in banking to detect fraudulent financial transactions.

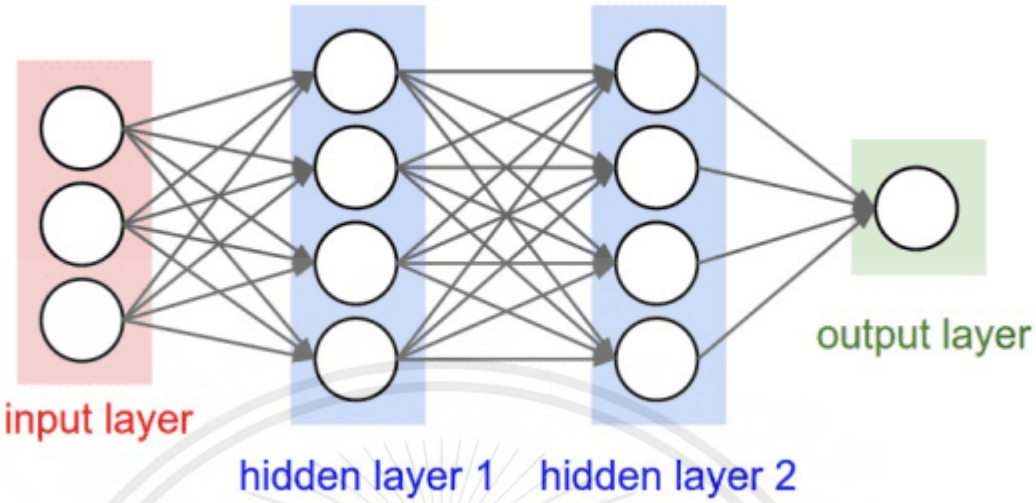


Fig 2.2: Feed-forward neural networks architecture [5]

- Convolutional neural networks (CNNs) are a type of feed-forward neural network, but this one is used to process the image. An example is the image classification between a cat and a dog. There are many ways to use CNN. In business, they used CNN to detect the company's logo on social media to see the potential for market opportunities. In the healthcare industry, they used CNN to scan the medical image to find abnormal things that might be the cause of the disease.

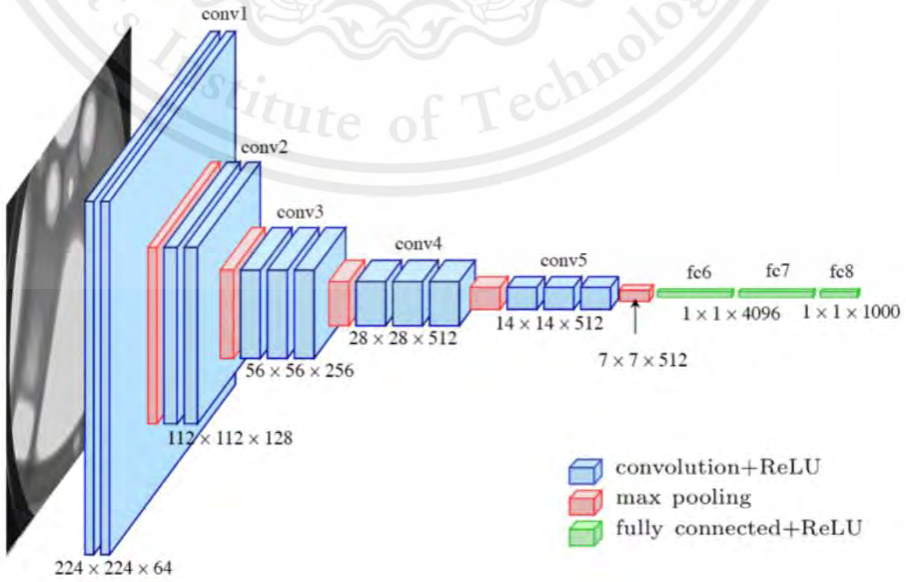


Fig 2.3: Convolutional neural networks (CNNs) architecture [6]

- Recurrent neural networks (RNNs) are artificial neural networks whose connections include loops, meaning the model both moves data forward and loops it backward to run again through previous layers [7]. In RNNs, the output of one data point becomes the input for the next, allowing the network to process sequences like stock price data over time. For example, in stock prediction, daily prices are considered. The first day's low price is fed to a neuron, which calculates weights and makes predictions. The output from the first day becomes part of the input for the second day, combined with the second day's price. This process continues for each subsequent day, making it a recurrent neural network. RNNs also have their limitations, which are the vanishing gradient and exploding gradient problems. That's why it has LSTM, or Long Short Term Memory Networks, that have the ability to deal with long-term memory better than RNN.

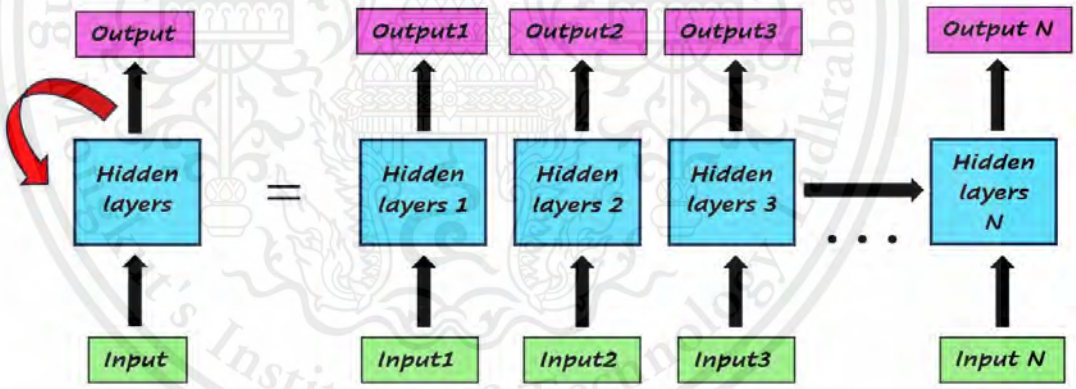


Fig 2.4: Recurrent neural networks (RNNs) architecture [7]

2.4 Convolution neuron network

CNN, or Convolution Neural Network, is one of the most impressive ANNs, or Artificial Neural Network which imitated from human neurons and is the most popular deep learning model. Human can learn and think with the neurons in our brains, so many scientists in the past tried to create that neuron network with mathematics, and that is Artificial Neural Network [5]. (Fig 2.5) One type of ANN that operates on images

is the convolutional neural network. The reason why CNN can work with images is because CNN processes the data with a grid-like topology. The CNN model will perform the picture recognition. Find the specific image feature to use in the image tasks, for example, image classification. In CNN architecture, we have a convolution layer, and pooling layer to find the important feature that can identify the image type, and fully connected layer to train our computer to know the features of different objects. A model that accurately categorizes one kind of image might not do so for a different image. The input layer is the layer to input the data. The output layer is the layer to get the output. (Fig 2.6) CNN models have an input layer, a convolution layer, a pooling layer, and a fully connected layer; the final layer is the output layer.

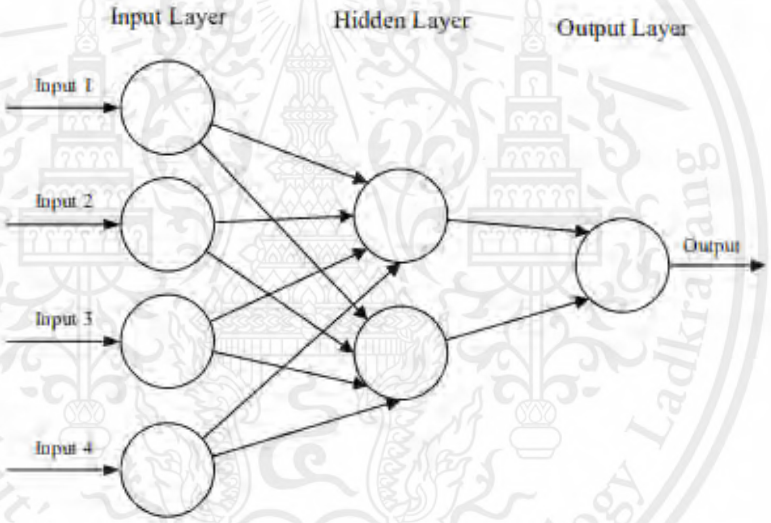


Fig 2.5: A simple three-layered feedforward neural network (FNN) [8]

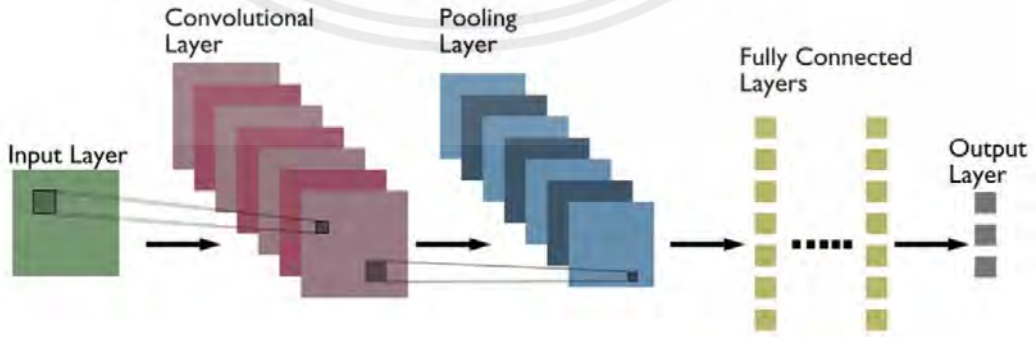


Fig 2.6: CNN layers, input layer, convolution layer, pooling layer, fully connected layer, and output layer [6]

2.4.1 Definitions of different layers

Convolution layers are used to detect a specific feature or pattern, which will be used to distinguish between each class later. These features can be edges, corners, textures, or more complex patterns. To do so, convolution layers used a filter or kenal to slide on the input image, sum the resulting multiplied values, and return the resulting value as the new value of the center pixel. With this method, the output convolution layer will have the same size as the filter or kernal. The output of the convolutional layer is called ‘feature map’.

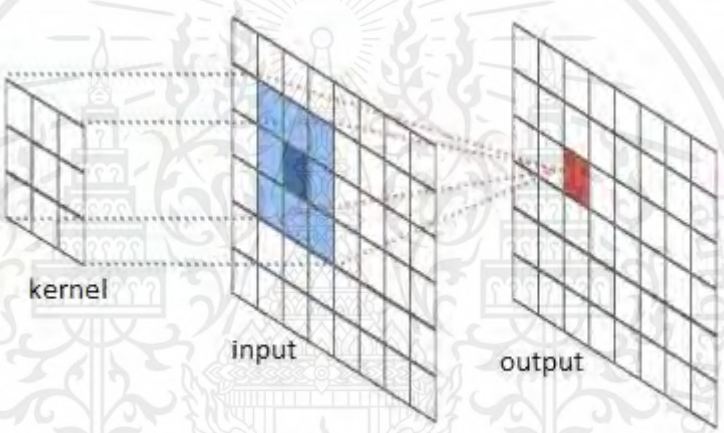


Fig 2.7: Convolution layers [9]

Pooling layers are used to reduce the spatial dimension of the input, making it easier to process and requiring less memory [6]. Pooling layers have two types. The first one is average pooling. The average pooling returns the average value of the pooling window (kernal). The second is max pooling, which returns the maximum value of the pooling window (kernal). Both of them are aimed at downsampling the data. Pooling can help solve overfitting problems. Without max pooling, the model cannot regconize features irrespective of small shifts or rotations. This would make the model less robust to variations in object positioning within the image, possibly affecting accuracy [6].

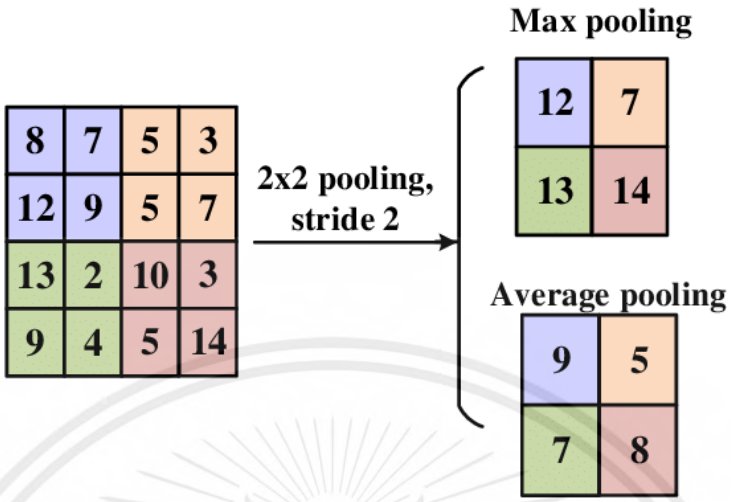


Fig 2.8: Convolution layers [10]

Fully connected layers are used to make a prediction by using the output of convolution layers and pooling layers. Fully connected layers connect every neuron together. The work of fully connected layers is to take the high-dimensional output from the previous layers and flatten it to get a single vector. This allows the network to extract the feature without considering localization and reach a classification decision.

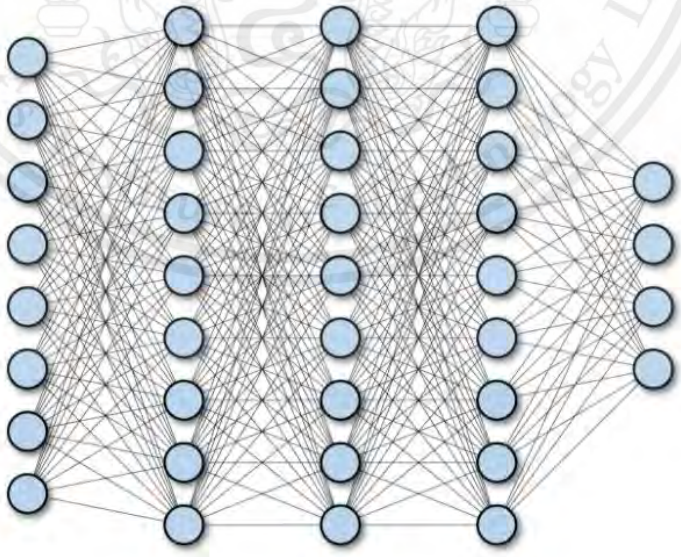


Fig 2.9: Fully connected layers [11]

2.4.2 Different types of CNN Architectures

The example of some CNN architectures:

LeNet is the CNN model that is used to distinguish handwritten letters. LeNet is very popular because it is the first CNN model, developed in 1998 by Yann LeCun, Corinna Cortes, and Christopher Burges [6]. LeNet used 5 convolution layers, followed by 2 fully connected layers. The problem of LeNet is the vanishing gradients problem, which can be solved by putting a max pooling layer between each convolution layer.

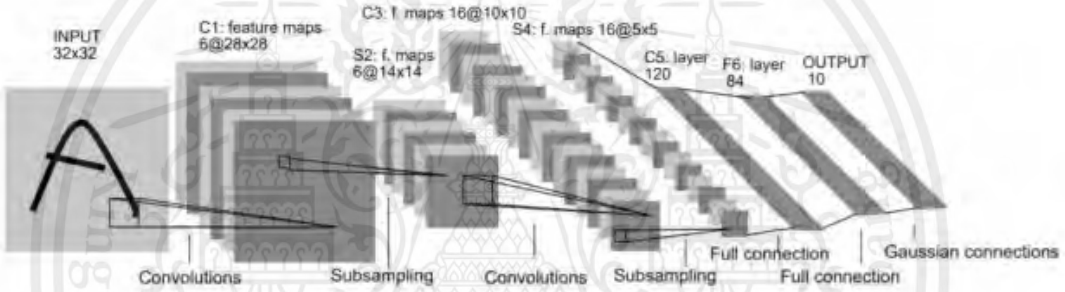


Fig 2.10: LeNet architecture [6]

VGG16 is one of the excellent vision model architecture. VGG16 was developed by Karen Simonyan, Andrew Zisserman et al. at Oxford University. VGGNet has 95 million parameters and was trained with one billion images (1000 classes). VGG16 called ‘16’ because it has 16 lyaers that have weight. All the convolution layers use 3x3 filter and 1 stride. All the max pooling layers use 2x2 filter and 2 stride. Using 2 fully connected layers followed by a softmax for the output.

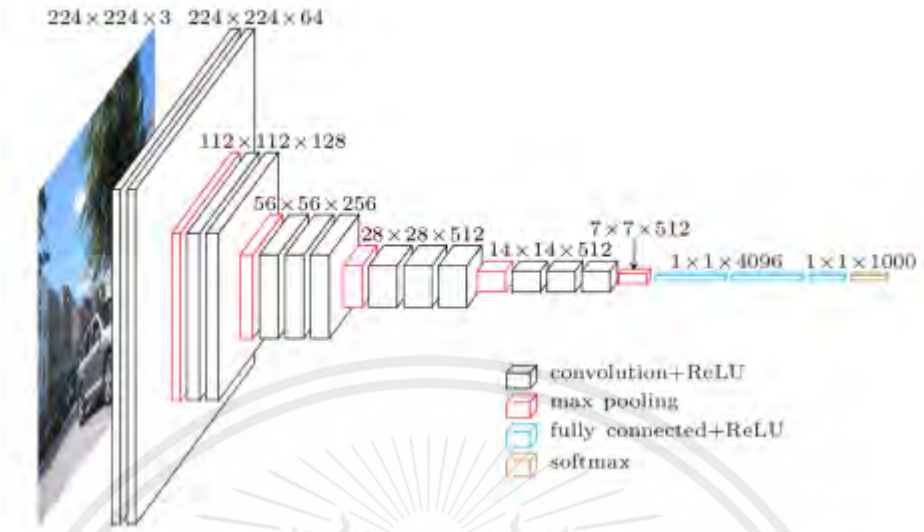


Fig 2.11: VGG16 architecture [12]

DenseNet was created to solve the vanishing gradients problem, which is a common problem in the CNN model. DenseNet is like its name because it is a connected dense block. Each dense box has convolution layers that connect to every other convolution layer in the same dense box. Inside the dense box, each convolution layer will take the input of the previous convolution layer, which means the input layer is the concatenation of the previous layer. This makes DenseNet able to reuse features, and that makes DenseNet more appropriate for fewer parameters.

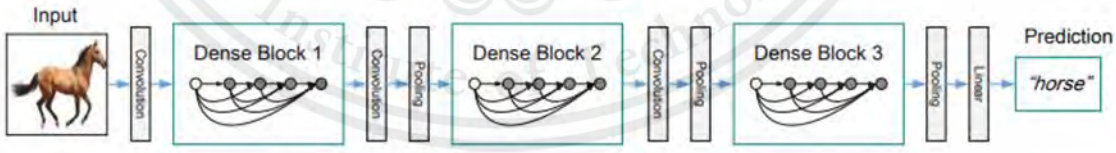


Fig 2.12: DenseNet201 architecture [13]

2.5 TensorFlow

TensorFlow, a rival to PyTorch and Apache MXNet frameworks, can train and run deep neural networks for image recognition, word embeddings, recurrent neural networks, sequence-to-sequence models for machine translation, natural language processing, and PDE (partial differential equation)-based simulations. The best part is

that TensorFlow uses the same models that were used for training to provide production prediction at scale. In this project, TensorFlow was used a lot to create the model and used its tools to improve the accuracy of the model.

2.6 Data Augmentation

Data augmentation is used to increase the image in the dataset by rotating and flipping the image. Data augmentation can also increase the variety of the dataset. With this method, the model can find more features and make sure that different angles of the image do not affect the model's classification.

2.7 Dataset

In this study, there are two datasets that have been used. The first is from Department of Industrial Engineering & Management, National Chin-Yi University of Technology, Taichung, Taiwan, which is the one that used to train the DenseNet201 model. The second one is the one that used to compare with the XAI. This one is provided with the bondary box which is the detection of the location of the cancer in the mammography. The source of the second dataset is robotflow.com, cancer image dataset.

2.7.1 Training dataset

The dataset is a dataset of Dataset of breast mammography images with masses. It is the set of mammography of breast cancer image both benign and malignant which come from Department of Industrial Engineering & Management, National Chin-Yi University of Technology, Taichung, Taiwan, available online on 25 June 2020 [14]. This dataset selects 106 breast mammography images with masses from INbreast database and after data augmentation, the number of breast mammography images was increased to 7632 [14]. Each image was marked with its corresponding breast density and the original images in INbreast database are DICOM files. We converted the DICOM files to PNG files through MATLAB [14]. Combining four breast density

categories and breast benign or malignant status, therefore, there are 8 categories in our classification task, so that it has eight categories but for our project we make a 2-class classification, benign and malignant. These all data is pass through the pre-processing; contrast limited adaptive histogram equalization (CLAHE).

The dataset was divided into three different datasets: a train dataset, a validation dataset, and a test dataset. The dataset has a ratio of 70%:20%:10%, or 5342:1526:764 in numerical order.

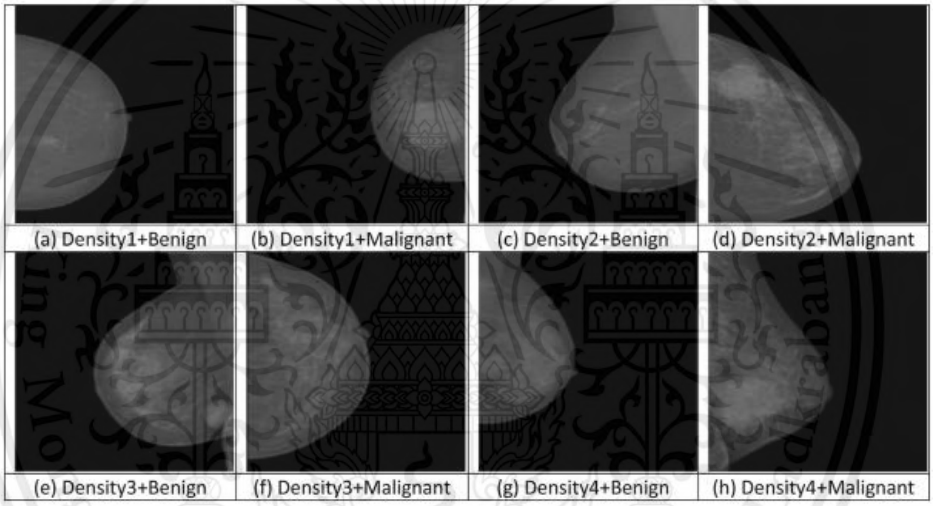


Fig 2.13: After CLAHE image processing [14]

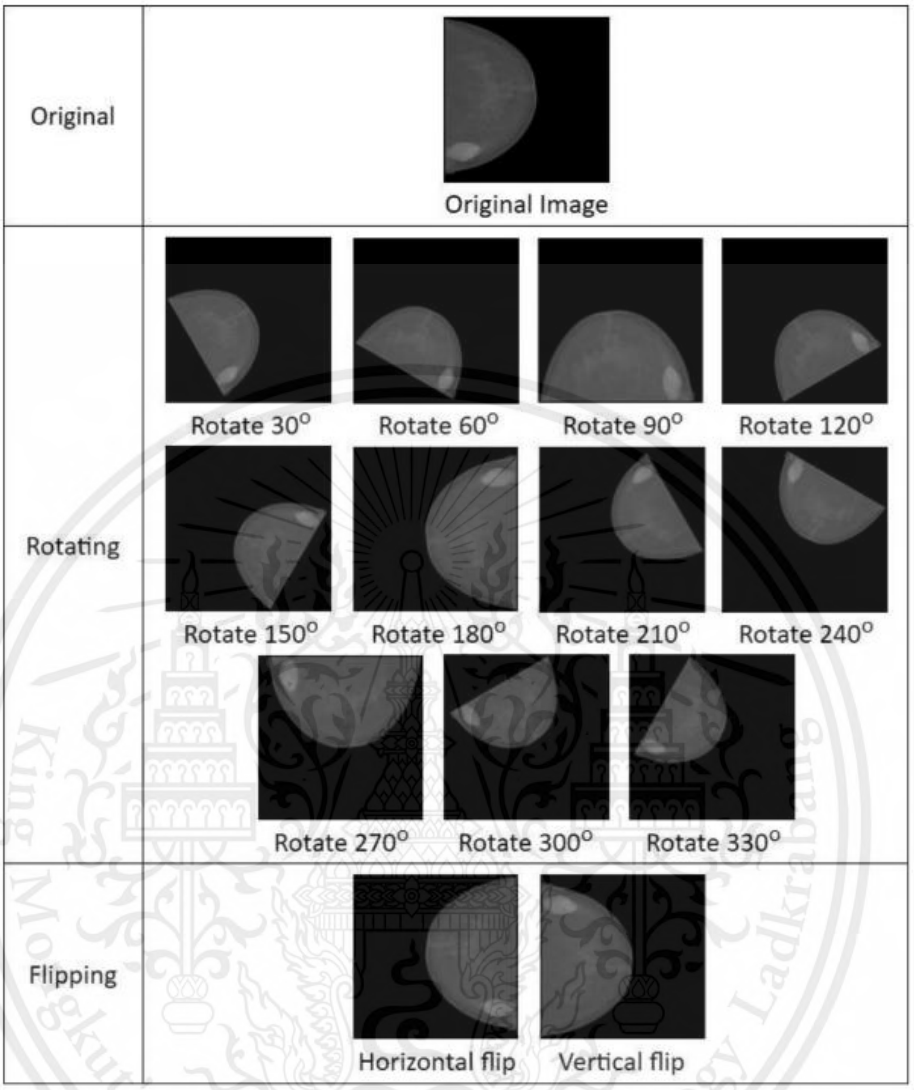


Fig 2.14: Example of the original image and after Augmentation [14]

2.7.2 Comparation dataset

This dataset is from roboflow.com, which is an open-source computer vision project. This dataset has 2427 images, which are separated into three datasets: training, validation, and testing. Training has 1696 images. Validation has 530 images. Testing has 201 tests. The image size is 640x640 without augmentation. Every image was provided with the text file, which is a label. This text file tells us the class of each image and also provides the number of bounding boxes. There are five numbers in there. The first number is the class, which is benign or malignant. The second and third numbers

are at the center of the bounding box. The fourth and fifth numbers are the height and width of the bounding box, respectively.

Breast cancer images in this dataset came from two different datasets, INbreast 2012 and MIAS Mammography. INbreast 2012 was acquired at the Breast Center in CHSJ, Porto, under the permission of both the hospital's Ethics Committee and the National Committee of Data Protection. The images were acquired between April 2008 and July 2010. There are 115 cases that were collected in total [15]. MIAS is from the Mammographic Image Analysis Society's Digital Mammogram Database. MIAS is an organization of UK research groups interested in the understanding of mammograms and has generated a database of digital mammograms. Films were taken from the UK National Breast Screening Programme [16]. MIAS consists of 161 cases in total.

2.8 Explainable AI

There are many AIs all around us, for example, smart phones that can recognize and track faces and online services that translate written text. AI is very popular and is used in so many businesses and industries. The popularity of AI comes from its potential to reduce costs and time in business processes. For example, the use of AI in drug discovery makes everything easier. In the past, it was hard to discover a new drug. Although we know that each molecule has its own special characteristics, making a new compound to get a good result is not easy work. AI can make this easier because the work that AI is good at is analysis. The way molecules interact has a lot of possibilities. Here, AI can predict the interactions between different substances and even the way a medicine would work against its intended target. The chemist can design the experiment based on AI's predictions. This method can increase the possibility of finding a new drug with the desired result. Using AI In the medical field, one of the most important things is the trust in AI of the users, doctors, and patients. It is very important to know the reason behind the AI's decision. To make the specialist accept the decision. (AI nowadays is used to help them, but the final decision is still in their hands) to increase trust in AI. Then why do physicians not trust AI? This is because of what we called the 'black box'.

The black box is the lack of understanding of AI's internal functioning. We have the input. We see the output, but the process that changes input to output is missing. Inside the black box, we cannot explain what happened. This limitation of AI is more serious when it comes to medical devices. Transparency and explainability need to be explained. Single wrong decisions can result in dangerous problems for the life and health of patients. Actually, AI also has other limitations.

These are three challenges of AI:

1. The large complexity and high energy demands of current deep learning models
2. The lack of robustness to adversarial attacks
3. The lack of transparency and explainability

The black box is the third challenge, the lack of transparency and explainability, which is the one that XAI came to solve. XAI, or Explainable Artificial Intelligence, has a role of supportive AI, which is to explain whatever they find to the doctor, like a radiologist does. Giving the reason behind the decision is better than giving only the decision to the doctor. Explaining the rationale behind one's decisions is an important part of human interaction. Explanations help to build trust in a relationship between humans and should therefore also be part of human-machine interactions. There are many types of XAIs as same as AI. The type of XAI depends on the information content, the recipient, and the proposal. Different recipient require different level of detail and with different information content. For example in image classification case, if the recipient is general user, they want to see the decision region. If the recioient is the AI researchers and developers, they want all the available information, including negative evidence, about the AI's decision in the highest resolution (e.g., pixel wise explanations), because only this complete information gives detailed insights into the (mal)functioning of the model. The proposal is information content, what the point of the explanation aim for. There are two type of this learn representation and individual prediction. Explaining learn representation aims to foster the understanding of the learned representations, ex Investigates the role of single neurons or group of neurons in encoding certain concepts. Explaining individual prediction aims to explain

individual prediction, ex explanations help to verify the predictions and establish trust in the correct functioning on the system.

Similar to AI, which has several varieties based on the type of data it works with, XAI offers numerous varieties to handle a wide range of data kinds. The dataset type in this study is an image (a breast cancer image). The most straightforward method to explain CNN model (mostly used with image classification) is to visualize layer activation. This method is called CAM, Class Activation Mapping. There are two more versions that develop from CAM, called Grad-CAM and Grad-CAM++. This series will be explained in detail in the next topic. Before going to those topics, let me introduce you to other attempts to explain AI's decision.

2.8.1 Other XAI methods

Deconvnet is a method to understand the higher layers of the neuron network. Zeiler & Fergus invented this method. They allowed the data flow from a neuron activation in higher layers back to the original image. We can determine which region of the input image has strongly activated that particular neuron.

Guided backpropagation is a technique that adapts to normal backpropagation. Backpropagation is typically employed in the training process, but Springenberg et al. employed it to comprehend the influence of an individual neutron.

LIME stands for Local Interpretable Model-Agnostic Explanations. LIME can make a local approximation to the complex decision surface of any deep learning model. The result of LIME is the decision region of the model. This method was developed by Ribeiro et al.

Contextual Explanation Networks (CENs) are a combination of deep learning neuron networks and context-specific probabilistic models. The result of this combination is the real-time explanation model. CENs can explain the decision of the

AI while the AI is making a classification decision of classification. The owner of this idea is Al-Shedivat et al.

2.9 Local Interpretable Model-Agnostic Explanations

LIME stands for Local Interpretable Model-Agnostic Explanations. Local is part of the overall goal of LIME, which is to “identify an interpretable model over an interpretable representation that is locally faithful to the classifier” [17]. Interpretable is to explain the weight of the neuron network in an understandable way to humans. Model-Agnigstic is any model that can work with word, image, or tubular data. Explanation is from artifacts that provide an understanding between input to a ML model and the model’s prediction.

LIME works by creating the perturbation image, which is a randomly generated black spot on the image’s pixels. The higher the perturbation number, the better the accuracy will be. Then use this perturbation image to predict the model and compute the distances between each randomly generated perturbation and the original image. Using perturbations, predictions, and weights to fit an explainable (linear) model The supperpixels that have larger coefficients (magnitude) for the prediction of that class will be figured out by using the magnitude of the coefficients to determine the most important features. [18]

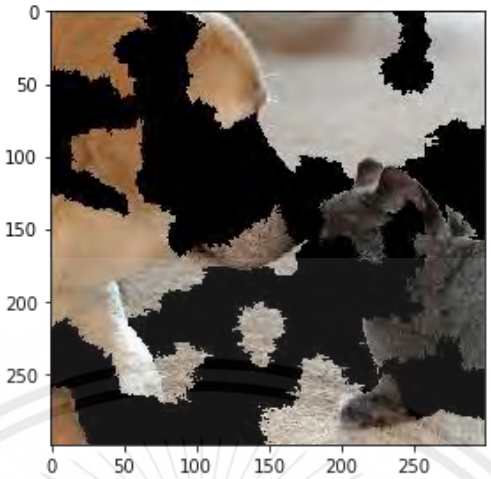


Fig 2.15: Example of perturbation image [18]

The result of LIME is dependent on the type of data. If the data is text, LIME will highlight the word that is the decision's reason for the model. If the data is tubular, LIME will provide the reason for the decision. If the data is an image, LIME will provide the decision region on the original image.

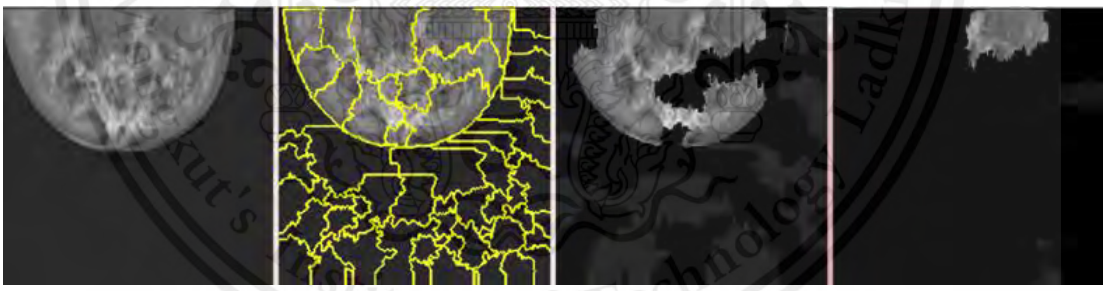


Fig 2.16: The process of LIME

2.10 Class Activation Mapping

CAM is from Class Activation Mapping, which is the first most popular XAI method to explain CNN models. They also said that CAM is the straightest way to explain AI, which is to visualize the activation map of the model. To visualize the activation map, CAM used the last convolution layer, which is the layer that has all the significant features that the algorithm uses to make a decision. The picture shows the architecture of CAM.

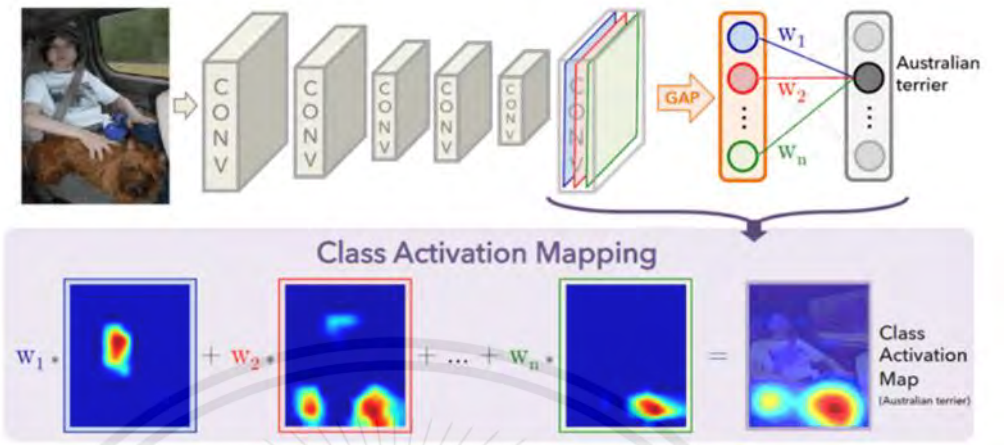


Fig 2.17: The architecture of CAM [19]

When you compute the CNN model, there will be two parts. The first one is convolutional layers, which are used to find the features of image classification. The second is the fully connected layer, which is used to predict the features obtained from convolution layers. In the last convolution layers, there is a feature map that shows the significant areas that AI uses to distinguish each class. We have to make this feature map a scalar and use that number to calculate it in the next step. To do that, CAM uses Global Average Pooling, or GAP. Next, do the training again and again to get the weight of each class. The class activation map is a summary of the multiplication of weight and feature maps. This class activation map is specific to each class.

CAM's equation:

$$L_{ij}^{c} = \sum_{k} w_{k}^{c} \cdot A_{ij}^{k}$$

Cam is the first evolution. There are more two improved versions which is Grad-CAM and Grad-CAM++. They tried to improve CAM because CAM have limitation. To get the weight, which is an important value to calculate CAM, the model have to retrain again and again for every class.

2.11 Gradient-Weighted CAM

Compared to CAM, Grad-CAM can solve the weak point about the retrain to find the weight. Then how does Grad-CAM find the weight to generate the class activation map? The answer is in its name, gradient-weighted CAM. Grad-CAM finds the weight through the gradient of the feature map. This can be possible because a gradient is actually a weight. The equation below proves this hypothesis.

$$w_k^c = \frac{1}{z}\sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

Now, Grad-CAM can solve the weakness of CAM, but Grad-CAM also has its own limitations; it cannot localize multiple occurrences of the same class, whereas Grad-CAM++ can.

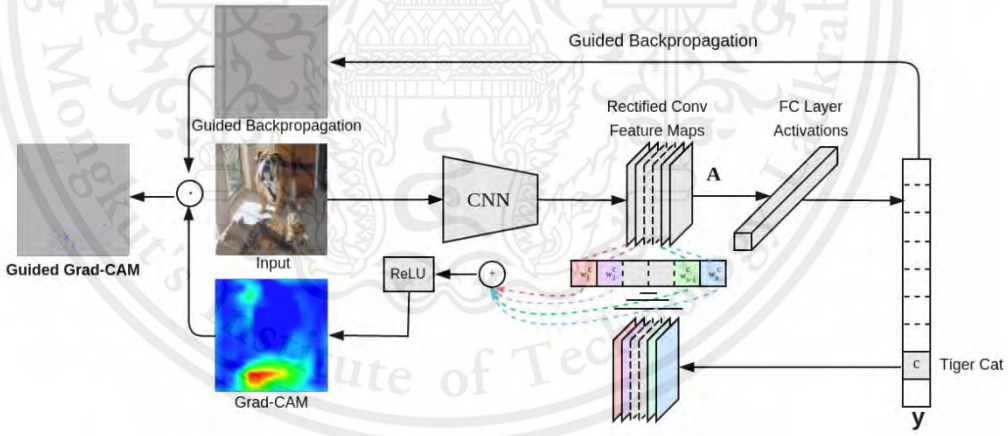


Fig 2.18: The architecture of Grad-CAM [20]

2.12 Grad-CAM++

Grad-CAM++ is the newest of these series. The concept of Grad-CAM++ is also like Grad-CAM but uses different equations. Grad-CAM++ can solve the weakness of Grad-CAM++, which is localizing multiple occurrences of the same class, so Grad-

CAM++ is more accurate for the model since it can localize the predicted class more precisely than Grad-CAM.

The method that Grad-CAM++ uses is a new equation to find the weight. The limitation of Grad-CAM is due to its smaller spatial footprint. This means some pixels in the image have become less significant over time. Grad-CAM scales all pixel gradients by the same factor, 1/Z. When the time passes, some pixels become less significant. Finally, those pixels will fade away. Grad-CAM++ solves this problem by multiplying the equation by the α_{ij}^{kc}, which is the value of α at pixel location (i,j) for the k-th feature map corresponding to the output class c. The new equation to find the weight and the equation to find the α_{ij}^{kc} are down below.

Equation to find the weight:

$$w_k^c = \frac{1}{z}\sum_i\sum_j \alpha_{ij}^{kc} \cdot relu\frac{\partial y^c}{\partial A_{ij}^k}$$

Equation to find α_{ij}^{kc}:

$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 y^c}{(\partial A_{ij}^k)^2}}{2 \cdot \frac{\partial^2 y^c}{\left(\partial A_{ij}^k\right)^2} + \sum_a \sum_b A_{ab}^k \frac{\partial^3 y^c}{(\partial A_{ij}^k)^3}}$$

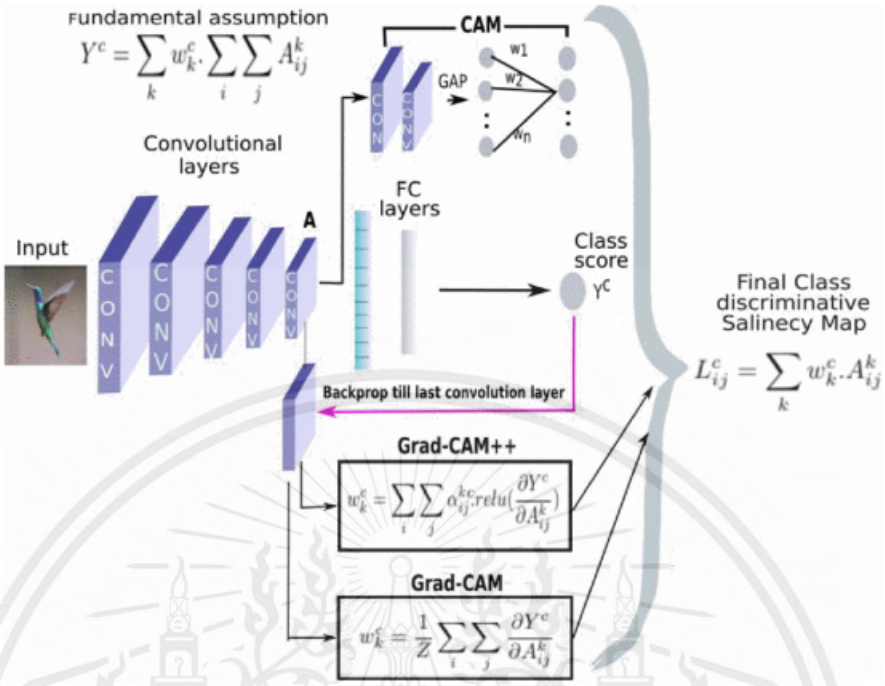


Fig 2.19: The architecture of Grad-CAM++ [21]

These three evaluations CAM has the same concept, which is to visualize the activation map. The difference is how to find the weight.

2.13 Mamography

A mammogram is a specific machine used to x-ray the breast because the breast is a special organ that has a special shape. It was hard to take a good-quality x-ray breast picture with another kind of x-ray machine. Mammography is a dedicated imaging modality for breast screening that uses a low-dose X-ray during breast examination. Mammography is currently the most effective tool for the early detection of breast cancer. It was recommended that women older than 40 should do the screening mammogram every year. Screening mammogram is one kind of mammogram that is used to detect and diagnose breast cancer early. If we find out about breast cancer too late, it cannot be cured, and that has caused the deaths of a lot of patients in the past. The screening mammogram helps a lot. Before mammograms came into Thailand, we used to use ultrasound at a 3 MHz transducer, but it was not appropriate for breast lesions. The image came out very white because 3 MHz is used to detect the deeper

organs, such as the abdominal and pelvic. For Asian women who have dense breasts, the detection of mammograms is difficult. Digital breast tomosynthesis (DBT) has the potential to improve early detection of breast cancer, leading to a better prognosis.

Usually, mammography images consist of many artifacts and noises, making it difficult to detect and understand the cancer at its primary stages. At this point, the CAD system will help in achieving high accuracy and sensitivity, which is beneficial for diagnosing mammography and also for patients. The hardness of mammography (the x-ray film from a mammogram) is that cancer can be a very tiny dust on the image, and that was hard to figure out. A radiologist who analyzes mammography will have two people double-check it. Reduce the errors that might occur from humans. Computer-aided diagnosis (CADx) is used to evaluate the structure identified by computer-aided detection (CADe), which is restricted to marking visible parts or structures in an image. The evaluation of CAD systems is measured by two major factors, such as sensitivity and specificity, and seeks suspicious structures. CAD systems may not be 100%, but their hit rate means sensitivity can be up to 98% these days. But the accuracy of the CAD depends on the conditions of the images used for training the system and factors like retrospective design. Image quality, conditions of mammography examination, radiologists' marks, type of lesion, and size and location of mass are highly influential.

CHAPTER 3
METHODOLOGY

3.1 Introduction

In Chapter 3, we described the way that we designed the experiment and how we did the experiment. This chapter provides three sections that will describe the project's design methodology (section 3.2), an interesting problem (section 3.3), and a proposed solution (section 3.4).

In Section 3.2, I describe and explain the methodology and code of each XAI method, including LIME, Grad-CAM, and Grad-CAM++. I also explain the CNN model that I used to apply with these XAIs, DenseNet201. In Section 3.3, I provide an example of the problems that I found during the experiment. The way, I tried to solve them, is provided in the section 3.4.

3.2 Design Methodology

The goal of the experiment is to examine three distinct XAI techniques on the CNN model DenseNet201: LIME, Grad-CAM, and Grad-CAM++. To compare these three methodologies of XAI, we used a breast cancer image from roboflow.com that already had a bounding box on the image. In this approach, we are able to compare the outcomes of these three XAIs in order to determine whether or not they represent the proper region of the AI's classification.

3.2.1 DenseNet201

The mammography is the medical image of the dataset that was used to train DenseNet201.The dataset from the Department of Industrial Engineering & Management, National Chin-Yi University of Technology, Taichung, Taiwan, does not provide the bounding box, so we have to use another dataset to compare the results of each XAI, using the dataset with the labels that already provide the bounding box of

the cancer in the image. This dataset is from roboflow.com. There are three datasets: training, validation, and testing. Training has 1696 images. Validation has 530 images. Testing has 201 tests. The image size is 640x640. Every dataset was provided with the text file. This text file tells us the class of each image and also provides the number of bounding boxes. This bounding box will be used to compare with the decision area that DenseNet201 predicts and generates using LIME, Grad-CAM, and Grad-CAM++.

DenseNet201 was written using a pre-train model. These DenseNet201 are from the kernel of TensorFlow. The mammography original size is 227x227. In the prepare data step, data augmentation was not applied because the dataset already provided a data augmentation version. The hyperparameters that we used in the training process include Adam Optimizer, 1e-4 learning rate, 30 epochs, and 4 batch sizes. Early stopping has also been used here to make sure that the model was saved with the best accuracy. The result of the model's testing is in the table below.

Table 3.1: The result of the DenseNet201 model to 3 groups of data, training, validation, and testing.

Data	Accuracy	Loss	Recall	precision	F1-score
Train	0.9993	0.0021	1.000	1.000	1.000
Validation	0.9974	0.0067	1.000	1.000	1.000
Test	1.000	3.325e-04	1.000	1.000	1.000

```python
# Load the DenseNet201 model with pre-trained weights
def create_model_DenseNet201(input_shape, n_out):
    input_tensor = Input(shape=input_shape)
    print(input_tensor.shape)
    base_model = applications.DenseNet201(weights= None, include_top=False,input_tensor=input_tensor)
    base_model.load_weights(r"D:\KMITL\4th year\Senior project\Dense201\pretrained_weights\densenet201_weights_tf_dim_ordering_tf_kernels_not
    x = GlobalAveragePooling2D()(base_model.output)
    x = Dropout(0.5)(x)
    x = Dense(1024, activation='relu')(x)
    x = Dropout(0.5)(x)
    final_output = Dense(n_out, activation='sigmoid', name='final_output')(x)
    model = Model(input_tensor, final_output)
    return model

model = create_model_DenseNet201(input_shape=(HEIGHT,WIDTH,CANAL), n_out=N_CLASSES)

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

for i in range(-5, 0):
    model.layers[i].trainable = True
model.summary()


for layer in model.layers:
    layer.trainable = True
```

Fig 3.1: The code of the DenseNet201 model

The concept of DenseNet201 is in the word “dense” in its name. Dense means the dense box, which contains some amount of covolution layers inside. For one DenseNet201 model, there are many dense boxes connected one by one, like a chain. Inside one dense box, the convolution layers are connected to each other. This means each convolution layer in a dense box can connect to all the convolution layers in the same dense box. Using this method, every covolution layer will have the previous data and top with its data, then send to the next one. There is no loss of features this way. You may observe that the features will be very large. That’s why it has a maximum pooling layer between each dense box. To make it smaller.

Looking at the function that was used to build DenseNet201, I used the pretrain model, which means this model already trained with a large dataset before learning our dataset; the weight and include_top are shown as none. This is because I do not want its weight (weight that trains from other types of images) and its fully connected layers. I want to get weight from our dataset. For the fully connected layers, the pretrain model has this prediction part with multiple classifications, but our project does only two class classifications. The new fully connected layers that I add in consist of global average pooling (GAP) and 1,024 dense layers. Because it already trains from other datasets, the accuracy of the model is the best among other CNN models in my previous work.

3.2.2 LIME

LIME has two main steps to take. The first step is to create a perturbation image. The second is to calculate the distance between each randomly generated perturbation and the original image. Then apply that distance to linear regression to get the zero and one numbers. One will be the pixels that have a signification for the model's decision, and zero is the pixel that's not related to the model's classification. On pixels with a value of 1, it will show the original image. On the other hand, pixels with a value of 0 will have a black color. Finally, the result of LIME will show only the decision area of the model.

```
def perturb_image(img,perturbation,segments):
  active_pixels = np.where(perturbation == 1)[0]
  mask = np.zeros(segments.shape)
  for active in active_pixels:
      mask[segments == active] = 1
  perturbed_image = copy.deepcopy(img)
  perturbed_image = perturbed_image*mask[:,:,np.newaxis]
  return perturbed_image
```

Fig 3.2: The code to create the perturbation image

```
#class_to_explain = 1
class_to_explain = img_class
simpler_model = LinearRegression()
simpler_model.fit(X=perturbations, y=predictions[:,class_to_explain], sample_weight=weights)
coeff = simpler_model.coef_[0]
coeff
```

Fig 3.3: The code of applying linear regression

In Fig. 3.2, the function that created the perturbation image The variable perturbation is the number of pixels that LIME is going to put the black color on as a perturbation. In this case, I used 600 pixels. More perturbation numbers can increase the accuracy of the LIME. After predicting the perturbation image with the DenseNet201 model, the next step is to measure the distance between each perturbation

point and the reference. The last step is located in Fig. 3.3, which is the application of linear regression to the distance that LIME measures from the perturbation image.

3.2.3 Grad-CAM

Grad-CAM generates a heatmap by making a summary of the multiplication of the gradient weight and feature map. These gradient weights are used instead of classification weights, which are actually the same thing. These methods of Grad-CAM try to make all the pixels in the image have the same significance, which can cause a lesser spatial footprint. The lesser spatial footprint occurs as time passes, and the significance of some pixels drops. The result of the Grad-CAM is to show the heatmap on the decision's area, but the lesser spatial footprint causes the limitation of the Grad-CAM, which cannot localize multiple occurrences of the same class.

Grad-CAM was written in two files. The first file in the.py file was used to create the grad-cam function. The input of this function includes: input_model (DenseNet201, which was saved after training the DenseNet201 model), image (the image that will be used to predict; this image should be prepared to be NumPy), layer name (the name of the last convolution layer), H (the height of the image, using the size that was used in the train process), and W (the width of the image, using the size that was used in the train process).

```
def grad_cam(input_model, image, layer_name,H=227,W=227):
    cls = np.argmax(input_model.predict(image))
    def normalize(x):
        """Utility function to normalize a tensor by its L2 norm"""
        return (x + 1e-10) / (K.sqrt(K.mean(K.square(x))) + 1e-10)
    """GradCAM method for visualizing input saliency."""
    y_c = input_model.output[0, cls]
    conv_output = input_model.get_layer(layer_name).output
    grads = tf.gradients(y_c, conv_output)[0]
    #grads = normalize(grads)
    gradient_function = K.function([input_model.input], [conv_output, grads])

    output, grads_val = gradient_function([image])
    output, grads_val = output[0, :], grads_val[0, :, :, :]

    weights = np.mean(grads_val, axis=(0, 1))
    cam = np.dot(output, weights)
    #print (cam)

    cam = np.maximum(cam, 0)
    #cam = resize(cam, (H, W))
    cam = zoom(cam,H/cam.shape[0])
    #cam = np.maximum(cam, 0)
    cam = cam / cam.max()
    return cam
```

Fig 3.4: The code to generate Grad-CAM function.

3.2.4 Grad-CAM++

Grad-CAM++ generates a heatmap by making a summary of the multiplication of the gradient weight and feature map. These gradient weights are used instead of classification weights, which are actually the same thing. This is the same method of Grad-CAM, which try to make all the pixels in the image have the same significance and cause a lesser spatial footprint. The lesser spatial footprint occurs as time passes, and the significance of some pixels drops. The result of the Grad-CAM is to show the heatmap on the decision's area, but the lesser spatial footprint causes the limitation of the Grad-CAM, which cannot localize multiple occurrences of the same class. Grad-CAM++ solve this problem by by multiplying the equation by the α_{ij}^{kc}, which is the value of α at pixel location (i,j) for the k-th feature map corresponding to the output class c.

Grad-CAM++ was written in two files. The first file in the.py file was used to create the grad-cam function. The input of this function includes: input_model (DenseNet201, which was saved after training the DenseNet201 model), image (the image that will be used to predict; this image should be prepared to be NumPy), layer name (the name of the last convolution layer), H (the height of the image, using the size that was used in the train process), and W (the width of the image, using the size that was used in the train process).

```
def grad_cam_plus(input_model, img, layer_name,H=227,W=227):
    cls = np.argmax(input_model.predict(img))
    y_c = input_model.output[0, cls]
    #cost = 全部のラベルの値。cost*label_indexでy_cになる
    conv_output = input_model.get_layer(layer_name).output
    #conv_output = target_conv_layer, mixed10の出力1,5,5,2048
    grads = tf.gradients(y_c, conv_output)[0]
    #grads = normalize(grads)

    first = K.exp(y_c)*grads
    second = K.exp(y_c)*grads*grads
    third = K.exp(y_c)*grads*grads*grads

    gradient_function = K.function([input_model.input], [y_c,first,second,third, conv_output, grads])
    y_c, conv_first_grad, conv_second_grad,conv_third_grad, conv_output, grads_val = gradient_function([img])
    global_sum = np.sum(conv_output[0].reshape((-1,conv_first_grad[0].shape[2])), axis=0)

    alpha_num = conv_second_grad[0]
    alpha_denom = conv_second_grad[0]*2.0 + conv_third_grad[0]*global_sum.reshape((1,1,conv_first_grad[0].shape[2]))
    alpha_denom = np.where(alpha_denom != 0.0, alpha_denom, np.ones(alpha_denom.shape))
    alphas = alpha_num/alpha_denom

    weights = np.maximum(conv_first_grad[0], 0.0)

    alpha_normalization_constant = np.sum(np.sum(alphas, axis=0),axis=0)

    mask = (alpha_normalization_constant == 0)
    alpha_normalization_constant[mask] = 1

    alphas /= alpha_normalization_constant.reshape((1,1,conv_first_grad[0].shape[2]))

    deep_linearization_weights = np.sum((weights*alphas).reshape((-1,conv_first_grad[0].shape[2])),axis=0)
    #print deep_linearization_weights
    grad_CAM_map = np.sum(deep_linearization_weights*conv_output[0], axis=2)

    # Passing through ReLU
    cam = np.maximum(grad_CAM_map, 0)
    cam = zoom(cam,H/cam.shape[0])
    cam = cam / np.max(cam) # scale 0 to 1.0
    #cam = resize(cam, (227,227))

    return cam
```

Fig 3.5: The code to generate Grad-CAM++ function.

Before applying images to these functions, preparation was needed. The first step of preparation is to load the image with a size of 227x227 (the image size that was used when training the model) and load it's in NumPy array type. These two functions

were used in the code below. Then expand the axis of the image by using the np.expand_dims function. The last step is to apply the np.preprocess_input function.

```
paths = [r"D:\KMITL\4th year\Senior project\Bounding box\cancer.v1i.yolov8\test\images\11_jpg.rf.29f9c7cbbc5540e31b5f8da6d02f0759.jpg"
for path in paths:
    path = os.path.join("image",path)
    orig_img = np.array(load_img(path,target_size=(227,227)),dtype=np.uint8)
    img = np.array(load_img(path,target_size=(227,227)),dtype=np.float64)
    img = np.expand_dims(img,axis=0)
    img = preprocess_input(img)
    predictions = model.predict(img)
    top_n = 1
    #top = decode_predictions(predictions, top=top_n)[0]
    #cls = np.argsort(predictions[0])[-top_n:][::-1]

    gradcam=gradcamutils.grad_cam(model,img,layer_name='conv5_block32_concat')
    gradcamplus=gradcamutils.grad_cam_plus(model,img,layer_name='conv5_block32_concat')
```

Fig 3.6: The code to apply Grad-CAM and Grad-CAM++ function.

3.3 Interesting Problems

The main problems that we always had were in the process of applying LIME, Grad-CAM, and Grad-CAM++ with DenseNet201. The other problem is how to write the bounding box using the text file.

3.3.1 TensorFlow version

The version of TensorFlow that we used is 2.5. TensorFlow that is newer than 2.0 has one feature that is different from the downgrade version. TensorFlow 2.0 and later versions have an eager execution feature. As opposed to creating a computational graph and delaying execution, as is the case in the typical TensorFlow execution mode, eager execution in TensorFlow allows operations to be done instantly as they are requested in Python. TensorFlow's API becomes more interactive and Pythonic with eager execution, much like NumPy.

3.3.2 Last convolution layer

Grad-Cam and Grad-CAM++ are used in the last convolution layer to calculate the activation map and weight, which are the two main elements needed to create a heatmap that will show the decision area of the DenseNet201 model. This last

convolution is important. Applying the wrong name to the last convolution layer causes an error.

3.3.3 Image preparation

In all of the XAI method, the original code is aways work with other CNN architecture. Some of them is VGG16, or RestNet, but the CNN model that is used in this study is DenseNet201. That's cause a lot of problems because the code is all related to each other. When change one variable, it is necessary to change all along the code. The part is the part that is the easiest to cause the error.

3.3.4 Image size to bounding box

There are a problem when ploting the bounding box. The bounding box is too small when compare to the original image size.

3.4 Proposed Solution

3.4.1 TensorFlow version

The solution to the problem has two ways. The first one is to use a different environment from the normal environment to downgrade the tensor flow in that environment. This can cause confusion in the future. The second way is to add one line of code. This is a code to close eager execution, so we can adjust by hand.

```
tf.compat.v1.disable_eager_execution()
```

Fig 3.7: The code to close eager execution

3.4.2 Last convolution layer

Apply the model.summary() fuction to see the name of the last convolurion layer. The last convolution layer is name “conv5_block32_concat”.

```
for i in range(-5, 0):
    model.layers[i].trainable = True
model.summary()


for layer in model.layers:
    layer.trainable = True
```

```
conv5_block32_2_conv (Conv2D)   (None, 7, 7, 32)    36864       conv5_block32_1_relu[0][0]
conv5_block32_concat (Concatena (None, 7, 7, 1920)  0           conv5_block31_concat[0][0]
                                                                conv5_block32_2_conv[0][0]
```

Fig 3.8: The code to summary the CNN model

3.4.3 Image preparation

For LIME, the original code of LIME was prepare for InceptionV3 model. That’s why to prepare the image, they provide Xi = (Xi - 0.5)*2, Xi is an image. When they want to show the image, which should plot on the original size, they will provide “skimage.io.imshow(Xi/2+0.5) “ For DenseNet201, it should be Xi = Xi/255. Another thing is the size of image, DenseNet’s input image size is 277x277. This is also different from the InceptionV3.

```
image = cv2.resize(Xi, (227,227))
print(image.shape)

img_preprocessed = image/255.0

image_array = np.expand_dims(img_preprocessed, axis=0)
print(image_array.shape)
```

```
(640, 640, 3)
```

Fig 3.9: Image preparation of LIME

3.4.4 Image size to bounding box

The solution is to multiply the number of the bounding box with the width and height of the original image.

```
x = x*640
y = y*640
width = width*640
height = height*640
```

Fig 3.10: Image preparation of LIME

3.5 Summary

The first step of this experiment is training the DenseNet201 model, which is the pretrain model, with a breast cancer image dataset from the Department of Industrial Engineering and Management, National Chin-Yi University of Technology, Taichung, Taiwan. After running this model and saving it at the best accuracy, I used it to predict breast cancer images. The results of these predictions are 0 and 1 classes. To explain where the decision area of breast cancer is located, I applied them to three XAI methods: LIME, Grad-CAM, and Grad-CAM++. The results of these three XAIs were compared with the bounding box from the original image. The dataset, which included labels from INBREAST 2012 and MIAS Mammography, supplied these bounding boxes.

The majority of the issues here were coding-related technical issues. Versions of Numpy, Tensorflow, and other tools don't match the code. Many errors were generated by the disparities in the picture types and sizes. In my opinion, the most crucial step is preparing the data before using it with the XAIs' methodologies.

CHAPTER 4
EXPERIMENTAL RESULT AND DISCUSSION

4.1 Introduction

Chapter 4 shows the results of three of XAI, LIME, Grad-CAM, and Grad-CAM++. The results of the tests are summarized in Section 4.2: Results and Discussion; Section 4.3: Summation of Results and Discussion.

In Section 4.2, I will show the comparison between each XAI method and the bounding box image. Then, show the comparison between all the results I get. followed by the discussion of each XAI method.

4.2 Result and Discussion

This section is a comparison between the results of LIME, Grad-CAM, and Grad-CAM++. The images used in this experiment are from robotflow.com. They all have the breast cancer image with the bounding box, which locates the location of the breast cancer inside the images.

There are three images from three different datasets: training, validation, and testing, in order. The bounding box on the left-hand side locates the cancer area with a red color. LIME on the right-hand side shows the decision region that the DenseNet201 model has.

Table 4.1: Original image and the result of LIME

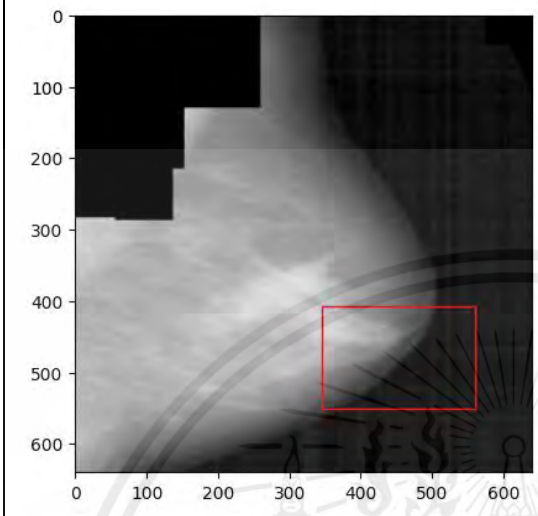
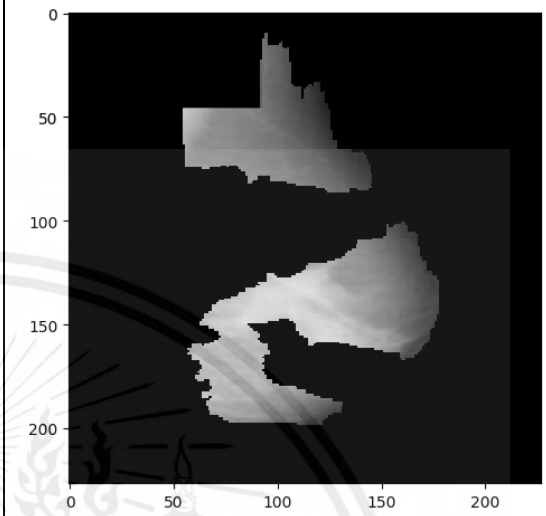
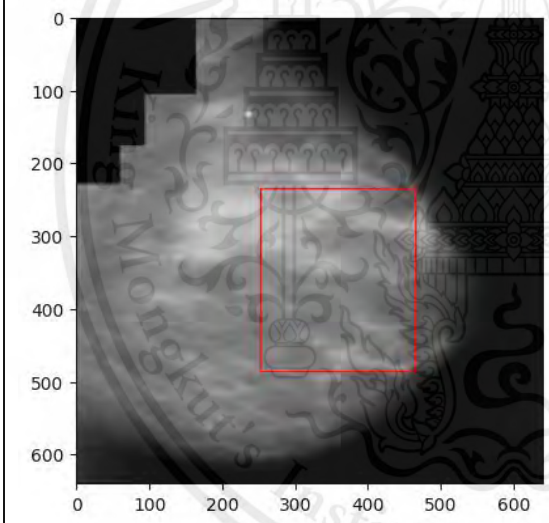
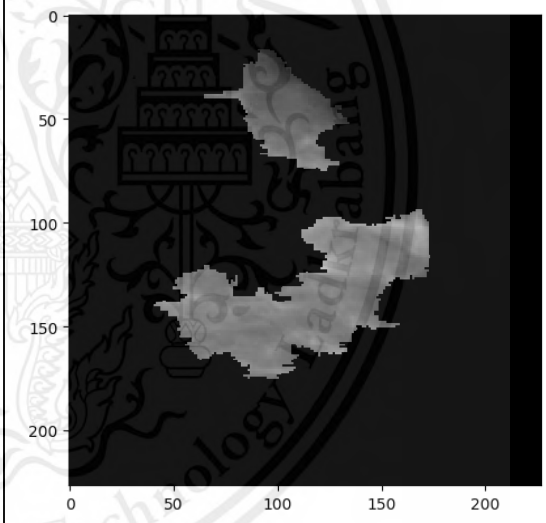
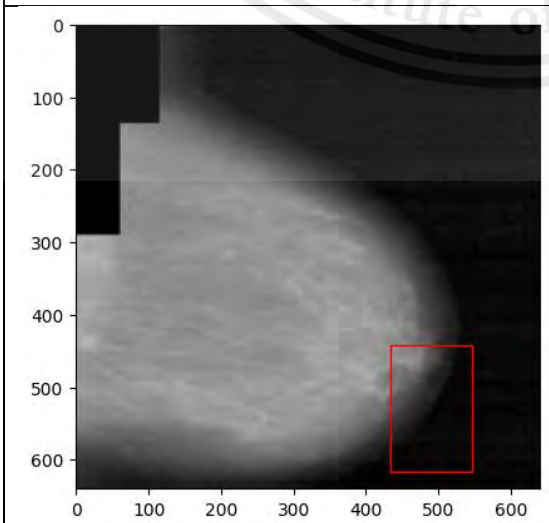
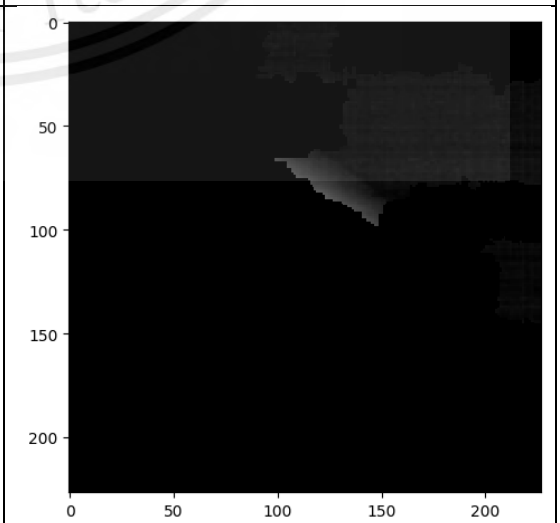
Original image	LIME
	
	
	

Table 4.2: Original image and the result of Grad-CAM

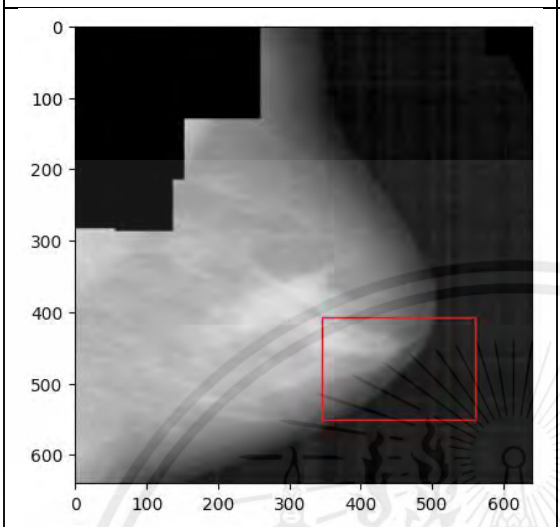
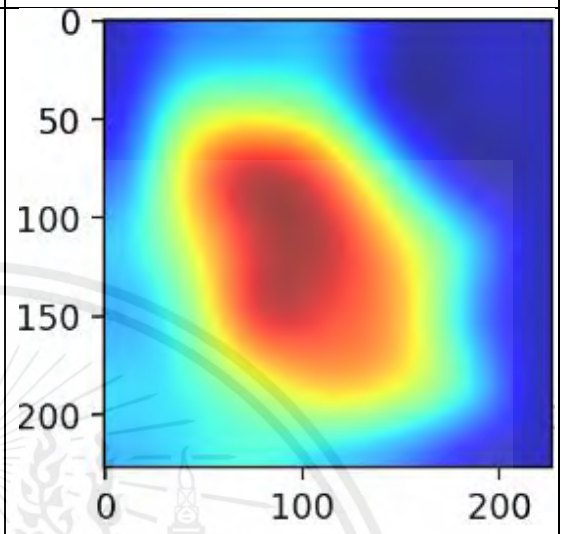
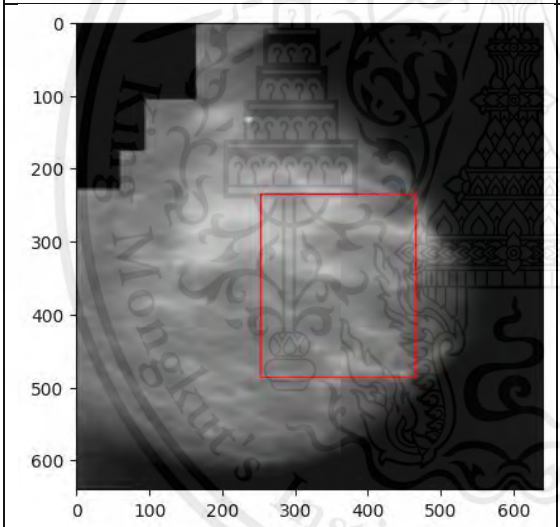
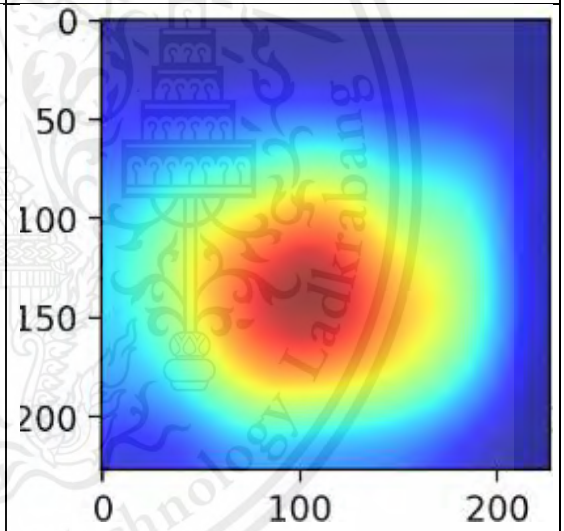
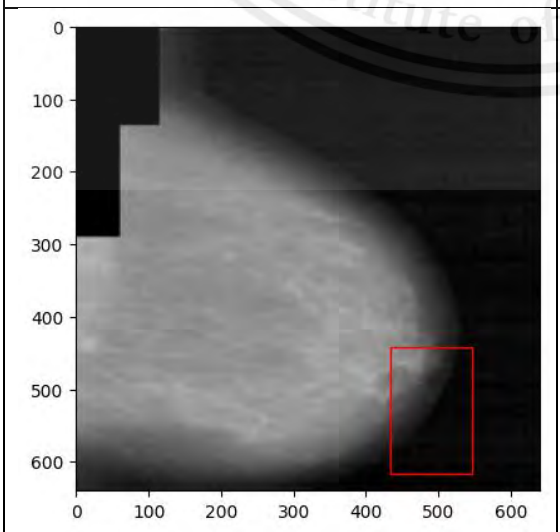
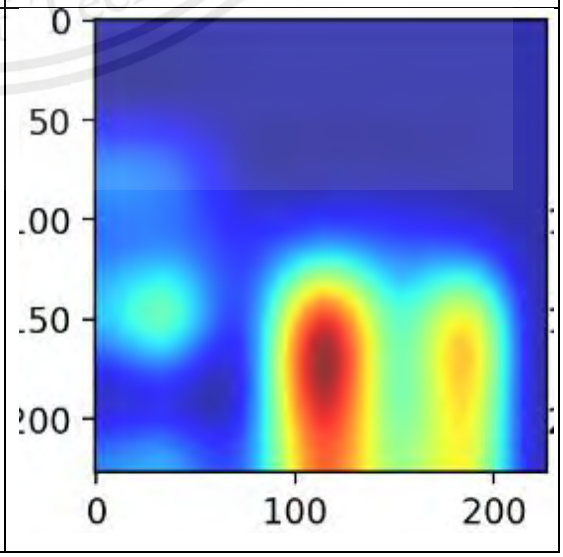
Original image	Grad-CAM
	
	
	

Table 4.3: Original image and the result of Grad-CAM++

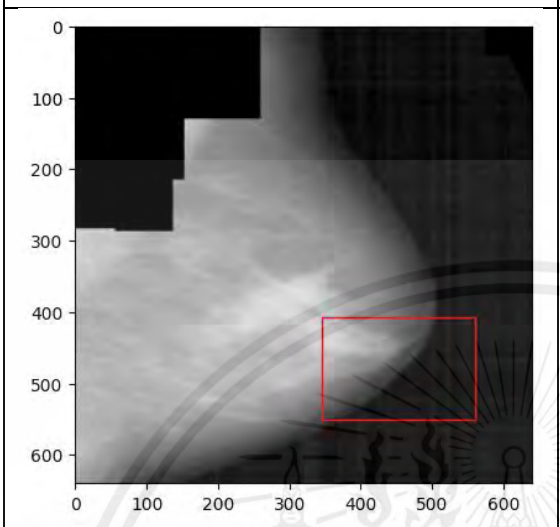
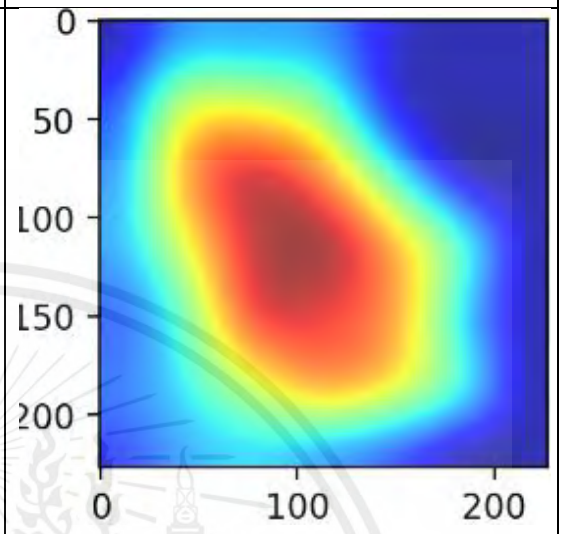
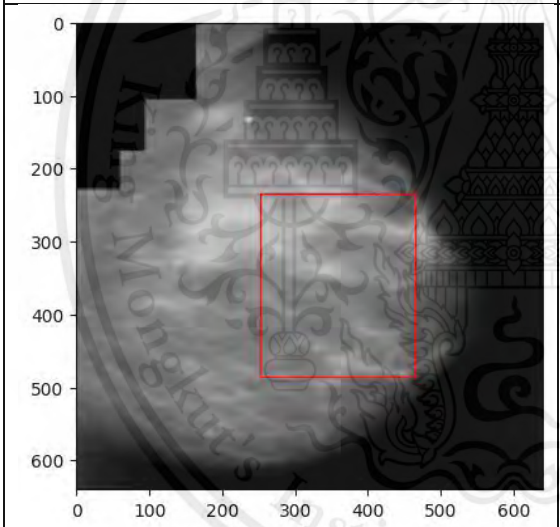
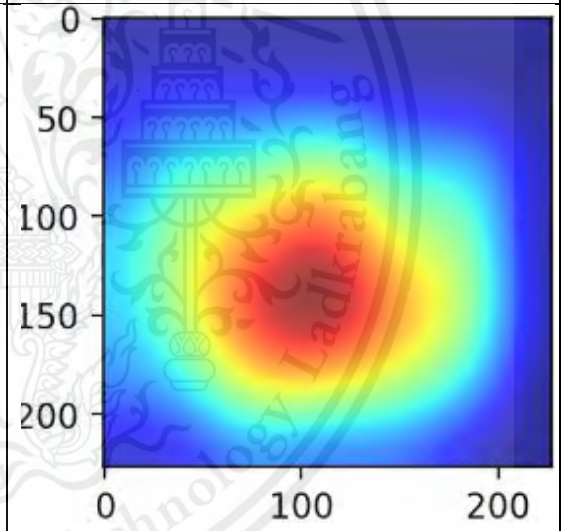
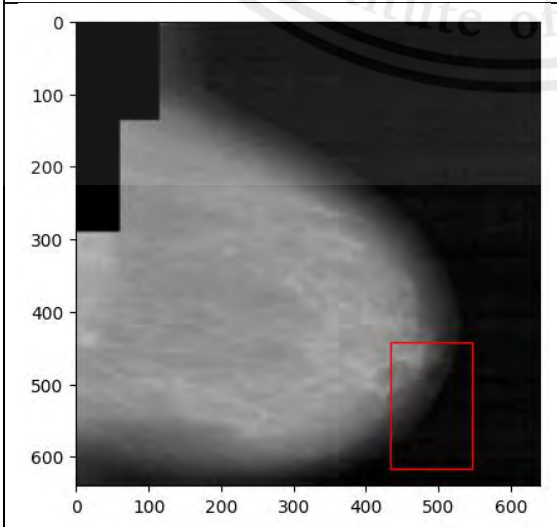
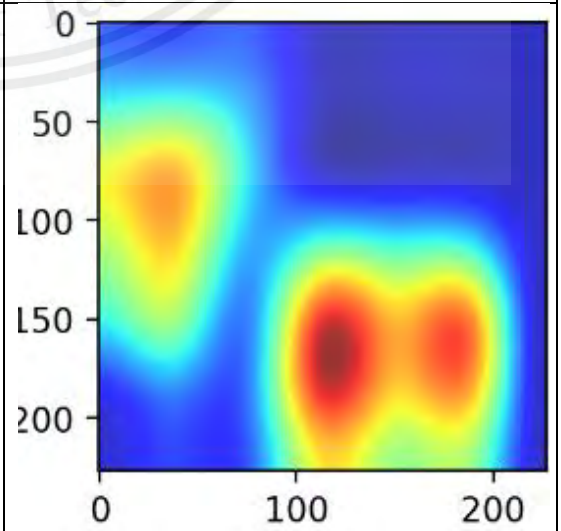
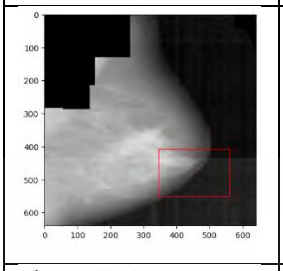
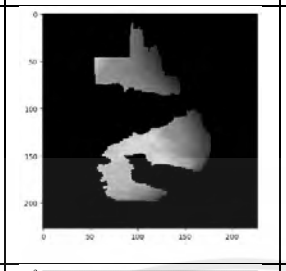
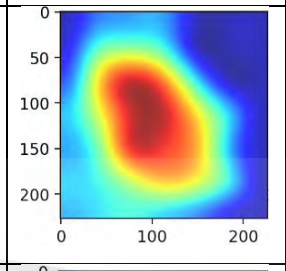
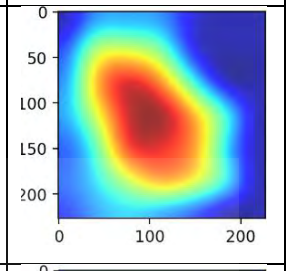
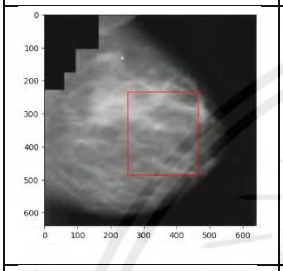
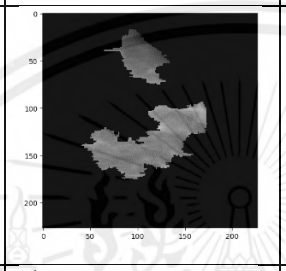
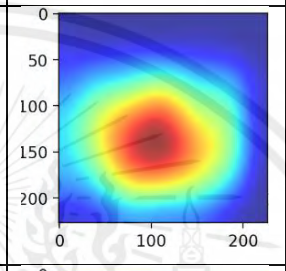
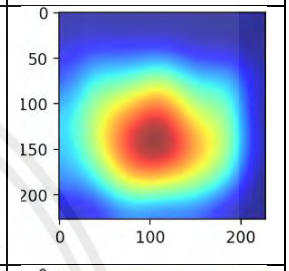
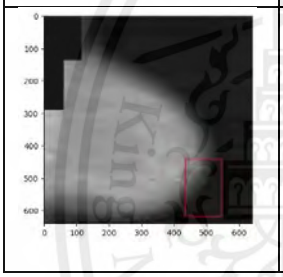
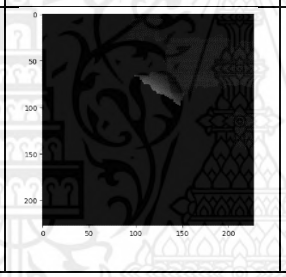
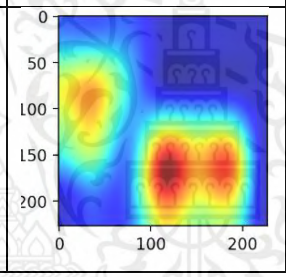
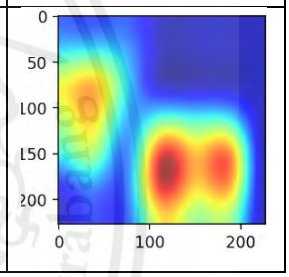
Original image	Grad-CAM
	
	
	

Table 4.4: Original image and the result of LIME, Grad-CAM, and Grad-CAM++

Original image	LIME	Grad-CAM	Grad-CAM++
			
			
			

4.2.1 LIME

LIME stands for Local Interpretable Model-Agnostic Explanations. LIME can make a local approximation to the complex decision area. The result of LIME shows the decision region of the model. Other spaces will be black spots. This is due to the linear regression used to compute the magnitude, or the coefficients, into zero and one. The larger coefficients (the area that the model pays attention to) will be one and will be shown in the final result. The area that has smaller coefficients will be zero and will be black on the final result. That's how LIME determines the most important features.

LIME works by creating the perturbation image, which is a randomly generated black spot on the image's pixels. The higher the perturbation number, the better the accuracy will be. In this experiment, the perturbation number is 600. To do the

perturbation image, LIME randomly creates zeros and ones to make a black spot on the image.

When comparing the results of LIME to the other two methods, Grad-CAM and Grad-CAM++, LIME can show a clearer result and be easier to interpret. Anyway, the objective of LIME is to show the weight of the model, which humans can understand, but the accuracy is lower than the other methods. I said that because some of the results of LIME are too much far from the bounding box region.

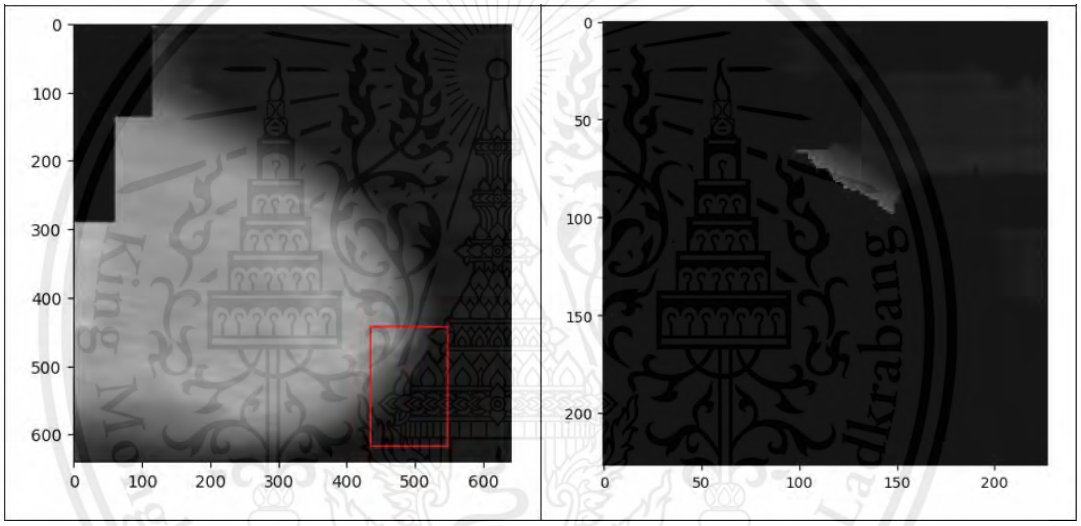


Fig 4.1: Original image with bounding box and LIME's result

4.2.2 Grad-CAM

Grad-CAM's result generated by the summation of each gradient-weight multiply with each feature map from last convolution layer. Grad-CAM can solve the weak point about the retrain to find the weight. Then how does Grad-CAM find the weight to generate the class activation map? The answer is in its name, gradient-weighted CAM. Grad-CAM finds the weight through the gradient of the feature map. This can be possible because a gradient is actually a weight.

Grad-Cam is one of the explainable AIs. It used to show the region of the image that the model used to make a decision. Compared to LIME, Grad-Cam will give us the spectrum of the possibility of the model's decision area, while LIME will give us the area only. Compared to Grad-CAM++, Grad-CAM provides a smaller region of decision. This makes Grad-CAM look better than Grad-CAM++, but actually, Grad-CAM++ can provide all the regions that are important to the model's decision. The accuracy of Grad-CAM is better than LIME but less than Grad-CAM++.

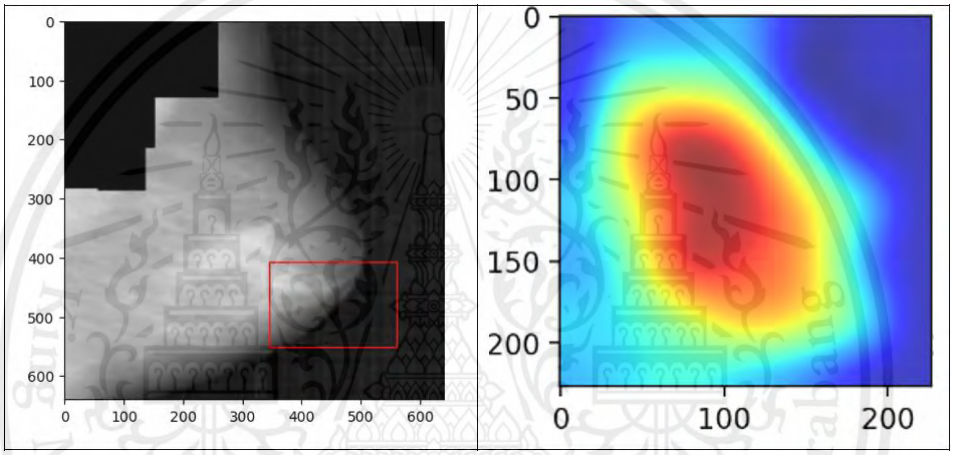


Fig 4.2: Original image with bounding box and Grad-CAM's result

Grad-CAM's weakness, which cannot localize multiple occurrences of the same class, is not relevant to this study because the mammography has only one occurrence anyway. The difference between Grad-CAM and Grad-CAM++ here is decreasing.

4.2.3 Grad-CAM++

Grad-CAM++ is the newest of these series. The concept of Grad-CAM++ is also like Grad-CAM but uses different equations. Grad-CAM++ can solve the weakness of Grad-CAM++, which cannot localize multiple occurrences of the same class, so Grad-CAM++ is more accurate for the model since it can localize the predicted class more precisely than Grad-CAM. Anyway, the weakness that cannot localize multiple occurrences of the same class is not an effect of this study. Mammography has only one

occurrence. The difference between Grad-CAM and Grad-CAM++ here is not that much.

Grad-CAM++ has the most accuracy. Here, you can observe on the result image that the red spot in Grad-CAM++ is located the closest to the bounding box. When compared to the other two methods, Grad-CAM++ makes it hard to see the details.

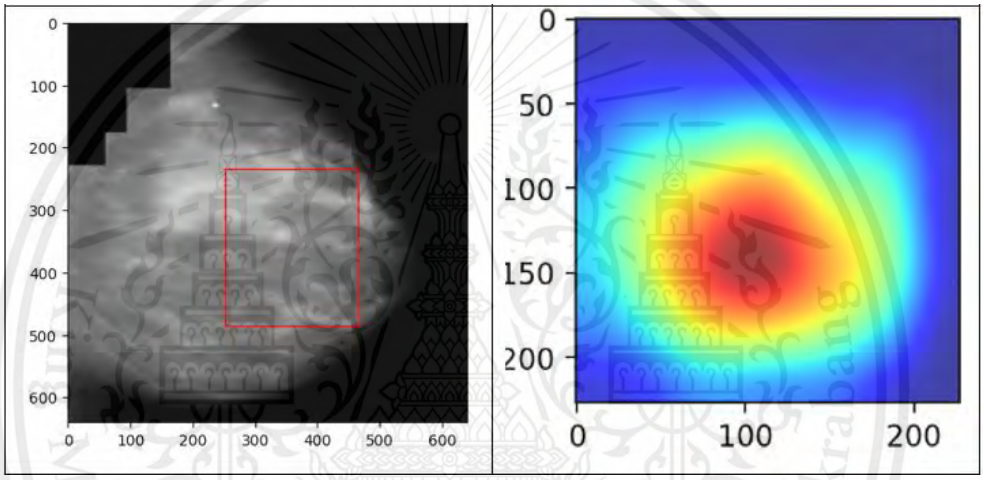


Fig 4.3: Original image with bounding box and Grad-CAM++'s result

4.3 Summary

Grad-CAM++ has the greatest accuracy among these three methods. Follow by Grad-CAM and LIME. The interpretation of AI's decision is the easiest one here, LIME. Then follow Grad-CAM and Grad-CAM++. The best method to use with breast cancer here is Grad-CAM++. The reason is that it is the most suitable one. LIME has less accuracy when compared to the other two methods, so it should not be LIME. No matter how easy it is to interpret the result, LIME has Then compare between Grad-CAM and Grad-CAM++. These two methods give almost the same results because mammography images have only one occurrence. Then, any Grad-CAM or Grad-CAM++ is good, so I will choose the one that can give me the best accuracy, which is Grad-CAM++.

This CNN model and all the XAI methods in this study cannot be used in the real world. This is just for the study because we used a small dataset to train DenseNet201.

CHAPTER 5
CONCLUSION

5.1 Introduction

In this chapter, we first summarize all the work in each chapter of this report (section 5.2). Then we conclude what I have learned during the project and the conclusion according to the objective in Section 5.3, and finally, in Section 5.4, we discuss future work.

5.2 Summary

This is a summary of each chapter from introduction to result and discussion.

Chapter 1 introduction

Artificial intelligence, or AI, is now widely used in many industries, for example, financial services, insurance, telecommunications, life sciences, and healthcare. The weakness of AI, especially in the medical industry, is the lack of transparency and explanation. AI is still not widely trusted, and its accuracy cannot be guaranteed. Knowing the rationale behind an AI decision will make it easier for you to choose whether to accept its response.

XAI, or explainable artificial intelligence, is an important tool to prove the reliability of AI, especially in medical fields. XAI is used to explain the reason behind AI's decision. There are various XAI methodologies and types. In this study, there are three XAI methodologies, including LIME, Grad-CAM, and Grad-CAM++. The objective of this project is to compare the different types of XAI and find the best XAI for image classification.

All of these XAI methodologies and the CNN model that was used in this study were created using Python in a Jupyter notebook, using TensorFlow, CV2, Matplotlib, etc.

Chapter 2 review theory related

There are 13 topics of theory related to this study, including artificial intelligence (AI), deep learning (DL) vs. machine learning (ML), convolutional neural networks (CNN), tensorflow, data augmentation, datasets, explainable artificial intelligence (XAI), class activation mapping (CAM), gradient-weighted CAM (Grad-CAM), Grad-CAM++, and mammography.

Chapter 3 methodology

This chapter is about the experiment and the problems while doing the experiment. There are 4 main parts: DeanseNet201, LIME, Grad-CAM, and Grad-CAM++. The experiment plan is to compare three different XAI methods, including LIME, Grad-CAM, and Grad-CAM++, on the CNN model DenseNet201 by using images from robotflow. The image provides the bounding box, so we can compare the region of the bounding box to the region of the decision area generated by three XAI methodologies.

Chapter 4 experimental result and discussion

LIME can show a clearer result and be easier to interpret, but its accuracy is the lowest among these two XAIs. Anyway, the objective of LIME is to show the weight of the model, which humans can understand, but the accuracy is lower than the other methods.

Grad-CAM's weakness, which cannot localize multiple occurrences of the same class, is not relevant to this study because the mammography has only one occurrence

anyway. The difference between Grad-CAM and Grad-CAM++ here is decreasing. The accuracy of Grad-CAM is better than LIME.

Grad-CAM++ has the greatest accuracy among these three methods. Follow by Grad-CAM and LIME. The interpretation of AI's decision is the easiest one here, LIME. Then follow Grad-CAM and Grad-CAM++. The best method to use with breast cancer here is Grad-CAM++. The reason is that it is the most suitable one. LIME has less accuracy when compared to the other two methods, so it should not be LIME. No matter how easy it is to interpret the result, LIME has Then compare between Grad-CAM and Grad-CAM++. These two methods give almost the same results because mammography images have only one occurrence. Then, any Grad-CAM or Grad-CAM++ is good, so I will choose the one that can give me the best accuracy, which is Grad-CAM++.

5.3 Conclusions

In this study, I learn the concept of each XAI methodology, including LIME, Grad-CAM, and Grad-CAM++. Then apply LIME, Grad-CAM, and Grad-CAM++ to the breast cancer image classification using the DenseNet201 model. The reason to do this is because we want to compare which XAI is the most suitable one for medical image classification. The model and the XAI methods in this study cannot be used in the real world because I use a small dataset.

To compare the results of these three XAI methodologies, I used the breast cancer image dataset from robotflow.com because this dataset provided the bounding box, which located the breast cancer area in the image. Now we can compare the area of the bounding box and the area of the decision region that was generated from LIME, Grad-CAM, and Grad-CAM++.

The result of LIME shows the decision region and blacks out an unrelated region. This way of expressing LIME makes the interpretation easy, but the accuracy of LIME is not good compared to Grad-CAM and Grad-CAM++. The results of Grad-

CAM and Grad-CAM++ are almost the same. This is because Grad-CAM and Grad-CAM++ have the same concept of visualizing the activation map as an explanation of the CNN model. The interpretation is harder than LIME. You can see from the result image that LIME is clean and clear. This makes LIME easier to understand, but Grad-CAM and Grad-CAM++ have more accuracy. When comparing Grad-CAM and Grad-CAM++, they are not that different. This is due to the limitation of Grad-CAM, which cannot localize multiple occurrences of the same class. Mammography has only one occurrence. Anyway, Grad-CAM++ can show more details than Grad-CAM. Grad-CAM++ also shows the region closest to the bounding box.

5.4 Future work

In this work, I used the bounding box as a reference to compare with the results of the three XAI methods, including LIME and Grad-CAM. Grad-CAM++. In this method, the decision area is unclear because the heatmap covers the entire image. The next project that I will do in the next semester can improve this point.

In my future work, I will move to work more about the detection model, such as the YOLO model, faster R-CNN, SSD, or mask R-CNN. I will combine the detection model with Grad-CAM++ or other interesting XAIs. This can improve the explainability of the localization of the cancer in the breast cancer image. This will generate the heatmap inside the bounding box, so it can show a more specific area of cancer in the image.

REFERENCES

[1] R. Winastwan, "Interpreting Image Classification Model with LIME," Medium. Accessed: Oct. 10, 2023. [Online]. Available: https://towardsdatascience.com/interpreting-image-classification-model-with-lime-1e7064a2f2e5

[2] "What is AI (Artificial Intelligence)? | McKinsey." Accessed: Sep. 24, 2023. [Online]. Available: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-ai

[3] A. Takyar, "AI Use Cases & Applications Across Major industries," LeewayHertz - AI Development Company. Accessed: Aug. 18, 2023. [Online]. Available: https://www.leewayhertz.com/ai-use-cases-and-applications/

[4] "Computer Aided Diagnosis - Medical Image Analysis Techniques | IntechOpen." Accessed: Jan. 10, 2023. [Online]. Available: https://www.intechopen.com/chapters/56615

[5] M. Rathi, "FeedForward Neural Networks." Accessed: Nov. 08, 2023. [Online]. Available: https://mukulrathi.com/demystifying-deep-learning/feed-forward-neural-network/

[6] A. Kumar, "Different Types of CNN Architectures Explained: Examples," Analytics Yogi. Accessed: Nov. 08, 2023. [Online]. Available: https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/

[7] Vivek Singh Bawa., "Basic architecture of RNN and LSTM," TAIL@ti Deep Learning & Computer Vision. Accessed: Oct. 21, 2023. [Online]. Available: http://pydeeplearning.weebly.com/1/post/2017/01/basic-architecture-of-rnn-and-lstm.html

[8] “Figure 4: A basic structure of a CNN (O’Shea & Nash, 2015),” ResearchGate. Accessed: Nov. 08, 2023. [Online]. Available: https://www.researchgate.net/figure/A-basic-structure-of-a-CNN-OShea-Nash-2015_fig2_349054891

[9] S. Pokhrel, “Beginners Guide to Understanding Convolutional Neural Networks,” Medium. Accessed: Jan. 12, 2023. [Online]. Available: https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d

[10] “Fig. 4. Pooling layer operation oproaches 1) Pooling layers: For the...,” ResearchGate. Accessed: Jan. 23, 2023. [Online]. Available: https://www.researchgate.net/figure/Pooling-layer-operation-oproaches-1-Pooling-layers-For-the-function-of-decreasing-the_fig4_340812216

[11] P. Mahajan, “Fully Connected vs Convolutional Neural Networks,” The Startup. Accessed: Jan. 23, 2023. [Online]. Available: https://medium.com/swlh/fully-connected-vs-convolutional-neural-networks-813ca7bc6ee5

[12] R. Thakur, “Step by step VGG16 implementation in Keras for beginners,” Medium. Accessed: Nov. 08, 2023. [Online]. Available: https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c

[13] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks.” arXiv, Jan. 28, 2018. Accessed: Nov. 08, 2023. [Online]. Available: http://arxiv.org/abs/1608.06993

[14] “Dataset of breast mammography images with masses | Elsevier Enhanced Reader.” Accessed: Apr. 29, 2023. [Online]. Available: https://reader.elsevier.com/reader/sd/pii/S2352340920308222?token=E9800B2F406C3AE99AF059D6E587E31054C9DAFDFD158E49FC420BDF4A2B7

A07F91601026AEAC47DDE3D64BD4AACC058&originRegion=eu-west-1&originCreation=20230429144345

[15] Ines C. Moreira, MSc student, Igor Amaral, MSc, Ines Domingues, MSc, Antonio Cardoso, MD, Maria Joao Cardoso, PhD, and Jaime S. Cardoso, PhD, "INbreast: Toward a Full-field Digital Mammographic Database," Academic Radiology, vol. 2011, Nov. 2011, doi: https://doi.org/10.1016/j.acra.2011.09.014.

[16] "Mammographic Image Analysis Homepage - Databases." Accessed: Nov. 25, 2023. [Online]. Available: https://www.mammoimage.org/databases/

[17] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier." arXiv, Aug. 09, 2016. Accessed: Nov. 09, 2023. [Online]. Available: http://arxiv.org/abs/1602.04938

[18] Cristian Arteaga, "Jupyter Notebook Viewer," arteagac.github.io. Accessed: Nov. 09, 2023. [Online]. Available: https://nbviewer.org/url/arteagac.github.io/blog/lime_image.ipynb

[19] S.-H. Tsang, "CAM: Learning Deep Features for Discriminative Localization (Weakly Supervised Object…," Medium. Accessed: Dec. 07, 2023. [Online]. Available: https://sh-tsang.medium.com/cam-learning-deep-features-for-discriminative-localization-weakly-supervised-object-7cab7b31f972

[20] M. Chetoui, "Grad-CAM- Gradient-weighted Class Activation Mapping," Medium. Accessed: Dec. 07, 2023. [Online]. Available: https://medium.com/@mohamedchetoui/grad-cam-gradient-weighted-class-activation-mapping-ffd72742243a

[21] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks," in 2018 IEEE Winter Conference on Applications of

Computer Vision (WACV), Mar. 2018, pp. 839–847. doi: 10.1109/WACV.2018.00097.