Developing AI approaches to automate the Pneumonia detection of chest X-ray images

Hongtao Yao

Master of Science
School of Informatics
University of Edinburgh
2022

Abstract

This project investigated the performance of four different models (VGG19, ResNet50, Inception-ResNet-v2 and Swin Transformer) for pneumonia classification. The innovation of this project is that it applied the Swin Transformer model on the dataset[1] for the first time and proved the feasibility of the new Transformer model (Swin Transformer) for medical imaging, which differs from the traditional Convolutional Neural Networks(VGG19, ResNet50 and Inception-ResNet-v2). A comparative analysis of the models was conducted by comparing their performance under different hyperparameters (learning rate and the number of epochs). The effects of transfer learning and different data enhancement methods on the models were investigated. We proved the pre-trained weights from other source tasks could bring better performance than model training from scratch. We also observed that contrast enhancement augmentation could provide more improvement than the traditional data augmentation method (including rotation and flipping) in medical imaging. The Grad-CAM heatmap was applied to analyse the convolutional neural networks' behaviour. Eventually, it was found that the VGG19 model was able to have the best performance in small data sets, but the Swin Transformer kept improving and it had the potential to outperform other models.

[1] Kaggle Chest X-ray Images [33]

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Hongtao Yao*)

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Heather Yorston. Thanks for accepting me to be involved in this meaningful project and supporting me in times of frustration and hardship. She has shown me the way in my projects and has always been there to help. She was always patient with me and encouraged me. Without her help, the whole process would have been very difficult.

I would also like to express my appreciation to Dr. Tobias Schwarz and Chavdar Chernev from the Royal School of Veterinary. They had also presented guidance in the project.

My parents are also the people I need to appreciate the most. Without them, I wouldn't be the person I am today. During my time studying abroad, they have always provided me with support and been my strongest backup.

Finally, I would like to thank my roommates, Tianlang Tan, Zhengxin Qian and Enshuo Zhang. They brought me happiness when I was down and provided me with moral support.

Table of Contents

Chapter 1

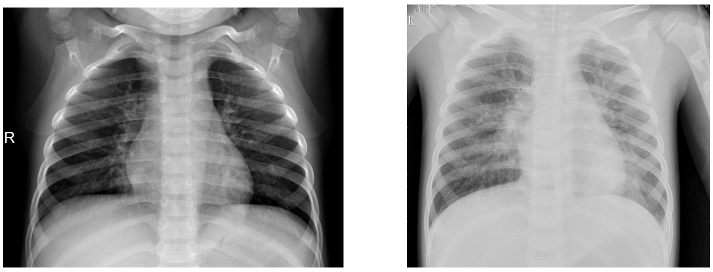
Introduction

1.1 Motivation

Over the past decades, artificial intelligence (AI) has developed rapidly with the combined influence of hardware and software. It is no longer just capable of executing a fixed program based on pre-arranged instructions and parameters, but can learn and automatically adjust its parameters to better perform the task. It could imitate human intelligence to perform tasks and iteratively improve itself based on the information it gathers. Artificial intelligence is now used in a broad range of fields, such as computer vision [49], natural language processing(NLP) [35], recommendation systems [12] and so on, and is used to assist people in making better decisions due to its promising performance.

Meanwhile, with the development of medical technology in the past decades, the technology of medical imaging has also been enhanced. The earliest medical images were X-ray, then ultrasound imaging and with the advent of computer technology, X-ray Computerized Tomography (CT) and Magnetic Resonance Imaging (MRI) have emerged. Although X-ray was first invented and used in the medical industry, it is still widely used today due to its low cost, high penetration and low radiation dose. The X-ray technique uses the fact that X-rays are highly penetrating and are absorbed differently when they pass through different tissues, so the amount of X-rays reaching the film varies, creating black and white contrast pictures. Nowadays, it is used by physicians to detect and assist in the diagnosis of clinical conditions such as orthopaedic, pulmonary, breast and cardiovascular diseases.

Chest radiology is currently the most commonly used radiological method for diagnosing diseases. Depending on the angle of capture, it could be divided into frontal and lateral views. The frontal chest X-rays could be used to show the blood vessels in the lungs, the shape and contours of the heart, or the bones in the chest. And the lateral ones are employed for more detailed observation of the heart condition. Pneumonia, as an infection of the lungs, is usually diagnosed by chest X-ray. A distinct chest X-ray with pneumonia will look fainter than normal and have large grey areas (Fig. 3.1). But in some cases, images of pneumonia do not have such distinctive features and make diagnosis difficult. What's worse, this disease causes the loss of a large number of lives each year. So it is important to develop an efficient approach to detect it.



(a) Health (b) Pneumonia

Figure 1.1: Heath and Pneumonia images

Since artificial intelligence can be utilized in the computer vision area, it becomes possible to apply this technique in chest radiology analysis. We hope that it will help physicians to better distinguish pneumonia so that patients can be detected and treated in a timely manner.

1.2 Problem Statement

The economy and ease of use have made chest X-rays the diagnostic choice of most physicians, but it also creates a tremendous workload for them. In 2006, there were approximately 128 million chest films generated in the United States alone [30] and the workload of radiologists in the same year amounted to 14,900 procedures, which had increased by 3% in the last three years and continued to grow [5]. A World Health Organization's report stated that pneumonia was responsible for 15% of deaths among infants under five years of age [34]. In 2018, 1.5 million people in the United States were diagnosed with pneumonia in emergency departments and 44,000 lives were lost as a result [16]. Therefore it is important to provide an effective diagnosis of pneumonia so that the pneumonia could be detected as early as possible and also ease the pressure on radiologists.

1.3 Aims and Objectives

This project aims to apply artificial intelligence methods to provide a pneumonia diagnosis system, which could classify healthy ones and those with pneumonia in chest X-rays. This kind of work could be called Computer-Aided Diagnosis (CAD). The research hypothesis could be listed as follow: (1) The system could accelerate the process of pneumonia diagnosis; (2) It could reduce the cost of diagnosis; (3) The accuracy of diagnosis could achieve human radiologists level.

The objectives of this project are: (1) Provide different artificial intelligence models that are capable of performing the classification task on chest X-rays dataset; (2) Innovatively utilise the Transformer model for medical image classification rather than traditional convolutional neural networks; (3) Attempt to use different methods to improve the performance of the models; (4) Analysis their performance and discuss how to extend the current work in the future.

1.4 Achieved Result

In this project, we not only utilized three common convolutional neural networks (the VGG19, ResNet50 and Inception-ResNet-v2) but also applied a novel Transformer model (the Swin Transformer) to the medical classification task. To the best of our knowledge, this is a novel study that the Swin Transformer was trained in a small medical dataset and achieved radiologist-level accuracy. We innovatively compared the Swin Transformer with other models with different hyperparameters. Although the VGG19 model performed best under certain conditions, the potential capacity of the Swin Transformer was the best. We also compared the common data augmentation methods (rotation, shifting and flipping) with the contrast enhancement method and concluded that the contrast enhancement was more suitable for processing X-rays images.

1.5 Dissertation Outline

This dissertation will be divided into seven chapters. Chapter 2 will present the related work about the most popular network architecture (Convolutional Neural Network) and the emerging architecture (Transformer). It also introduces the concept of transfer learning and related medical image classification projects. The data information will be introduced in Chapter 3 and the related augmentation will also be presented. In Chapter 4, the methodologies used in this project will be displayed, such as the models, visualisation methods and evaluation metrics. After that, the experiment design and the results will be exhibited in Chapter 5. Chapter 6 would be used for discussing and analysing the results. Finally, the conclusion and future work will be discussed in Chapter 7.

Chapter 2

Related Work

2.1 Convolutional Neural Network

Convolutional neural network (CNN) is a popular architecture in the deep learning area for recent years and performs well in many machine learning tasks like classification, detection or segmentation. The common elements that comprise a CNN are convolutional layers, max pooling layers and fully connected layers, which could be displayed via the configuration of the AlexNet [25] (Fig. 2.1). The convolutional layers are used to extract features from the input data, which is a matrix of the same size as the input image. It uses a filter with k kernels and a receptive field (represents the region in the input space that will affect feature extraction) size of nxn to scan the input in a certain stride and multiple the corresponding input space with each kernel on the filter to get a new matrix, which is the extracted feature. Then the max pooling layers follow for downsampling to reduce the feature size as successive convolutional layers can increase the data size. Finally, the fully connected layers make a final classification of the results based on the features obtained earlier.

The concept of CNN was introduced as early as 1998 by Yann LeCun [27]. But due to a lack of successful practice, this excellent concept did not really have a profound impact on the field of deep learning until the advent of AlexNet. AlexNet was created by Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton together in 2012 [25]. This model won the 2012 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge), which was one of the top events in the field of computer vision and represents the cutting edge of deep learning in the field of imaging. Its success brought CNN into the public eye and since then CNN has gained tremendous momentum.

A series of excellent CNN models after that have been proposed and have achieved

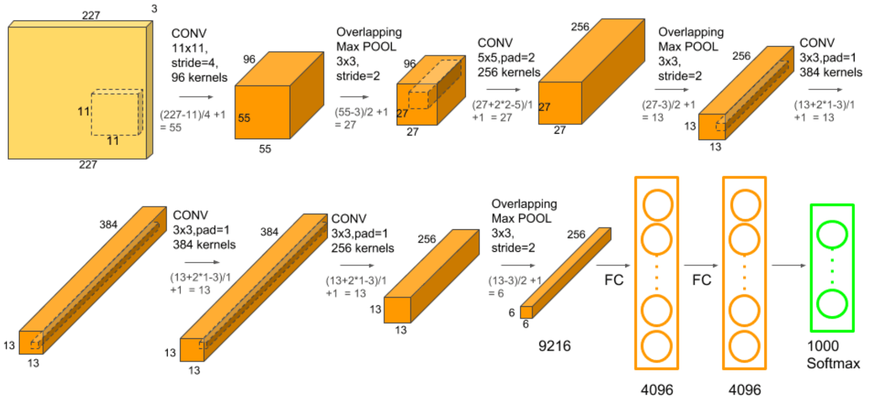


Figure 2.1: AlexNet network configurations

excellent results in this competition. For example, VGG is a convolutional neural network introduced by the group called Visual Geometry Group from Oxford University [43]. It achieved first and second place respectively in the ILSVRC localization and classification tasks in 2014 by virtue of its network depth. Kaiming He and his team solved the problem of model degradation caused by excessive depth increase and proposed the ResNet [19]. This model had an extraordinary 152-layer network but still maintained exceptional performance and won the 2015 ILSVRC. Another well-known convolutional neural network is GoogLeNet, which is also known as Inception [46]. It applied a method called Inception Module to fuse the image features obtained from different kernels to combine them to obtain a better result. This modification increased the width of the model and also helped it win the 2014 ILSVRC. Inception model also absorbed the advantages of VGG, ResNet's model and evolved InceptionResNet, which made it converge faster and easier to train.

2.2 Attention and Transformer

In addition to the traditional CNN models for image tasks, the Transformer model also had a profound impact on deep learning in the last few years. A Transformer model is a novel network architecture that discards the structure of the traditional CNN or RNN[1] model and instead uses the attention mechanism for learning [48]. So before introducing the Transformer, it is necessary to have a basic understanding of attention.

[1] The Recurrent Neural Network can be used to process temporal data, passing the output of the previous moment as an input to the next moment

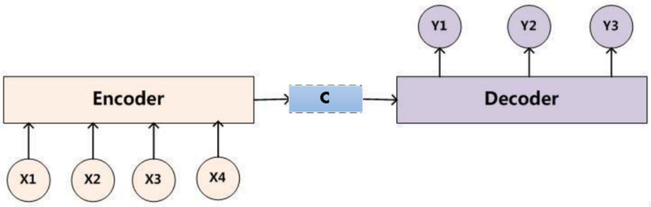


Figure 2.2: Traditional Encoder-Decoder

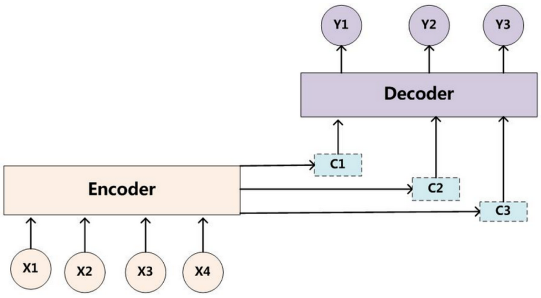


Figure 2.3: Encoder-Decoder with Attention

Attention was first used for the task of machine translation. In the traditional machine translation task, the RNN Encoder-Decoder structure (Fig. 2.2) was popular as it could handle the common problem of unequal lengths of input and output. The encoder is responsible for encoding the variable-length input sequence into a fixed-length vector, while the decoder is used to decode the vector into a variable-length output sequence [7] and this structure is also applied by a Transformer. However, the experiments have shown that the performance of this approach deteriorates dramatically with increasing sentence length. This is because with longer sentences, the information tends to be lost as the gradient vanishes during transmission. It is also very difficult to generalize all the semantic details of a long sentence with a fixed length vector. So Bahdanau et al. [4] proposed a mechanism called attention to address this bottleneck. The most important modification was that the intermediate vector would no longer be encoded as a fixed-length. It allows the decoder to review the entire words or segments of the input sentence depending on what is currently being processed, and then generate a new vector ($C1$, $C2$ and $C3$ in Fig. 2.3) for the current output ($Y1$, $Y2$ and $Y3$ in Fig. 2.3). Each context vector is used to indicate the correlation (or weights) between one element in the output sequence and all elements in the input sequence and a high correlation indicates that two elements have a strong contextual connection. These various context vectors could help the model pay attention to high correlation elements and obtain more accurate semantic information and thus improve the accuracy of the

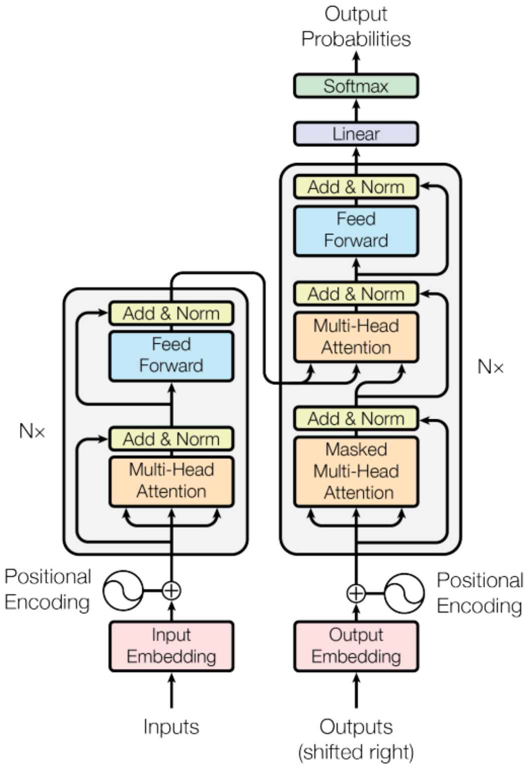


Figure 2.4: The Transformer Encoder-Decoder [48]

translation.

A Transformer is a novel network architecture that discards the structure of the traditional CNN or RNN model and instead uses the attention mechanism for learning [48]. It was first applied to the field of machine translation and then later extended to the field of vision (Vision Transformer). Self-attention is the key component of a Transformer. In self-attention, each input element would be embedded and this embedded vector would produce three vectors, which are query, key and value. By comparing the similarity of one query with other keys, the result (as the weight) is multiplied with the corresponding value. The sum of all multiplied values is the final output. Unlike the aforementioned attention in the RNN, which is used to establish a link between input and output, self-attention is used to apply attention within the input; hence the term self-attention. In Fig. 2.4, there are two similar layers with '*N*x' (*N* means this layer could be applied *N* times) next to them. The left part is the encoder and the right part is the decoder. The input of the encoder would be first embedded and converted to vectors, then a positional encoding would be applied to let the transformer know the position of each input so that it has temporal awareness like an RNN. Then the embedding vectors would be converted to queries, keys and values and transmitted to the

Multi-Head Attention. The Multi-Head Attention is composed of many self-attention layers, similar in function to the channel in CNN, that projects the inputs to lower dimensions to learn various features for better performance. Layer normalization is applied [3] to distribute values normally and improve the training efficiency. The new values would be delivered to the feed-forward layer, which is an activation function that is used to strengthen the representation of the data. From Fig. 2.4, it is noticeable that the arrows skip some layers. This is due to the use of ResNet's shortcut to address the degradation problem. The difference between the encoder and decoder is that the decoder used masked Multi-Head Attention to hide future outputs so that the training and predictions are consistent. And the decoder would take the encoder's keys and values with its own queries as an input of attention for further processing. This kind of transformer architecture reduces computational complexity and enables parallel computing. It also provides a new architecture for other machine learning tasks [48].

The Transformer, in addition to being able to compete with the state-of-the-art RNN models for NLP tasks, has been explored as to whether it can be compared to CNN models in the field of computer vision. By replacing the convolutional layer in ResNet with self-attention, Ramachandran et al. [39] found that the new model is comparable to the baseline in ImageNet classification or COCO detection tasks, thus demonstrating that the attention layer can be stand-alone in computer vision tasks. Cordonnier et al. [9] further explored the relationship between attention and the convolutional layer and proved that the attention layer also performs convolution. However, text and images have two different dimensions of data, where text is one-dimensional and image is two-dimensional. Therefore it is important to pre-process the image input to transform it into a NLP-compliant input vector. There are several methods to achieve that. An Image Generative Pre-trained Transformer (iGPT) [6] would resize the input image to a low resolution and then reshape it into a one-dimensional sequence. It then finishes processing this input through a similar approach to that used in NLP tasks. Cordonnier et al. [9] and Dosovitskiy et al. [14] provided an alternative approach. This is to split the input image into a fixed-size patch and embed each patch and also add positional embedding. Nevertheless, both approaches could only handle small images because the computation complexity of their global self-attention is quadratic with image size. Also such resizing inevitably results in a loss of information, whereas the Swin Transformer [28] was created to handle larger images in general databases by using Shifted Window Attention, such as 224x224 in ImageNet [13].

2.3 Transfer Learning and Fine-Tuning

Transfer learning is the approach that applies knowledge learned in a previous domain or task to a different but related area to avoid starting from scratch. This concept was proposed since it was thought that the model would only perform well if the training data and the test data had the same feature space and distribution. In practice, however, there are often cases where the distribution or feature space has changed due to a lack of data or outdated data, and it is very expensive to recollect data and train a model. Therefore, a technique was needed that could transfer knowledge to solve this problem [37]. Not all situations are suitable for transfer learning, considering the different feature spaces or distributions, transfer learning is avoided in domains that are significantly disparate, as this would result in a negative transfer and affect the training of the model. In Pan et al.'s survey [37], they summarized different transfer learning methods. The first one is instance-based transfer learning, which is to select the data in the source domain that can be reused in the target domain [15] [21]. A second case is feature-based transfer which is concerned with identifying common feature representations between the source and target domains, and then using these features for knowledge transfer [11]. The last one is referred to as parameter-based transfer and it is about sharing the model parameters or prior knowledge of the source task and the target task [17].

With the development of deep learning, the most widespread approach to model training today is to pre-train a model and then fine-tune it. Pre-training enables the model to obtain initial parameters for the target task that have some effect and thus gain some performance improvement in the target task training [32]. And fine-tuning is to modify the pre-trained parameters by using data from the target domain to make the model more suitable for the task. Yosinski et al. found out that the layers close to the input layer do not change dramatically as the dataset changes by fine-turning or freezing part of the layers [52]. And Kumar et al. further discovered that the inappropriate use of fine-tuning resulted in distorted pre-trained feature extraction and degraded model performance [26]. Therefore, the performance of the model can only be improved with appropriate fine-tuning or freezing on the pre-trained model.

2.4 Medical Image Classification

In recent years, the success of deep learning in the field of computer vision has led researchers to turn their attention to the medical field. And medical image classification has become an important task in computer vision as it can assist physicians in the analysis of those chest radiography images. In 1995, Lo et al. constructed a simple convolutional neural network to determine whether a chest X-ray image containing lung nodules or not and at that time it took 15 seconds to evaluate each radiography image [29]. Their work validated the feasibility of artificial intelligence for radiography diagnosis. Considering different pathology has different features, Avni et al. applied the Bag-of-Visual-Words (BoVW) [10] model to extract keypoints of features to classify healthy and pathology cases [1]. Rajpurkar et al. provided a 121-layer convolutional neural network called CheXNet based on ResNet [38]. This model was trained on a dataset of more than 100,000 frontal-view chest X-ray images with 14 different diseases and it outperformed an average physician with state-of-the-art performance. In the work of Ayan et al, they compared the performance of different CNN models in a Pneumonia X-ray classification task [2]. They trained a Xception model [8] (a variant of Inception [46]) model and a VGG16 [43] with transfer learning and noticed that the VGG model achieved a slightly higher accuracy than the Xception.

Covid-19, a highly contagious and dangerous disease that could also infect lungs, has caught the attention of researchers since its widespread in 2019 and a lot of work has been done on detecting it. Ozturk et al. proposed a CNN model with 17 convolutional layers for early detection of Covid-19 and it reached 98% accuracy on binary classification and 87% on multiple classification [36]. Hemdan et al. presented a model called COVIDX-Net which contains several different model architectures such as VGG and GoogLeNet [20]. Due to a shortage of Covid-19 X-ray images early in the pandemic, this model was only validated on 50 X-ray images with 25 positive Covid-19 cases and achieved 90% accuracy which was very impressive. In the work of Sethy et al., they applied the ResNet50 model to extract features from Covid-19 X-ray images and then used a Support Vector Machine (SVM) for classification and this combination achieved 95% accuracy [42].

Chapter 3

Dataset

3.1 Dataset Introduction

In this project, the dataset used is a public chest X-ray dataset from Kaggle [33], which contains 5,863 frontal chest X-ray images (JPEG format) and is divided into two categories (Normal/Pneumonia). However, in this dataset, there is an unbalanced distribution of data, with roughly three times as many pneumonia images as normal images. So the original dataset was refined by randomly selecting images and building a balanced dataset (training data [70%] was used to fit the model with the learnt hyperparameters; validation data[20%] was used to evaluate the model and tune the hyperparameters; test data [10%] was used to evaluate the final model performance) as follows:

	Normal	Pneumonia
Train	1151	1526
Validation	288	306
Test	144	153

Table 3.1: Data Distribution

3.2 Data Augmentation

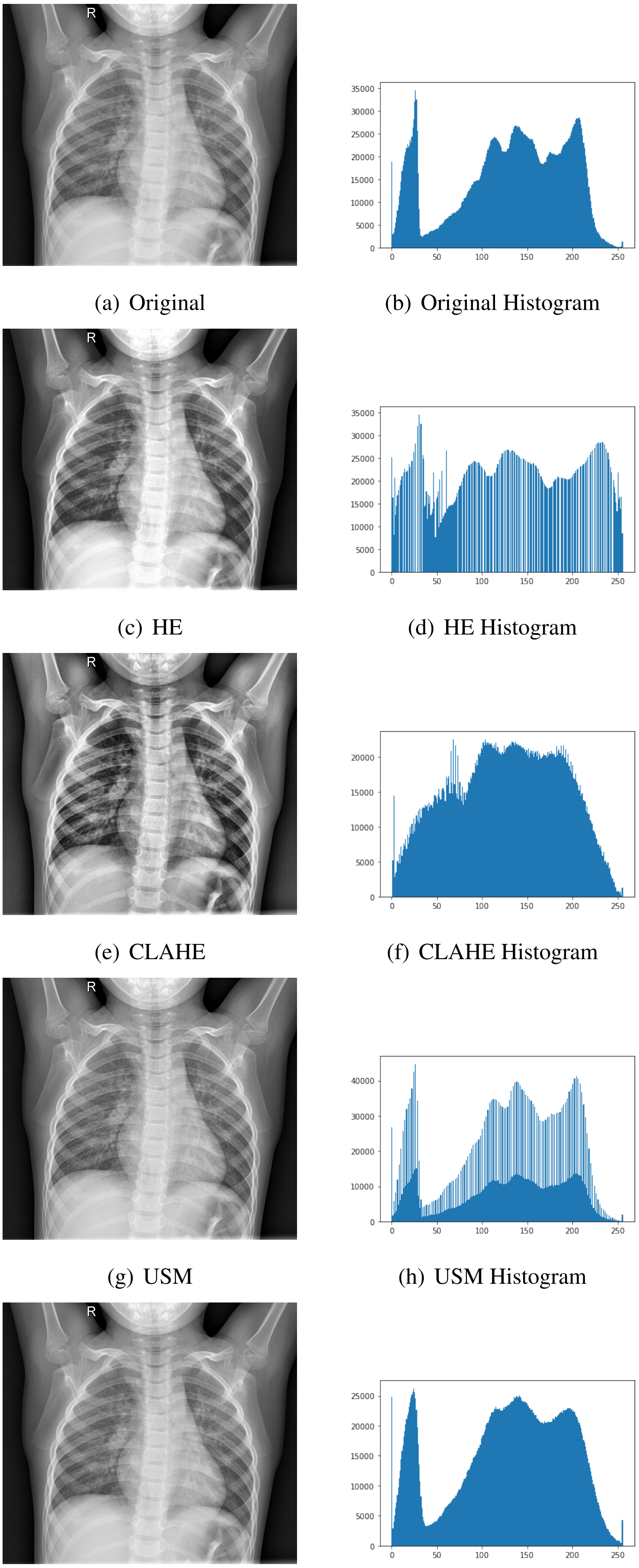
The learning of deep models in the field of computer vision usually requires powerful computing resources and large amounts of data as a model usually has millions or even billions of parameters to be trained [22]. However, in the medical field, collecting enough data is a challenge, so increasing the amount of data through data augmentation is required. In addition to the common methods, such as rotation, cropping, flipping and shifting used, considering X-ray images which are grayscale with different tissues having various grayscales, we also applied methods to enhance the images such as contrast and edge sharpening [40]. In this project, there are four methods used, the first two for contrast enhancement and the last two for edge sharpening.

The first one is Histogram Equalization (HE)(Fig. 3.1(c)), which spreads out the most frequent intensity values to expand the intensity range according to a probability distribution. The algorithm first finds the frequency of each pixel value in a grayscale image. Then it calculates the cumulative frequency of each pixel value. Finally, the cumulative frequency is divided by the overall number of pixels and multiplied by the maximum number of greyscales in the image. This has the advantage of being effective in enhancing images with a uniform distribution of grayscales.

The second method is Contrastive Limited Adaptive Equalization (CLAHE)(Fig. 3.1(e)), which divides the whole image into several tiles and applies the HE method to each tile. Then, if a histogram bin is above a contrast threshold, those pixels would be clipped and distributed evenly to other bins before applying the HE method. This method enhances the local contrast of the image and reduces the interference of noise.

The third one is Unsharpen Mask sharpening (UMS)(Fig. 3.1(g)) which is used to enhance the contrast of neighbour pixels. It first applies a Gaussian blur on the original image, then subtracts the Gaussian image from the original one. Finally it restores the pixel value to the normal range (0-255). This can remove some minor details of interference and noise and enhance the edge in the image.

The final method is Laplace sharpening(Fig. 3.1(i)). It is rotationally invariant and can handle the sharpening of images in different directions by using the Laplace operator. This is also the only difference from the UMS method.



(a) Original (b) Original Histogram

(c) HE (d) HE Histogram

(e) CLAHE (f) CLAHE Histogram

(g) USM (h) USM Histogram

(i) Laplace (j) Laplace Histogram

Figure 3.1: Original and augmented images [23]

Chapter 4

Methodology

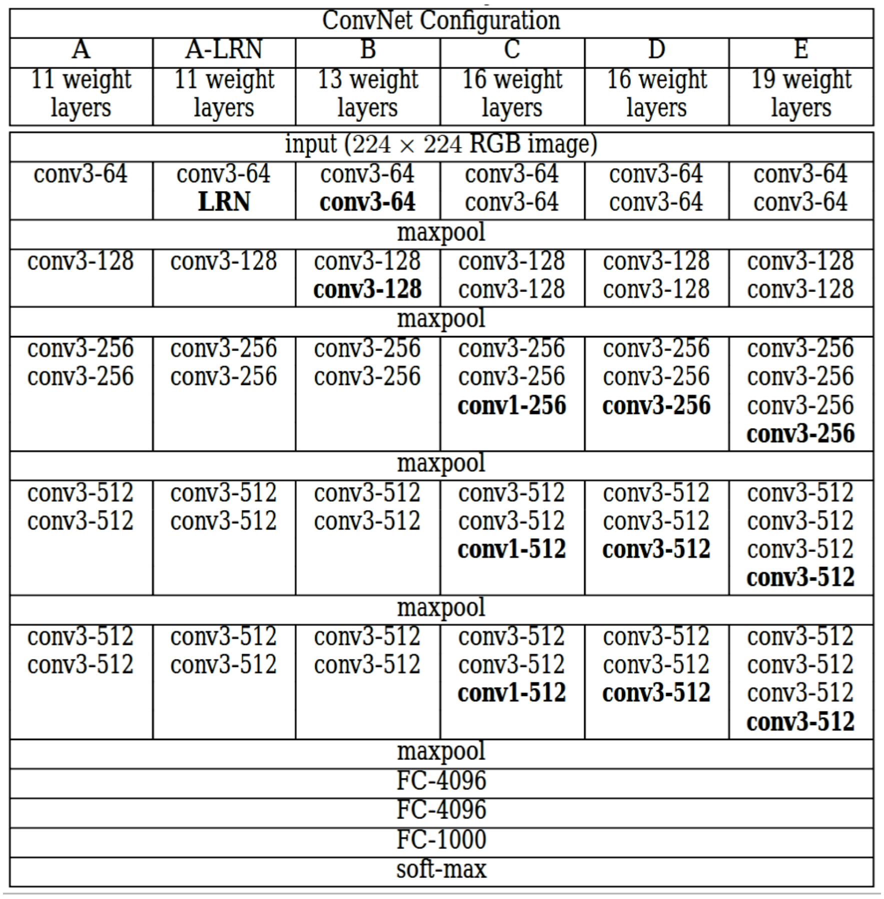
4.1 Classification Models

4.1.1 VGG19

The VGG19 model has 19 layers and the rightmost E configuration from Fig. 4.1 illustrates its architecture. Its main layers are convolutional layers with a kernel size of 3, which means it applies a 3x3 kernel to filter the input and extract the features. Before this, previous models could not perform well after 10 layers and their kernel sizes were relatively large. For example, the sizes of the kernels in AlexNet are 11x11 or 7x7 [25]. However, it is not the case that the larger the kernel size, the better the model. In [43], the VGG teams replaced a 5x5 kernel with two 3x3 kernels while keeping the receptive field size the same. This modification not only increased the depth, but also reduced the number of parameters to be learned as 2x3x3 is less than 5x5. As a result, this architecture allowed the model to learn more complex patterns with deeper layers, while keeping more efficient computations with fewer parameters. After each section of convolutional layers, a max-pooling layer was applied to sample those extracted features compressing them into a reduced dimension to speed up the computation. At the end, three fully connection (FC) layers were added, which were used to project the extracted features to the corresponding label space for classification. The final softmax layer was used to distribute the probabilities of all labels for eventual classification.

4.1.2 ResNet

The ResNet model was introduced by Kaiming He and his team to address the problem of network degradation caused by increasing depth [19]. They noticed that a well-trained



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 4.1: Six VGG network configurations with different layers [43]. The depth of increases from left to right (from 11 to 19 layers), and the parameters of each layer is displayed as conv$\langle kernel\ size \rangle - \langle number\ of\ channels \rangle$.

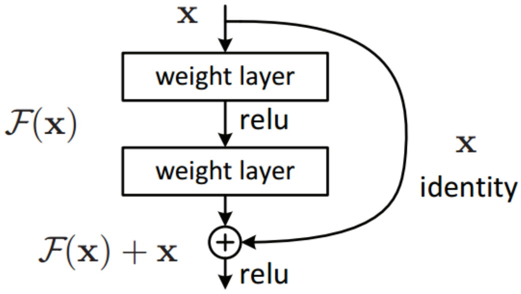

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 4.2: Residual block [19]

shallow model could not be improved by stacking more identity mapping (underlying mapping) layers and these added layers could not perform identity mapping so that instead the performance was reduced. A residual mapping was provided by them to replace the original underlying mapping and address the degradation problem. The core concept is that in a residual block (Fig. 4.2), the layers do not learn the complex underlying mapping ($H(x)$ in eq. (4.1)) directly, but learn the comparatively simple residual mapping ($F(x)$ in eq. (4.1), where x is the input). This shortcut connection that skips one or more layers can still perform identify mapping and gain better performance without introducing extra parameters or increasing the computational complexity. In this project, ResNet50 was applied and the dimensions of each layer differed with depth. When the input and output dimensions are different in one residual block, a 1x1 convolutional layer would be applied for the linear projection (W_s in eq. (4.2)) or extra zeros are be added to the increased dimensions. If this identity shortcut is applied to feature maps of two sizes, it is common to use with a stride of 2 to match the two sizes.

$$H(x) = F(x) + x \tag{4.1}$$

$$H(x) = F(x) + W_s x \tag{4.2}$$

4.1.3 Inception-ResNet

The Inception-ResNet model is a combination of the ResNet and Inception models. In the ResNet model, the residual block would handle the degradation problem caused by increasing the depth, but the impact of increased width had also attracted the attention of researchers. Initially, Szegedy et al. [46] noticed that any uniform increment of the number of two chained convolutional layers' filters would result in a quadratic increase in the amount of computation resource. Also with a larger model it was easier to overfit when using limited data and this caused computation inefficiency. To solve such bottlenecks, they provided a filter-level sparse structure, codenamed the Inception module. The first version of the Inception module is displayed in Fig. 4.3. It executes a convolution operation on the input by using 3 filters of different sizes (1x1, 3x3, 5x5) and furthermore performs a maximum pooling. This combination of sub-layers can parallelise the processing of the input and increase computation efficiency. Several 1x1 convolutional layers are added before the 3x3 and 5x5 convolution operations and after max polling is utilized to reduce the number of channels of input and thus the

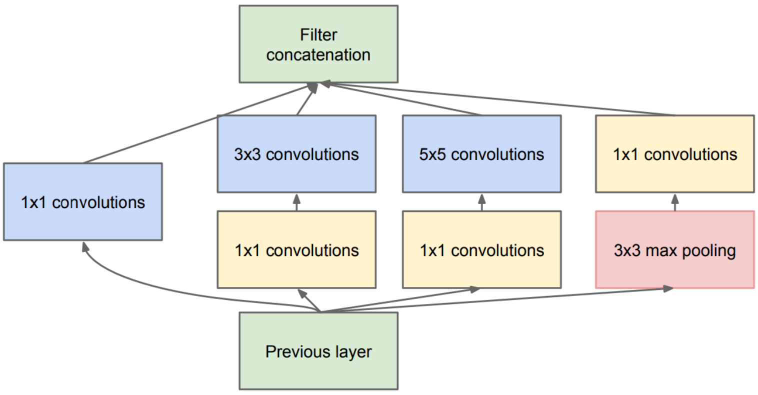
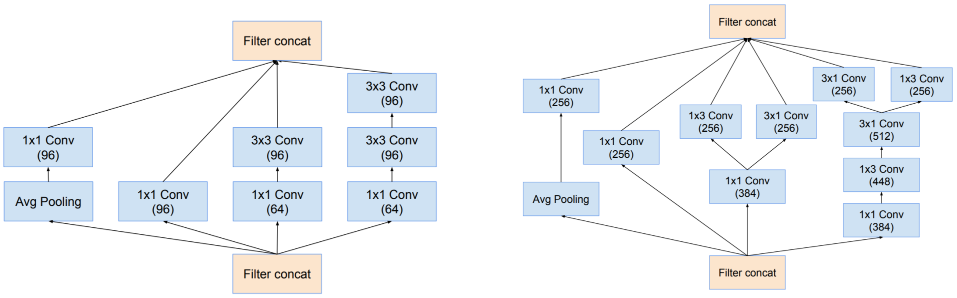


Figure 4.3: Inception v1 module [46]

computational cost of operations. The outputs of all sub-layers are finally cascaded and passed to the next Inception module. In Szeged et al.'s further work [47], they modified the Inception module to make it more efficient, without leading to a loss of expressiveness, by applying convolution factorization, which is the same as replacing one large convolutional layer with two smaller ones as mentioned in the VGG part (Fig. 4.4). For example, they proposed that the original 5x5 convolutional layer was replaced by two 3x3 convolutional layers (Fig. 4.4(a)) and a nxn convolutional layer could also be replaced by a 1xn convolution followed by a nx1 convolutional layer (Fig. 4.4(b)). This parallel structure, with asymmetric convolutional kernels, allows for a reduction in computational effort while ensuring that information loss is sufficiently small. The 1*1 convolution kernel in the structure is also used for dimensionality reduction and increases the nonlinearity.

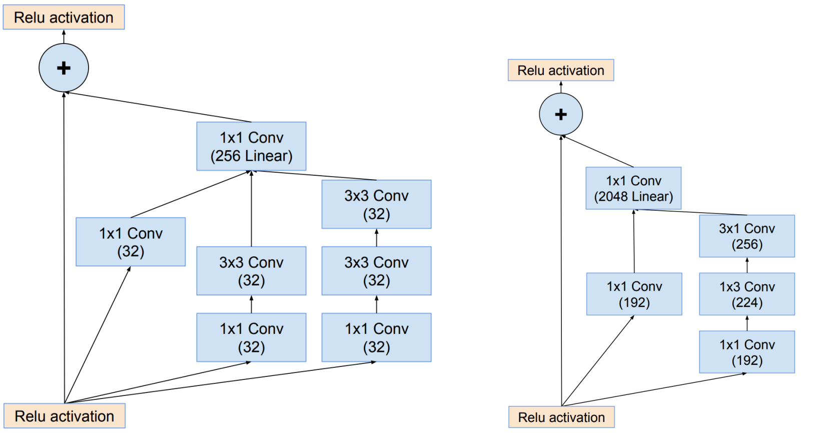


(a) Inception example 1 [45]

(b) Inception example 2 [45]

Figure 4.4: Inception Module examples

Also in their work, Szeged et al. [45] combined a residual block with the Inception module and got an architecture where each block has an identity shortcut connection that connects directly to the plus sign from the previous layer's activation result, and the right parts of Fig. 4.5(a) and Fig. 4.5(b) remains the same structure as the Inception module.



(a) Inception-ResNet example 1 [45]

(b) Inception-ResNet example 2 [45]

Figure 4.5: Inception-ResNet Module examples

4.1.4 Swin Transformer

The Swin Transformer model is a new vision Transformer provided by Liu et al. and is considered as a general-purpose backbone for computer vision [28]. Its crucial aspect introduces a method called Shift Window Attention that applies self-attention in each small window so that large-scale images could be processed. It first divides the image equally into non-overlapping windows with *mxm* patches (Layer l in Fig. 4.6) and then applies multi-head self-attention in each window to reduce the computation cost. In order to create connections between neighbour patches in different windows, the windows would be shifted and then the image would be divided into new windows again

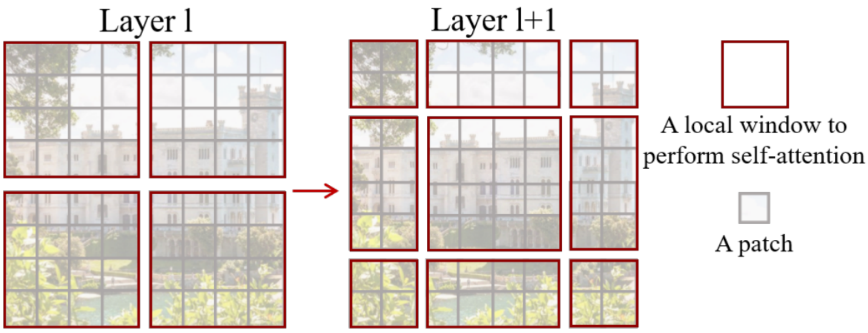


Figure 4.6: Shifted Window approach [28]

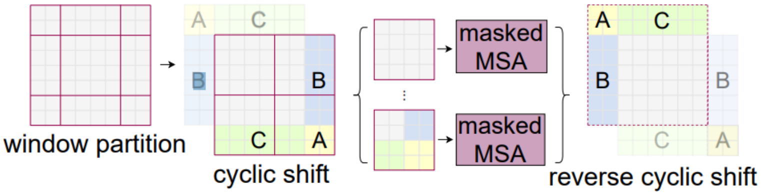


Figure 4.7: Cyclic shift [28]

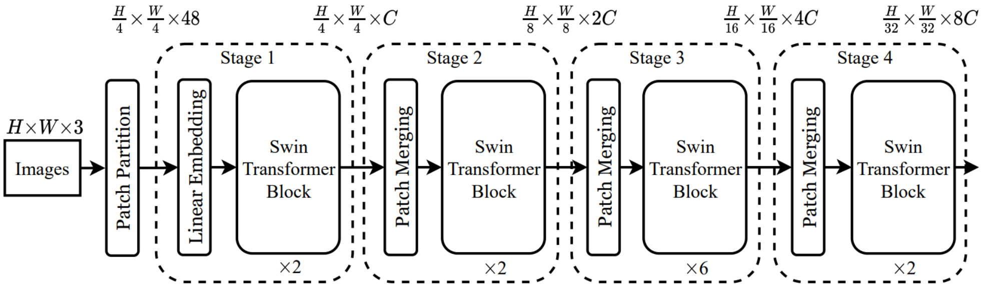


Figure 4.8: Swin Transformer architecture where H, W and C are the height, width of the input and an arbitrary dimension [28]

(Layer l+1 in Fig. 4.6) so that patches among the new windows could do Multi-Head Self-Attention cross the previous boundary. However, the number of windows after shifting is changed and the numbers of patches in windows differ. A cyclic shift method is utilized to move the smaller windows to form a normal-sized window, and mask the unrelated (not adjacent in the original image) areas and then perform the self-attention calculation (Fig. 4.7).

The Swin Transformer's architecture is displayed in Fig. 4.8. First, the input image is partitioned into several 4x4 patches and thus the feature dimension is 4x4x3 = 48. Then a linear embedding is applied to convert patches into fixed-sized (C) vectors and the result is passed to a Swin Transformer Block. In the block, layer normalization is applied and Shifted Window attention is used with an Unmasked Multi-Head Self-Attention and a Masked Multi-Head Self-Attention. Patch merging is used to merge adjacent patches for down-sampling and to increase the receptive field while acquiring multi-scale features. As the rate of down-sampling is 2, the output's height and width would be half of the input (e.g. H to H/2 and W to W/2) and the number of channels in the C dimension would be 4 (a HxW token would be divided into 4 H/2xW/2 tokens) times of the original (e.g. C to 4C). In order to match VGG and ResNet, the number of channels in the C dimension should be twice the input and a linear mapping is used to convert 4C to 2C. This hierarchical design allows the model to gain features in different scales. After several patch merging and transformer blocks, the output is processed by global average pooling to classify the result.

4.2 Visualization Method

Gradient-based Class Activation Mapping (Grad-CAM) is utilized to visualize the behaviour and determination of neural networks and can provide an explanation of the model [41]. The procedure of generating Grad-CAM is to first get the input image and the desired category, then forward propagate the input image through the CNN model and obtain rectified convolutional feature maps (A, usually the output of the last convolutional layer output after ReLU) and the prediction score of all categories (before softmax). Then the score of the specific category in the prediction would be backpropagated to the feature maps and the gradient information would be produced. The gradient is global-average-pooled to obtain the importance of each feature map. Then these weights are multiplied by the feature maps obtained and summed, and finally the Grad-CAM is obtained by the ReLU activation function.

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{4.3}$$

$$L_{Grad-CAM}^c = ReLU(\sum_k \alpha_k^c A^k) \tag{4.4}$$

The weights for each channel would be generated through eq. (4.3). The specific category prediction score (y^c) would be extracted from the prediction result (y) according to the category (c). And it would be backpropagated to the feature map (A) at channel k (A^k). The gradient of y^c and A^k would be calculated for each position in the feature map (ij is the position of data at a height of i and width of j) and its result would be applying a global-average-pooling that divided by Z (the result of multiplying the height and the width of the feature map). As a result, the weight (α_k^c) for channel k is obtained. And eq. (4.4) is by Grad-CAM to visualize the area where the classification is made. Each feature map (A^k) and its corresponding weight (α_k^c) are multiplied and accumulated in all the products. The result would be applied a ReLU activation function to retain only the information that is useful for this category.

4.3 Evaluation Methods

In the image classification tasks, the predicted result is classified into four categories: true positive (TP), true negative (NP), false positive (FP) and false negative (FN). Here true or false means if the model predicts the category correctly or not, and positive or negative represents the predicting category. For example, in a cat and dog classification problem, if predicting a dog is the positive class, TP stands for the correct prediction of a dog and FN represents the model mistaking a dog for a cat. These values would be used to perform the following operations to evaluate the model's performance.

- Accuracy = (TP+TN)/(TP+TN+FP+FN): This is the ratio of the number of correctly classified to the whole number of predicted data. A higher score means that the model's classification performance is better.
- Error: In this project, the Cross Entropy Loss was applied to measure the error value, which represents the classification error in the tasks. If this error value is smaller, the better the performance of the model.
- F1-score = 2(Recall * Precision) / (Recall + Precision): Precision (= TP/(TP+FP)), this represents the percentage of samples with a positive prediction that are truly positive and Recall (= TP/(TP+FN)) is the amount of positive cases in the sample that were correctly predicted. The F1-score combines two indicators to avoid extreme situations and a higher value represents better performance.

Chapter 5

Experiment and Results

5.1 Experiment Environment

In this project, VGG19, ResNet50, Inception-ResNet-v2 (IR) and Swin Transformer (SW) were used in this pneumonia classification task. All of them are implemented by using Tensorflow 2.9 with its Keras library[1]. The code was written in Python. All experiments were run on an NVIDIA GeForce RTX 3070 GPU with CUDA 11.3.

5.2 Experiment Settings

In the experiments, the optimizer used was Adam [24], which helps the model to converge better. To measure the difference (or error) between the predicted and true values, the Cross Entropy Loss function was used consistently. Early stopping was also applied to avoid overfitting (this only performs well on the training set but poorly on the test set) so that if the value of the error did not decrease in the next 5 epochs, the model would stop training and save the best weights. We explored the effects of the learning rate, the number of epochs, transfer learning and data augmentation. Since it is stated in the article [28] that the Swin Transformer requires a large dataset for training and would be difficult to be trained by one individual, in our subsequent experiments it was used pre-trained.

Learning rate is one of the hyperparameters, where hyperparameters are the parameters used to configure the process of model training. It refers to the step size in each iteration that moves in the direction of the minimum of the loss function. If the learning rate is too large, it might miss the minimum loss, while if it is too small it will

[1] Tensorflow is an open source machine learning platform and Keras is a high-level API of Tensorflow

take a long time to reach the optimal result. In this experiment, different learning rates were compared (1e-3, 1e-4 and 1e-5). Another hyperparameter we fine-tuned was the number of epochs, which represents the number of times the model has been trained in the entire training data. During the training process, as the epoch grows, the training results of the model will go from underfitting (the model performs poorly on the training set), to fitting (performs well on both training and testing sets), and if the number of epochs is too large, it will produce overfitting results. We tried different numbers of epochs (20, 50) without applying early stopping to visualise the most realistic changes in the performance of the models. We also applied transfer learning (see Section 2.3) in our experiments so we could initialize the model with pre-trained parameters and hope that a good initialization of the model would improve the performance. The corresponding pre-trained weights are included in the Keras library (VGG19, ResNet50 and Inception-ResNet-v2) or downloaded from the official Github repository (Swin Transformer [31]) and they are trained on non-medical images. According to the aforementioned research, fine-tuning should be utilized appropriately and layers closer to the output are mainly used to learn the characteristics of the current data. A certain percentage (0%, 50% and 100%, where in 0% situation, the last fully-connected layer would be trainable for classification) of the layers from back to front were trainable to study the variation in the performance of the same model while other layers were frozen and untrainable. The improvement of using different data augmentation methods was also explored. Some data augmentation approaches that are common are applying a rotation (rotation_range from 0 to 10 degrees), horizontal flipping and shifting (both height_shift_range and width_shift_range from 0 to 0.2). The other data augmentation methods used are mentioned in Section 3.2 and use contrast enhancement (CE) methods. These include Histogram Equalization (HE), Contrastive Limited Adaptive Equalization (CLAHE), Unsharpen Mask sharpening (USM) and Laplace sharpening.

In the experiments, the results are measured by accuracy, error and F1-score values on the test set. For the analysis of the final model, we used the Grad-CAM heatmap images except for the Swin Transformer[2].

[2] The Grad-CAM heatmap is applied on the convolutional layers but the Swin Transformer does not have convolutional layers

5.3 Results of different Learning Rates for models

The effect of different learning rates is explored and displayed in Tab. 5.1. The number of epochs was set to 20 and no data augmentation methods were applied. As the training of the Swin Transformer requires a large dataset (14M-300M images), it is difficult for an individual to train a model from scratch. Therefore, the Swin Transformer model was trained using the weights obtained from pre-training with all parameters trainable, while others were trained from scratch. We can see that decrement in the learning rate could improve the performance of the model and the VGG19 achieved the best performance among all models.

Model	Learning Rate	Test Accuracy	Test Error	F1-score
VGG19	1E-3	0.515	0.701	0
VGG19	1E-4	0.925	0.172	0.905
VGG19	**1E-5**	**0.946**	**0.134**	**0.950**
ResNet50	1E-3	0.851	0.449	0.862
ResNet50	1E-4	0.888	0.446	0.777
ResNet50	1E-5	0.929	0.226	0.928
IR	1E-3	0.932	0.162	0.934
IR	1E-4	0.919	0.266	0.910
IR	1E-5	0.932	0.171	0.916
SW	1E-3	0.515	0.704	0
SW	1E-4	0.925	0.165	0.935
SW	1E-5	0.936	0.139	0.921

Table 5.1: Learning rate experiments. The number of epochs was set to 20 and no pre-trained (except for SW) or data augmentation methods were applied (IR means the Inception-ResNet-v2 and SW means the Swin Transformer)

5.4 Results of larger Epochs for models

According to the previous experiment, the learning rate was set to be 1E-5 so we could achieve the best performance for each model. In this part, the effect of the larger number

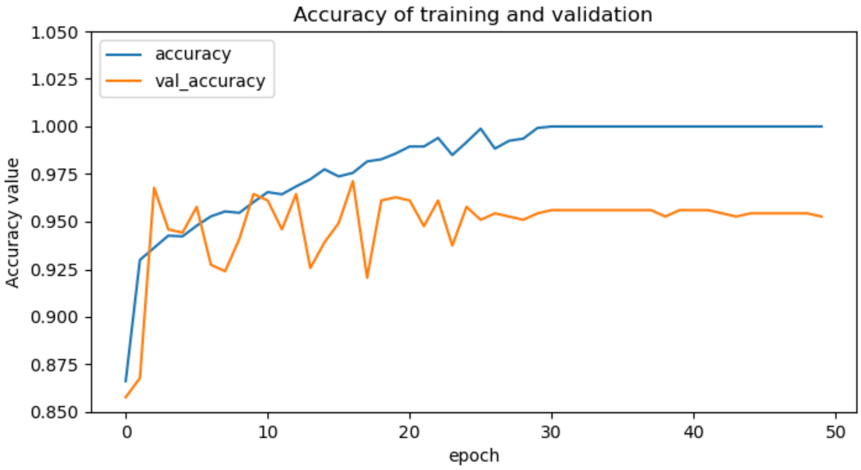


Figure 5.1: VGG19

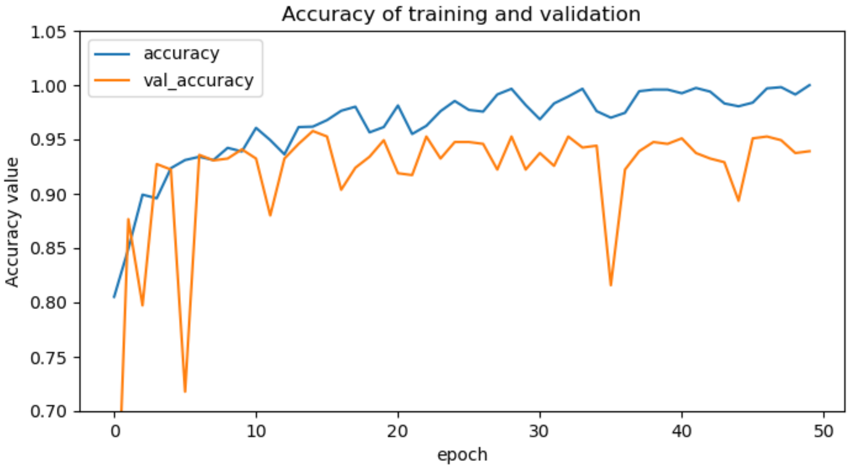


Figure 5.2: ResNet50

of epochs was explored, so early stopping was not applied. The Swin Transformer was still equipped with pre-trained weights while others were not. The result was shown in Tab. 5.2 without data augmentation.

5.5 Results of different Trainable Percentage for models

In this experiment, we experimented with the effects of transfer learning with different trainable percentages (0%, 50% and 100%, where the trainable percentage of 0% means only the last fully-connected layer was trainable). The result was displayed in Tab. 5.3.

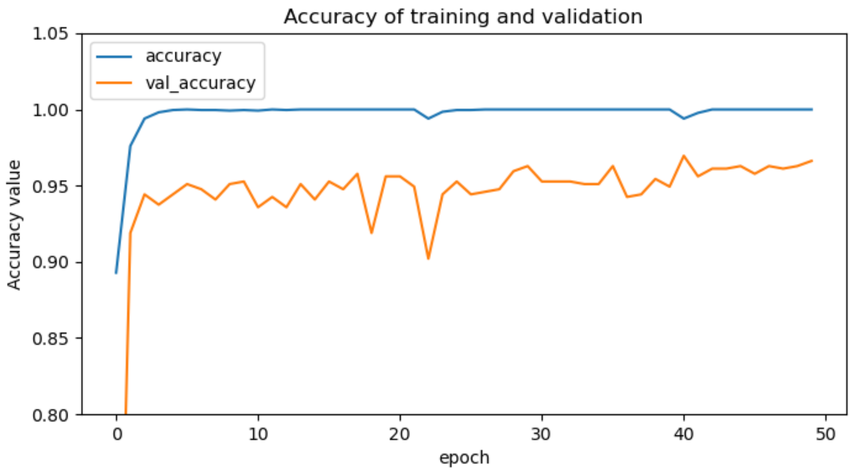


Figure 5.3: Inception-ResNet-v2

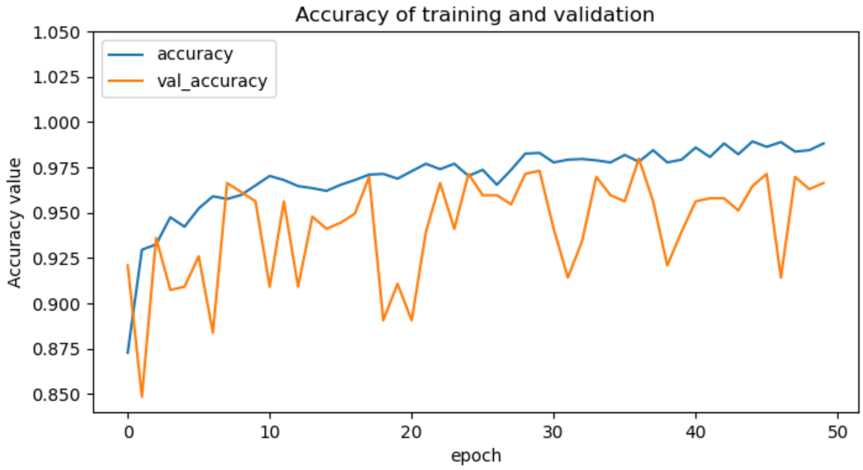


Figure 5.4: Swin Transformer

Model	Epochs	Test Accuracy	Test Error	F1-score
VGG19	20	0.946	0.134	0.950
VGG19	50	0.929	0.455	0.924
ResNet50	20	0.929	0.226	0.928
ResNet50	50	0.932	0.291	0.928
IR	20	0.932	0.171	0.916
IR	50	0.952	0.172	0.920
SW	20	0.936	0.139	0.921
SW	**50**	**0.952**	**0.115**	**0.950**

Table 5.2: Epochs Experiments. The learning rate was set to 1e-5 and no pre-trained (except for SW) or data augmentation methods were applied (IR means the Inception-ResNet-v2 and SW means the Swin Transformer)

5.6 Results of different Data Augmentations for models

Normal data augmentation (flipping, shifting and rotation) and contrast enhanced data augmentation (HE, CLAHE, USM and Laplace sharpening) were applied to explore the effect of augmentation methods. And in this part, we utilized the early stopping and transfer learning with a 100% trainable percentage.

5.7 Grad-CAM results of final models

This section used Grad-CAM heatmaps to indicate the area of interest of the model on the image, with a redder colour indicating that this area is getting greater attention. Since Grad-CAM was applied on the convolutional layers whereas the Swin Transformer does not have these layers, we only displayed the other three models' Grad-CAM heatmaps.

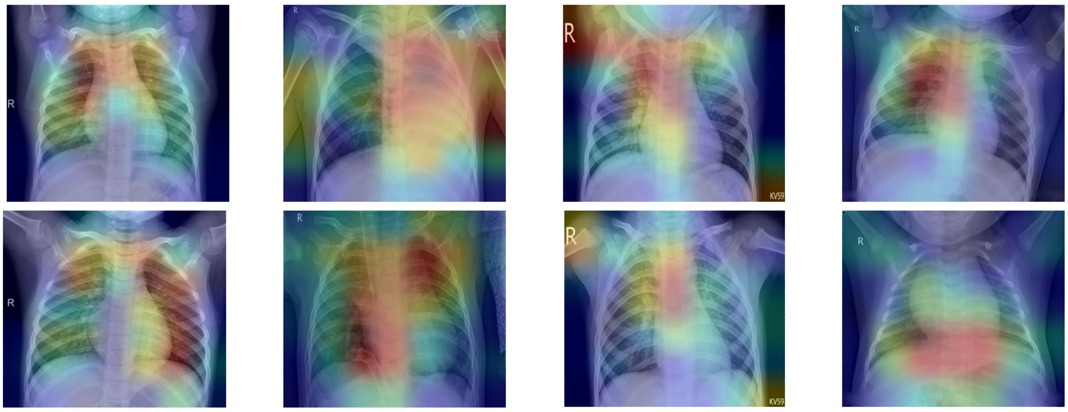


Figure 5.5: The VGG19 Grad-CAM heatmap (0.983 accuracy with the contrast enhancement). Four columns from left to right are TP (2 normal images predicted successfully), TN (2 pneumonia images predicted successfully), FP (2 normal images but considered to be pneumonia) and FN (2 pneumonia images but considered to be normal)

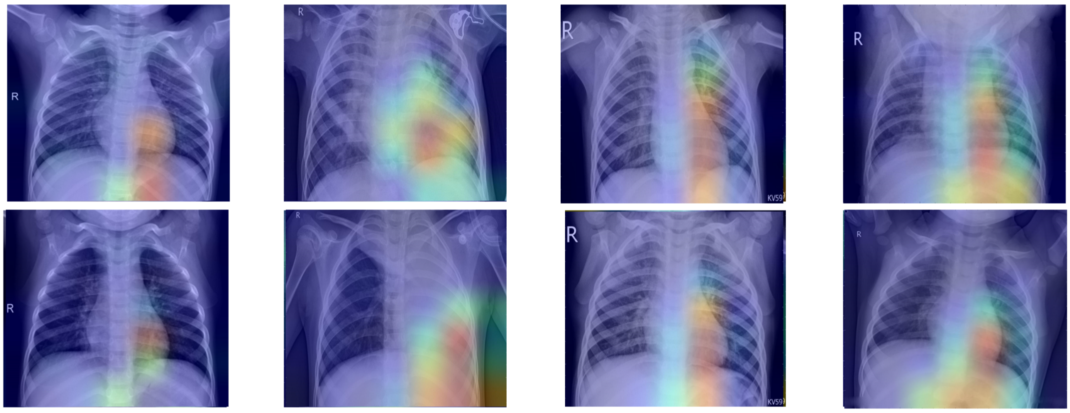


Figure 5.6: The ResNet50 Grad-CAM heatmap (0.976 accuracy with the contrast enhancement). Four columns from left to right are TP (2 normal images predicted successfully), TN (2 pneumonia images predicted successfully), FP (2 normal images but judged to be pneumonia) and FN (2 pneumonia images but considered to be normal)

Model	Trainable Percentage	Test Accuracy	Test Error	F1-score
VGG19	0%	0.956	0.098	0.930
VGG19	**50%**	**0.976**	**0.087**	**0.957**
VGG19	100%	0.973	0.080	0.982
ResNet50	0%	0.942	0.296	0.917
ResNet50	50%	0.945	0.176	0.920
ResNet50	100%	0.969	0.093	0.960
IR	0%	0.973	0.078	0.934
IR	50%	0.956	0.094	0.949
IR	100%	0.942	0.424	0.940
SW	0%	0.949	0.131	0.948
SW	50%	0.937	0.135	0.932
SW	100%	0.936	0.139	0.921

Table 5.3: Transfer Learning Experiments. The learning rate was set to 1e-5 and the number of epochs was 50 with early stopping. No data augmentation methods were applied (IR means the Inception-ResNet-v2 and SW means the Swin Transformer)

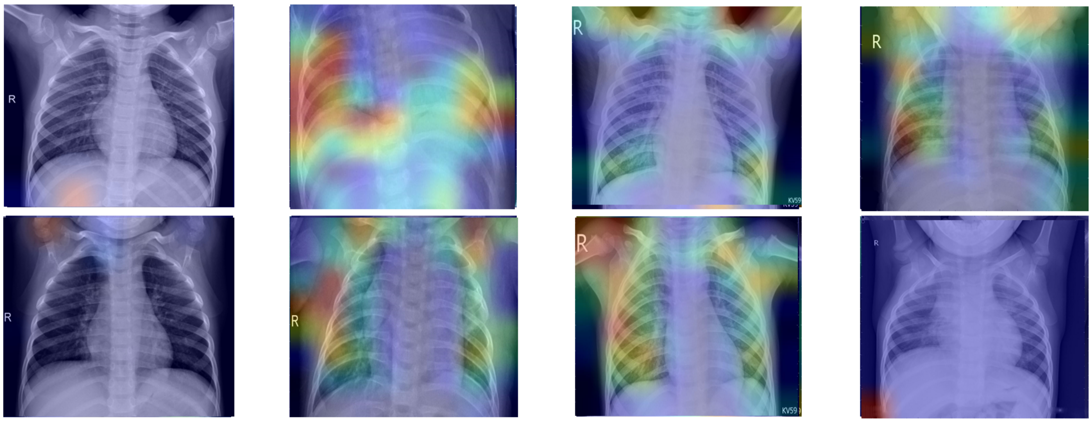


Figure 5.7: The Inception-ResNet-v2 Grad-CAM heatmap (0.956 accuracy with the contrast enhancement). Four columns from left to right are TP (2 normal images predicted successfully), TN (2 pneumonia images predicted successfully), FP (2 normal images but considered to be pneumonia) and FN (2 pneumonia images but considered to be normal)

Model	Method	Test Accuracy	Test Error	F1-score
VGG19	No	0.973	0.080	0.982
VGG19	Normal	0.983	0.059	0.971
VGG19	**CE**	**0.983**	**0.085**	**0.972**
ResNet50	No	0.969	0.093	0.960
ResNet50	Normal	0.932	0.285	0.628
ResNet50	CE	0.976	0.088	0.934
IR	No	0.942	0.424	0.940
IR	Normal	0.949	0.136	0.942
IR	CE	0.956	0.269	0.946
SW	No	0.936	0.139	0.921
SW	Normal	0.959	0.134	0.957
SW	CE	0.959	0.116	0.958

Table 5.4: Data Augmentation Experiments. The learning rate was set to 1e-5 and the number of epochs was 50 with early stopping and all layers were trainable (IR means the Inception-ResNet-v2 and SW means the Swin Transformer, and CE means the contrast enhancement).

Chapter 6

Discussion and Analysis

From the above results, we can see the importance of hyperparameters to the model. From Tab. 5.1, the VGG19 and Swin Transformer models with a learning rate of 1E-3 performed poorly and their F1-scores were zeros. The reason was that both of them could only consider the data as pneumonia and were not able to correctly classify the normal images, resulting in a zero number of correctly identified normal images. There were two possible reasons for this, one was that the learning rate is too large and the model was unable to learn the correct information, so it tended to miss the minimum loss and thus failed to converge. The other was that the features of the normal images were not as distinct as those of the pneumonia ones, so it was not as easy to distinguish them as the pneumonia images. However, when the learning rate decreased, most of the models improved, as evidenced by increased accuracy and reduced errors on the test set and higher F1-score scores. This indicates that they were able to correctly identify normal and pneumonia photos as well as possible.

Tab. 5.2 illustrates that the larger number of epochs could provide better performance as it could allow for a larger number of training sessions and thus the opportunity for better convergence. However, the VGG19 showed the opposite behaviour. From Fig. 5.1, it was evident that the performance of VGG19 on the validation set was not improving and even tended to decrease. This might be caused by the fact that the model was overfitting and the performance was decreasing. The validation accuracy (Fig. 5.2, 5.3 and 5.4) of the other three models was still increasing with the Inception-ResNet-v2 and Swin Transformer being the most obvious. The reason could be that the Inception-ResNet-v2 had the largest amount of parameters, about twice as many as the other models and the Swin Transformer was a transformer rather than a convolutional neural network, which itself requires a huge dataset to be trained in order to have a significant improvement, so by increasing epochs a similar effect can be achieved.

From the transfer learning experiments (Tab. 5.3), we noticed that performance of VGG19 and ResNet50 increased with the growing trainable percentage and most models improved after applying transfer learning (compared with Tab. 5.2). However, for the Inception-ResNet-v2 and Swin Transformer this was the reverse. The reasons for this might also be related to the structure of the model and the source task of transfer learning. Since the VGG19 and ResNet50 have relatively simpler architecture among the four models, so fewer parameters needed to be fine-tuned. They could achieve better performance with limited data and the pre-trained weights, which were not even trained in the medical radiology area. But for the Inception-ResNet-v2 and Swin Transformer, even if all parameters were trainable, it could be hard to get improved performance with the limited data and would even degrade the model.

Tab. 5.4 demonstrates the effect of the data augmentation and both methods could improve the performance. Although the VGG19 with contrast enhancement could achieve the best performance, the Swin Transformer with contrast enhancement received the largest improvement (2.3% improvement over using no augmentation). Comparing the two different methods of data augmentation, it can be observed that contrast enhancement could achieve more improvement than the traditional method (rotation, flipping or shifting). This might be due to the fact that in medical diagnosis, such images would be captured from a similar perspective. Therefore, if the image was rotated too much or shifted too much, this might result in inaccurate data and affect the model performance. However, contrast enhancement would not modify such information and it could highlight the contrast and features in the image. The Receiver Operating Characteristic (ROC) curves of each model trained with the contrast enhancement were generated, where the ROC was used to reflect both sensitivity and specificity of the model. If the Area Under the Curve (AUC) is larger (closer to 1), the better the model performance. All models obtained a 3-5% improvement in the AUC values and reached close to 0.98 after the use of the contrast enhancement. And the improvement of using normal methods would be small.

For a better understanding of the outcome of employing the convolutional neural networks, a Grad-CAM heatmap was applied. From the previous results, knew that the VGG19 had the best performance (the highest accuracy of 0.983). Its Grad-CAM heatmaps (Fig. 5.5) shows that it had a strong focus (red areas) on the lungs for correctly discriminated images, while for those incorrectly classified images, its attention moved away from the lungs to below or above them. Through further analysis of the results, we found it was able to distinguish the two categories almost correctly so that it obtained a high F1-score (0.972). However, for the ResNet50 and Inception-ResNet-v2, their attention was not as strong as the VGG19. Fig. 5.6 and Fig. 5.7 illustrates that their area of focus was small. The ResNet50 seemed to be focused on a smaller part of the lungs and was more aware of pneumonia and pneumonia-like areas. The Inception-ResNet-v2 gave less attention to the normal images and the red areas are not even on the lungs and for those misclassified images, its attention even went beyond the lungs to the shoulders. By analysing their classification results, we found both of them considered a lot of normal images as pneumonia ones while the amount of misclassified pneumonia (real pneumonia) was small. Therefore, such a bias caused the model to perform less well on the F1-score than the VGG19.

Although the Swin Transformer (0.959 accuracy with CE) could not be analysed with the Grad-CAM heatmap, we could obtain some important information by analysing its final classification results. We found that in the misclassified images, there was no bias as in the previous models (the ResNet50 and Inception-ResNet-v2). The two categories had almost equal proportions. From this, it was inferred that the Swin Transformer was more capable of learning adequate features for unbiased classification. And if sufficient data were available, it might outperform other models.

Chapter 7

Conclusions and Future works

7.1 Conclusions

In this project, we applied four different artificial intelligence methods for pneumonia diagnosis. In addition to the traditional CNN models (VGG19, ResNet50 and Inception-ResNet-v2), we also innovatively applied the latest Transformer model (Swin Transformer) and proved its feasibility in the medical imaging field. By adjusting the hyperparameters (learning rate, the number of epochs and trainable percentage in transfer learning), we investigated the differences between the models and found that large models (with many parameters or layers) or models with complex structures (the Transformer structure) require a much larger training volume than small models. We also proved that in transfer learning, using pre-trained weights from different source tasks (real-word images) could also improve the model performance on medical diagnosis. By comparing different data augmentation methods, we concluded that contrast enhancement of data augmentation was more appropriate for processing medical images such as X-rays. From the final models' performance[1], all of them had exceeded the radiographers (0.95 accuracy) [50] and it took only 16 milliseconds to analyse one image. So we proved that artificial intelligence could accelerate the diagnosis process and achieve the human radiologist level. Finally, the VGG19 model was considered to be the best model, but we inferred that the Swin Transformer might outperform them with sufficient data as it performed unbiased classification and had a more capacity to learn with a longer training process.

[1] The models trained with the contrast enhancement

7.2 Future works

In this project, the normal data augmentation could alleviate the problem caused by the lack of medical images for training methods, but the need for large amounts of data to train well-performing models remains high. So in future work, we might apply artificial intelligence models to generate more medical synthetic images. The Generative Adversarial Networks (GAN [18]) have been applied to generate some simple images for normal computer vision tasks [51], so we suggest that a good GAN model could be trained to generate more medical data for training with the help of well-trained physicians. Alternatively, the architecture of the Swin Transformer could be enhanced to reduce the model's need for huge data while maintaining good performance.

In the project, we only processed 2D X-rays, but a lot of 3D CT images have also been used in diagnosis, and researchers have applied CNNs to process these tasks [44], but no Transformers have been used. We postulate that 3D information could also be embedded into a Transformer in a suitable format for processing so that the Swin Transformer could be trained on 3D medical data in the future.

Bibliography

[1] Uri Avni, Hayit Greenspan, and Jacob Goldberger. X-ray categorization and spatial localization of chest pathologies. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 199–206. Springer, 2011.

[2] Enes Ayan and Halil Murat Ünver. Diagnosis of pneumonia from chest X-ray images using deep learning. In *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, pages 1–5. Ieee, 2019.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[5] Mythreyi Bhargavan, Adam H Kaye, Howard P Forman, and Jonathan H Sunshine. Workload of radiologists in united states in 2006–2007 and trends since 1991–1992. *Radiology*, 252(2):458–467, 2009.

[6] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.

[7] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[8] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[9] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*, 2019.

[10] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[11] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 210–219, 2007.

[12] Aminu Da'u and Naomie Salim. Recommendation system based on deep learning methods: a systematic review and new directions. *Artificial Intelligence Review*, 53(4):2709–2748, 2020.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Wei Fan, Ian Davidson, Bianca Zadrozny, and Philip S Yu. An improved categorization of classifier's sensitivity on sample selection bias. In *Fifth IEEE international conference on data mining (ICDM'05)*, pages 4–pp. IEEE, 2005.

[16] National Center for Immunization and Respiratory Diseases. Pneumonia can be prevented—vaccines can help. `https://www.cdc.gov/pneumonia/prevention.html`. Accessed July 29, 2022.

[17] Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 283–291, 2008.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Ezz El-Din Hemdan, Marwa A Shouman, and Mohamed Esmail Karar. Covidx-net: A framework of deep learning classifiers to diagnose COVID-19 in X-ray images. *arXiv preprint arXiv:2003.11055*, 2020.

[21] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19, 2006.

[22] Zeshan Hussain, Francisco Gimenez, Darvin Yi, and Daniel Rubin. Differential data augmentation techniques for medical imaging classification tasks. In *AMIA annual symposium proceedings*, volume 2017, page 979. American Medical Informatics Association, 2017.

[23] Daniel Kermany, Kang Zhang, Michael Goldbaum, et al. Labeled optical coherence tomography (oct) and chest X-ray images for classification. *Mendeley data*, 2(2), 2018.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[26] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[29] S-CB Lo, S-LA Lou, Jyh-Shyan Lin, Matthew T Freedman, Minze V Chien, and Seong Ki Mun. Artificial convolution neural network techniques and applications for lung nodule detection. *IEEE transactions on medical imaging*, 14(4):711–718, 1995.

[30] Fred A Mettler Jr, Mythreyi Bhargavan, Keith Faulkner, Debbie B Gilley, Joel E Gray, Geoffrey S Ibbott, Jill A Lipoti, Mahadevappa Mahesh, John L McCrohan, Michael G Stabin, et al. Radiologic and nuclear medicine studies in the united states and worldwide: frequency, radiation dose, and comparison with other radiation sources—1950–2007. *Radiology*, 253(2):520–531, 2009.

[31] Microsoft. Swin transformer pretrained weight. `https://github.com/microsoft/Swin-Transformer`. Accessed July 12, 2022.

[32] Dmytro Mishkin and Jiri Matas. All you need is a good init. *arXiv preprint arXiv:1511.06422*, 2015.

[33] PAUL MOONEY. Chest X-ray images (pneumonia). `https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia`. Accessed June 20, 2022.

[34] World Health Organization. Pneumonia. `https://www.who.int/zh/news-room/fact-sheets/detail/pneumonia`. Accessed July 29, 2022.

[35] Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2):604–624, 2020.

[36] Tulin Ozturk, Muhammed Talo, Eylul Azra Yildirim, Ulas Baran Baloglu, Ozal Yildirim, and U Rajendra Acharya. Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Computers in biology and medicine*, 121:103792, 2020.

[37] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[38] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest X-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[39] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. *Advances in Neural Information Processing Systems*, 32, 2019.

[40] SK Savitha and NC Naveen. Algorithm for pre-processing chest-X-ray using multi-level enhancement operation. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 2182–2186. IEEE, 2016.

[41] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[42] Prabira Kumar Sethy and Santi Kumari Behera. Detection of coronavirus disease (COVID-19) based on deep features. 2020.

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[44] Satya P Singh, Lipo Wang, Sukrit Gupta, Haveesh Goli, Parasuraman Padmanabhan, and Balázs Gulyás. 3D deep learning on medical images: a review. *Sensors*, 20(18):5097, 2020.

[45] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[46] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[47] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[49] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[50] Nick Woznitza, Keith Piper, Stephen Burke, and Graham Bothamley. Chest x-ray interpretation by radiographers is not inferior to radiologists: a multireader, multicase comparison using jafroc (jack-knife alternative free-response receiver operating characteristics) analysis. *Academic Radiology*, 25(12):1556–1563, 2018.

[51] Jianwei Yang, Anitha Kannan, Dhruv Batra, and Devi Parikh. Lr-gan: Layered recursive generative adversarial networks for image generation. *arXiv preprint arXiv:1703.01560*, 2017.

[52] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.