DIABETIC RETINOPATHY DETECTION

Final Report

Information Technology Capstone Project

COMP5703

Group Members

1. Jingjing Wu (480509446)
2. Boya Liu (460285007)
3. Yilin Wang (490142480)
4. Janet Hong (460243708)
5. Ziang Zhang (480563057)

ABSTRACT

In computer science, artificial intelligence (AI) has been gathering lots of attention in the digital era. With the growing popularity in AI, one of the most unignorable applications is AI technology in the medical industry. It uses complex machine learning algorithms to detect diseases, diagnose, and propose a treatment, which is quicker, more efficient and with a lower cost.

Nowadays, more and more people focus on the topic of blindness from diabetic retinopathy, which is the most common complications of diabetes. To detect the level of diabetic retinopathy, it requires specialists to examine the images of the patient's retina, and check hard exudates, haemorrhages, abnormal growth of blood vessels, aneurysm and “cotton wool” spots. Being motivated by this, it will be beneficial if an automated detection system can be developed to diagnose the level of diabetic retinopathy. It is believed that this technology contributes to faster and more accurate diagnosis and the reduction of human errors. Hence, the aim of the project is to design an automated DR grading system and provide diagnosis accuracy which is consistent with the realistic clinical potentials.

Machine learning and deep learning algorithms are applied to construct the detection system. As CNN is specialized in capturing the details of image features, it is employed to establish a classification model. Retinal images are pre-processed at the first stage and then go into the neural network. The neural network can learn the features from input images. The output of the model is the level of each image which represents the patient’s current severity of DR. Once the model achieves a satisfactory accuracy, the system can be operated in the real-life medical industry of diabetic retinopathy detection.

After different experiments, this project finally achieved an accuracy of 91% with a speed of 0.06 second per image.

Table of Contents

1. Introduction

In the recent era, healthcare systems are facing a rigorous circumstance with an increasing need for clinical services, a rising burden of illness along with incremental health expenditures (Atun, 2015). However, it is difficult to solve the problem by just changing the current health system and achieve a balanced healthcare coverage by 2030 (Panch, Szolovits, & Atun, 2018). Whereas there is a demand, it comes the supply. Therefore, in order to remit the situation, artificial intelligence (AI) and robotics are rapidly moving into the field of medical care, especially for some key medical functions, including but not limited to diagnosis and medical treatment (Danks & LaRosa, 2018). It is invisibly changing medical practice. AI aims to mimic human cognitive functions by deriving knowledge from the dataset. It brings a paradigm shift to healthcare, supported by the increasing number of healthcare data and booming development of analytical tools (Jiang, Jiang & Zhi, 2011). To successfully build up an artificial intelligence system, it involves three steps, including hypothesis generation, hypothesis testing and putting into practice if succeed. While AI is the broad science that mimics human abilities, machine learning is a sub-discipline of AI, which trains a machine to learn from data. Deep learning is the next evolution of machine learning, which is based on models with fewer assumptions about the underlying data. In health-care fields especially for medical imaging, deep learning presents considerable promised for medical diagnostics.

As we noticed, there is a rapid rise in the global prevalence of diabetes. The World Health Organization estimates that 422 million people worldwide have diabetes (World Health Organization, 2018). One of the most common complications of diabetes is Diabetic Retinopathy (DR), which is the leading cause of blindness. The estimated prevalence rate of the disease was 28% in the United States and 18% in India (Zhang et al., 2010). Clinical practice guidelines for avoiding DR have been carried out worldwide. An effective screening program can identify the patients who requires close follow-up and prompt treatment, and who needs annual screening. Patients with Type 1 diabetes should take annual screening for DR at least in the first five years after the onset of the disease. For those who with Type 2 diabetes, an immediate examination should be conducted at the time of diagnosis, followed by yearly examinations (Solomon et al., 2017). However, there are significant differences between the current available guidelines in the methods for examination and experts included in screening and evaluation (Chakrabarti, Harper, & Keeffe, 2012). The experts and resources are limited in some countries where the prevalence of diabetes in local populations is high and DR detection and management is most needed.

The DR classification problem is highly time-consuming for clinicians because clinicians need to extract symptom features, compute weighting of numerous features, and check the location of those features. Artificial Intelligence are able to achieve much quicker classification after training. As given machine ability to learn, it contributes to diagnosis and aid clinicians to classify the DR level in real-life cases.

For the management of DR, this project constructed an automated and comprehensive DR detection model and grading system. Machine learning techniques are frequently applied to automated classification of DR, it implements classification tasks by given data, however, it focuses on feature extraction. In this case, we develop a model with Convolutional neural networks (CNN) architecture, which is a branch of deep learning. CNN algorithm is already widely used in digital imaging processing and vision. It has very good performance and learning of model is fast. In this study, the input data is a large set of high-resolution retina images with different levels of DR. After model automatically learning features, the output is the predicted DR level for each image. The purpose of this work is to use deep learning algorithm for DR diagnostics and assist clinicians to diagnose faster and more accurate. Hence, patients can receive efficient and timely treatment.

The paper is organized as follows. Firstly, the paper describes the related literature about this project. And then, the next section highlights the project objectives and problems, as well as the project scope. For Methodology section, it describes publicly available database that used for DR detection and mentions all methods involved with this project. And then it follows by the results and discussions, these two parts details the results during training and analysis the results, implementations and significance of the project contributions. Finally, the last section focuses on the limitations and future work for DR detection.

2. Related Literature

Diabetic Retinopathy (DR) is regarded as the most common cause of blindness in developing and developed countries for the last 50 years. It is an eye disease where diabetes affects the status of blood vessels in human retina, which is one of the main causes of vision impairment (Akram, 2013).

Diabetes mellitus (DM) is a chronic disease caused by impaired metabolism of glucose and insulin deficiency. It leads to hyperglycemia and vascular or neuropathic abnormalities. The prevalence of diabetes population is forecasted to increase from 2.8% to 4.4% globally in the coming of 2030. There are two types of DM where Type I is recognized as absolute insulin deficiency and Type II is judged to be insulin resistance and absolute insulin deficiency (Singh et al., 2008). Both types of DM have the potential of causing DR to some extent. Experiments have shown that almost all of type I DM patients and two-thirds of type II DM patients have symptoms of DR. Their shared symptoms are flashes, floaters, immediate loss of vision as well as blurred vision (Akram, 2013). The symptoms commonly happen after being diagnosed with DM at a rate of 50% in 10 years and 90% in 30 years. Moreover, since patients who suffered from Proliferative DR have higher risk exposure to heart attack, diabetic nephropathy and even death, it raises the public awareness of the early detection of DR (Fadzil et al., 2011).

The infection of DR has raised some retinal abnormalities including the followings:

I. Microaneurysms (MA): They are visible in the early stage of retinal damage. The formation of MA is usually caused by abnormal permeability or non-perfusion of retinal blood vessels (Williams et al., 2004). They look like red spots and has sharp margins (Early Treatment Diabetic Retinopathy Study Research Group [ETDRS],1991).

II. Hard exudates: The lipoproteins or other proteins leaking through retinal vessels (Singh et al., 2008). It looks like small white or yellowish-white spots or deposits with sharp margins (ETDRS,1991).

III. Haemorrhages: They can be small red dots or larger blot with an uneven density that appear within the deeper layer of retina (ETDRS,1991). They are caused by leakage of weak capillaries (Singh et al., 2008).

IV. Neovascularization (NV): It is the abnormal growth of retinal blood vessels on the inner surface (Patz, 1980).

V. Macular edema (ME): It results from leakage of fluid from broken blood-retinal barriers (Williams et al., 2004). It affects the central vision (Frank, 1984).

VI. Cotton Wool Spots (CWS): They occur when occlusion of arterioles happens. They appear as fluffy cotton-like lesions in the retinal nerve fibre layer (McLeod, 2005).

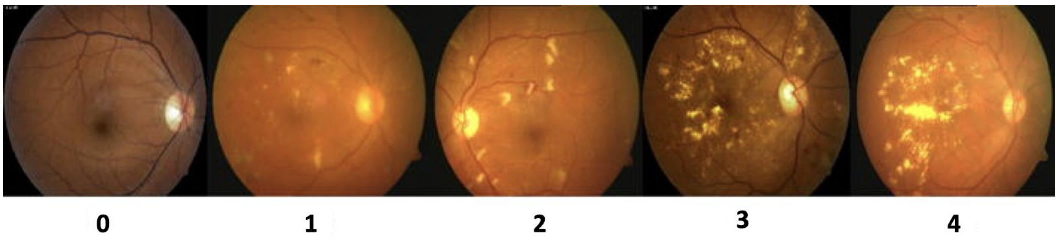


Figure 2.1: Retina Fundus images of (0) normal, (1) mild, (2) moderate, (3) severe, (4) PDR

Broadly, DR can be classified as nonproliferative diabetic retinopathy (NPDR) and proliferative diabetic retinopathy (PDR). The differences between different stages vary among the severity of abnormalities of the above said features in the fundus. A normal retina of the eye does not have any of the above abnormalities. In NPDR stage, it can be differentiated to mild, moderate and severe stage based on the above features except for less growth of new retinal blood vessels. In the PDR stage, the abnormalities of the above said features are more advanced with the growth of new blood vessels. The figure below illustrates the fundus image of different stages of DR where 0 is normal eyeball, 1 to 3 are three various stages of NPDR and 4 is PDR (Nayak et al., 2008).

Traditionally, the detection of DR relies on clinical examinations. It is fairly accurate since a professional ophthalmologist is able to detect most cases of DR through disc and macula photograph. However, the insufficiency of expert and expensive cost lead to the termination of such methods (Schachat et al., 1993). Therefore, the idea of automated detection is proposed. Since 1982, the quantification of DR and the detection of relevant abnormal features such as hard exudates and cotton wool spots on fundus images had been taken into research. The detection of individual features on the above fundus features based on relevant computer algorithms were developed (Nayak et al., 2008).

There are two different criteria to measure the performance of computational algorithms in the identification of DR, i.e., lesion-based and image-based. In the lesion-based criterion, every single lesion is regarded as an individual connected region and every retinal image is composed of numbers of lesion regions. By applying proper segmentation techniques to retinal images, a dataset of lesion regions is created. The lesion accuracy can be calculated by lesions specificity and sensitivity in comparison to the clinical results. However, it is not the number of exudes found that is vital to the diagnosis. The drawback of this criteria is that it may have a good statistical result but a poor diagnostic performance. Therefore, the image-based level is considered. Each image is regarded as a whole to decide whether the image has some evidence of DR. The system's accuracy is calculated based on the number of correctly identified images of the tested normal-abnormal images (Osareh et al., 2009).

In the initial stage, scientists came up with an algorithm used to recognize the optic disk, blood vessels and fovea. After all, it could only be applied to normal eye-ball images and not be used in the detection of large variations in the abnormalities of retinal images. Later, many studies have been put forward to develop computer-based detections, but none of them could provide accurate classification on various stages of DR (Nayak et al., 2008).

Currently, most algorithms existed for the detection of DR are based on segmentation techniques for each of the abnormal features mentioned above. Agurto and her colleagues come up with a new method which could help avoiding the difficulties encountered in the segmentation of retinal abnormal features. That is amplitude-modulation-frequency-modulation (AM-FM) methods to characterize retinal structures. A significant improvement for this method is its rigorous characterization of different retinal structures based on instantaneous frequency (IF) and instantaneous amplitude (IA). As is discussed, various retina is encoded with different AM-FM features. After applying this method, the accuracy of classification can achieve up to 92% (Carla et al., 2010).

In addition to the mentioned methods, a series of related studies have been carried out using neural networks. Artificial neural network (ANN) is thought to be efficient as the system applies mathematical weights to determine the probability of the given input belonging to an expected level. In ophthalmology, neural networks have been widely used in the diagnosis of visual field defects (Gardner et al., 1996). It is an information processing paradigm inspired by the biological nervous system. Image data are passed through the network, layer by layer, until reaching the outputs. Yet, there is no feedback between each layer. Therefore, it is also regarded as feedforward neural network (Nayak et al., 2018).

Still, ANN is not very suitable for image classification since it may easily lead to overfitting due to the size of the image. In the world of deep learning, Convolutional Neural Network (CNN) is considered as a more advanced method in image classification and object detection. The difference between CNN and ANN is the last layer of the neural network. The last layer of CNN is fully connected while in ANN, each neuron is connected to other hidden neurons in the network (Gogul & Kumar, 2017).

In the current stage, CNN is viewed as the optimal technique to use. CNNs are designed for processing data that comes in the form of multiple matrix or multiple arrays. Colour images can be converted into 3 layers of 2D arrays. CNNs are popular and developed rapidly for its four key ideas: local connections.

weights, pooling and the use of many layers (LeCun, Bengio, & Hinton, 2015). The first few layers of CNN are convolutional layers and pooling layers. Convolutional layers are composed of feature maps, in which each unit is connected to local patches of the previous layer through a set of weights. The result will then pass through a nonlinear function called activation function. A pooling layer in the network computes the maximum or minimum of a local patch in one or more feature maps. Feature maps, nonlinear function and pooling are connected and then followed by other convolutional layers or fully-connected layers. Convolutional layers are potential for extracting deep features for image classification and object detection (Chen et al., 2016).

Since CNNs are specialized in capturing the details of image features such as barely distinguishable lesions, our proposed approach is to build up a proper DR detection system using the principles of CNN models.

3. Project Problems

3.1. Project Aims & Objectives

In view of the increasing prevalence of Diabetic Retinopathy (DR), this project aims to develop an automated DR grading system based on the presence of lesions associated with the vascular abnormalities. If DR is detected in time, there are great opportunities to slow or avert the progression to vision impairment. However, patients with DR often show few symptoms until a very late stage. Without regular eye screening checks, patients are less likely to catch DR at a more treatable stage. Therefore, it is necessary for patients to have a regular eye screening check. Manually detecting DR is time-consuming for clinicians, and an accurate evaluation highly relies on the skilled experts and equipment, which are lacking in remote areas. Therefore, the automated DR grading system is designed to offer significant potential benefits to DR screening programs, including increasing efficiency, accessibility and affordability.

To achieve the aim, we collected retinal photographs dataset from Kaggle and built a classification model using deep learning and image classification knowledge. The images in this dataset were labelled, then the deep learning models were trained by using these large sets of labelled images and convolutional neural network (CNN) architectures that learned features directly from the data without the need for manual feature extraction. The CNN can extract features directly from the input images. The relevant features are learned while CNN trains on the large sets of images. This process increases the model behaviour of grading DR. With numbers of hidden layers, CNN can learn to detect different features of input images. The complexity of the learned input features is increased as the number of hidden layers increase. With digital colour fundus photography as input of our model, the goal of this project is to develop a model with realistic clinical potential.

3.2. Project Questions

There are some questions regarding the DR detection and fundus photography:

❑ By examining some sample images, we noticed that most patients have the same severity level of DR on both eyes or only one level difference between the two eyes. This brings the question of whether the lesions of one eye will affect another eye or not?

❑ As most patients with DR have no obvious symptoms in the early stages, can we predict the severity level of DR although it is asymptomatic?

❑ On ophthalmic fundus exam, “cotton wool” spots may appear as very small lesions, and some abnormalities may be difficult to capture. Is the difficulty of recognizing vascular abnormalities including “cotton wool” spots become an issue when detecting and evaluating DR?

❑ Under real-world screening conditions, the quality of retinal photographs varies considerably. This brings the question of whether the quality of images will affect the diagnosis accuracy of the automated detection system.

❑ Currently, most of the detection systems were developed using the colour fundus photographs from publicly available datasets. However, they do not take the patient’s characteristics into consideration. Will the characteristics such as gender or age impact on the detection behaviour for DR?

3.3. Project Scope

The primary goal of our project is to build a powerful neural network algorithm for detecting the DR by training digital colour fundus photos of the retina. This includes preprocessing input images to enhance important features, like lesions associated with the vascular abnormalities, as well as augmenting data to significantly increase the diversity of input data. The resulting model is able to identify and classify the input fundus photographs into five severity levels.

The model was designed to mimic the human brain, and this technology becomes more efficient in identifying diagnosis. This translates to faster and more accurate diagnoses. Meanwhile, it contributes to reducing human errors. As the DR detection technology is supposed to be broadly used even in remote areas, the diagnosis accuracy in model needs to be ideally consistent with the realistic clinical potentials. The expectation of this project is to achieve 90% accuracy when predicting the level of DR. At the same time, the diagnostic result should be provided in a reasonable time, that requires the model runs within a feasible time in the backstage. Therefore, the model is supposed to predict 100 results within 1 minute, As the dataset is unbalanced, accuracy and Cohen’s Kappa scores are used to evaluate the outcome performance. It will take around four months to complete this project.

4. METHODOLOGIES

4.1. Methods

For better understanding and stratifying the project into tasks, we had gone through the literature and journals of recent AI (Artificial Intelligence) application in the medical field, especially the studies on Diabetic Retinopathy Detection. We had implemented a work-breakdown structure (WBS) to divide our project into smaller components. It was perfect for us to organize the team's work into manageable sections. To track our progress during this semester, we have used the Gantt Chart and updated it every week to envision the consequences of certain delays and be prepared for changes. Every group member has managed to update their weekly individual progress report for self-reflecting and summarizing achievements.

For all algorithms we have done for this project, we have used Python as the coding language. Also, Tensorflow and Keras are the open-source API we used for building and evaluating models. They help us to build deep learning models more easily and effectively. Keras application has the most popular DL models that are made available alongside pre-trained weights. Built-in functions such as ImageDataGenerator by Keras makes our model read image data faster than inputting data from scratch. Furthermore, lots of these built-in functions can be customized if needed. For our project, if we want to increase our model complexity, it is quite easy for us to just add more layers after our determined pre-trained.

4.2. Data Collection

The Diabetic Retinopathy Detection data is collected from data modelling and analyzing contest platform Kaggle (California Healthcare Foundation [CHF], 2015). This dataset consists of a csv of image_id s and their corresponding classes.

❑ trainLabels.csv:

This csv file contains the name of images under the “image” column and their levels under the “level” column. The levels of images are discrete from 0 to 4.

❑ resized_train:

This is the folder where the images are. The image data collected was simply resized by the publisher into 1024x1024 if bigger than this size. Else, they remain their size. This dataset contains 35,126 high-resolution retina images of 17,563 different subjects’ eyes taken under different screening conditions. Each image is for one eye. For example, for the left eye of subject No. 10, the image_id is “10_left”.

4.3. Data Analysis

4.3.1. Exploratory data analysis

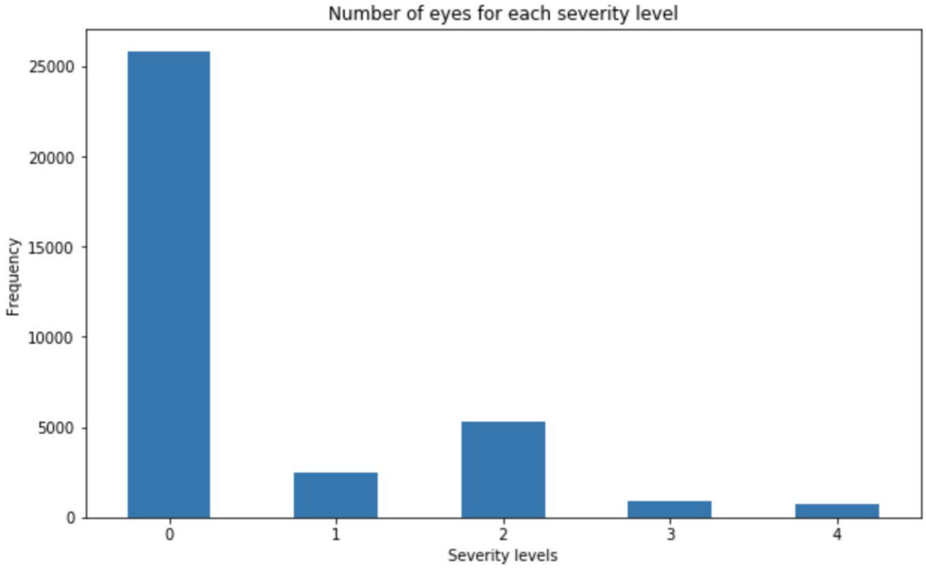
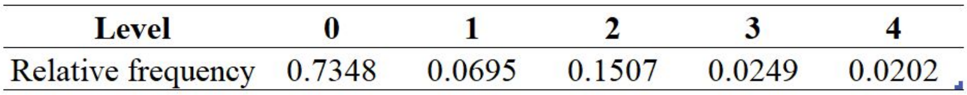


Figure 4.3.1.1: Distribution of severity levels



Level	0	1	2	3	4
Relative frequency	0.7348	0.0695	0.1507	0.0249	0.0202

Table 4.3.1.1: Percentages of the severity level distribution

For better understanding the dataset, we plotted a bar chart and a frequency table to show the distribution of the severity levels in our dataset. As shown in Figure 4.3.1.1 and Table 1, our dataset is highly unbalanced. More than 70% of data from severity level 0 while only 2% of data belong to level 4. Lacking inputs from other severity levels had a strong impact on our classification model. Our model tended to predict level 0 due to the imbalanced inputs. Therefore, we considered some approaches to deal with the issue, which would be discussed in later sections.

4.3.2. Data preprocessing

One of the most effective ways to improve the performance of our model is to improve the quality of input data. The first thing we took into account was the lighting conditions. These colour fundus photographs were taken under various environmental conditions, which lead to different lighting conditions and some are hard to visualize. Thus, we used the cv2 module to convert the original image (a) into grayscale (b) and RGB (c) format. It is clear that the colour distraction is removed and the images are much more understandable. To further improve the lighting condition, GaussianBlur and addWeighted function were applied to sharpen our input images. The GaussianBlur function first smooths images by suppressing most of the high-frequency components. Then, the smoothed images were subtracted from the original images using the addWeighted function. The outputs have most of the high-frequency components that are blocked by the smoothing filter. By adding back the output to the original can enhance the high-frequency components, which result in sharpened images. After sharpening the images, the blood vessels and many important details in the eyeball (d) and (e) were clearer and easier to identify. The sigmaX parameter in GaussianBlur stands for the kernel standard deviation in X direction. As shown in Figure 4.3.2.1, the exposure of images is increasing gradually from 10 to 50.

Another way to improve the performance of our model is to crop uninformative areas. The black edges around the retina sphere are useless, which takes unnecessary time for training the model. Therefore, we removed the uninformative areas by searching rows and columns that have at least one pixel along rows and columns that is larger than the pixel value for black areas, 0. Then, we indexed into image data to extract the useful bounding box. In this case, it is necessary to identify a tolerance value which tells the function how many areas we want to remove. A lower tolerance value (a,b,c) will keep more black, uninformative areas. A higher tolerance value (d,e,f) might result in losing information.

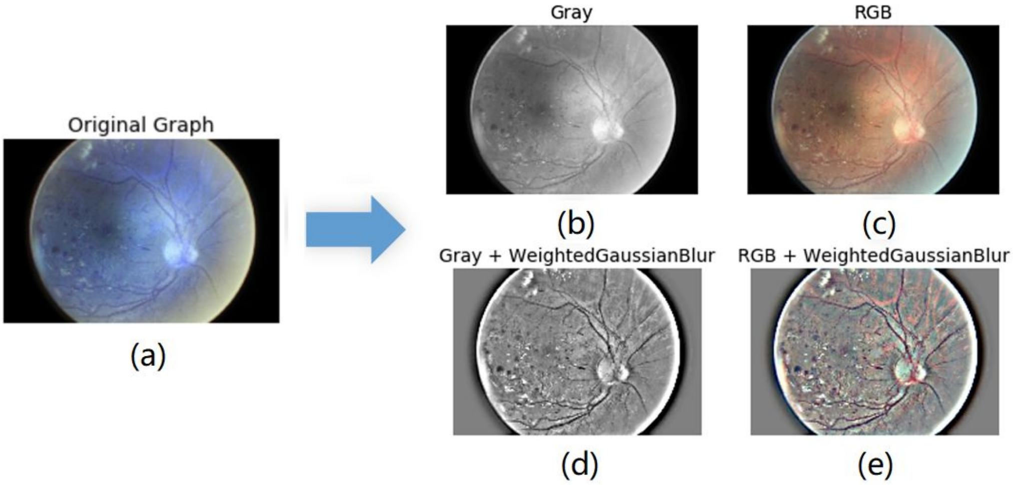


Figure 4.3.2.1: image preprocessing process of (a) original, (b) Gray, (c) RGB, (d) Gray and Sharpen, (4) RGB and Sharpen

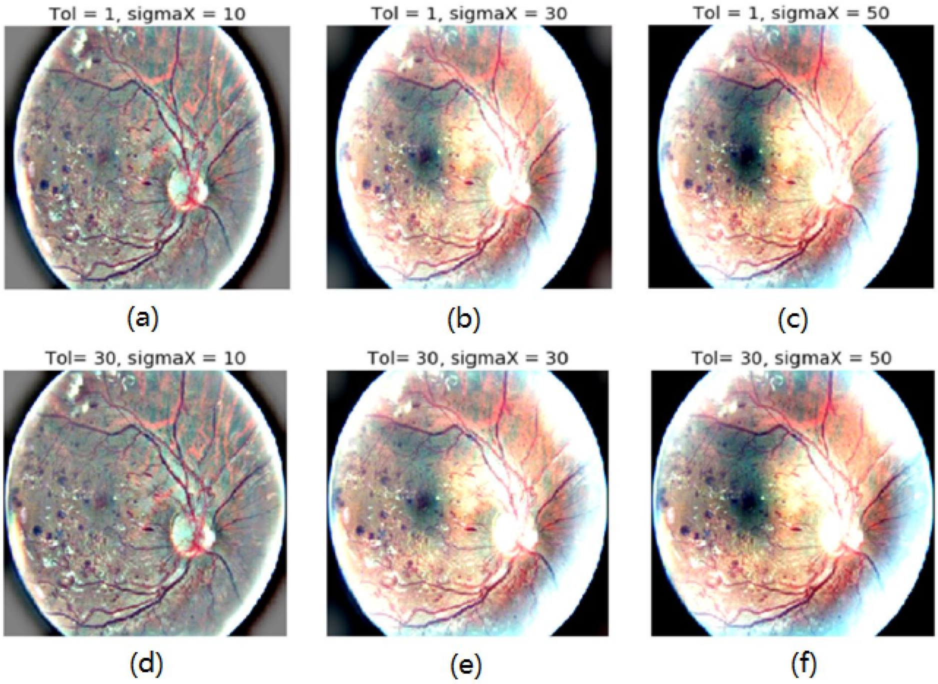


Figure 4.3.2.2: Image preprocessing process with different tolerance and sigmaX

4.3.3. Data Augmentation

Data augmentation is a strategy to increase the diversity of data available in training dataset. It helps to reduce manual intervention to develop more images and enhance data quality. In this project, four augmentation arguments applied to our training dataset are shown as following:

- ❏ **Rescale** : As the input data are images, the pixels are in the range of [0, 255]. Rescale 1./255. is to transform every pixel value in the range of [0,1]. It helps to treat all images in the same manner, so that images can make more evenly contributes to the total loss.
- ❏ **Brightness range**: The brightness of the images is augmented by adjusting different lighting effect, and this argument intent to generalize images by either randomly darken or brighten images.
- ❏ **Shear Range**: Shear mapping is to displace each point of the image in a certain direction. And it can be configured by the shear_range argument. The length of shifting is proportional to vertical (or horizontal) distance from point to x-axis.
- ❏ **Rotation**: This approach rotates image data in a certain direction to increase the variety of data. However, some information may be cropped off after rotation.

4.3.4. Balanced data

For image classification problem, distribution of data has a high impact on neural network model training and prediction. If a certain class of data outnumber other classes, the neural network tends to learn image features of the majority class. And the model likely returns the majority class during prediction. Therefore, several techniques including adding class weights and oversampling were using to eliminate the effect of unbalanced data.

- ❏ **Undersampling:** Random undersampling aims to balance class distribution by randomly eliminating majority class examples so that all classes have the same number of instances. Nevertheless, this method would potentially discard useful and important information.

❑ **Oversampling**: Oversampling is a robust balancing approach. After splitting the training and validation set, it creates copies of minority classes in the training set so that they would have the same number of candidates as the majority has. However, this method would lead to overfitting and increase training time. Therefore, the undersampling and oversampling methods were combined so as to retain important information without increasing training time. Half data from the majority class was removed randomly in the training set before doing the oversampling.

❑ **Class Weight Balancing**: This is to balance the data by altering the weight that each training example carries when computing the loss. Weights are inversely proportional to the size of classes. The value would pass to class_weight parameter while fitting the model. This can be useful to tell the model to "pay more attention" to samples from an under-represented class.

4.4. Modelling

4.4.1. Pretrained Model Selection

Xception is the pre-trained model we are using.It is an "extreme" version of inception model that would separately map the spatial correlation of every input channel and then use 1x1 convolution to map correlations across all channels. These two steps are called depthwise separable convolutions that created to reduce the time complexity of a standard convolutional layer. The architecture of Xception is a model with depthwise separable convolutions and residual connection linearly stacked.

As Figure 4.4.1.1 the workflow of Xception architecture, the data will first go through the entry flow, then repeat the middle flow 8 times, and finally into the exit flow.

The architecture of Xception is entirely based upon depth-wise separable convolutions. Depthwise separable convolutions break a standard convolutional layer into two parts: depthwise convolution and pointwise convolution.

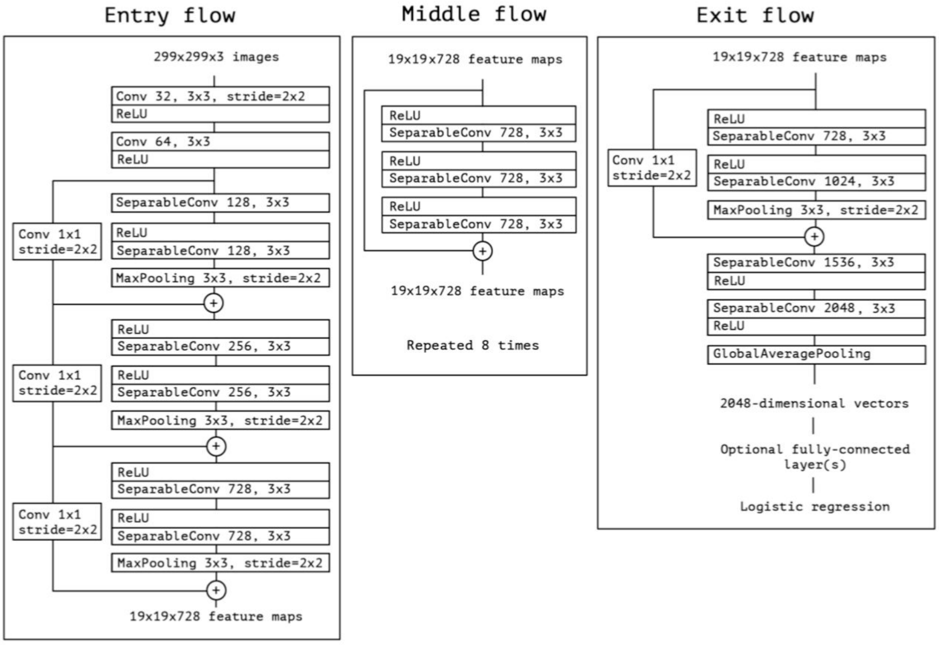


Figure 4.4.1.1: Xception Architecture (Chollet, 2016)

Depthwise Convolution: A filtering stage which performs individually over each channel of an input, unlike standard convolution which is applied throughout the entire depth. If the input data has its height and its width of size D_i and M number of input channels, the output should have same M number of channels, but the height and width $\mathrm{D_g}$ of output will change depending on the filter size $D_k * D_k$.

The total number of multiplication of this first step will be

$$D_k^2 * D_g^2 * M$$

Pointwise Convolution: A 1*1 convolution that linearly combines all M layers and project them onto a new channel space of N.

The total number of multiplication of this second step will be

$$D_g^2 * M * N$$

To get the total number of multiplications, we just need to add these two numbers.

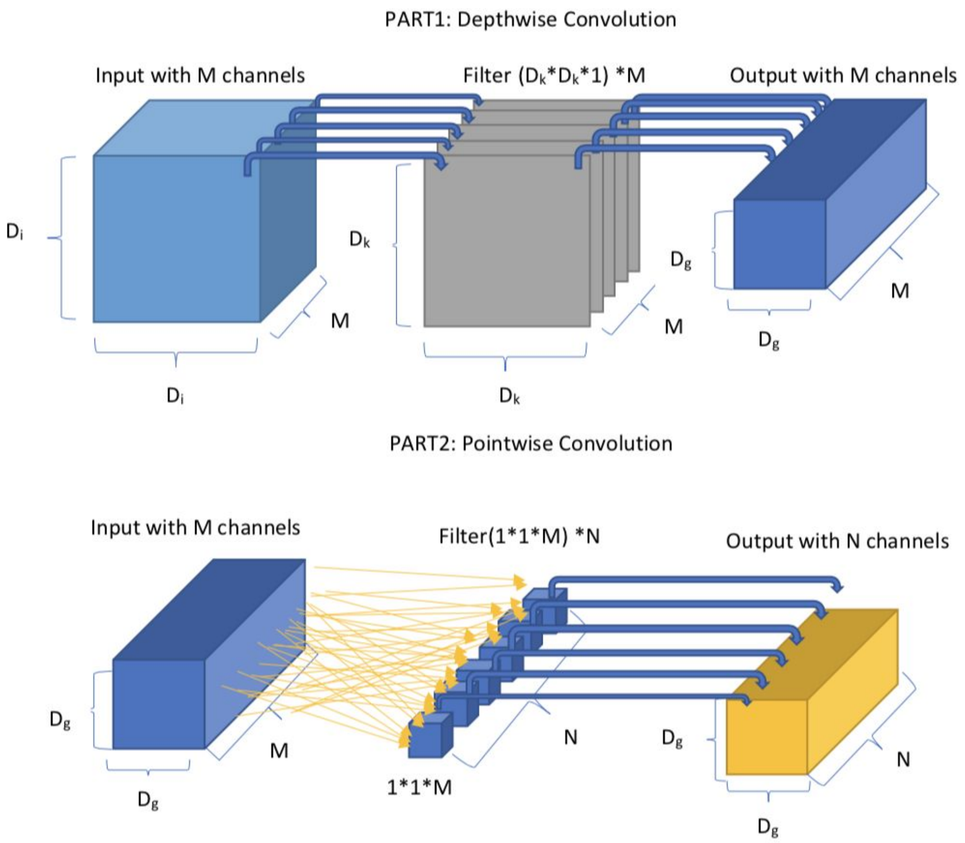


Figure 4.4.1.2. Depthwise separable convolution workflow

4.4.2. Hyperparameter tuning

- **Optimizer:**

Optimizers are algorithms used to update the weight and bis parameters of the neural network to minimize the loss function. In this project, there are three popular optimizers applied in our neural network - SGD, Adam and Nadam.

Stochastic Gradient Descent (SGD) performs frequent updates of model parameters, hence, it converges in less time. However, SGD has a high variance that causes the loss function heavily fluctuated. There might be an overshooting problem even after achieve a global minimum.

Adam stands for Adaptive Moment Estimation which is one of the most popular gradient descent optimization algorithms. It computes the adaptive learning rate for each parameter. Adam works well in practice as it converges rapidly and the learning speed is quite fast and efficient. However, it is computationally costly. It stores an exponentially decaying average of the past squared gradients, as well as an exponentially decaying average of the past gradients (Kingma & Ba, 2014).

Nadam (Nesterov-accelerated Adaptive Moment Estimation) combines Adam and NAG (Nesterov Accelerated Gradient), which performs a more accurate step in the gradient direction.

- ❑ **Loss function:**

In the machine learning area, loss function serves a purpose of guiding optimizer and evaluating the model. From the loss returned by the loss function, we can see if the model has a good fit on the data. Moreover, based on the loss function, optimizer can find the local minimum of loss and adjust parameters to improve the model with lower loss.

For a multi-label classification problem, the most common loss function is categorical cross-entropy. Compared to other loss functions, cross-entropy can precisely describe the distance between real distribution of label and distribution of predicted label, which means lower cross-entropy indicates that predicted distribution is closer to real distribution. Sparse categorical cross-entropy has the same function as categorical cross-entropy, the difference between them is that sparse categorical cross-entropy requires the target as an integer. On the other hand, categorical cross-entropy requires a one-hot encoded target.

Categorical cross-entropy is defined as:

$$Cross\ Entropy = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

❑ **Activation function:**

In a neural network, the output of the neural unit is the sum of weighted input and activation function can decide whether to activate those output or not. Multiple activation functions have been used in the model. ReLU activation functions have been added after each dense layer to avoid vanish of gradients or explosion of gradients issues. The softmax activation function is located at the end of the model, it projects output of dense layer to an interval between 0 and 1, which can be treated as a probability.

$$ReLU: f(x) = \max(0, x)$$

$$Softmax: f(x) = argmax(x)$$

❑ **Avoid Overfitting:**

With the increase of model complexity, overfitting issue is more likely occurred. To prevent overfitting, dropout and batch normalization techniques have been applied in the model.

❑ Dropout

Dropout is a regularization technique for neural network, it randomly drops neural units to avoid overfitting caused by a large number of parameters. Meanwhile, the reduced size of the network comes with a faster training process.

❑ Batch Normalization

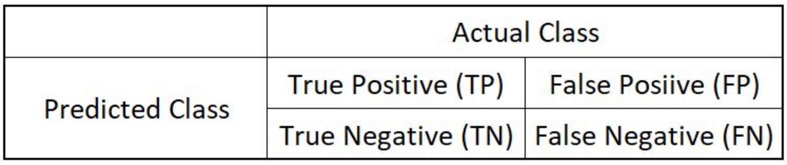
During the training of the neural network model, the distribution of input at each layer changes as the previous layer parameter changes. This causes vanishing of gradients and slows down the convergence. By calculating each mini-batches’ statistics, batch normalization changes the distribution of input into a standard normal distribution which is more sensitive to the activation function. Advantages of batch normalization include reducing internal covariate shift, accelerating convergence and avoiding overfitting.

4.5. Evaluation

Model evaluation is an integral part of the model development process. In this project, various evaluation metrics including accuracy, confusion matrix, precision, recall, F-1 score and Cohen’s kappa metrics had been used for assessing model’s performance. Since the project is dealing with a multi-class classification problem, metrics of overall classes are computing by maco-average approach.

❑ **Accuracy:** This is the rate of correct classification among all prediction in an independent test set. However, classification accuracy cannot explain performance well with unbalanced data sets and multi-class classification problem.

❑ **Confusion matrix:** Confusion matrix is a performance measurement for the machine learning classification problem. It is a table with four different combinations of predicted and actual values. The entries of the confusion matrix include true positive, true negative, false positive, false negative. By computing confusion matrix, metrics including Precision, Recall, Accuracy can be measured.



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

Table 4.5.1: Confusion Matrix

❑ **Precision**: The fraction of true positives among all of the examples which were predicted to belong to a certain class. Compared to accuracy, precision explained more about the model’s ability to predict the correct class.

$$Precision = \frac{TP}{TP + FP}$$

- ❑ **Recall:** Recall refers to the percentage of the total number of correctly classified class by the model.

$$Recall = \frac{TP}{TP + FN}$$

- ❑ **F-1 Score:** F-1 score is a harmonic mean of precision and recall. For a multi-label classification problem, precision and recall can not be maximized at the same time. Therefore, F-1 score is considered as a metric that balance precision and recall.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

- ❑ **Cohen’s Kappa Metric**: While making predictions on imbalanced data, the prediction performance of majority class will have a greater impact than minority class on the single, scalar metric like precision or recall. This problem is addressed by Cohen Kappa metric. Cohen’s kappa metric combines observed accuracy and expected accuracy together to evaluate classifiers. In addition, it takes into account expectation of model’s prediction, which means that it is less misleading than other classifiers. Pr(a) represents the observed accuracy while Pr(e) represents the expected accuracy. The formula is shown below:

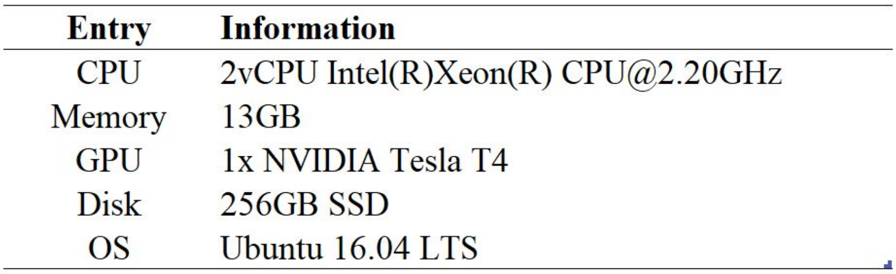
$$Cohen's\ kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

5. Resources

5.1. Hardware & Software

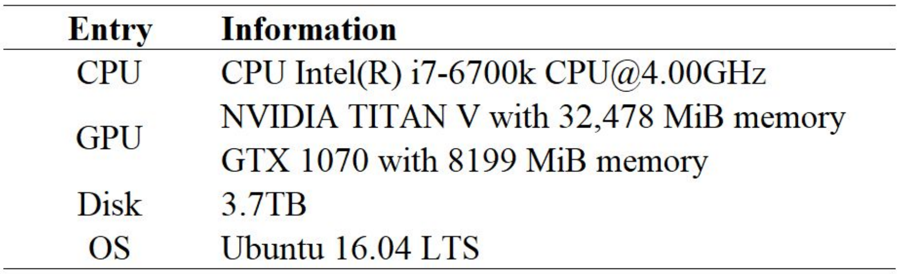
The hardware for coding is MacBook Pro (13-inch, 2018, Four Thunderbolt 3 Ports) of which the processor is 2.3 GHz Intel Core i5.

The hardware for model training is Google Cloud Platform and USYD lab server. The configuration of Google Cloud Platform and the USYD lab server refer to Table 5.1.1 and Table 5.1.2 respectively.



Entry	Information
CPU	2vCPU Intel(R)Xeon(R) CPU@2.20GHz
Memory	13GB
GPU	1x NVIDIA Tesla T4
Disk	256GB SSD
OS	Ubuntu 16.04 LTS

Table 5.1.1: GCP configurations



Entry	Information
CPU	CPU Intel(R) i7-6700k CPU@4.00GHz
GPU	NVIDIA TITAN V with 32,478 MiB memory GTX 1070 with 8199 MiB memory
Disk	3.7TB
OS	Ubuntu 16.04 LTS

Table 5.1.2: USYD lab server configurations

The software for coding is Google Colab which is an IDE for python. It consists of a range of free libraries and resources and can run on google servers without installing anything. It also provides free GPU as a hardware accelerator. Additionally, it allows us to share files seamlessly with project collaborators.

The software used for document sharing and data storing is Google Drive. The reason we choose it is because of its convenience due to which we can share and edit documents easily.

Smartsheet is used for creating Work Breakdown Structure (WBS) and visualizing Gantt chart.

The following python library is used for raising productivity:

❏ *Tensorflow.keras.preproecssing.image.ImageDataGenerator*: preprocessing image data and generate batches of tensor image data with real-time data augmentation.

❏ *Tensorflow.keras.application*: providing quick access of pre-trained models such as Densenet121, Xception and ResNet50.

❏ *Sklearn.metrics.cohen_kappa_score* and *accuracy_score*: evaluating model performance by computing Cohen's Kappa score and accuracy.

❏ *Tensorflow.keras.callbacks.TensorBoard*: tracking loss and accuracy, and visualizing the model graph.

5.2. Materials

The input images data are retrieved from dataset of Kaggle's competitions, listing as follows:

❏ Diabetic Retinopathy Detection Competition (California Healthcare Foundation [CHF], 2015)

❏ APTOS 2019 Blindness Detection Competition (Asia Pacific Tele-Ophthalmology Society [APTOS], 2019)

5.3. Roles & Responsibilities

❏ Project Stakeholder - Dr Matloob Khushi

Dr Matloob Khushi is the client of the project. As the project stakeholder, he provides advice and risks for the project including potential issues and functional requirements such as access to resources for the project.

❏ Functional Manager - Tin Lai

Tin is the tutor of the project. He provides suggestions to help the team to resolve existing tackles and communicates with the project stakeholders timely. He is also responsible for the evaluation of stage accomplishments and works with the group members to set and coach on the project goals.

❏ Project Manager - Jingjing Wu

Jingjing acts as a team leader in the group to ensure the completion of the project successfully. This includes conceptualization, planning, executing-controlling-monitoring and closing out. She also guarantees that the project proceeds within the specified time frame and assigns jobs to each individual in the group.

❏ Model Developer - Janet

Janet is responsible for developing models including model architecture design, code integration and technical strategy advisement. She builds up an appropriate model architecture and ensures that models are operable on different platforms. Furthermore, she migrates codes between data engineer and model trainer.

❏ Cloud Computing Engineer - Yilin

Yilin is responsible for providing expertise on cloud environment and results visualization. She provides help on cloud environment setup for team members and realizes experiments results visualization. Moreover, she makes advice on technical strategies, policy and procedures.

❏ Model Trainer - Boya

Boya is in charge of model training, hyperparameter optimization and training progress documentation. She creates model training and testing plans per the requirements and conducts procedures of model optimization once the model architecture is built up.

❏ Data Engineer - Ziang

Ziang is responsible for database management and technical support. He offers help in processing the data to meet the model’s needs during the architecture setup process. He is also responsible for communication between members in the project’s project documentation.

6. Milestones / Schedule

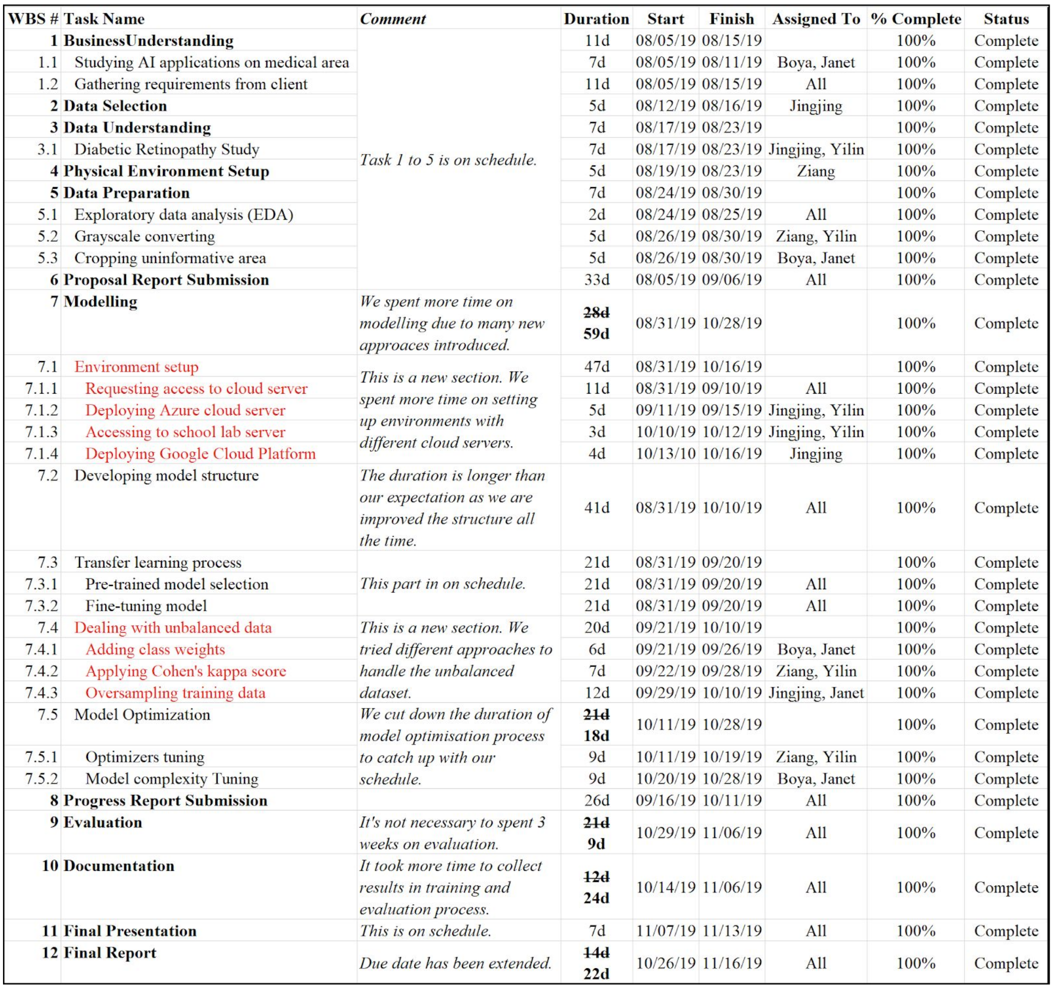
To deliver jobs on time and monitor the progress of the project, schedule and milestone were established after identifying the major functional deliverables and updated along with the progress. The schedule is recorded in a work breakdown structure and shown as a Gantt Chart.

Work breakdown structure (WBS) decomposes the task into many sub-tasks and visualizes them as a hierarchical table. Definition and details of each task are outlined in the table which included the time arrangement, personnel assignment, degree of completion and status. The WBS of this project is shown in Table 6.1.

Gantt Chart is a bar chart which horizontal axis is time and the vertical axis is the task. It demonstrates the duration of tasks and the link between tasks. From the Gantt chart, it is clear to show the visualization of project plan. Also, it provides effective time management and it is easy for project members to check project status. The Gantt Chart is created based on WBS and the result refers to Appendix - Gantt Chart.

The entire duration of our project has taken 103 days. While most of our assigned deadlines were met on time, there were a few changes made to the original plan.

Most changes of schedule were made in section seven, where we added two additional sections into our schedule due to project needs. First section is environment setup, because the training of neural network is highly reliant on the processor and we do not have adequate hardware to perform training, thus we decided to utilize cloud server resources to train. To begin, we asked for access of Azure server, which took 11 days, then we spent 5 days deploying the Azure cloud server and then we found out that azure server does not have GPU, which resulted in no significant increases in training speed. Thus, we spent three and four days respectively to access to school lab server and deploying Google cloud platform and finally set up the environment successfully. Through the work we performed in this section, we established a foundation base for later training sessions and also increase efficiency for future work.



WBS #	Task Name	*Comment*	Duration	Start	Finish	Assigned To	% Complete	Status
1	**BusinessUnderstanding**	*Task 1 to 5 is on schedule.*	11d	08/05/19	08/15/19		100%	Complete
1.1	Studying AI applications on medical area		7d	08/05/19	08/11/19	Boya, Janet	100%	Complete
1.2	Gathering requirements from client		11d	08/05/19	08/15/19	All	100%	Complete
2	**Data Selection**		5d	08/12/19	08/16/19	Jingjing	100%	Complete
3	**Data Understanding**		7d	08/17/19	08/23/19		100%	Complete
3.1	Diabetic Retinopathy Study		7d	08/17/19	08/23/19	Jingjing, Yilin	100%	Complete
4	**Physical Environment Setup**		5d	08/19/19	08/23/19	Ziang	100%	Complete
5	**Data Preparation**		7d	08/24/19	08/30/19		100%	Complete
5.1	Exploratory data analysis (EDA)		2d	08/24/19	08/25/19	All	100%	Complete
5.2	Grayscale converting		5d	08/26/19	08/30/19	Ziang, Yilin	100%	Complete
5.3	Cropping uninformative area		5d	08/26/19	08/30/19	Boya, Janet	100%	Complete
6	**Proposal Report Submission**		33d	08/05/19	09/06/19	All	100%	Complete
7	**Modelling**	*We spent more time on modelling due to many new approaces introduced.*	~~28d~~ **59d**	08/31/19	10/28/19		100%	Complete
7.1	Environment setup	*This is a new section. We spent more time on setting up environments with different cloud servers.*	47d	08/31/19	10/16/19		100%	Complete
7.1.1	Requesting access to cloud server		11d	08/31/19	09/10/19	All	100%	Complete
7.1.2	Deploying Azure cloud server		5d	09/11/19	09/15/19	Jingjing, Yilin	100%	Complete
7.1.3	Accessing to school lab server		3d	10/10/19	10/12/19	Jingjing, Yilin	100%	Complete
7.1.4	Deploying Google Cloud Platform		4d	10/13/10	10/16/19	Jingjing	100%	Complete
7.2	Developing model structure	*The duration is longer than our expectation as we are improved the structure all the time.*	41d	08/31/19	10/10/19	All	100%	Complete
7.3	Transfer learning process	*This part in on schedule.*	21d	08/31/19	09/20/19		100%	Complete
7.3.1	Pre-trained model selection		21d	08/31/19	09/20/19	All	100%	Complete
7.3.2	Fine-tuning model		21d	08/31/19	09/20/19	All	100%	Complete
7.4	Dealing with unbalanced data	*This is a new section. We tried different approaches to handle the unbalanced dataset.*	20d	09/21/19	10/10/19		100%	Complete
7.4.1	Adding class weights		6d	09/21/19	09/26/19	Boya, Janet	100%	Complete
7.4.2	Applying Cohen's kappa score		7d	09/22/19	09/28/19	Ziang, Yilin	100%	Complete
7.4.3	Oversampling training data		12d	09/29/19	10/10/19	Jingjing, Janet	100%	Complete
7.5	Model Optimization	*We cut down the duration of model optimisation process to catch up with our schedule.*	~~21d~~ **18d**	10/11/19	10/28/19		100%	Complete
7.5.1	Optimizers tuning		9d	10/11/19	10/19/19	Ziang, Yilin	100%	Complete
7.5.2	Model complexity Tuning		9d	10/20/19	10/28/19	Boya, Janet	100%	Complete
8	**Progress Report Submission**		26d	09/16/19	10/11/19	All	100%	Complete
9	**Evaluation**	*It's not necessary to spent 3 weeks on evaluation.*	~~21d~~ **9d**	10/29/19	11/06/19	All	100%	Complete
10	**Documentation**	*It took more time to collect results in training and evaluation process.*	~~12d~~ **24d**	10/14/19	11/06/19	All	100%	Complete
11	**Final Presentation**	*This is on schedule.*	7d	11/07/19	11/13/19	All	100%	Complete
12	**Final Report**	*Due date has been extended.*	~~14d~~ **22d**	10/26/19	11/16/19	All	100%	Complete

Table 6.1 Work Breakdown Structure table (red colour indicates additional section on initial plan)

The other section deals with unbalanced data. In the beginning, we did not realize that there is a huge impact of unbalanced data on the end result. As training progressed, we realized that unbalanced data heavily influenced model training, thus we used methods of adding class weight and oversampling training data in order to decrease these impacts. We also used Cohen's Kappa score to evaluate the results.

Environment set up and development of model structure happened simultaneously. As mentioned earlier, we realized that Azure Cloud Server did not increase training efficiency drastically after we performed modelling, and as a result, our proposed 28 days for modelling was influenced and extended to 59 days. We cut down our evaluation process by 12 days because when we reached that stage we realized that evaluation did not need 3 weeks to complete. Accordingly, our documentation process took longer because we recognized that it took longer to collect results in the training and evaluation process. And the duration from the documentation is from an original 12 to the doubled 24 days. Our final report date has also been extended because the due date was extended.

7. Results

7.1. Experiments

The experiments include four parts. The first step is to select the best pre-trained model. There are three pre-trained models used in the tuning process - ResNet50, InceptionV3 and Xception. And then, the model is trained with 3 different optimizers - SGD, Adam, and Nadam. The speed of convergence and generalization are two aspects to determine the optimizer efficiency. Thirdly. the model is constructed with two different balanced approaches. And finally, we increased the model complexity by adding more layers.

The epoch accuracy and loss graphs are created to present the comparison results. Meanwhile, the testing performance is shown with testing kappa scores, and this is considered in model assessment as well.

7.1.1. Pretrained Model Selection

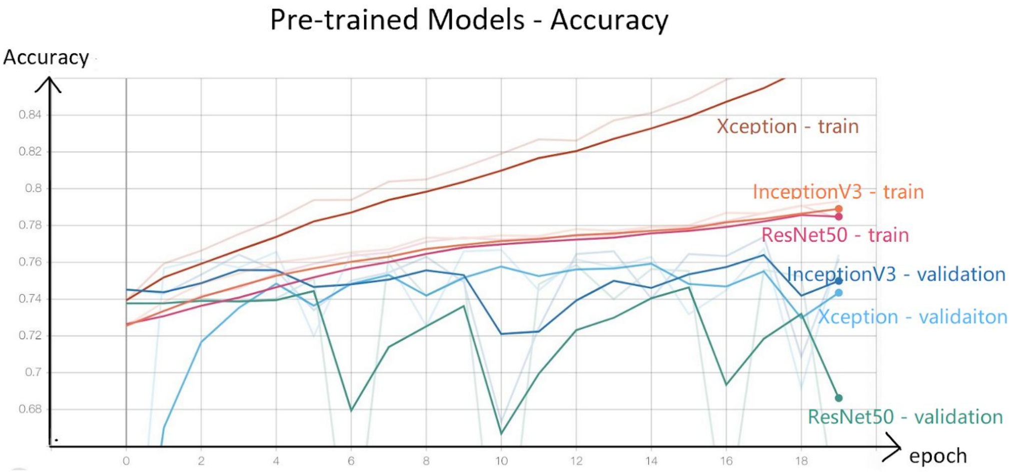


Figure 7.1.1.1: Epoch accuracy for pre-trained models

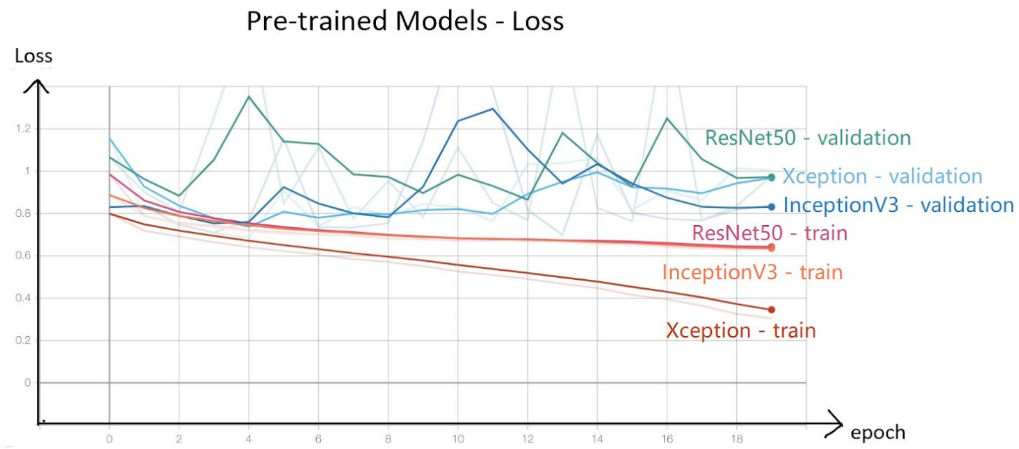


Figure 7.1.1.2: Epoch loss for pre-trained models

Pre-trained Model	Testing Kappa Score (%)
ResNet50	54.80
InceptionV3	57.60
Xception	**58.90**

Table 7.1.1.1: Testing results for pre-trained models

The graphs and table above indicates the results for 3 different pre-trained model. From Figure 7.1.1.1 and Figure 7.1.1.2, it shows that Xception has the highest accuracy in training dataset and the loss is decreasing as expected. And for ResNet50 and InceptionV3, they have similar performance. The two lines are overlapping in both accuracy and loss graphs. As the dataset is unbalanced, Cohen’s Kappa score is used to measure the performance on testing dataset. Table 7.1.1.1 indicates the testing Kappa scores on the different pre-trained model. We can figure out that Xception has better performance than the other two pre-trained models. As a result, Xception is selected as the pre-trained model in the rest of the tuning process.

7.1.2. Optimizer Tuning

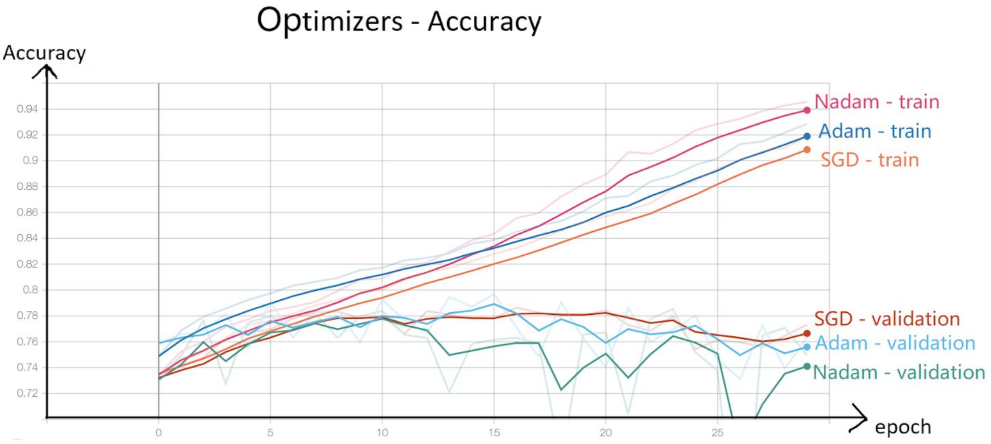


Figure 7.1.2.1: Epoch accuracy for optimizers

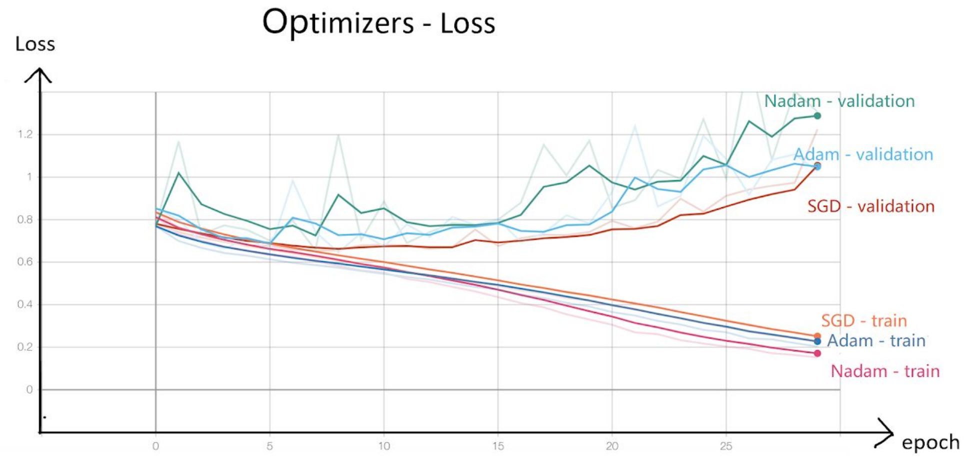


Figure 7.1.2.2: Epoch loss for optimizers

Optimizer	Testing Kappa Score (%)
SGD	51.89
Adam	54.02
Nadam	**58.90**

Table 7.1.2.1: Testing results for optimizers

When tuning the optimizers, the learning rate is one of the most important hyperparameter to configure the neural network. The learning rate is set as 0.01 for SGD, 0.001 for Adam, and 0.002 for Nadam. However, Adam and Nadam support adaptive learning rate, which accelerate training and reduce the pressure on choosing a learning rate. The accuracy and loss of optimizer training are shown in Figure 7.1.2.1 and Figure 7.1.2.2 There is no big difference among three optimizers. But when looking at the test performance on testing dataset, it shows that Nadam has highest Kappa score of 58.9%. So Nadam is applied as the optimizer to training model.

7.1.3. Balanced Methods

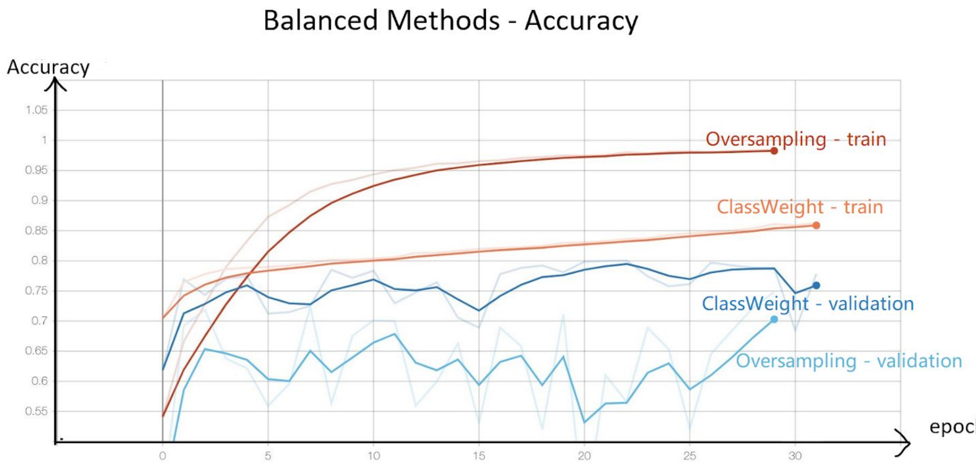


Figure 7.1.3.1: Epoch accuracy for balanced methods

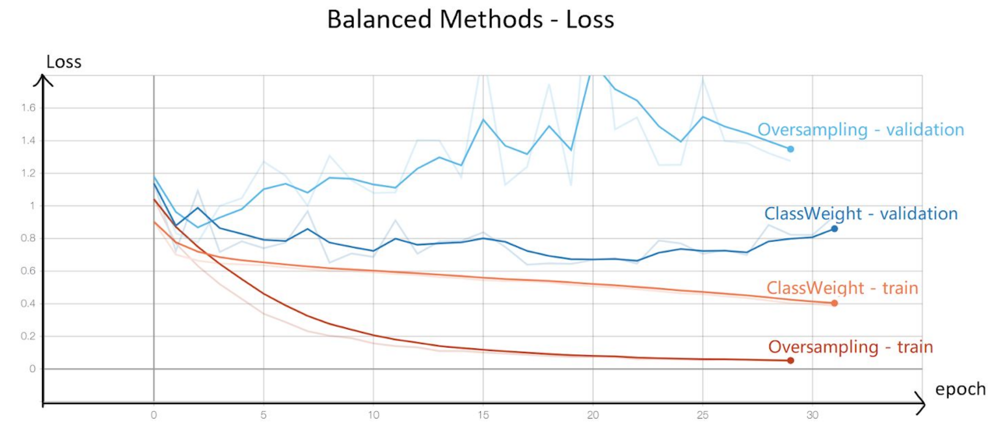
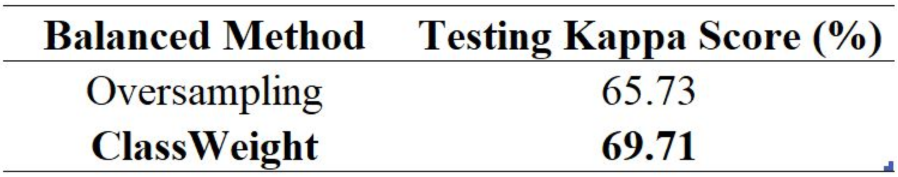


Figure 7.1.3.2: Epoch loss for balanced methods

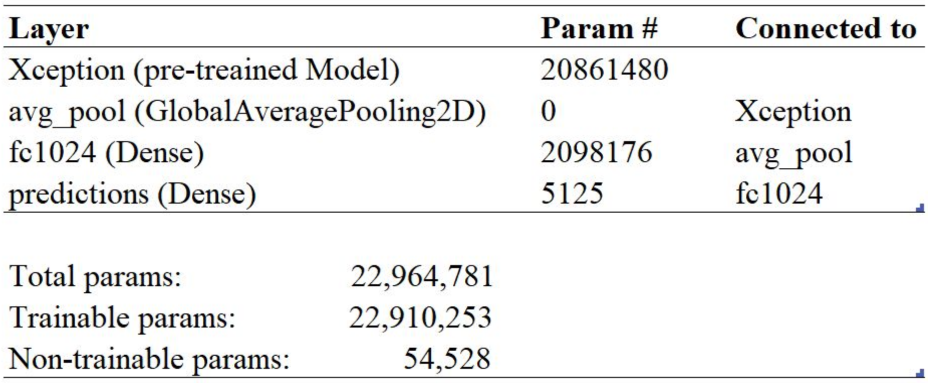


Balanced Method	Testing Kappa Score (%)
Oversampling	65.73
ClassWeight	**69.71**

Table 7.1.3.1: Testing results for balanced methods

From Figure 7.1.3.1 and Figure 7.1.3.2, it is clear that oversampling method achieves higher training accuracy and lower loss as compared to the class weight method. However, the oversampling method leads to serious overfitting. The validation loss is increasing after 3rd epoch. The model with class weight method has comparable performance on both train and validation datasets. As shown in Table 7.1.3.1, the class weight method has a higher kappa score on testing dataset. Therefore, we use class weight method to deal with our unbalanced dataset.

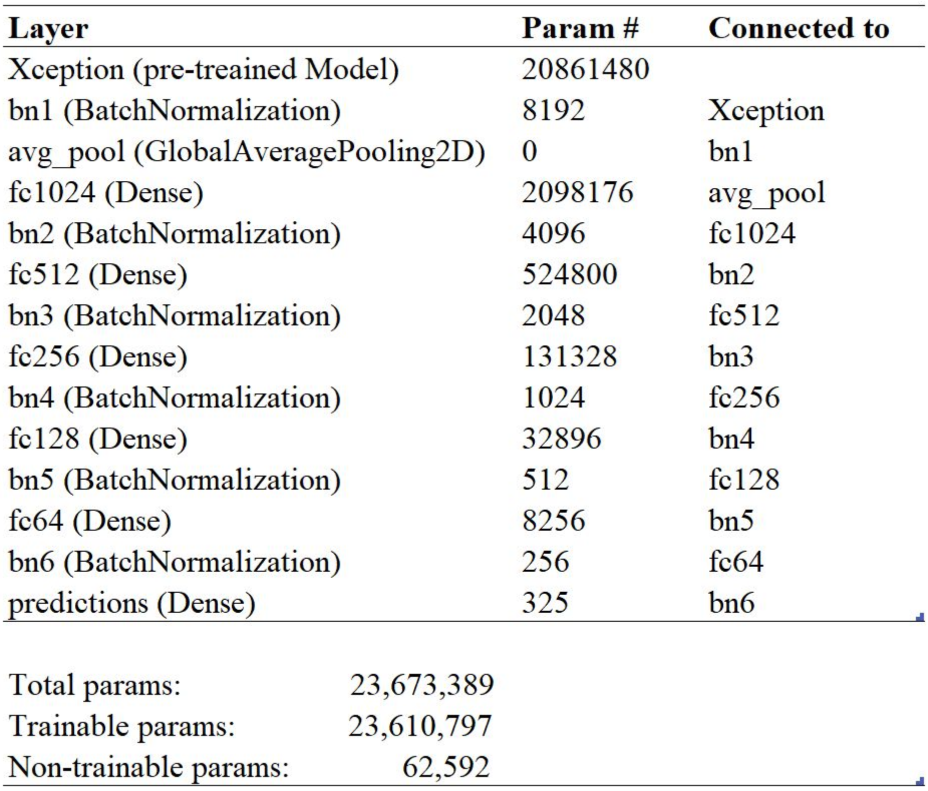
7.1.4. Model Complexity



Layer	Param #	Connected to
Xception (pre-treained Model)	20861480	
avg_pool (GlobalAveragePooling2D)	0	Xception
fc1024 (Dense)	2098176	avg_pool
predictions (Dense)	5125	fc1024

Total params:	22,964,781
Trainable params:	22,910,253
Non-trainable params:	54,528

Table 7.1.4.1: Base model summary



Layer	Param #	Connected to
Xception (pre-treained Model)	20861480	
bn1 (BatchNormalization)	8192	Xception
avg_pool (GlobalAveragePooling2D)	0	bn1
fc1024 (Dense)	2098176	avg_pool
bn2 (BatchNormalization)	4096	fc1024
fc512 (Dense)	524800	bn2
bn3 (BatchNormalization)	2048	fc512
fc256 (Dense)	131328	bn3
bn4 (BatchNormalization)	1024	fc256
fc128 (Dense)	32896	bn4
bn5 (BatchNormalization)	512	fc128
fc64 (Dense)	8256	bn5
bn6 (BatchNormalization)	256	fc64
predictions (Dense)	325	bn6

Total params:	23,673,389
Trainable params:	23,610,797
Non-trainable params:	62,592

Table 7.1.4.2: Complex model summary

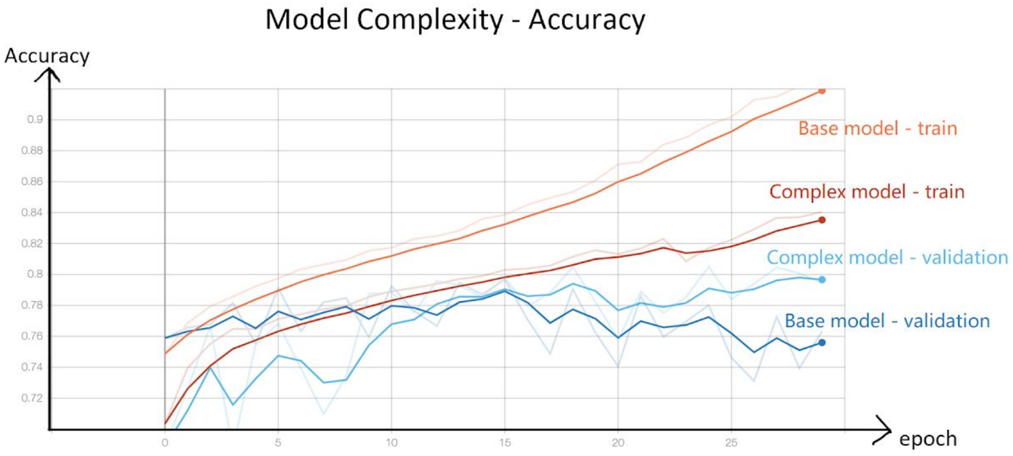


Figure 7.1.4.1: Epoch accuracy for model complexities

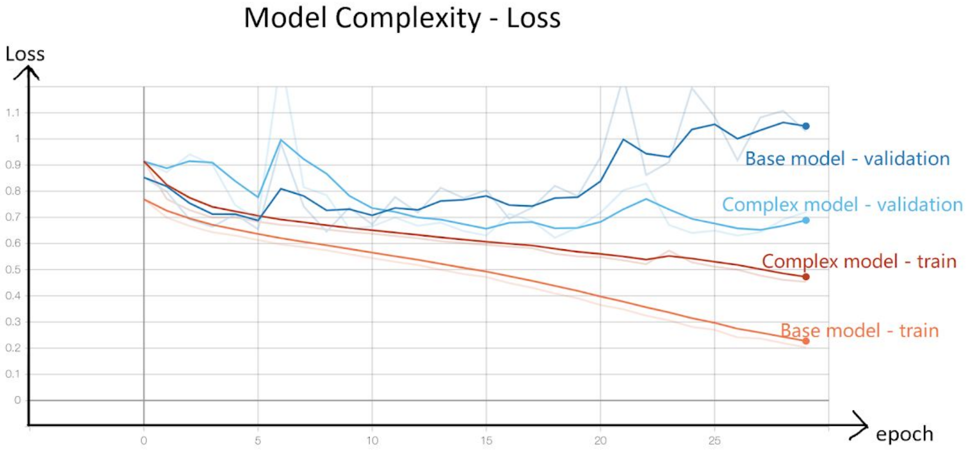


Figure 7.1.4.2: Epoch loss for model complexities

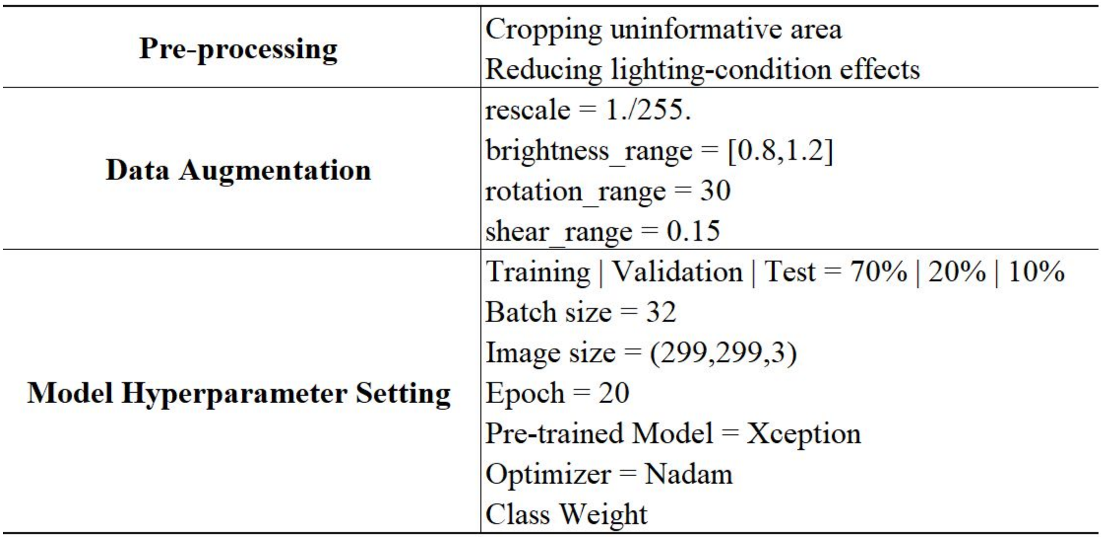
Model Complexity	Trainable Parameters	Testing Kappa Score (%)
Base model	22,910,253	69.71
Complex model	**23,610,797**	**79.71**

Table 7.1.4.3: Testing results for model complexities

In order to compare the performance of different model complexity, we constructed two models. The base model (Table 7.1.4.1) only has one dense layer of 1024 neurons before the output layer. The complex model (Table 7.1.4.2) contains 5 dense layers with decreasing neurons from 1024 to 64 in each layer. The trainable parameters for the base model and complex models are about 23 million and 24 million respectively. From Figure 7.1.4.1, the training accuracy for the base model is much higher than the complex model. However, the validation accuracy is not improved at all. In contrast, both training and validation accuracies for the complex model are increased gradually. As shown in Table 7.1.4.2, the complex model achieves a more satisfying kappa score at 79.71%. Thus, the complex model is used as our final model.

7.2. Final Architecture

After going through all these experiments, the final model was constructed with the following methods and parameters as shown in Table 7.2.1.



Pre-processing	Cropping uninformative area Reducing lighting-condition effects

Data Augmentation	rescale = 1./255.
	brightness_range = [0.8,1.2]
	rotation_range = 30
	shear_range = 0.15

Model Hyperparameter Setting	Training | Validation | Test = 70% | 20% | 10% Batch size = 32 Image size = (299,299,3) Epoch = 20 Pre-trained Model = Xception Optimizer = Nadam Class Weight

Table 7.2.1: Model Training Setting

Table 7.2.1 shows the visualizing dataflow graphs of our final model. The input is our DR images, and the input shape of our image is 299 x 299 x 3. Firstly, all pre-processed images are passed into the pre-trained model - Xception. The output shape is 10 x 10 x 2048. Then we add batch normalization layer after the pre-trained model. Next layer is the global average pooling layer, which is similar to the fully connected layer. It performs linear transformations of the vectorized feature map (Lin, Chen & Yan, 2013) and the output shape is 2048. After this, 5 dense layers are added with neurons from 1024 to 64 in a decreasing order. The dense layers are narrower and deeper towards the end. Meanwhile, Batch Normalization layers are added before each dense layer to reduce overfitting problems. This contributes to dramatically accelerate the training process of the neural network. And ReLU activation function is applied after each dense layer. This is more computationally efficient and the networks with ReLU tend to have better convergence performance. The output layer is with 5 neurons because we are supposed to have the same number of neurons as classification levels. And then the resulting vector is fed into the Softmax layer. Softmax assigns decimal probabilities to each class, and the target class has the highest probability.

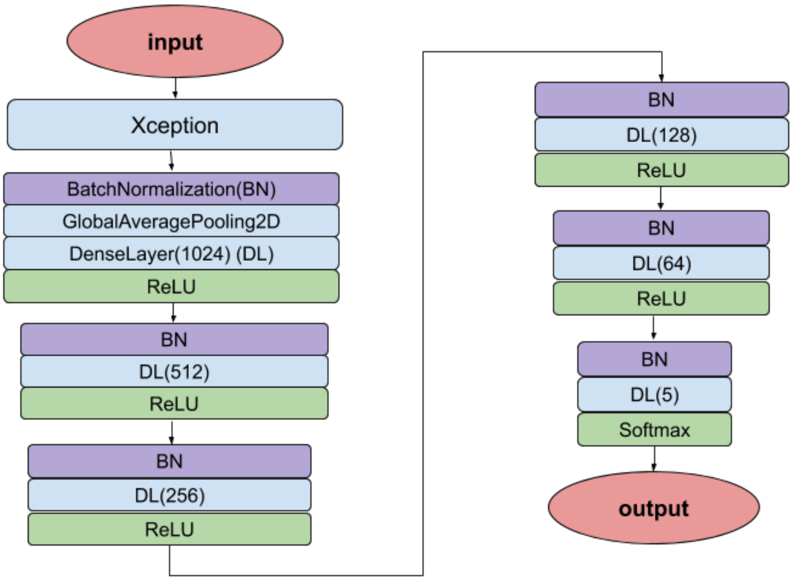


Figure 7.2.1: Final Model Architecture

As shown in Table 7.1.4.2, It indicates that layers of the final model and number of parameters each layer has. In pre-trained model Xception, there are totally 20,861,480 parameters. The model does not include the top layer of Xception, but each layer inside is trainable. After pre-trained model, there are 13 layers. The total number of parameters in the final model are 23,673,389 while the number of trainable parameters is 23,610,797.

7.3. Model Performance Evaluation

To evaluate if our model overfits the current data by learning too many details and noises, a new dataset was introduced. The new dataset APTOS 2019 Blindness Detection (Asia Pacific Tele-Ophthalmology Society [APTOS], 2019) is collected from the Kaggle 2019 competition. There are 3662 images in the training dataset. And the data is split into training, validation, and testing as well. Although the data size is smaller, the image quality is much higher compared with data from Kaggle 2015 competition. Due to limited time and resources, only data from Kaggle 2015 competition was used for research and study. However, we tried the final model architecture on both dataset.

To assess the model performance, we made predictions on the testing dataset for both 2015 competition data and 2019 competition data. And classification report and confusion matrix are used to visualize and summarize the number of correct and incorrect predictions with count values and stratified by each class.

7.3.1. Classification Report

	precision	recall	f1-score	No. of samples
0	0.88	0.98	0.93	2571
1	1.00	0.00	0.01	256
2	0.72	0.65	0.68	527
3	0.45	0.58	0.50	91
4	0.89	0.35	0.51	68
Total				3513
accuracy			0.84	
macro avg	0.79	0.52	0.53	
weighted avg	0.85	0.84	0.80	

Table 7.3.1.1: Classification report for 2015 Competition data

	precision	recall	f1-score	No. of samples
0	1.00	0.99	1.00	181
1	0.93	0.76	0.84	37
2	0.83	0.97	0.89	117
3	0.56	0.82	0.67	11
4	1.00	0.14	0.25	21
Total				367
accuracy			0.91	
macro avg	0.86	0.74	0.73	
weighted avg	0.92	0.91	0.90	

Table 7.3.1.2: Classification report for 2019 Competition data

The classification report displays the main classification metrics on a per-class basis, which provides us a deeper intuition of the model behavior in each class, especially for unbalanced dataset. As shown in Table 7.3.1.1, although the overall accuracy for predicting 2015 competition data is 0.84, the model is unable to differentiate levels 1 to 4 very well, especially for level 1. It only performs well on predicting level 0, with a high F1-score at 0.93. However, the model performs much better on 2019 competition dataset (Table 7.3.1.2). The overall accuracy achieves 0.91 with the F1-score for first three levels all over 0.8. It has relatively poor performance on detecting level 3 and 4. By treating all classes equally, the macro average of F1-score for these two datasets are 0.53 and 0.73 respectively.

7.3.2. Confusion Matrix

Confusion matrix clearly indicates the distribution of our predicted levels across all the actual outcomes, which is much more detailed representation of what is going on with our labels. According to Figure 7.3.1.1, our model is successful in identifying 98% of candidates in level 0 correctly, with only 2% are marked as level 2. It tends to predict other levels into level 0 as well. 87% of candidates in level 1 are misclassified into level 0, while 25% of candidates in level 2 are labeled as level 0. Based on the colour for the rest levels, the model is unable to label them very well. In contrast, Figure 7.3.1.2 clear shows that the model has better performance on 2019 competition data. The colours in the diagonal elements are much darker than those in Figure 7.3.1.1, except for level 4. The high true positive rates in level 0 to 3 indicating that there are many correct predictions.

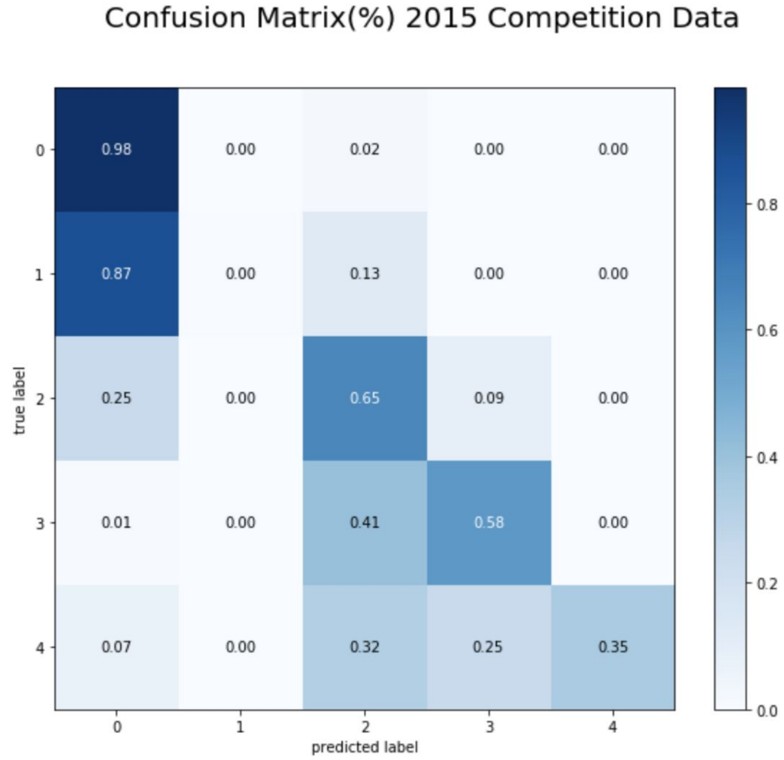


Figure 7.3.2.1: Confusion matrix for 2015 Competition data

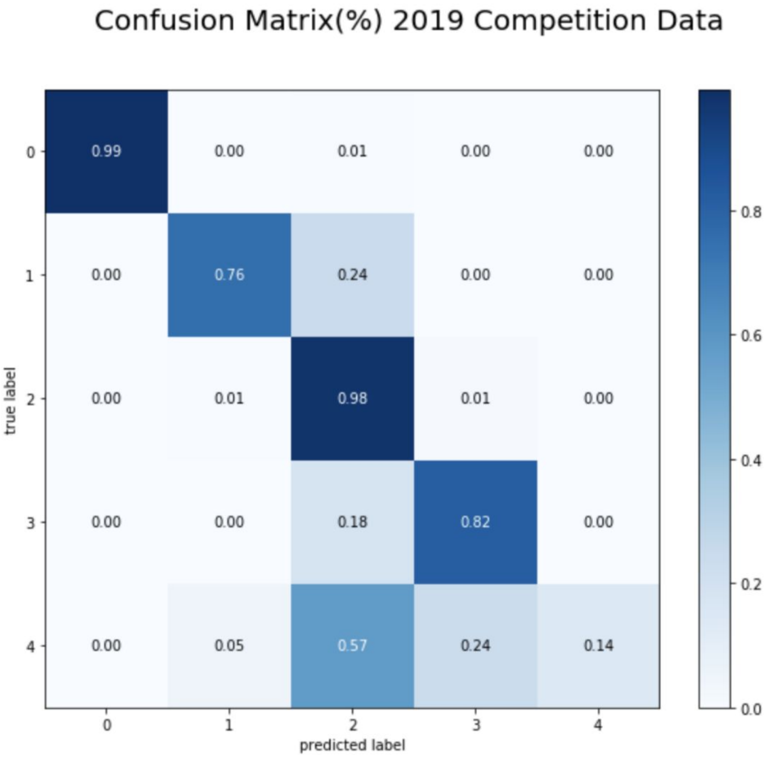


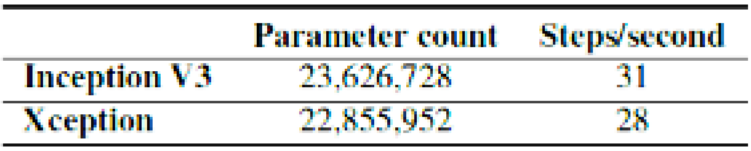
Figure 7.3.2.2: Confusion matrix for 2019 Competition data

8. Discussion

8.1. Pre-trained Model Selection

All evaluations were run with our inputs images first gone through the preprocessing method. For pre-trained model selection, we tested ResNet-50, Inception V3, and Xception that included a dense layer of 1024 units before the softmax layer for classification. In general, Xception outperforms the other two. According to Google's original paper that first introduced Xception model, this deep learning architecture performs even better than Inception V3 on both ImageNet ILSVRC and JFT datasets. And it vastly outperforms ResNet-50 on ImageNet ILSVRC dataset. It has the modified depthwise separable convolution layers that was motivated by the inception module in Inception V3 and it also has residual connections that was originally proposed by ResNet (Chollet, 2017). We believe these should be reasons for its good performance.

Furthermore, due to its featured depthwise separable convolution, it is not needed to perform convolution across all channels. Besides, Xception sports the smallest weight serialization at only 91MB. Considering all these facts of it, Xception actually has fewer connections and lighter structure, which makes it faster for training.



	Parameter count	Steps/second
Inception V3	23,626,728	31
Xception	22,855,952	28

Table 8.1.1. Size and training speed comparison. (Chollet, 2017)

8.2. Optimizer Tuning

Compared to classical momentum, NAG has both the momentum step and the gradient step depend on the current gradient. On the other hand, Adam defines momentum(m) as a decaying mean rather than a decaying sum that NAG does over the previous gradients and it includes an adaptive learning rate. This helps the algorithm to change the direction while the learning rate has changed. It also directly corrects for the “initialization bias” that caused by initialized momentum vector as 0.

Nadam improves mostly on Adam’s algorithm with consideration of current gradient in both the momentum term and the gradient term. Stochastic Gradient Descent (SGD) as a variant of gradient descent usually has a stable performance with a good initialization and learning rate. Besides, it also needs a longer time to converge to the minimum. In order to do that, more time is needed.

Regarding our base model using different optimizers, SGD looks relatively stable on validation loss when compared to Adam and Nadam’s being fluctuating a lot. However, our time is limited and we were training models all in 20 epochs which can be considered insufficient for SGD to converge. Therefore, we only tried SGD with learning rate 0.01 and the testing result was not as good as those of Adam and Nadam. Considering the time constraint and the testing results in Table 7.1.2.1, Nadam was the best option.

8.3. Balanced Methods

Class weights directly modify the loss function by giving more or less penalty to the class with more (or less) frequency. Therefore, when the model is using the loss function to approximate our observations, it will sacrifice some ability to predict the majority classes in the unbalanced data that carry lower weights.

Oversampling and undersampling techniques basically give more weights to some classes by duplicating data of some particular classes which will also double the penalty to those classes. This ideally gives us sufficient number of data to learn. However, it may also easily lead to overfitting problem.

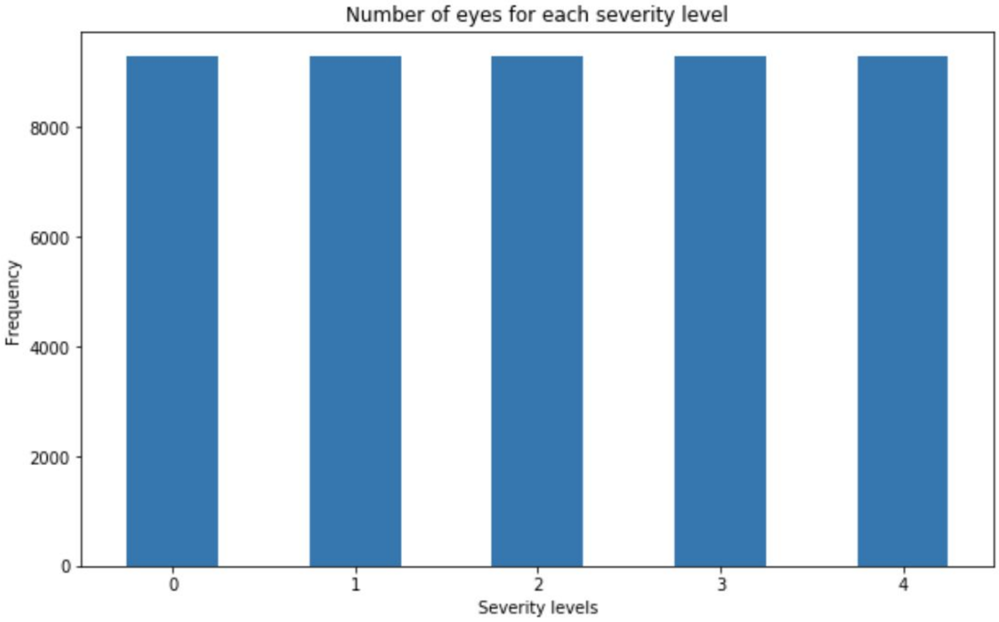


Figure 8.3.1: Level distribution after oversampling

We implemented class weight and oversampling method for our data imbalance issue. From our plots of loss and accuracy history during the training process of both models (Figure 7.1.3.1& 7.1.3.2), class weight method outperforms oversampling method that obviously leads to overfitting problem.

From our perspective, for image classification task, it is very likely to overfits the training set when using the oversampling method because it makes exact copies of existing observations. It does not actually increase the data diversity. When our data is oversampled, level 1, 2, 3, and 4 data were duplicated many times, lesions looking exactly the same would be trained again and again. That does not help our model to learn features from these monotonous images. Furthermore, since we undersampling our data first, losing important information can happen when half of zero are cut from our data randomly. Then, our model cannot develop the ability to differentiate level 0 and other labels. Therefore, class weight is selected as our balancing data method.

8.4. Model Complexity

The structures of our base model and complex model are shown as Table 7.1.4.1 and 7.1.4.2 in section 7.1.4. The base model only has one hidden layer with 1024 neurons between the output of pre-trained model and the output layer. Hidden layers are the magic of neural nets. CNN model learn to detect various features of input images using numbers of hidden layers. Every hidden layer increases the complexity of the learned input features. They provide the discrimination that is necessary to be able to separate our training dataset. However, increasing the number of hidden layers does not always result in improving the model accuracy, which really depends on the complexity of the problem. To increase the dimensional complexity of the data we can learn, we tried to add more hidden layers to our model. The network might overfit to the training dataset when the number of hidden layers much more than sufficient number of layers. In our case, the complex model has the appropriate number of hidden layers to accurately represent the data and to generalize the unseen data.

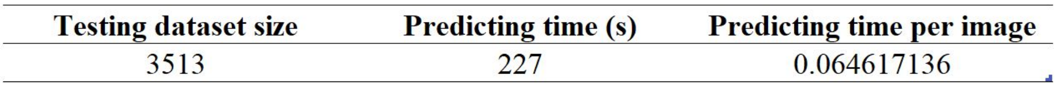
Moreover, the number of neurons in the hidden layers is also a very important part of deciding our overall neural network architecture. Too many neurons in the hidden layers might lead to overfit the training dataset as well. As shown in Figure 7.1.4.1 and 7.1.4.2, the base model is overfitting, which is probably because the neurons decreases dramatically from 1024 in hidden layer to 5 in the output layer. Therefore, while adding 5 more hidden layers to the complex model, we also modified the number of neurons in each layer. The number of neurons decrease gradually from 1024 to 64, before passing to the output layer. Considering the epoch accuracy and loss, and the testing kappa score, the complex model achieve better performance on our dataset.

8.5. Final Model Conclusion

From the two classification reports of our final model on both versions of data, it shows a large improvement from the result of CHF 2015 dataset to APTOS 2019 dataset. Figure 7.3.1.1(2015) shows comparatively high precision score. But that does not necessarily mean our model is doing well on predicting DR for the reason that all classes have lower recall which measure the quality of the prediction with respect to the mistake we made. Lower recall score indicates our model did not manage to predict some levels right to some extent. For instance, level 1 has precision 100% but recall 0%. Precision can be 100% even if one of the actual 1 is correctly identified and all other data are not predicted as 1. This may lead to the problem that other 255 data of true 1s are not correctly identified. Therefore, the value for recall is close to zero. The diagonal elements in the confusion matrix represents the rates for which the predicted label is equal to the true label, which is the recall. With the colorbar in confusion matrix, it is much clear representation of how strong our model can make predictions for each class. The darker the color in the diagonal elements, the higher the recall. Recall can be more important than precision in our study because we want all the fundus images be accurately predicted as where they should belong to. Even though the difference between the 0 and 1 is subtle, accurate prediction can allow early identification of patients at high risk of getting PDR to have timely referral to specialists.

8.6. Implication and Significance

Our work represents a progress in the application of AI to ophthalmology. The DL algorithm introduced here is differed from other DR diagnosis procedure that is time consuming for grading DR. Our model that achieve 227 seconds for predicting 3513 images can increase the efficiency of DR detection and solve the issue that human resources for eye services and skilled graders in developing countries are extremely short.



Testing dataset size	Predicting time (s)	Predicting time per image
3513	227	0.064617136

Table 8.6.1. Testing Performance

In general, manual grading of DR would reference some grading systems. For example, following the instructions by Scottish Grading Protocol, the graders should first evaluate the fundus photography quality on the basis of the sharpness and clarity of nerve fibres. Good Images will then go through systematic inspection from optic disc to macula, and then other areas. Then, more image editing including the essential red filter will be applied to highlight the subtle features and improve visualization of the graphs. The size of blot haemorrhages are also required to be measured for grading (Zachariah, Wykes, & Yorston, 2015). There is no agreement on which grading system is the best but each step of these systems needs extra time. Our model can improve the efficiency of this process by directly predict each fundus image in 0.06 second after screening with 91% accuracy.

According to the American Diabetes Association (ADA) guidelines, each patient that is diagnosed with diabetes mellitus needs to do an immediate retinal examination and another examination following it biannually until they no longer have DR. In general, this means every patient needs to have a retinal examination once a year. In Bangladesh, almost 13 million patients need to be screened for DR detection (American Diabetes Association, 2013). However, the fact is that human resources for these services and DR graders are inadequate. For millions of people who needs retinal examination, more medical doctors need specialized education and training when compared to the targeted ratio of ophthalmologist-population was set 1:100,000. But, in fact, most countries within South-East Asia Region did not achieve the target (World Health Organization, 2009).

Being cognizant of the limitations on collecting and grading fundus photos, our model using deep learning algorithm can be an alternative for those professional graders to save the cost and time for training them, especially for the low resource countries.

9. Limitations and Future Works

This report presented an automated DR detection system of color fundus images using convolutional neural networks. Beyond current research stage, there are still many insufficiencies of our project even a high accuracy is obtained.

Firstly, one constraint in our study is the limitation of computing resources in model training. The training of deep learning models requires a highly configured graphics processing unit (GPU). At first, the faculty of the university only provided us the Microsoft Azure Lab Service, which was merely equipped with CPU. However, it is a waste of time and unrealistic to train the model simply with CPU. With CPU, the training time took more than 24 hours even for only one single epoch in our model. This caused a serious delay of schedule in the project. Therefore, we finally gave up training models in the server provided and decided to train model in Google Cloud Platform and GPU server of University of Sydney provided by our tutor. Here we want to thank him for his selfless help. With GPU server, the training time of our model took only 20 minutes for each epoch which was a dramatic improvement compared to before. Furthermore, since we are building the neural network model in TensorFlow environment, we can use multiple GPUs to train the model to improve the training efficiency. In general, to successfully build up the classification system, GPU server is an indispensable constitution.

Secondly, the limitation of time for the project is also one of the barriers we have encountered. The duration of this project only lasts for 4 months. There are many insufficiencies of our project. Due to the shortage of time, we only build up the system based on convolutional neural networks algorithms. The reason we decided to go with this architecture is that after many background research, CNN is currently regarded as a leading approach to solve relevant problems. However, the project should have concluded comparison between different algorithms to be more completed. In the future of the study, we will try to implement other algorithms such as SVM, K Nearest Neighbours, and Random Forest. It is necessary to have more models of various methods for this topic so that we can compare the behaviour of these models and have a better conclusion.

Thirdly, the quality of fundus images has a significant influence on the model performance. The model cannot accurately detect features on low-quality images. As discussed above, the higher the quality of the retinal images, the better the results. This is also a limitation of our model. In order to improve the model, relevant image pre-processing techniques can be applied to enhance the features. Sopharak and his colleagues suggested a mathematical morphology method to detect DR. It uses mathematical morphological equations to create marker images. The marker images can help to enhance the existence of exudates by applying different mathematical morphological equations (Sophrak, 2008). They used this method in exudates detection. From Figure 9.1, it can be found that relevant features are more obvious when compared to the initial image. We can use this method as an image preprocessing technique in our model in the future and see whether it can improve the model performance.

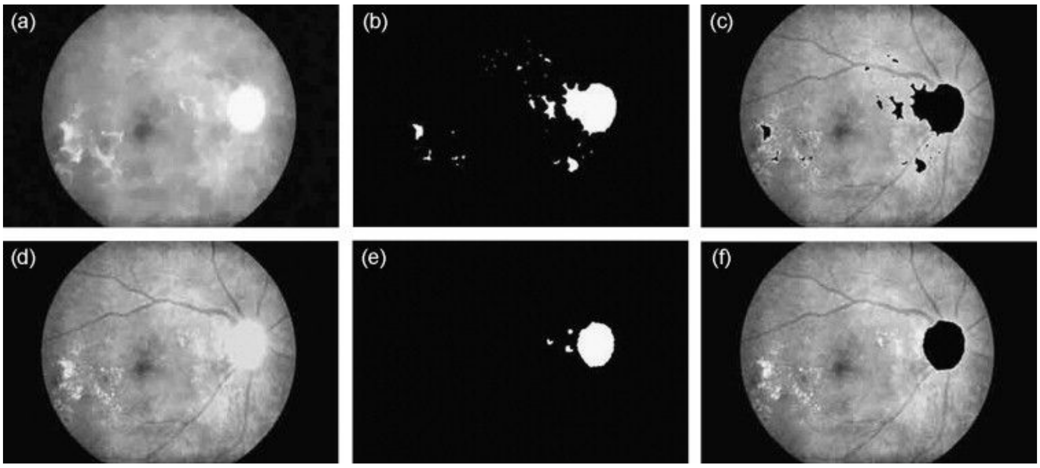


Figure 9.1: (a) Intensity image after closing, (b) thresholded image, (c) marker image, (d) reconstructed image, (e) thresholded result of difference image and (f) optic disc area eliminated from the contrast enhanced image. (Sophorak, 2008)

Besides preprocessing techniques, the development of retinal screening tools also can also boost the performance of our model. Since the initial motivation of this study is to scale the efforts of ophthalmologists through technology, especially in some remote areas, the popularity of retinal imaging techniques should also be considered. Retinal images of our dataset comes from

EyePACS, which is a free platform for diabetic retinopathy screening. It is equipped with many advanced cameras in retinal screening. It is strongly believed that with the development of screening techniques, the model's performance will improve as well.

Fourthly, our prediction model was based on inspection of central retina. All of our input image data are center-based. However, as indicated by Arcadu, lateral areas of a retina may also contain predictive information of DR before pathology has led to abnormalities of fundus (Arcadu, 2019). Ultrawide Field (UWF) images have an excellent result in detecting haemorrahges and macular within early treatment diabetic retinopathy study (ETDRS). UWF can identify 49.8% more haemorrahges and maculars than ETDRS photography, which provides a more accurate result (Pablo, 2017). Therefore, in our system, we can apply relevant techniques to enhance the prediction results.

Moreover, when the reliability of the system is considered, it is worth thinking limitation that their judgement may not be 100% correct. Therefore, this raises a question about how this system can outperform diagnosis from orphthologists. Our given dataset is based on the diagnosis from experts in this field. However, there is a question that how our system can outperform human detection. One idea that we have in mind is to use generative adversarial network (GAN) (Goodfellow et al., 2014) to generate retinal images and learn features from them. This method of GANs makes promising resources in the field of copying data and can preserve patients' privacy in some extend (Yi, X., et.al., 2019). To apply GANs' techniques in our model, we can first separate each level as an individual. Then for each severity level, we can use GAN to generate images of the same level and let the model to learn features from those generated images. This method can cast the system off restrictions from manual diagnosis of ophthalmologists.

Regarding the limitations of the project, there are still many scopes that we can improve. At the beginning of the project, we have set up several research questions for our study, but we only accomplish few of them due to the limitation of time. Hence, in the future of our study, we are going to extend the study into a deeper field including the following based on our original research questions:

- ❏ Most patients of diabetic retinopathy have different severity levels for left and right eyes. It is worth studying the relationship of lesions between two eyes. In other words, does the severity level of one eye influence another eye for an individual?
- ❏ Diabetic retinopathy can evolve into several abnormalities of the retina. Some of the features are more easily detected when compared to others. For example, exudates with colour properties are more obvious while microaneurysms and "cotton wool" are more difficult to segment. This raises the problem of whether the difficulty of recognizing these features leads to an issue in DR detection. We can enhance those inconspicuous features of abnormalities in the preprocessing section in our automated detection system to improve the model's performance.

There is a variability of the colors of fundus worldwide. The dataset is mainly adopted from a single ethnicity and mainly include retinal of a single color. In our experiments, we tested and validated the model performance based on the given dataset. However, in order to popularize the reliability of the system universally, we can adopt dataset from different races and generalize the application of the system multi-ethnically. This can help testing the results of our system in fundus of different colours. This idea is inspired by Li and his fellows. They have built up a deep learning algorithm (DLA) for the detection of DR and validated their results with images from population-based dataset of Caucasian Australians, Indigenous Australians and Malays. In their artificial intelligence-based DLA, the results seem no difference (Li, et al., 2018). In spite of that, we still need to validate our system multi-racially to generate the scientific credibility.

When taking into consideration of the real-world applications of the system, it is far more than just building up the model with a satisfactory result. Currently, this is still a research topic instead of a practical application. In real-life cases, expenses, operability and accessibility are all considerable elements. There are many approaches that we can do to put it into practice. For example, in the hospital, we can have detecting instruments to identify DR.

In spite of professional instruments in areas of expertise, the initial motivation of our project is to enhance the convenience in the detection of DR. At the 2019 annual meeting of The Association of Research in Vision and Ophthalmology, researchers came up with the idea of the combination of a smartphone-based device which can take high-quality retinal images and artificial intelligence software to detect DR (Zalewski, 2019). To concentrate on the most cost-effective cases, devices that exceed $10000, weigh over 5 pounds or are nonportable are not considered (Micheletti, 2016). Traditional retinal cameras are expensive while smartphone-based platform is cheaper. Therefore, smartphone-based devices, “all-in-one” devices and related technological equipment are encouraging to accommodate in the future.

In addition, web-based screening for DR can also be developed. As indicated by Michael, the web detection technology is based on an online web server system using Java programming language including image processing, data security and data storage. All patients’ personal information are stored on a need-to-know basis and only individuals with access can reach related information. This can protect patients’ privacy to some extent. The protocol also have a brief questionnaire, visual acuity measurement and several retinal photographs to grasp the retinal images (Michael, 2005). This is a method that we use for reference in further applications.

Our project is a medical related topic. In our current model, we classify the input dataset based on the overall performance of all features. In the medical field, simply predicting the severity level of DR is far more than enough. Ophthalmologists need to identify the damaged retinal areas. Schlemper and his fellows proposed a novel attention gate (AG) model for medical image analysis which can learn features of target structures by itself. AGs can be easily integrated into CNNs. AG models can be evaluated on various tasks including medical segmentation and image classification. It can suppress irrelevant features and highlight the salient features of an input image. Figure 9.2 shows the mechanism of AG (Schlemper, 2019). We can apply similar techniques in our model by including AGs in our CNN model. It can highlight the features leading to the prediction level in the image. The highlighted areas can help ophthalmologists to diagnose the major retinal abnormalities and apply suitable treatment.

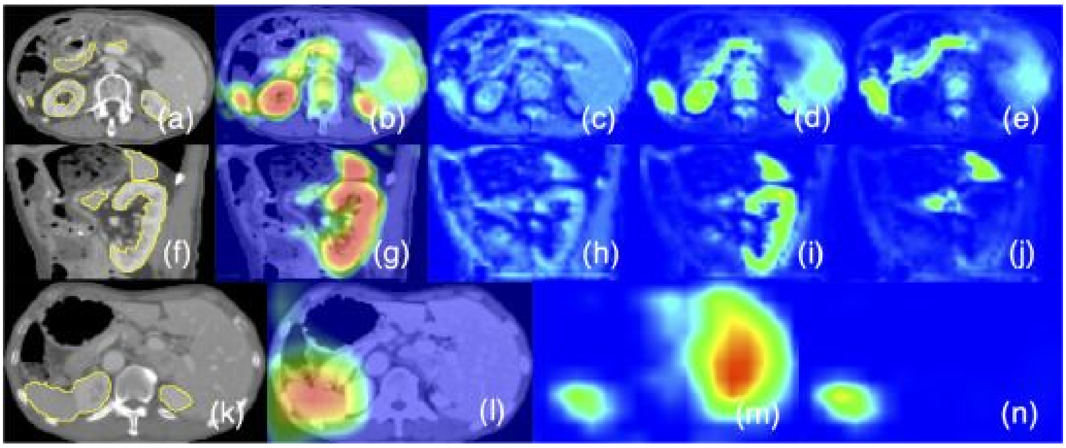


Figure 9.2: Axial (a) and sagittal (f) views of a 3D CT scan, (b,g) attention coefficients, image feature activations before (c,h) and after attention gating (d,e,i,j). Similarly, (k-n) visualise the gating on a coarse scale skip connection. The filtered feature activations (d,e,i,j) are collected from multiple AGs, where a subset of organs is selected by each gate and activations consistently correspond to specific structures across different scans. (Schlemper, 2019)

10. Conclusion

In this project, we proposed an automated detection system of diabetic retinopathy using deep learning algorithms based on retinal images. The proposed system overall have shown an outstanding performance to identify normal, mild, moderate, severe and PDR in a time-efficient and cost-economic manner. It has shown a good result - 91% accuracy, at a speed of 0.06s per image. It is focused on reliable detection of retinal abnormalities. The result of this model has shown a potential to help ophthalmologists to detect DR automatically at an early stage to prevent late treatment. In the future, further research can be invested on our model to produce a more robust and accurate algorithm to automatically detect DR.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our client, Dr Matloob Khushi, for his invaluable guidance and support throughout the course of this research. Meanwhile, we are also grateful to our tutor, Tin Lai, for providing us with valuable and selfless advice. Our capstone project would not have accomplished successfully without their general encouragement and help.

References

American Psychological Association (APA). (2010). *Publication Manual of the American Psychological Association* (6th Ed.). Washington, DC: Author.

Abramoff, M. D., & Suttorp-Schulten, M. S. (2005). Web-based screening for diabetic retinopathy in a primary care population: the EyeCheck project. *Telemedicine Journal & e-Health*, *11*(6), 668-674.

Agurto, C., Murray, V., Barriga, E., Murillo, S., Pattichis, M., Davis, H., ... & Soliz, P. (2010). Multiscale AM-FM methods for diabetic retinopathy lesion detection. *IEEE transactions on medical imaging*, *29*(2), 502-512.

Akram, M. U., Khalid, S., & Khan, S. A. (2013). Identification and classification of microaneurysms for early detection of diabetic retinopathy. *Pattern Recognition*, *46*(1), 107-116.

American Diabetes Association. (2013). Standards of medical care in diabetes—2013. *Diabetes care*, *36*(Supplement 1), S11-S66.

Arcadu, F., Benmansour, F., Maunz, A., Willis, J., Haskova, Z., & Prunotto, M. (2019). Deep learning algorithm predicts diabetic retinopathy progression in individual patients. *NPJ digital medicine*, *2*(1), 1-9.

Asia Pacific Tele-Ophthalmology Society. (2019, June). *APTOS 2019 Blindness Detection.* Retrieved from https://www.kaggle.com/c/aptos2019-blindness-detection/

Atun, R. (2015). Transitioning health systems for multimorbidity. *The Lancet*, *386*(9995), 721-722.

California Healthcare Foundation (2015, February). *Diabetic Retinopathy Detection*. Retrieved from https://www.kaggle.com/c/diabetic-retinopathy-detection/

Chakrabarti, R., Harper, C. A., & Keeffe, J. E. (2012). Diabetic retinopathy management guidelines. *Expert Review of Ophthalmology*, *7*(5), 417-439.

Chen, Y., Jiang, H., Li, C., Jia, X., & Ghamisi, P. (2016). Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(10), 6232-6251.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251-1258).

ETDRSR Group. (1991). Grading diabetic retinopathy from stereoscopic color fundus photographs: An extension of the modified air-lie house classification, etdrs report number 10. *Ophthalmology*, *98*, 786-806.

Fadzil, M. A., Izhar, L. I., Nugroho, H., & Nugroho, H. A. (2011). Analysis of retinal fundus images for grading of diabetic retinopathy severity. *Medical & biological engineering & computing*, *49*(6), 693-700.

Frank, R. N. (1984). On the pathogenesis of diabetic retinopathy. *Ophthalmology*, *91*(6), 626-634.

Friedman, D. S., Ali, F., & Kourgialis, N. (2011). Diabetic retinopathy in the developing world: how to approach identifying and treating underserved populations. *American journal of ophthalmology*, *151*(2), 192-194.

Gardner, G. G., Keating, D., Williamson, T. H., & Elliott, A. T. (1996). Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool. *British journal of Ophthalmology*, *80*(11), 940-944.

Gogul, I., & Kumar, V. S. (2017, March). Flower species recognition system using convolution neural networks and transfer learning. In *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)* (pp. 1-6). IEEE.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., ... & Wang, Y. (2017). Artificial intelligence in healthcare: past, present and future. *Stroke and vascular neurology*, *2*(4), 230-243.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

LaRosa, E., & Danks, D. (2018, December). Impacts on trust of healthcare AI. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 210-215). ACM.

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

Li, Z., Keel, S., Liu, C., He, Y., Meng, W., Scheetz, J., ... & Taylor, H. (2018). An automated grading system for detection of vision-threatening referable diabetic retinopathy on the basis of color fundus photographs. *Diabetes care*, *41*(12), 2509-2516

Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.

McLeod, D. (2005). Why cotton wool spots should not be regarded as retinal nerve fibre layer infarcts. *British Journal of Ophthalmology*, *89*(2), 229-237.

Micheletti, J. M., Hendrick, A. M., Khan, F. N., Ziemer, D. C., & Pasquel, F. J. (2016). Current and next generation portable screening devices for diabetic retinopathy. *Journal of diabetes science and technology*, *10*(2), 295-300.

Nayak, J., Bhat, P. S., Acharya, R., Lim, C. M., & Kagathi, M. (2008). Automated identification of diabetic retinopathy stages using digital fundus images. *Journal of medical systems*, *32*(2), 107-115.

Osareh, A., Shadgar, B., & Markham, R. (2009). A computational-intelligence-based approach for detection of exudates in diabetic retinopathy images. *IEEE Transactions on Information Technology in Biomedicine*, *13*(4), 535-545.

Panch, T., Szolovits, P., & Atun, R. (2018). Artificial intelligence, machine learning and health systems. *Journal of global health*, *8*(2).

Patz, A. (1980). Studies on retinal neovascularization. Friedenwald Lecture. *Investigative ophthalmology & visual science*, *19*(10), 1133-1138.

Schachat, A. P., Hyman, L., Leske, M. C., Connell, A. M. S., Hiner, C., Javornik, N., & Alexander, J. (1993). Comparison of diabetic retinopathy detection by clinical examinations and photograph gradings. *Archives of ophthalmology*, *111*(8), 1064-1070.

Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., & Rueckert, D. (2019). Attention gated networks: Learning to leverage salient regions in medical images. *Medical image analysis*, *53*, 197-207.

Silva, P. S., El-Rami, H., Barham, R., Gupta, A., Fleming, A., van Hemert, J., ... & Aiello, L. P. (2017). Hemorrhage and/or microaneurysm severity and count in ultrawide field images and early treatment diabetic retinopathy study photography. *Ophthalmology*, *124*(7), 970-976.

Singh, R., Ramasamy, K., Abraham, C., Gupta, V., & Gupta, A. (2008). Diabetic retinopathy: an update. *Indian journal of ophthalmology*, *56*(3), 179.

Solomon, S. D., Chew, E., Duh, E. J., Sobrin, L., Sun, J. K., VanderBeek, B. L., ... & Gardner, T. W. (2017). Diabetic retinopathy: a position statement by the American Diabetes Association. *Diabetes care*, *40*(3), 412-418.

Sopharak, A., Uyyanonvara, B., Barman, S., & Williamson, T. H. (2008). Automatic detection of diabetic retinopathy exudates from non-dilated retinal images using mathematical morphology methods. *Computerized medical imaging and graphics*, *32*(8), 720-727.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

Williams, R., Airey, M., Baxter, H., Forrester, J. K. M., Kennedy-Martin, T., & Girach, A. (2004). Epidemiology of diabetic retinopathy and macular oedema: a systematic review. *Eye*, *18*(10), 963.

World Health Organization. (2009). *VISION 2020* (No. SEA-Blindness-1). WHO Regional Office for South-East Asia.

World Health Organization. (2018, October 30). *Diabetes*. Retrieved from https://www.who.int/news-room/fact-sheets/detail/diabetes

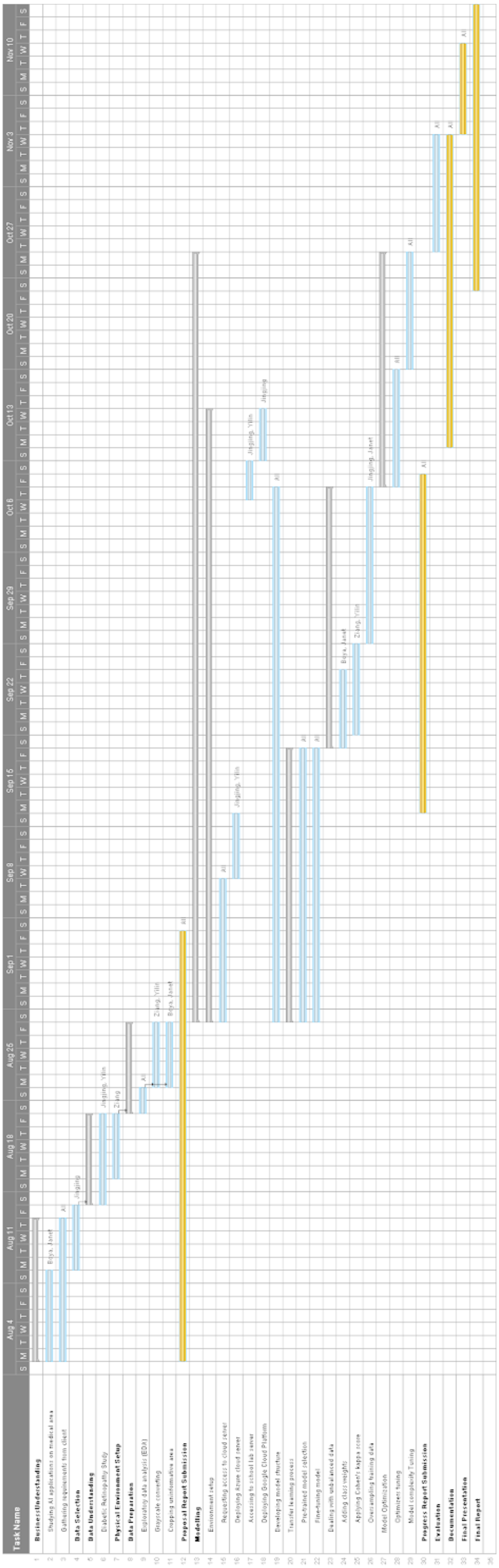
Yi, X., Walia, E., & Babyn, P. (2019). Generative adversarial network in medical imaging: A review. *Medical image analysis*, 101552.

Zachariah, S., Wykes, W., & Yorston, D. (2015). Grading diabetic retinopathy (DR) using the Scottish grading protocol. *Community eye health*, *28*(92), 72.

Zalewski, S. (2019,Apri,29). Earlier Detection of Diabetic Retinopathy with Smartphone AI. *Michigan Medicine*.Retrieved from https://labblog.uofmhealth.org/health-tech/earlier-detection-of-diabetic-retinopathy-smartphone-ai

Zhang, X., Saaddine, J. B., Chou, C. F., Cotch, M. F., Cheng, Y. J., Geiss, L. S., ... & Klein, R. (2010). Prevalence of diabetic retinopathy in the United States, 2005-2008. *Jama*, *304*(6), 649-656.

Appendix - Gantt Chart


Task Name
1 BusinessUnderstanding
2 Studying AI applications on medical area
3 Gathering requirements from client
4 Data Selection
5 Data Understanding
6 Diabetic Retinopathy Study
7 Physical Environment Setup
8 Data Preparation
9 Exploratory data analysis (EDA)
10 Grayscale converting
11 Cropping uninformative area
12 Proposal Report Submission
13 Modelling
14 Environment setup
15 Requesting access to cloud server
16 Deploying Azure cloud server
17 Accessing to school lab server
18 Deploying Google Cloud Platform
19 Developing model structure
20 Transfer learning process
21 Pre-trained model selection
22 Fine-tuning model
23 Dealing with unbalanced data
24 Adding class weights
25 Applying Cohen's kappa score
26 Oversampling training data
27 Model Optimization
28 Optimizers tuning
29 Model complexity Tuning
30 Progress Report Submission
31 Evaluation
32 Documentation
33 Final Presentation
34 Final Report
Aug 4
Aug 11
Aug 18
Aug 25
Sep 1
Sep 8
Sep 15
Sep 22
Sep 29
Oct 6
Oct 13
Oct 20
Oct 27
Nov 3
Nov 10
S M T W T F S
Beiya, Janet
All
Jingling
Jingling, Yilin
Zhang
All
Zhang, Yilin
Beiya, Janet
All
All
Jingling, Yilin
All
Jingling, Yilin
Jingling
All
All
All
Beiya, Janet
Zhang, Yilin
Jingling, Janet
All
All
All
All
All
All