**THE UNIVERSITY OF DANANG**

**DANANG UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**FACULTY OF INFORMATION TECHNOLOGY** 

**GRADUATION PROJECT THESIS**

**MAJOR: INFORMATION TECHNOLOGY**

**SPECIALTY: …………………………………**

**PROJECT TITLE:**

**BUILDING AN APPLICATION FOR**

**DIABETIC RETINOPATHY DIAGNOSIS AID**

Instructor: **NINH KHANH DUY, Ph.D**

Student: **TRAN THI THANH THAO**

Student ID: **102160195**

Class: **16TCLC1**

**Da Nang, 12/2020**

{Trang trắng này dùng để dán bản Nhận xét của người hướng dẫn, hoặc thay trang này bằng Nhận xét của người hướng dẫn}

{Trang trắng này dùng để dán bản Nhận xét của người phản biện, hoặc thay trang này bằng Nhận xét của người phản biện}

**SUMMARY**

Topic title: Building a diabetic retinopathy diagnosis aid application for Vietnamese.

Student name: Tran Thi Thanh Thao

Student ID: 102160195 Class: 16TCLC1

{Nội dung tóm tắt trình bày tối đa trong 1 trang} {Font: Time New Roman; thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified}

………………………………………………………………………………………… ..

………………………………………………………………………………………… ..

………………………………………………………………………………………… ..

………………………………………………………………………………………… ..

………………………………………………………………………………………… ..

………………………………………………………………………………………… ..

………………………………………………………………………………………… ..

|  |  |
| --- | --- |
| DA NANG UNIVERSITY  **UNIVERSITY OF SICIENCE AND TECHNOLOGY**  FALCUTY………………………………………… | **THE SOCIALIST REPUBLIC OF VIETNAM**  Independence - Freedom - Happiness |

**GRADUATION PROJECT REQUIREMENTS**

Student Name: Tran Thi Thanh Thao Student ID: 102160195

Class: 16TCLC1 Faculty: Information Technology Major:

1. *Topic title:*

*Building an application for diabetic retinopathy diagnosis aid*

1. *Project topic:* ☐*has signed intellectual property agreement for final result*
2. *Initial figure and data:*

……………………………………………………………………………….….…….……......…………………………………………………………………………….………………………………………………………………………………………………...…

*Content of the explanations and calculations:*

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

1. *Drawings, charts (specify the types and sizes of drawings):*

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

|  |  |
| --- | --- |
| 1. *Name of instructor:* | *Content parts:* |
|  |  |
|  |  |
|  |  |

1. *Date of assignment: ……../……./201…..*
2. *Date of completion: ……../……./201…..*

|  |  |
| --- | --- |
|  | *Da Nang, date month year 201* |
| **Head of Division**…………………. | **Instructor** |

# PREFACE

{Font: Time New Roman; thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified} ………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

Note: Student can describe “Acknowledgement” in “Preface”

# ASSURRANCE

We would like to assure you that the project “Building a diabetic retinopathy aid application for Vietnamese” is my research due to the support and guidance of Ninh Khanh Duy, Ph. D. All references, used in this graduation project, are cited with the name of the author, the name of the journal, or the study, and also have been listed in the reference section.

The data and the results of the project are completely honest. If there are invalid copies, which violate training regulations, I will accept all responsibilities.

Student Performed

Tran Thi Thanh Thao

# TABLE OF CONTENT

[PREFACE 1](#_Toc56693816)

[ASSURRANCE 2](#_Toc56693817)

[TABLE OF CONTENT 3](#_Toc56693818)

[LIST OF PICTURES 6](#_Toc56693819)

[LIST OF TABLES 8](#_Toc56693820)

[LIST OF SYMBOLS, ACRONYM 10](#_Toc56693821)

[INTRODUCTION 11](#_Toc56693822)

[1. Introduction 11](#_Toc56693823)

[1.1. Purpose 11](#_Toc56693824)

[1.2. Significance 11](#_Toc56693825)

[2. Scope 11](#_Toc56693826)

[3. Method 12](#_Toc56693827)

[4. Structure 12](#_Toc56693828)

[Chapter 1: OVERVIEW 13](#_Toc56693829)

[1.1. Context 13](#_Toc56693830)

[1.2. Related work 14](#_Toc56693831)

[1.3. Application structure 16](#_Toc56693832)

[1.4. Conclusion 17](#_Toc56693833)

[Chapter 2: THEORETICAL FOUNDATION AND TECHNOLOGY 18](#_Toc56693834)

[2.1. Machine learning 18](#_Toc56693835)

[2.1.1. Definition 18](#_Toc56693836)

[2.1.2. Machine learning algorithms category 18](#_Toc56693837)

[2.1.3. Machine learning life cycle 19](#_Toc56693838)

[2.2. Artificial neural network 20](#_Toc56693839)

[2.2.1. Definition 20](#_Toc56693840)

[2.2.2. Architecture 20](#_Toc56693841)

[2.3. Deep neural network 21](#_Toc56693842)

[2.3.1. Deep learning 21](#_Toc56693843)

[2.3.2. Convolutional neural network 22](#_Toc56693844)

[a. Definition 22](#_Toc56693845)

[b. Architecture 22](#_Toc56693846)

[2.3.3. EfficientNet 25](#_Toc56693847)

[2.3.4. ResNet 26](#_Toc56693848)

[2.3.5. DenseNet 27](#_Toc56693849)

[2.6. Technology 28](#_Toc56693850)

[2.6.1. Python 28](#_Toc56693851)

[a. Definition 28](#_Toc56693852)

[b. Characteristics 28](#_Toc56693853)

[c. Flask Python 28](#_Toc56693854)

[2.6.2. JavaScript 29](#_Toc56693855)

[Chapter 3: SYSTEM DESIGN ANALYSIS 30](#_Toc56693856)

[3.1. Application overview 30](#_Toc56693857)

[3.2. System design analysis 30](#_Toc56693858)

[3.2.1. Object analysis 30](#_Toc56693859)

[3.2.2. System diagram 30](#_Toc56693860)

[a. Activity diagram 30](#_Toc56693861)

[b. Use case 31](#_Toc56693862)

[3.2.3. Define data model 32](#_Toc56693863)

[3.4. Conclusion 32](#_Toc56693864)

[Chapter 4: EXPERIMENT AND RESULT 33](#_Toc56693865)

[4.1. Solution 33](#_Toc56693866)

[4.1.1. Environment and machine learning framework 33](#_Toc56693867)

[4.1.2. Training process 34](#_Toc56693868)

[4.1.3. Imbalanced data 35](#_Toc56693869)

[a. Data collection 35](#_Toc56693870)

[b. Solution 37](#_Toc56693871)

[4.1.4. Feature extraction in the data processing 38](#_Toc56693872)

[4.1.5. Evaluation method 40](#_Toc56693873)

[a. Confusion matrix 40](#_Toc56693874)

[b. Cohen’s kappa score 41](#_Toc56693875)

[4.2. Training model implementation 42](#_Toc56693876)

[4.2.1. Overview of model training and results prediction 42](#_Toc56693877)

[4.2.2. Model with balanced dataset 43](#_Toc56693878)

[a. Data 43](#_Toc56693879)

[b. Model architecture 43](#_Toc56693880)

[c. Training 46](#_Toc56693881)

[d. Result 46](#_Toc56693882)

[4.2.3. Model with imbalanced dataset 47](#_Toc56693883)

[a. Data 47](#_Toc56693884)

[b. Model architecture 48](#_Toc56693885)

[c. Training 52](#_Toc56693886)

[d. Result 53](#_Toc56693887)

[4.2.4. Improving accuracy 57](#_Toc56693888)

[a. Data 57](#_Toc56693889)

[b. Model architecture 59](#_Toc56693890)

[c. Training 59](#_Toc56693891)

[d. Result 60](#_Toc56693892)

[4.2.5. Conclusion 70](#_Toc56693893)

[4.3. Server development 71](#_Toc56693894)

[4.3.1. Dependency 71](#_Toc56693895)

[4.3.2. Implementation 71](#_Toc56693896)

[a. Keycloak 71](#_Toc56693897)

[b. Application 71](#_Toc56693898)

[4.4. Evaluation result 72](#_Toc56693899)

[4.5. Conclusion 72](#_Toc56693900)

[CONCLUSION 73](#_Toc56693901)

[1. Achievement 73](#_Toc56693902)

[2. Disadvantage 73](#_Toc56693903)

[3. Future development 73](#_Toc56693904)

[REFERENCES 75](#_Toc56693905)

[APPENDIX1 1](#_Toc56693906)

# LIST OF PICTURES

Picture 1.1 Application structure 16

Picture 2.1 The relationship between machine learning and AI 18

Picture 2.2 Neural network architecture 21

Picture 2.3 Neural network architecture with hidden layers 21

Picture 2.4 The difference between deep learning and machine learning 22

Picture 2.5 Convolutional neural network architecture 23

Picture 2.6 Example of convolutional layer 23

Picture 2.7 Example of pooling layer 24

Picture 2.8 Example of fully-connected layer 25

Picture 2.9 Comparison EfficientNets with other existing CNNs on ImageNet 26

Picture 2.10 Structure of a basic residual block 27

Picture 2.11 Structure of a basic dense blocks 27

Picture 2.12 Python logo 28

Picture 2.13 Flask Python logo 29

Picture 2.14 HTML, CSS and JavaScript logo 29

Picture 3.1 Activity diagram 30

Picture 3.2 Use case 31

Picture 3.3 Application data model 32

Picture 4.1 Fine tuning in transfer learning 34

Picture 4.2 Classes distribution in EyePACs dataset 35

Picture 4.3 Classes distribution in APTOS2019 dataset 36

Picture 4.4 Classes distribution in selective dataset 36

Picture 4.5 The sample fundoscopy photography images from dataset 37

Picture 4.6 Data before feature extraction 39

Picture 4.7 Data after feature extraction 39

Picture 4.8 Overview of model training and result prediction 42

Picture 4.9 Data distribution for training, validation and test 43

Picture 4.10 Class distribution in training dataset 47

Picture 4.11 Class distribution in test dataset 48

Picture 4.12 Confusion matrix of DenseNet169 - 1 evaluated on test dataset 54

Picture 4.13 Confusion matrix of DenseNet169 - 2 evaluated on test dataset 54

Picture 4.14 Confusion matrix of EfficientNetB5 - 1 evaluated on test dataset 55

Picture 4.15 Confusion matrix of EfficientNetB5 - 2 evaluated on test dataset 55

Picture 4.16 Confusion matrix of ResNet50 - 1 evaluated on test dataset 56

Picture 4.17 Confusion matrix of ResNet50 - 2 evaluated on test dataset 56

Picture 4.18 Class distribution in imbalanced test dataset 58

Picture 4.19 Class distribution in balanced test dataset 59

Picture 4.20 Confusion matrix of fold-0 model on train dataset 60

Picture 4.21 Confusion matrix of fold-0 model on validation dataset 61

Picture 4.22 Confusion matrix of fold-0 model on imbalanced test dataset 61

Picture 4.23 Confusion matrix of fold-1 model on train dataset 62

Picture 4.24 Confusion matrix of fold-1 model on validation dataset 62

Picture 4.25 Confusion matrix of fold-1 model on imbalanced test dataset 63

Picture 4.26 Confusion matrix of fold-2 model on train dataset 63

Picture 4.27 Confusion matrix of fold-2 model on validation dataset 64

Picture 4.28 Confusion matrix of fold-2 model on imbalanced test dataset 64

Picture 4.29 Confusion matrix of fold-3 model on train dataset 65

Picture 4.30 Confusion matrix of fold-3 model on validation dataset 65

Picture 4.31 Confusion matrix of fold-3 model on imbalanced test dataset 66

Picture 4.32 Confusion matrix of fold-4 model on train dataset 66

Picture 4.33 Confusion matrix of fold-4 model on validation dataset 67

Picture 4.34 Confusion matrix of fold-4 model on imbalanced test dataset 67

Picture 4.35 Confusion matrix of median model on train dataset 68

Picture 4.36 Confusion matrix of median model on validation dataset 68

Picture 4.37 Confusion matrix of median model on imbalanced test dataset 69

Picture 4.38 Confusion matrix of median model on balanced test dataset 69

[Appendix Picture 1.1 Illustrative image for mild NDR 2](#_Toc56692556)

[Appendix Picture 1.2 Illustrative image for moderate NDR 2](#_Toc56692557)

[Appendix Picture 1.3 Illustrative image for severe NDR 3](#_Toc56692558)

[Appendix Picture 1.4 Illustrative image for severe Proliferative DR 4](#_Toc56692559)

# LIST OF TABLES

[Table 3.1 Actor analysis in system 30](#_Toc56692560)

[Table 4.1 Grade distribution of the diabetic retinopathy images dataset 37](#_Toc56692561)

[Table 4.2 Basic example of confusion matrix 40](#_Toc56692562)

[Table 4.3 Kappa coefficient interpretation 41](#_Toc56692563)

[Table 4.4 Grade distribution of the diabetic retinopathy images in balanced dataset 43](#_Toc56692564)

[Table 4.5 DenseNet169 architecture of the severity grading DR detection model 44](#_Toc56692565)

[Table 4.6 The number of DenseNet169 network parameters 44](#_Toc56692566)

[Table 4.7 EfficientNetB5 architecture of the severity grading DR detection model 44](#_Toc56692567)

[Table 4.8 The number of EfficientNetB5 network parameters 45](#_Toc56692568)

[Table 4.9 ResNet50 architecture of the severity grading DR detection model 45](#_Toc56692569)

[Table 4.10 The number of ResNet50 network parameters 46](#_Toc56692570)

[Table 4.11 The settings of network training hyperparameters 46](#_Toc56692571)

[Table 4.12 Parameters during each step in training process 46](#_Toc56692572)

[Table 4.13 DR classification model with balanced dataset evaluation based on accuracy and loss analysis 47](#_Toc56692573)

[Table 4.14 Threshold distribution in grading diabetic retinopathy 48](#_Toc56692574)

[Table 4.15 DenseNet169 - 1 architecture with imbalanced dataset 49](#_Toc56692575)

[Table 4.16 The number of DenseNet169 - 1 parameters with imbalanced dataset 49](#_Toc56692576)

[Table 4.17 DenseNet169 - 2 architecture with imbalanced dataset 49](#_Toc56692577)

[Table 4.18 The number of DenseNet169 -2 parameters with imbalanced dataset 49](#_Toc56692578)

[Table 4.19 EffdicientNetB5 - 1 architecture with imbalanced dataset 50](#_Toc56692579)

[Table 4.20 The number of EfficientNetB5 - 1 parameters with imbalanced dataset 50](#_Toc56692580)

[Table 4.21 EffdicientNetB5 - 2 architecture with imbalanced dataset 50](#_Toc56692581)

[Table 4.22 The number of EfficientNetB5 - 2 parameters with imbalanced dataset 51](#_Toc56692582)

[Table 4.23 ResNet50 - 1 architecture with imbalanced dataset 51](#_Toc56692583)

[Table 4.24 The number of ResNet50 - 1 parameters with imbalanced dataset 51](#_Toc56692584)

[Table 4.25 ResNet50 - 2 architecture with imbalanced dataset 51](#_Toc56692585)

[Table 4.26 The number of ResNet50 - 2 parameters with imbalanced dataset 52](#_Toc56692586)

[Table 4.27 The settings of networks training hyperparameters 52](#_Toc56692587)

[Table 4.28 DenseNet169 with imbalanced dataset evaluation based on accuracy and loss analysis 53](#_Toc56692588)

[Table 4.29 EfficientNetB5 with imbalanced dataset evaluation based on accuracy and loss analysis 53](#_Toc56692589)

[Table 4.30 ResNet50 with imbalanced dataset evaluation based on accuracy and loss analysis 53](#_Toc56692590)

[Table 4.31 Data distribution in each fold 57](#_Toc56692591)

[Table 4.32 EfficientNetB5 architecture 59](#_Toc56692592)

[Table 4.33 The setting of hyperparameters during training 60](#_Toc56692593)

[Table 4.34 5-fold evaluation based on accuracy and loss analysis 60](#_Toc56692594)

[Table 4.35 5-fold evaluation based on the quadratic weighted kappa 70](#_Toc56692595)

[Table 4.36 List APIs implemented in application 72](#_Toc56692596)

# LIST OF SYMBOLS, ACRONYM

|  |  |  |
| --- | --- | --- |
| **No.** | **Symbol** | **Explanation** |
| 1 | DR | Diabetic Retinopathy |
| 2 | ML | Machine Learning |
| 3 | DL | Deep Learning |
| 4 | CNN | Convolutional Neural Network |
| 5 | AI | Artificial Intelligence |

# INTRODUCTION

## 1. Introduction

In this day and age, diabetic retinopathy has been still diagnosed in a traditionally manual way. There are so many factors affecting this disease diagnosis including time, effort, and qualification. Therefore, using artificial intelligence in the process of diabetic retinopathy diagnosis is a new way to advance the time and quality of healthcare service. It is an important step that helps reducing rates of blindness in the world.

### 1.1. Purpose

Building a diabetic retinopathy diagnosis aid application is one of main the project's purposes.

This application also provides some knowledge about diabetic retinopathy - one of the most common complications of diabetes mellitus. In addition, this is an improvement of medicine, which can reduce time, effort along with the workload on ophthalmologists.

### 1.2. Significance

Diabetic retinopathy diagnosis aid application is one of the first steps in applying artificial intelligence to recognize and support the diagnosis of medical diseases in general and diabetic retinopathy in particular. In addition, the problems during diabetic retinopathy diagnosis can be brought into research and development in the future.

Diabetic retinopathy early detection and treatment can significantly reduce the risk of vision loss. Therefore, this application is a positive contribution to the promotion and improvement efficiency in healthcare, reduce both diseases related to eye and rates of blindness in Viet Nam.

## 2. Scope

Diabetic retinopathy occurs when changes in blood glucose levels cause changes in retinal blood vessels. In some cases, these vessels will swell up (macular edema) and leak fluid into the rear of the eye.

Like many conditions of this nature, the early stages of diabetic retinopathy may occur without symptoms and without pain. An actual influence on the vision will not occur until the disease advances.

Macular edema can result from maculopathy and affect vision occurs if leaking fluid causes the macular to swell. New vessels on the retina can prompt bleeding, which can also block vision in some cases.

Detection and diagnosis one stage of diabetic retinopathy depends on so many factors, including soft and hard exudates, macular edema, microaneurysms, hemorrhages, abnormal vein-artery crossovers, abnormal growth of blood vessels, etc. Each stage has each own symptoms and characteristics, depending on the situation of each patient. Therefore, within the time allowed by the project, the scope of application focuses on 4 characteristics: exudates, macular edema, microaneurysms, and hemorrhages.

The application input data is a retinal image. The ophthalmologist uses special tools to see inside the fundus of the eye and other structures with an ophthalmoscope. It is crucial in determining the health of the optic disc, vitreous humor, and retina. The accuracy of the application depends on the quality of the input image, meaning if the retinal input is clear, there is a high chance that the prediction is accurate. By contrast, if the image input is low quality, the precision is not guaranteed.

## 3. Method

* Deep learning method.
* Meta-analysis from documents and data from various sources (books, magazines, blog sites, forums, ...) method.
* Object-oriented design analysis method.
* Testing and evaluating result method.

## 4. Structure

The report includes the following contents:

*Introduction*

*Chapter 1: OVERVIEW*

*Chapter 2: THEORETICAL FOUNDATION AND TECHNOLOGY*

*Chapter 3: SYSTEM DESIGN ANALYSIS*

*Chapter 4: EXPERIMENT AND RESULT*

*Conclusion and future development*.

# Chapter 1: OVERVIEW

## 1.1. Context

More recently, human health is a topic that has always been concerned and ameliorated. It can’t be denied that investing in health is a zero-loss investment. The medical treatment is now being advanced in quality to meet the human health protection need.

The eye is one of the five important senses, performing the functions of seeing, observing, capturing images of objects and colors to transfer to the brain for processing and storage. In a little more detail, from a biological perspective, eyes are sensitive parts of the body to environmental influences, and help humans respond appropriately to all changes in surroundings. It's also an organization that supports human communication without language. Through eyes, humans can contact, exchange information instead of words. What is mentioning is that any eye disease significantly influences the human quality of daily life. According to statistics, some diseases often appear on fundoscopy: vitreous hemorrhage, retinal detachment, diabetic retinopathy, age-related macular degeneration, retinal degeneration, retinal vessel occlusion, etc. Retinal diseases are on increase and become the main cause of vision loss, blindness if not treated promptly. These diseases diminish the patient's quality of life. It is worth saying that they can be a society burden if suffering from vision impairment or blindness.

Diabetic retinopathy is one of the retinal diseases affecting greater or less degree in the human eyes as well as health. It's considered as the most popular cause of blindness for people having age less than 50 years. A person having diabetes is more prone to the risk of diabetic retinopathy. It is a systemic disease which is affecting up to 80 percent of humans for more than 10 years. Many researchers acknowledged that 90 percent of diabetic patients could be saved from this disease through early diagnosis. According to the 108 Military Central Hospital survey, there are over 285 million people with diabetes in the world (2010), estimated to be 439 million people with diabetes (2030). In 2017, about 425 million people worldwide have diabetes and this number is approximated to increase to 642 million by 2040. Every year, about 1.8 million people are blind due to diabetic retinopathy, including about 20 percent of people with diabetes who have retinopathy complications in various degrees. If not detected and treated early, the lesions of the disease will be very heavy such as microaneurysms, hemorrhages, soft and hard exudates, and so on. This is not a small but alarming statistic.

Frankly speaking, diagnostic error in the medical industry is a serious risk to quality and safety in healthcare. Major diagnostic errors are found in 10% to 20% of autopsies, suggesting that 40,000 to 80,000 patients die annually in the United States from diagnostic errors. Patient surveys confirm that at least one person in three has firsthand experience with a diagnostic error, and researchers have found that diagnostic errors - not surgical mistakes, or medication overdoses - account for the largest fraction of malpractice claims, the most severe patient harm, and the highest total of penalty payouts. With a view to improving diagnostic quality, time, and effort of traditional diagnostic methods, artificial intelligence technology has been widely implemented. On June 19, 2020, Vingroup Big Data Institute tested the AI-VinDr solution in medical imaging, assisting doctors to give an accurate and fast diagnosis. The solution was implemented in major hospitals such as 108 Military Central Hospital, Hanoi Medical University Hospital, and Vinmec Times City International Hospital. It concentrated on: lung disease diagnosis on chest X-ray and breast cancer diagnosis on a mammogram. This is a vast step forward in the Vietnamese medical industry. Furthermore, a recent study published in the Journal of the United Kingdom National Cancer Institute shows that the AI system has achieved a breast cancer detection accuracy comparable to an average breast radiologist. Both radiologists and the AI system have shown 95 percent confidence intervals. With the ability of AI networks to train themselves continually, there are big chances that their performance will be significantly improved in the nearest future. In other words, an equivalent problem to breast cancer detection - diabetic retinopathy detection solved by artificial intelligence is possible.

## 1.2. Related work

According to a research by Professor Alastair Denniston of University Hospitals Birmingham NHS Foundation Trust [1], AI can effectively diagnose cancer and other diseases as a medical professional. The study examined data from 14 samples, which found that the AI accurately detected 87% of the cases, while the figure of doctors was only 86%. AI also excludes patients without disease with an accuracy rate of up to 93%, slightly higher than medical experts (91%).

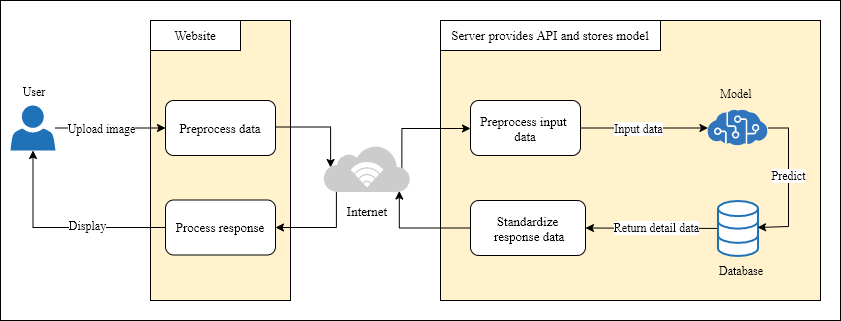
Recently, deep learning (DL) has been widely used in diabetic retinopathy detection and classification. It can successfully learn the features of input data even when many heterogeneous sources are integrated [2]. There are many DL-based methods such as restricted Boltzmann Machines, convolutional neural networks (CNNs), auto encoder, and sparse coding [3].

R. Pires et al. [4] built their own CNN architecture to determine whether an image was referable to DR. The CNN was trained on the Kaggle dataset and was tested by the Messidor-2 and DR2 dataset. The classes of the training dataset were balanced using data augmentation. The work achieved an area under the ROC curve of 98.2% when testing the Messidor-2. The study of H. Jiang et al. [5] integrated three pre-trained CNN models, namely, Inception V3, Inception-Resnet-V2 and Resnet152 to classify their own dataset as referable DR or non-referable DR. In CNNs training, Adam optimizer was used to update their weights. The work obtained an accuracy of 88.21% and area under the curve (AUC) of 0.946. In general, the two studies mentioned above demonstrate the possibility of diabetic retinopathy detection by using deep learning. Nevertheless, the aforementioned researchers who classified DR images into two classes did not consider the five DR stages. The DR stages are important to determine the exact stage of DR to treat the retina with a suitable process as well as to prevent visual deterioration and blindness.

To have a correct assessment to detection of DR, in 2018, X. Wang et al. [6] studied the performance of the three available pre-trained architectures of CNN, VGG16, AlexNet, and InceptionNet V3, to detect the five DR stages in the Kaggle dataset. The dataset only contains 166 images. They reported an average accuracy of 50.03% in VGG16, 37.43% in AlexNet, and 63.23% in InceptionNet V3; however, they trained the networks with a limited number of images, which could prevent the CNN from learning more features and the images required more preprocessing functions to improve them. Also, only one dataset was used to evaluate their study. At the same time, a study conducted by Refs. [7] used CNN to segment the blood vessels in RGB retina images. The images were fed to the CNN for segmentation and to condition a random field model [8] to consider the non-local correlations during segmentation. After that, the vessel map was rebuilt, and morphological operations were applied. Their CNN contains 16 CONV layers and five dilated CONV layers. The STARE [9], HRF [10], DRIVE [11], and CHASE DB1 [12] datasets contain 20, 45, 40, and 28 images, respectively. Accuracy of 0.9634, 0.9628, 0.9608, and 0.9664 was achieved for the DRIVE, STARE, HRF, and CHASE DB1, respectively.

All of the studies mentioned in the current work manipulated the diabetic retinopathy screening system using deep learning techniques. The need for reliable diabetic retinopathy screening systems became a critical issue recently due to the increase in the number of diabetic patients. Using DL in DR detection and classification overcomes the problem of selecting reliable features for ML; on the other hand, it needs a huge data size for training. Most studies used large numbers of images and need to apply data augmentation to increase them. Covering the current work, the studies used public datasets, a combination of two or more public datasets to overcome the problem of data size and to evaluate the DL methods on many datasets. One of the limitations of the usage of deep learning with the medical field is the size of the datasets needed to train the DL systems, as DL is required a large amount of data. The results of DL systems depend heavily on the size of the training data as much as its quality and balance its classes. There is no denial to say that data imbalance is considered a serious problem in this study, arising very often in practice in classification problems in general and diabetic retinopathy classification in particular. A dataset is considered to be imbalanced if one of its classes plays a huge dominance over the rest of the classes. Another important thing is, the overfitting phenomenon can occur easily because the model tries to fit data in the dominant class. Definitely, the amount of people without diabetic retinopathy is higher than the number of ones with diabetic retinopathy and multiple times higher than the one in each DR stage. Another data imbalance problem that is quite famous is Credit Card Fraud Detection. The study [13] conducted by experts from University of Novi Sad indicates that the dataset of this popular problem contains 284,807 transactions where 492 transactions were frauds and the rest were genuine. Considering the numbers, we can see that this dataset is highly imbalanced, where only 0.173% of transactions are labeled as frauds. They also showed that this problem was solved by using Naive Bayes, logistic regression, and artificial neural networks. Hence, viewed from all aforementioned sides, diabetic retinopathy detection with imbalanced dataset is a possibly solved problem.

## 1.3. Application structure



Picture 1.1 Application structure

The figure 1.1 above describes the user's manipulation with the application. The system of application will take over the following works:

* Web application collects user's data.
* Process and send requests to the server containing APIs.
* The server receives the request from the Web application.
* Standardization of input and output data.
* Trained model predicts disease through data users uploaded.
* Retrieve information from the database and respond.
* Web application receives response and displays the result.

The model will be stored on the server. Processing of the submitted data from the user, making predictions, and returning the resulting data to the user will also be taken over on the server. The web application is responsible for displaying, processing user actions and sending them to the server. The application architecture consists of 3 parts:

* Web application.
* The server provides the API.
* Model for DR diagnosis aid.

## 1.4. Conclusion

Chapter 1 provides readers with the disadvantages of traditional diagnostic methods and the importance of applying artificial intelligence in the diagnosis of diabetic retinopathy in Vietnam. Therefore, building a diabetic retinopathy diagnosis aid application is the first step of using artificial intelligence in research and development of the medical industry.

Chapter 1 also shows the scope and the goal to be achieved when building this application for the Vietnamese.

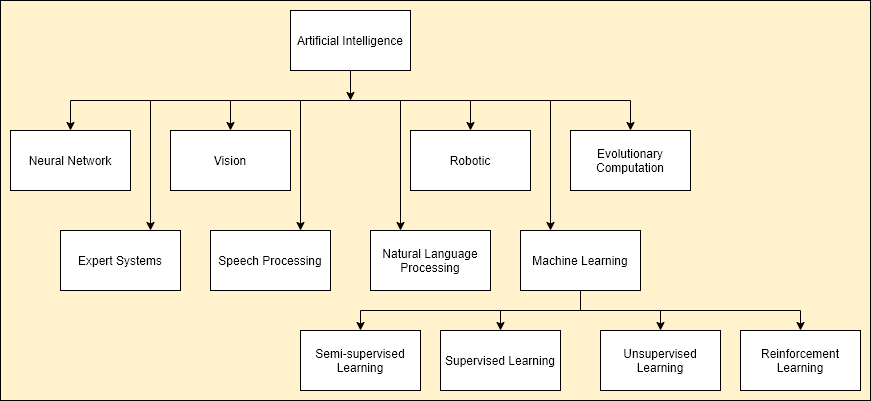
# Chapter 2: THEORETICAL FOUNDATION AND TECHNOLOGY

## 2.1. Machine learning

### 2.1.1. Definition

Machine learning is a field of ​​artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. In a little more detail, machine learning is a method for creating computer programs by analyzing data sets. Machine learning is hugely related to statistics, as both fields study data analysis, but different from statistics, Machine learning focuses on the complexity of algorithms in performing the computation.

Machine learning is highly applicable in data tracing, diagnostic in the medical industry, detecting fake credit cards, analyzing the stock market, speech and writing recognition, etc.



Picture 2.1 The relationship between ML and AI

### 2.1.2. Machine learning algorithms category

Machine learning algorithms are classified according to the desired outcome of the algorithm. Common types of algorithms include:

* **Supervised machine learning algorithms**: can apply what has been learned in the past to new data using labeled data to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The supervised machine learning algorithms are those algorithms that need external assistance. The input dataset is divided into train and test datasets. The training dataset has an output variable that needs to be predicted or classified. All algorithms learn some kind of patterns from the training dataset and apply them to the test dataset for prediction or classification. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
* **Unsupervised machine learning algorithms**: on the contrary, are used when the data used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.
* **Semi-supervised learning algorithms**: is a technique which combines the power of both supervised and unsupervised learning. It can be fruit-full in those areas of machine learning and data mining where the unlabeled data is already present and getting the labeled data is a tedious process. They use both labeled and unlabeled data for training, typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it or learn from it.
* **Reinforcement learning**: is a type of learning which makes decisions based on which actions to take such that the outcome is more positive. The learner has no knowledge which actions to take until it’s been given a situation. The action which is taken by the learner may affect situations and their actions in the future. This method allows developing a self-sustained system that, throughout contiguous sequences of tries and fails, improves itself based on the combination labeled data and interactions with the incoming data.

Machine learning enables the analysis of massive quantities of data. Although it generally gives faster, more accurate results, it may also require additional time and resources to train it properly. Combining machine learning with AI and other technologies can make it even more effective in processing large volumes of information.

### 2.1.3. Machine learning life cycle

Five major steps in machine learning life cycle: problem definition, data gathering, data preparation, train model and evaluate model. Any machine learning problem requires data to train, we can consider it as a prerequisite. Data after collecting needs:

* **Normalization**: All input data need to be standardized for the computer to be able to process. The normalization process includes digitizing data, scaling parameters to suit the problem. This normalization directly affects training speed as well as training efficiency. Specifically, we will discuss this in another article.
* **Split**: The fact that the model is selected matches very well with existing data sets. Nevertheless, there may be a situation where real data does not match. This problem in machine learning is known as overfitting. Therefore, when training, one must divide the data into 3 categories to partially verify the model's generality. They are:
* Training dataset: is used for training.
* Validation dataset: is used to verify data when training.
* Test dataset: is used for evaluate model performance after training.

## 2.2. Artificial neural network

### 2.2.1. Definition

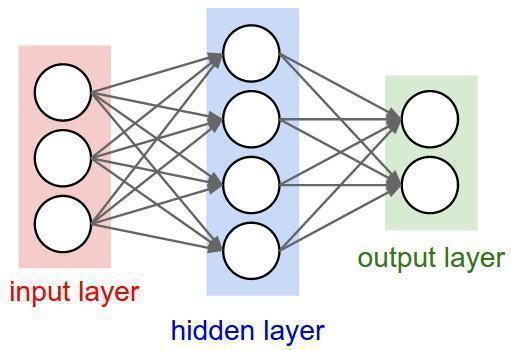
Artificial neural network or neural network for short (ANN) is a mathematical model or a computational model built by biological neural networks. It consists of a group of artificial neural (nodes) connected together and processing information by transmitting on the connections and calculating new values at the nodes. In many cases, an artificial neural network is an adaptive system that changes its own structure based on external or internal information flowing through the network during learning. ANN has been in existence long ago, but has only recently come into light under Artificial Intelligence due to the applications that make it more preferable. These include: image processing, language processing and translation, route detection, speech recognition and forecasting.

ANN is currently being used to solve many complex problems and the demand is increasing with time. The wide number of applications starting from face recognition to making decisions are being handled by neural networks. The more it is exposed to real-time examples, the more it adapts. ANN are capable of learning from faults thereby increasing its capacity to perform well. Hence, neural networks are being preferred more for complex problem-solving.

### 2.2.2. Architecture

ANN is simply made from many layers. Three major ANN's layers including:

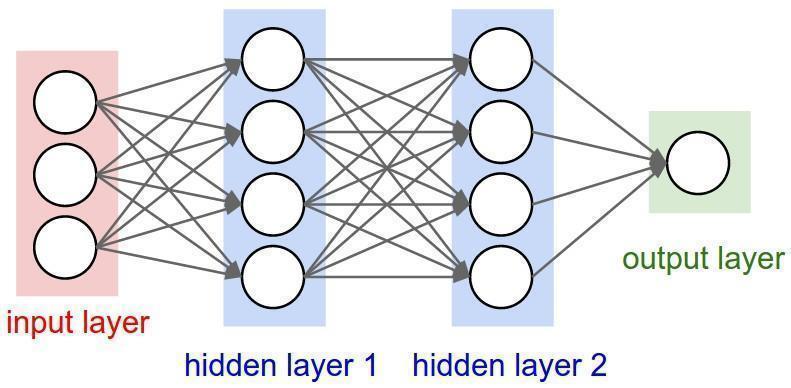
* Input layer: includes the network's input data.
* Hidden layer: comprises one or more layers, these layers act as logic and inference of the network.
* Output layer: Displays the network's output.



Picture 2.2 Neural network architecture

Between the layers, there are so many connections. In each layer, there can be different numbers of neurons as well as connections. If putting in input values, there will be a corresponding output. The neural network will then act as a black box, simply receiving input and giving output.

A neuron has only 1 input and 1 output layer but can have many hidden layers.



Picture 2.3 Neural network architecture with hidden layers

## 2.3. Deep neural network

### 2.3.1. Deep learning

Deep learning is part of machine learning methods based on artificial neural networks. Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. Deep learning has enabled many practical applications of machine learning and extended AI's subfields. These methods have dramatically improved state-of-the-art speech recognition, visual object recognition, object detection, and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep learning has also brought about breakthroughs in processing images, video, speech, and audio.



Picture 2.4 The difference between deep learning and machine learning

### 2.3.2. Convolutional neural network

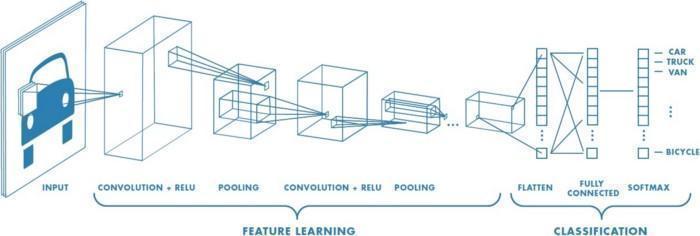
#### a. Definition

Convolutional neural networks are one of the most advanced deep learning models. They have applications in image and video recognition, recommendation systems, image classification, medical image analysis, natural language processing, financial time series and more. Large companies such as Facebook, Google, or Amazon have even used them in their systems for better customer satisfaction.

CNN is now the go-to model on every image related problem. In terms of accuracy, they blow the competition out of the water. The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision.

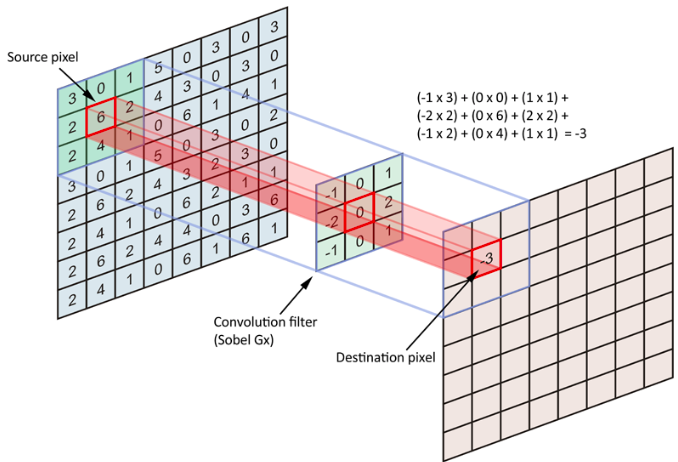
#### b. Architecture

CNN is a supervised method. A Convolutional Neural Network (CNN) consists of an input and an output layer, as well as multiple hidden layers. These layers are generally divided into three types: convolutional layer, nonlinearity layer, pooling layer, and fully-connected layer, and change in quantity and arrangement for building suitable training models for each different problem. The input and output of every stage are sets of arrays referred to as feature maps. In the case of a colored image, every feature map would be a 2D array containing a color channel of the input image, a 3D for a video, and a 1D for audio input. The output stage represents characteristics extracted from all locations on the data.



Picture 2.5 Convolutional neural network architecture

**Convolutional layer** plays a significant role in how CNN operates, is the core building block of a Convolutional Network that does most of the computational heavy lifting. Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as an image matrix, filter, or kernel.



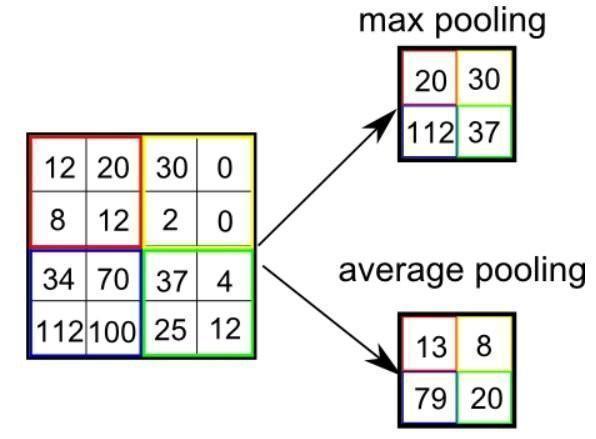
Picture 2.6 Example of convolutional layer

In the example above, the filter used is a 3x3 matrix. This filter is moved through each image area until the entire image scan is completed, creating a new image smaller or equal to the input image size.

**Nonlinearity layer** applies the non-saturating activation function. It will increase the nonlinear properties of the choice function and of the overall network that are desirable for multi-layer networks while not affecting the receptive fields of the convolution layer. The activation functions are usually sigmoid, tanh, and ReLU. Among these functions, ReLU is most selected due to the simple implementation, fast processing speed. Specifically, the calculation of the ReLU function is simply converted all negative values give 0:

*(x) = max(0, x)*

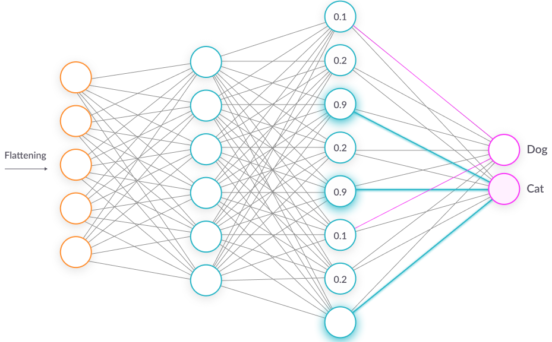
**Pooling layer** are often used between convolutional layers, to reduce data size but retaining important properties. The reduced data size helps to reduce the computation in the model. Similar to the convolution layer, the pooling layer also uses a sliding window to scan all the areas in the image and does a sampling instead of convolution - that is, choosing to store a single unique value that represents all information of that image area.



Picture 2.7 Example of pooling layer

There are two types of pooling: max pooling and average pooling. Max Pooling returns the maximum value while average pooling returns the average of all the values from the portion of the image covered by the kernel.

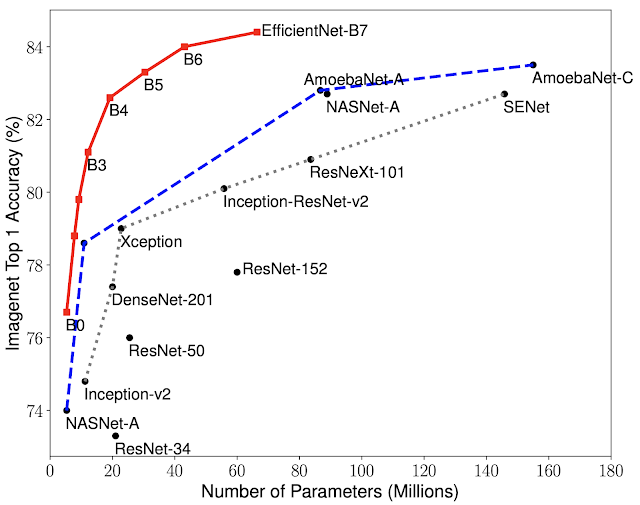
**Fully-connected layer** is designed in a very similar way as in traditional neural networks, all pixels are fully connected to the node in the next layer. Compared with traditional neural networks, the input images of this layer have been greatly reduced in size, while ensuring important information for identification.



Picture 2.8 Example of fully-connected layer

### 2.3.3. EfficientNet

Convolutional neural networks (CNNs) are commonly developed at a fixed resource cost and then scaled up in order to achieve better accuracy when more resources are made available. For instance, ResNet can be scaled up from ResNet-18 to ResNet-200 by using more layers. Google recently published both a very exciting paper and source code for a newly designed CNN (convolutional neural network) called EfficientNet, that set new records for both accuracy and computational efficiency. This was not a minor improvement but rather an accuracy improvement of up to 6% while on the order of 5–10x more efficient than most current CNN’s.



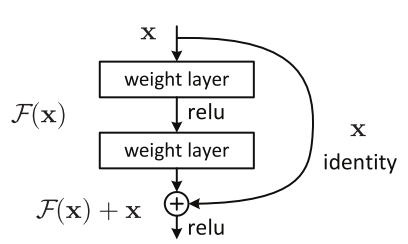
Picture 2.9 Comparison EfficientNets with other existing CNNs on ImageNet

Compared EfficientNets with other existing CNNs on ImageNet, in general, the EfficientNet models achieve both higher accuracy and better efficiency over existing CNNs, reducing parameter size and by an order of magnitude. For example, in the high-accuracy regime, EfficientNet-B7 reaches state-of-the-art 84.4% top-1 / 97.1% top-5 accuracy on ImageNet, while being 8.4x smaller and 6.1x faster on CPU inference than the previous Gpipe. Compared with the widely used ResNet-50, EfficientNet-B4 improves the accuracy from 76.3% of ResNet-50 to 82.6% (+6.3%).

### 2.3.4. ResNet

ResNets or Residual Networks are a type of Convolutional Neural Network. ResNet makes it possible to train up to hundreds or even thousands of layers and still achieves compelling performance. Taking advantage of its powerful representational ability, the performance of many computer vision applications other than image classification has been boosted, such as object detection and face recognition.

When training Deep CNN models (a large number of layers, a large number of params, ...) we often have problems with vanishing gradient or exploding gradient. In fact, when the number of layers in the CNN model increases, the accuracy of the model also increases, but when the number of layers is too large (> 50 layers), the accuracy decreases. This problem of training very deep networks has been alleviated with the introduction of ResNet or residual networks and these Resnets are made up from Residual Blocks.

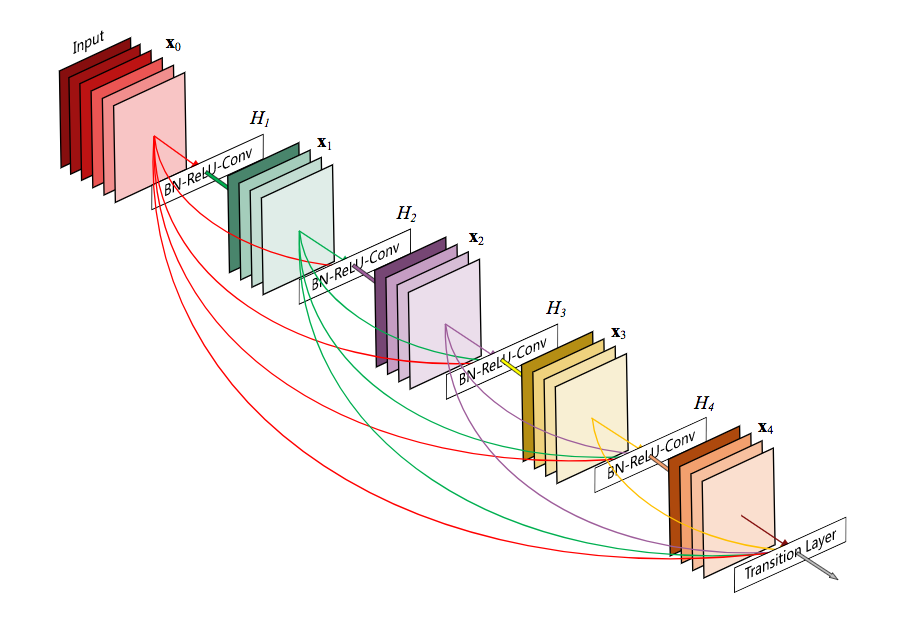


Picture 2.10 Structure of a basic residual block

With residual block, it is completely possible to train CNN models with larger size and complexity without worrying about vanishing gradients. The key of the Residual block is that after every 2 layers, we add input with output: F (x) + x. ResNet is a CNN network consisting of many small residual blocks.

### 2.3.5. DenseNet

DenseNet is one of the new discoveries in neural networks for visual object recognition. DenseNet is quite similar to ResNet with some fundamental differences. DenseNet was developed specifically to improve the declined accuracy caused by the vanishing gradient in high-level neural networks. DenseNet is composed of Dense blocks. In those blocks, the layers are densely connected together: Each layer receives in input all previous layers output feature maps.



Picture 2.11 Structure of a basic dense blocks

## 2.6. Technology

### 2.6.1. Python

#### a. Definition

Python is an interpreted, object-oriented, and high-level programming language designed to be easy to read and simple to implement. It is open-source, which means it is free to use, even for commercial applications. Python can run on Mac, Windows, and Unix systems and has also been ported to Java and .NET virtual machines.



Picture 2.12 Python logo

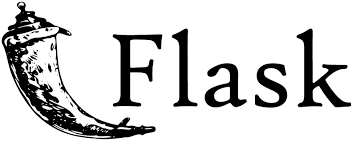
Python is considered a scripting language, like Ruby or Perl, and is often used for creating Web applications and dynamic Web content. It is also supported by a number of 2D and 3D imaging programs, enabling users to create custom plug-ins and extensions with Python. Examples of applications that support a Python API include GIMP, Inkscape, Blender, and Autodesk Maya. Moreover, the Python community has developed many modules and packages to help programmers implement machine learning.

#### b. Characteristics

* Simple grammar, easy to understand.
* Extensive support libraries.
* Open source and community development.
* User-friendly data structures.
* Productivity and speed.
* Presence of third-party modules.

#### c. Flask Python

Flask allows building from simple to complex web applications. It can build small API, web applications such as web pages, blogs, wiki pages, or a time-based website or even a commercial website. Flask provides you with the tools, libraries, and technologies to help you do all things above.



Picture 2.13 Flask Python logo

Flask is a micro-framework. It means that Flask is a standalone environment with no database abstraction layer, form validation, or any other components of pre-existing third-party libraries. Hence, Flask has the advantage of being lightweight, having very few errors due to less dependency as well as easily detecting, handling security errors.

### 2.6.2. JavaScript

JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.



Picture 2.14 HTML, CSS and JavaScript logo

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behavior, and all major web browsers have a dedicated JavaScript engine to execute it.

# Chapter 3: SYSTEM DESIGN ANALYSIS

## 3.1. Application overview

Application is built for diabetic retinopathy diagnosis aid by using deep learning and website platforms. Moreover, application also provides users with information about lesions and influences in each disease stage.

## 3.2. System design analysis

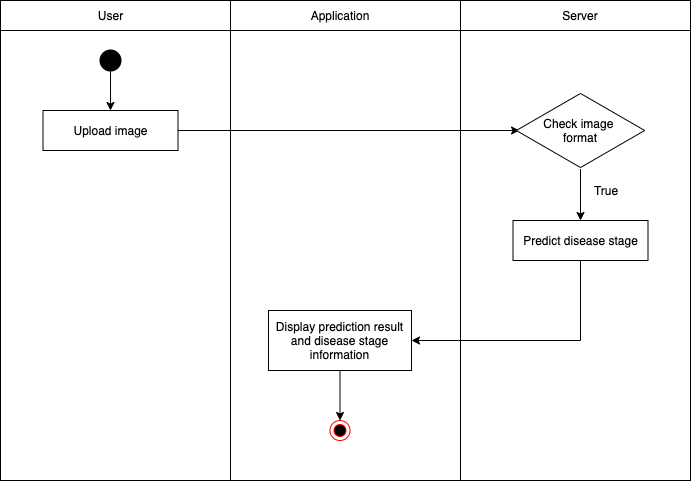
### 3.2.1. Object analysis

Table 3.1 Actor analysis in system

|  |  |
| --- | --- |
| **Actor** | **Description** |
| User | * Upload a fundoscopy image of a patient suspected of suffering from diabetic retinopathy. * Research and read knowledge about diabetic retinopathy. * Retrieve prediction result based on information upload. * Track the rate of patients suffering from diabetic retinopathy. |

### 3.2.2. System diagram

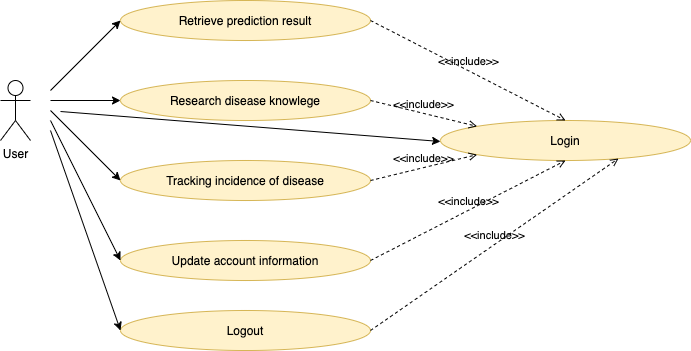
#### a. Activity diagram



Picture 3.1 Activity diagram

The activity diagram fully describes the process of operations that occur when the user submits data and receives the predicted results. Users can choose a fundoscopy image of a patient to upload to the server, the server after receiving data will check the data type. The image will be preprocessed and put through a model for prediction. After getting the prediction results from the model, the server will encapsulate the results and return the application to display to the user. If the user wants to know more information about the rate of patients suffering from diabetic retinopathy, the user can send a query through the user interface of the application. The query request after being posted to the server will be processed and returned results.

#### b. Use case



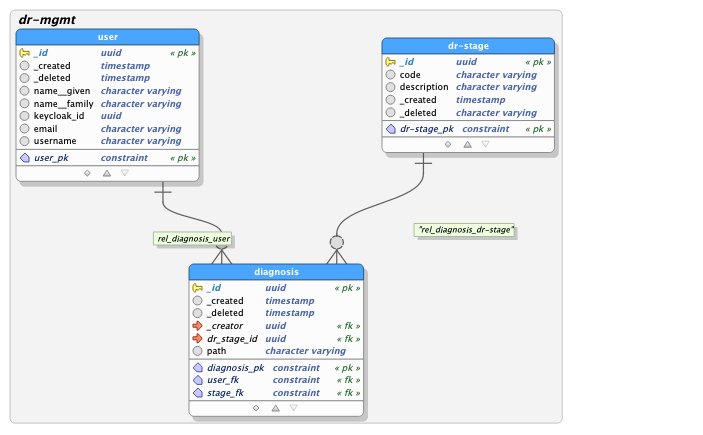
Picture 3.2 Use case

This use case shows a list of actions in general in system. All actions mentioned below need login authentication to perform.

* Retrieve prediction result: users after upload fundoscopy images can retrieve prediction result via application.
* Research disease knowledge: users can update themselves information about diabetic retinopathy.
* Tracking incidence of disease: users access dashboard to update statistics of patients suffering from diabetic retinopathy in hospital.
* Update account information.
* Logout.

### 3.2.3. Define data model

Data models define how the logical structure of a database is modeled and how data is connected to each other as well as how they are processed and stored inside the system. The application for diabetic retinopathy diagnosis aid has 3 models for user, stages of diabetic retinopathy and diagnosis. The data types of fields in each model are described in figure 3.3.



Picture 3.3 Application data model

## 3.4. Conclusion

Chapter 3 shows an overview structure of the application and the way user interacts with application. Moreover, organize and store data are also summarized in this chapter.

# Chapter 4: EXPERIMENT AND RESULT

## 4.1. Solution

Aiming to classify each fundus photograph accurately, neural networks using conventional deep CNN architecture are built. This research starts with basic and popular networks: DenseNet, EfficientNet, and ResNet. Applying a balanced and imbalanced dataset on networks to find a network that gives better accuracy and results. After that, this network is improved to achieve expected accuracy.

### 4.1.1. Environment and machine learning framework

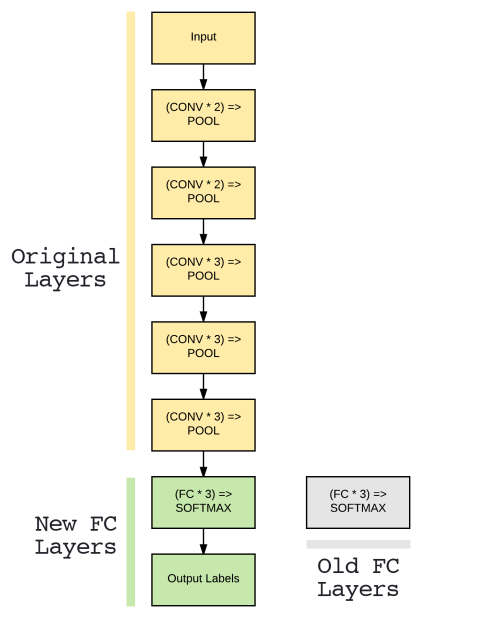
Kaggle is a website for sharing ideas, getting inspired, competing against other data scientists, learning new information, and coding tricks, as well as seeing various examples of real-world data science applications. There are plenty of datasets that can be utilized for anything as simple as video game sales, to something more complex and important like air pollution data. This data is real and referenced, so you can train and test your models on projects that could eventually help real people. Besides, Kaggle supports the user exploring and running machine learning code with Kaggle Notebooks. Being developed based on Jupyter Notebook, using Kaggle Notebook is similar to using Jupyter Notebook. It is an ideal tool for us to practice our programming skills with Python through deep learning libraries. Kaggle Notebook pre-installed libraries that are very common in Deep Learning research like PyTorch, TensorFlow, Keras, and OpenCV.

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides. Keras has advantages:

* Keras has broad adoption in the industry and the research community.
* Keras makes it easy to turn models into products.
* Keras has strong multi-GPU & distributed training support.
* Keras is at the nexus of a large ecosystem: rapid model prototyping, extra layers, losses, metrics, callbacks...

### 4.1.2. Training process

The solution for training is using a technique called transfer learning. Transfer learning consists of taking features learned on one problem and leveraging them on a new, similar problem. Initializing the feature extractor with weights from Imagenet pre-trained CNN. Heads are initialized with corresponding weights.



Picture 4.1 Fine tuning in transfer learning

Multiple steps with different settings are used in each training stage:

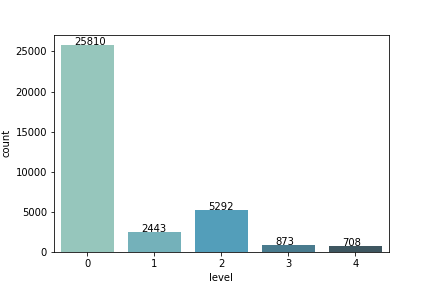
* **Warm-up step**: Warmup steps are just a few updates with a low learning rate before the main training. After this warm up, using the regular learning rate (schedule) to train the model to convergence. The idea that this helps the network to slowly adapt to the data intuitively makes sense.
* **Fine-tune step**: Fine-tuning is one approach to transfer learning. It's the last step, which consists of unfreezing the entire model obtained above (or part of it) and re-training it on the new data with a very low learning rate. This can potentially achieve meaningful improvements, by incrementally adapting the pre-trained features to the new data.

### 4.1.3. Imbalanced data

#### a. Data collection

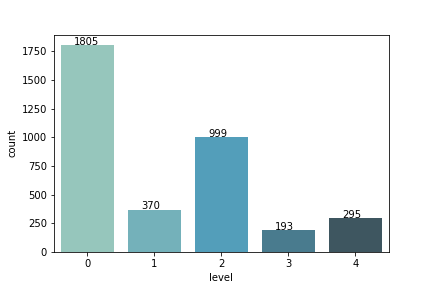
Dataset was collected from:

* **EyePACs**: EyePACs is a license-free Web-based DRS system designed to simplify the process of image capture, transmission, and review. The system provides a flexible platform for collaboration among clinicians about diabetic retinopathy. Since 2006, EyePACs has been expanded to over 120 primary care sites throughout California and elsewhere recording over 34,000 DRSs. With the database of the retinal image comprising over 5 million images, in 2015, EyePACs collaborated with the California Healthcare Foundation to hold a competition on the detection of diabetic retinopathy. This competition published a large number of retinal images and attracted so many teams in the world.



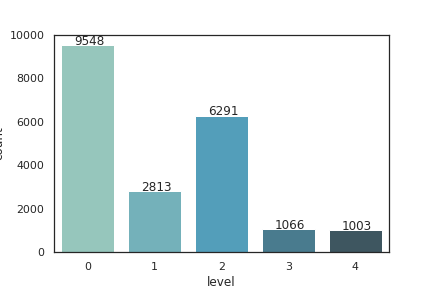
Picture 4.2 Classes distribution in EyePACs dataset

* **Asia Pacific Tele-Ophthalmology Society (APTOS) competition 2019**: At the 4th APTOS Symposium, experts shared how emerging, innovative digital technologies are used in different parts of the world to enhance ophthalmic care. A goal of this competition is to capture algorithms across the world and train new data scientists to help prevent blindness and may be used to detect other sorts of diseases in the future, like glaucoma and macular degeneration. The competition provided thousands of images collected in rural areas to help identify diabetic retinopathy automatically.



Picture 4.3 Classes distribution in APTOS2019 dataset

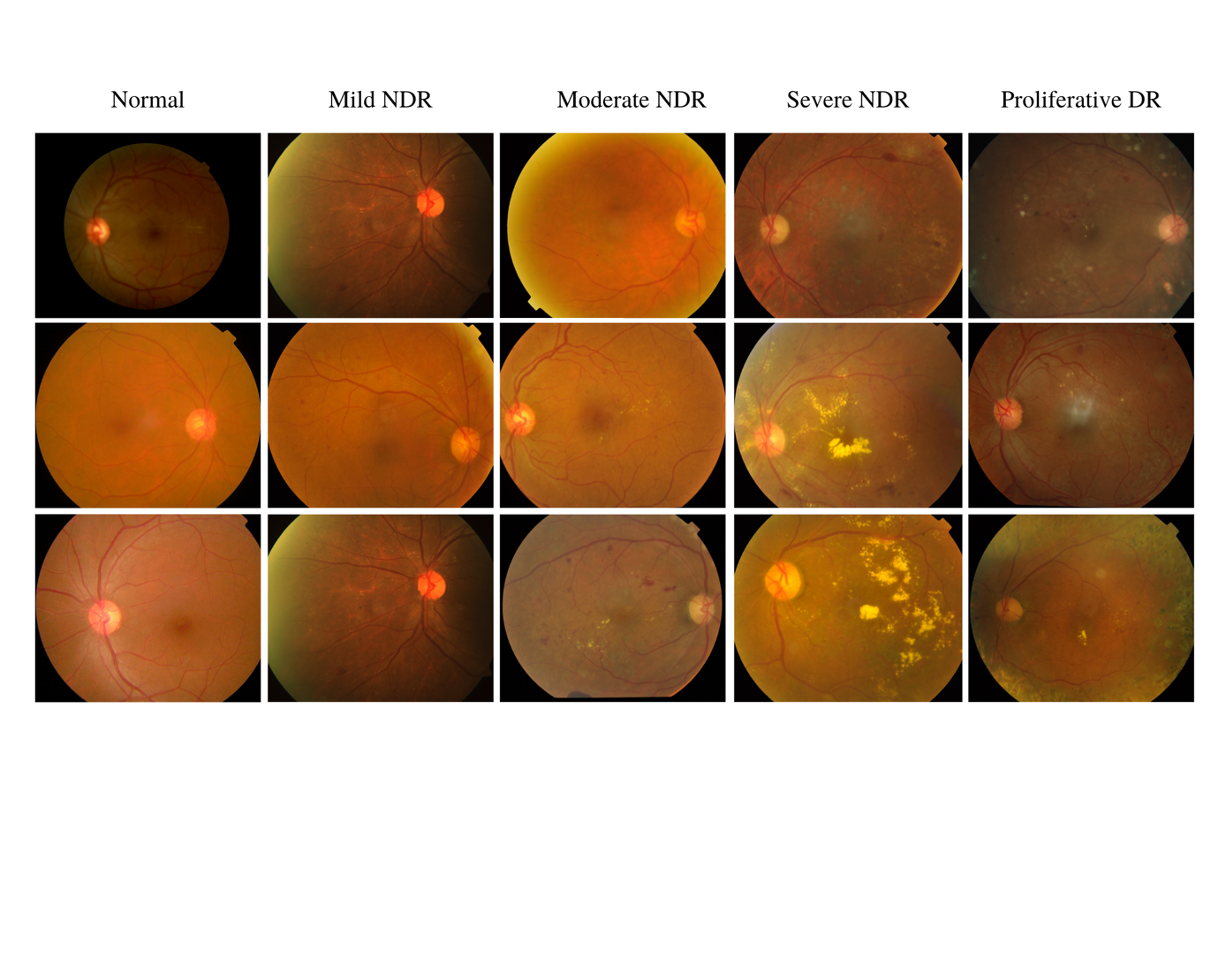
The dataset after collecting is selective for training and testing. Dataset retains only 30% of the number of images in label 0 (Normal) from the **EyePACs** dataset and combines with the rest.



Picture 4.4 Classes distribution in selective dataset

Dataset includes 20,721 images categorized into 5 labels:

* Normal – 0 (9548 images)
* Mild nonproliferative diabetic retinopathy - 1 (2813 images)
* Moderate nonproliferative diabetic retinopathy - 2 (6291 images)
* Severe nonproliferative diabetic retinopathy - 3 (1066 images)
* Proliferative diabetic retinopathy - 4 (1003 images)



Picture 4.5 The sample fundoscopy photography images from dataset

#### b. Solution

Model perhaps confused disease stages with each other if the difference in the number of images in labels is too large. It can be seen that this difference in data can lead to overfitting, the model tries to fit the data with the highest quantity. The dataset summary below shows the difference between the quantity of 5 labels.

Table 4.1 Grade distribution of the diabetic retinopathy images dataset

|  |  |  |
| --- | --- | --- |
| **DR grade** | **Name** | **Total images** |
| 0 | Normal | 9548 |
| 1 | Mild nonproliferative DR | 2813 |
| 2 | Moderate nonproliferative DR | 6291 |
| 3 | Severe nonproliferative DR | 1066 |
| 4 | Proliferative DR | 1003 |

The highest amount of data is label 0 with more than 9000 images, while the lowest is label 4 with more than 1000 images. This discrepancy up to thousands of images can influence model performance. Therefore, the solution given is to use a technique, called k-fold cross-validation.

The k-fold cross-validation procedure is a standard method for estimating the performance of a machine learning algorithm or configuration on a dataset. Different splits of the data may result in very different results. Repeated k-fold cross-validation provides a way to improve the estimated performance of a machine learning model. This involves simply repeating the cross-validation procedure multiple times and reporting the mean result across all folds from all runs. This mean result is expected to be a more accurate estimate of the true unknown underlying mean performance of the model on the dataset, as calculated using the standard error.

The k-fold cross-validation procedure can be implemented easily using the scikit-learn machine learning library. It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split. The general procedure is as follows:

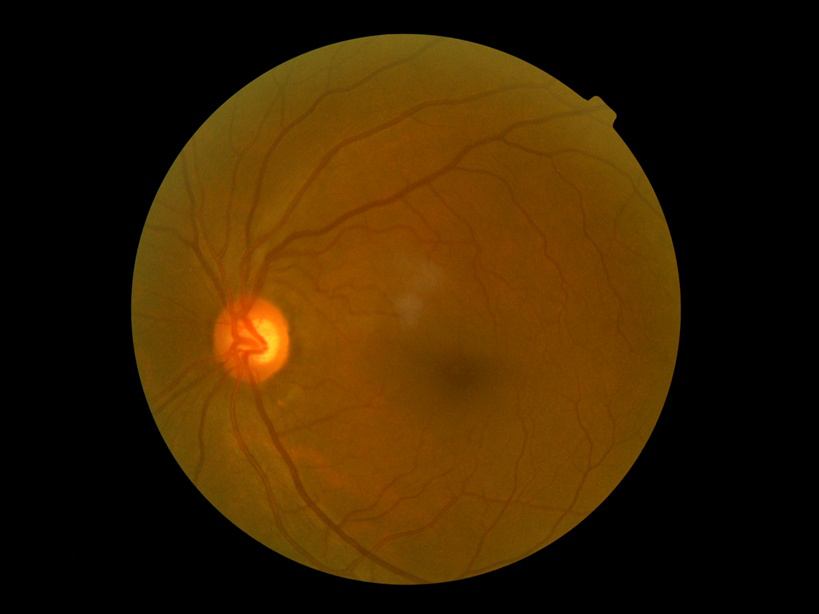
* Shuffle the dataset randomly.
* Split the dataset into k groups.
* For each unique group:
  + Take the group as a hold out or test data set.
  + Take the remaining groups as a training data set.
  + Fit a model on the training set and evaluate it on the test set.
  + Retain the evaluation score and discard the model.
* Summarize the skill of the model using the sample of model evaluation scores.

Importantly, each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set 1 time and used to train the model k-1 times.

### 4.1.4. Feature extraction in the data processing

Diabetic retinopathy detection is based on the features of fundoscopy images. In other words, they are the lesions in each disease stage. Each stage causes lesions with different characteristics, including hemorrhage, aneurysm, edema. macular, exudate. The lesions are not too different, model possibly mistakes in prediction between the adjacent stages. Thus, preprocessing data before training is very important.

The different preprocessing steps to perform on the input dataset before giving it to the model. The preprocessing steps in the study were kept minimal to maintain better generalization of the image conditions. The fundus photography images of the dataset have a random resolution, so they were down-scaled to the same resolution, 224 × 224 pixels, to match with the input image dimension. The images were converted to RGB channels and removed black areas from original square images, and also cropped a circular area based on the image center.



Picture 4.6 Data before feature extraction



Picture 4.7 Data after feature extraction

A basic augmentation step was performed which used the built-in preprocessing function of Keras, ImageDataGenerator. Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. Training deep learning neural network models on more data can result in more skillful models, and the augmentation techniques can create variations of the images that can improve the ability of the fit models to generalize what they have learned to new images. The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class. Functions used in data augmentation include:

* **Rescale (divide by 255)**: Image is scaled down by factor 255 before feeding to the model. Since 255 is the maximum pixel value. Rescale 1/255 is to transform every pixel value from range [0,255] -> [0,1]. Some images are high pixel range, some are low pixel range. A higher pixel range image results in higher loss and should use smaller learning rates, lower pixel range image will need a larger learning rate. Rescale in data augmentation helps the images share the same model, weights, and learning rate.
* **Random rotation 360**: A rotation augmentation randomly rotates the image clockwise by a given number of degrees from 0 to 360. The rotation will likely rotate pixels out of the image frame and leave areas of the frame with no pixel data that must be filled in.
* **Random flip (horizontal and vertical)**: An image flip means reversing the rows or columns of pixels in the case of a vertical or horizontal flip respectively. The flip augmentation is specified by a horizontal flip or vertical flip argument to the ImageDataGenerator class constructor.

### 4.1.5. Evaluation method

To objectively evaluate model performance, several evaluation methods were applied. The model evaluation was performed for multi-label on the dataset used in this study by evaluating the accuracy, quadratic weighted kappa indices, and confusion matrix. Also, loss and accuracy curves were plotted to keep track of the performance of the model concerning the number of epochs.

#### a. Confusion matrix

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). The name stems from the fact that it makes it easy to see if the system is confusing two classes, commonly mislabeling one as another.

Each row in a confusion matrix represents an actual class, while each column represents a predicted class. In this confusion matrix below, of the 8 cat pictures, the system judged that 3 were dogs, and of the 5 dog pictures, it predicted that 2 were cats. All correct predictions are located in the diagonal of the table (highlighted in bold), so it is easy to visually inspect the table for prediction errors, as they will be represented by values outside the diagonal.

Table 4.2 Basic example of confusion matrix

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Predicted class | |
| Cat | Dog |
| Actual class | Cat | **5** | 3 |
| Dog | 2 | **3** |

#### b. Cohen’s kappa score

Cohen’s kappa measures the agreement between two raters who each classify N items into C mutually exclusive categories. A simple way to think this is that Cohen’s Kappa is a quantitative measure of reliability for two raters that are rating the same thing, corrected for how often that the raters may agree by chance.

The value for kappa can range between 0 - 1. A score of 0 means that there is random agreement among raters, whereas a score of 1 means that there is a complete agreement between the raters. However, a score that is less than 0 means that there is less agreement than chance.

Table 4.3 Kappa coefficient interpretation

|  |  |  |
| --- | --- | --- |
| **Value** | **Level of agreement** | **Data that are reliable (%)** |
| 0 – 0.20 | None | 0 - 4 |
| 0.21 – 0.39 | Minimal | 4 – 15 |
| 0.40 – 0.59 | Weak | 15 – 35 |
| 0.60 – 0.79 | Moderate | 35 – 63 |
| 0.80 – 0.90 | String | 64 – 81 |
| Above 0.90 | Almost perfect | 82 – 100 |

Sklearn library supports calculating Cohen’s kappa score by using cohen\_kappa\_score function in metrics. This function computes score expressing the level agreement between two annotators on a classification problem. It is defined as:

Where is the empirical probability of agreement on the label assigned to any sample (the observed agreement ratio), and is the expected agreement when both annotators assign labels randomly. is estimated using a per-annotator empirical prior over the class labels.

## 4.2. Training model implementation

### 4.2.1. Overview of model training and results prediction



Picture 4.8 Overview of model training and result prediction

With the growing popularity of deep learning-based approaches, several methods that apply CNNs to this problem appeared, including making transfer learning with CNN architectures. In this work, to find the best suitable architecture model for this problem, the dataset is separated and implemented on balance and imbalance type. In balance dataset implementation, the image used in this research was taken from the two highest labels in the dataset with the same quantity. The images are evaluated by 3 popular architectures of CNNs: ResNet50, DenseNet169, and EfficientNetB5. These models will also perform on an imbalanced dataset with 5 labels. The model shows the highest accuracy on the test dataset will be chosen to improve architecture and enhance precision.

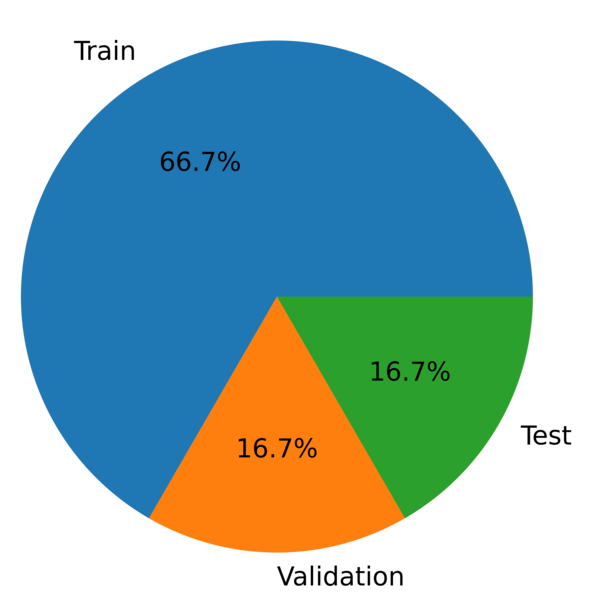
### 4.2.2. Model with balanced dataset

#### a. Data

The two highest labels in quantity were chosen to evaluate in this research: normal (label 0) and moderate nonproliferative diabetic retinopathy (label 2).

Table 4.4 Grade distribution of the diabetic retinopathy images in balanced dataset

|  |  |  |  |
| --- | --- | --- | --- |
| **Label** | **Train** | **Validation** | **Test** |
| 0 | 2003 | 488 | 509 |
| 2 | 1997 | 512 | 491 |
| Total | 4000 | 1000 | 1000 |



Picture 4.9 Data distribution for training, validation and test

#### b. Model architecture

**DenseNet169**: The Dense Convolutional Network is a network architecture where each layer is directly connected to every other layer, in a feed-forward fashion. For each layer, the feature maps of all other layers are treated as separate inputs, whereas its feature maps are passed on as inputs to all subsequent layers. The advantages of DenseNet to alleviate the vanishing gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters, have the edge over other deep convolutional networks. DenseNets obtain significant improvements over the state-of-the-art on most of them while requiring less memory and computation to achieve high performance. After retrieving the image features using the pre-trained model, DenseNet169 with no top is treated as an input to a new convolutional neural network by adding layers later to match desired output data. This table below illustrates the final custom layers of the DenseNet architecture.

Table 4.5 DenseNet169 architecture of the severity grading DR detection model

|  |  |  |
| --- | --- | --- |
| **Layer type** | **Output shape** | **Param** |
| RELU - Activation | (7, 7, 1664) | 0 |
| GlobalAveragePooling2D | (1664) | 0 |
| Dropout | (1664) | 0 |
| Dense | (2048) | 3409920 |
| Dropout | (2048) | 0 |
| Dense | (2) | 4098 |

Table 4.6 The number of DenseNet169 network parameters

|  |  |
| --- | --- |
| **Trainable params** | 16,056,898 |
| **Non-trainable params** | 15,898,498 |
| **Total params** | 16,056,898 |

**EfficientNetB5**: EfficientNet was released this June (2019) by Google AI and is the new state-of-the-art on ImageNet. It introduces a systematic way to scale CNN (Convolutional Neural Networks) in a nearly optimal way. EfficientNet achieves state-of-the-art and uses a lot fewer parameters than most modern CNN architectures. For this work, the B5 version was used. After retrieving the image features using the pre-trained model, Imagenet pre-trained weights EfficientNetB5 with no top is treated as an input to a new convolutional neural network by adding layers later to match the desired output data. This table below illustrates the final custom layers of the EfficientNetB5 architecture.

Table 4.7 EfficientNetB5 architecture of the severity grading DR detection model

|  |  |  |
| --- | --- | --- |
| **Layer type** | **Output shape** | **Param** |
| RELU - Activation | (7, 7, 1664) | 0 |
| GlobalAveragePooling2D | (1664) | 0 |
| Dropout | (1664) | 0 |
| Dense | (2048) | 4196352 |
| Dropout | (2048) | 0 |
| Dense | (2) | 4098 |

Table 4.8 The number of EfficientNetB5 network parameters

|  |  |
| --- | --- |
| **Trainable params** | 4,200,450 |
| **Non-trainable params** | 28,513,520 |
| **Total params** | 32,713,970 |

**ResNet50**, short for Residual Networks, is a classic neural network used as a backbone for many tasks of computer vision. This model was the winner of the ImageNet challenge in 2015. The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+layers successfully. Prior to ResNet training very deep neural networks were difficult due to the problem of vanishing gradients. AlexNet, the winner of ImageNet 2012 and the model that apparently kick-started the focus on deep learning had only 8 convolutional layers, the VGG network had 19 and Inception or GoogleNet had 22 layers and ResNet 152 had 152 layers. ResNet-50, a smaller version of ResNet 152 and frequently used as a starting point for transfer learning. After retrieving the image features using the pre-trained model, Imagenet pre-trained weights ResNet with no top is treated as an input to a new convolutional neural network by adding layers later to match the desired output data. This table below illustrates the final custom layers of the ResNet50 architecture.

Table 4.9 ResNet50 architecture of the severity grading DR detection model

|  |  |  |
| --- | --- | --- |
| **Layer type** | **Output shape** | **Param** |
| RELU - Activation | (7, 7, 1664) | 0 |
| GlobalAveragePooling2D | (1664) | 0 |
| Dropout | (1664) | 0 |
| Dense | (2048) | 4196352 |
| Dropout | (2048) | 0 |
| Dense | (2) | 4098 |

Table 4.10 The number of ResNet50 network parameters

|  |  |
| --- | --- |
| **Trainable params** | 27,735,042 |
| **Non-trainable params** | 53,120 |
| **Total params** | 27,788,162 |

#### c. Training

The training of the proposed network was performed with a multi-label classification method for the grading of diabetic Retinopathy among five severity levels. The hyperparameters settings were kept constant for each of the three networks. The networks were trained with 2 epochs in the warm-up step to avoid early overfitting before completing the fine-tuning step with 30 epochs. However, it could be early stopped if the model performance stopped improving on the validation loss for 5 epochs (reducing overfitting). The summary of the settings of the training hyperparameters is given in the table below.

Table 4.11 The settings of network training hyperparameters

|  |  |
| --- | --- |
| **Parameters** | **Value** |
| Loss function | Categorical cross entropy |
| Optimizer | Adam |
| Batch size | 16 |

Table 4.12 Parameters during each step-in training process

|  |  |  |
| --- | --- | --- |
| **Step** | **Learning rate** | **Epoch** |
| Warm up | 1e-3 | 2 |
| Fine tune | 1e-4 | 30 |

#### d. Result

The calculations were performed on Kaggle kernel having 1 CPU core with 18 GB RAM and 1 GPU core with 15.9 GB RAM. The model evaluation was performed for the multi-label method used in this study by evaluating the accuracy and loss of 4000 images for training and 1000 images for validation. Also, loss and accuracy were tracked on a test dataset with 500 images. The table below shows results on a dataset of three mentioned networks: DenseNet169, EfficientNetB5 and ResNet50.

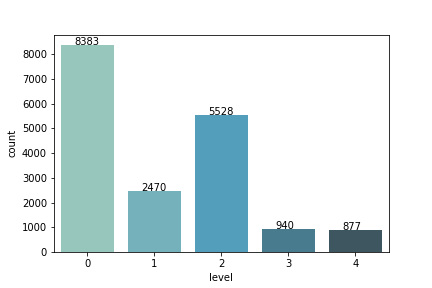
Table 4.13 DR classification model with balanced dataset evaluation based on accuracy and loss analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **ResNet50** | **EfficientNetB5** | **DenseNet169** |
| Best train accuracy | 0.8397 | 0.8345 | 0.8322 |
| Best train loss | 0.3796 | 0.3662 | 0.3820 |
| Best validation accuracy | 0.7230 | 0.7470 | 0.7300 |
| Best validation loss | 0.5960 | 0.5255 | 0.5440 |
| Test accuracy | 0.7210 | 0.7470 | 0.7480 |

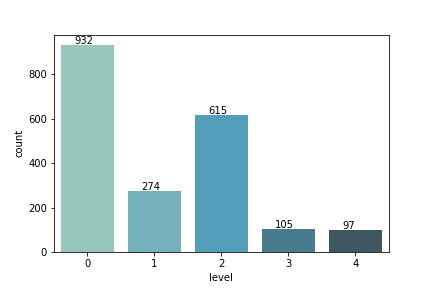
### 4.2.3. Model with imbalanced dataset

#### a. Data

Evaluate the networks in section an on unbalanced data categorized into 5 labels: 0, 1, 2, 3, 4:

****

Picture 4.10 Class distribution in training dataset

****

Picture 4.11 Class distribution in test dataset

#### b. Model architecture

In this section, three networks in section a performed an assessment of an imbalanced dataset. However, other networks with the same layers but the difference in the output layer were added to evaluate. In more detail, the networks including ResNet50 - 1, DenseNet169 - 1, EfficientNetB5 - 1, were held as the same architectures in section a, the output of the last layer of 5 instead of 2, the activation function was softmax and the loss function were still categorical\_crossentropy. The networks including ResNet50 - 2, DenseNet169 - 2, and EfficientNetB5 - 2, were preserved as architectures in section a, however, the difference here is that the output of the last layer was 1, the activation function is changed to a linear, and the loss function was mean\_squared\_error. The predicted results were compared with the given thresholds to distribute data to the correct labels. This means enlarging the distribution threshold for labels, instead of 0 is threshold for class 0, it could be expanded to 0.5).

Table 4.14 Threshold distribution in grading diabetic retinopathy

|  |  |
| --- | --- |
| **Label** | **Threshold** |
| 0 | < 0.5 |
| 1 | < 1.5 |
| 2 | < 2.5 |
| 3 | < 3.5 |
| 4 | >= 3.5 |

Table 4.15 DenseNet169 - 1 architecture with imbalanced dataset

|  |  |  |
| --- | --- | --- |
| **Layer type** | **Output shape** | **Param** |
| GlobalAveragePooling2D | (1664) | 0 |
| Dropout | (1664) | 0 |
| Dense | (2048) | 340992 |
| Dropout | (2048) | 0 |
| Dense | (5) | 10245 |

Table 4.16 The number of DenseNet169 - 1 parameters with imbalanced dataset

|  |  |
| --- | --- |
| **Trainable params** | 4,420,165 |
| **Non-trainable params** | 12,642,880 |
| **Total params** | 16,063,045 |

Table 4.17 DenseNet169 - 2 architecture with imbalanced dataset

|  |  |  |
| --- | --- | --- |
| **Layer type** | **Output shape** | **Param** |
| GlobalAveragePooling2D | (2048) | 0 |
| Dropout | (2048) | 0 |
| Dense | (2048) | 4196352 |
| Dropout | (2048) | 0 |
| Dense | (1) | 2049 |

Table 4.18 The number of DenseNet169 -2 parameters with imbalanced dataset

|  |  |
| --- | --- |
| **Trainable params** | 4,198,401 |
| **Non-trainable params** | 28,513,520 |
| **Total params** | 32,711,921 |

Table 4.19 EffdicientNetB5 - 1 architecture with imbalanced dataset

|  |  |  |
| --- | --- | --- |
| **Layer type** | **Output shape** | **Param** |
| GlobalAveragePooling2D | (2048) | 0 |
| Dropout | (2048) | 0 |
| Dense | (2048) | 4196352 |
| Dropout | (2048) | 0 |
| Dense | (5) | 10245 |

Table 4.20 The number of EfficientNetB5 - 1 parameters with imbalanced dataset

|  |  |
| --- | --- |
| **Trainable params** | 4,206,597 |
| **Non-trainable params** | 28,513,520 |
| **Total params** | 32,720,117 |

Table 4.21 EffdicientNetB5 - 2 architecture with imbalanced dataset

|  |  |  |
| --- | --- | --- |
| **Layer type** | **Output shape** | **Param** |
| GlobalAveragePooling2D | (1664) | 0 |
| Dropout | (1664) | 0 |
| Dense | (2048) | 4196352 |
| Dropout | (2048) | 0 |
| Dense | (2) | 4098 |

Table 4.22 The number of EfficientNetB5 - 2 parameters with imbalanced dataset

|  |  |
| --- | --- |
| **Trainable params** | 27,735,042 |
| **Non-trainable params** | 53,120 |
| **Total params** | 27,788,162 |

Table 4.23 ResNet50 - 1 architecture with imbalanced dataset

|  |  |  |
| --- | --- | --- |
| **Layer type** | **Output shape** | **Param** |
| GlobalAveragePooling2D | (2048) | 0 |
| Dropout | (2048) | 0 |
| Dense | (2048) | 4196352 |
| Dropout | (2048) | 0 |
| Dense | (5) | 10245 |

Table 4.24 The number of ResNet50 - 1 parameters with imbalanced dataset

|  |  |
| --- | --- |
| **Trainable params** | 27,741,189 |
| **Non-trainable params** | 53,120 |
| **Total params** | 27,794,309 |

Table 4.25 ResNet50 - 2 architecture with imbalanced dataset

|  |  |  |
| --- | --- | --- |
| **Layer type** | **Output shape** | **Param** |
| GlobalAveragePooling2D | (2048) | 0 |
| Dropout | (2048) | 0 |
| Dense | (2048) | 4196352 |
| Dropout | (2048) | 0 |
| Dense | (2) | 2049 |

Table 4.26 The number of ResNet50 - 2 parameters with imbalanced dataset

|  |  |
| --- | --- |
| **Trainable params** | 4,198,401 |
| **Non-trainable params** | 23,587,712 |
| **Total params** | 27,786,113 |

#### c. Training

The training of the proposed network was performed with a multi-label classification method for the grading of diabetic retinopathy among five severity levels. The hyperparameters settings were kept constant for each of the three networks. The networks were trained with 5 epochs in the warm-up step to avoid early overfitting before completing the fine-tuning step with 30 epochs. However, it could be early stopped if the model performance stopped improving on the validation loss for 5 epochs (reducing overfitting). The summary of the settings of the training hyperparameters is given in the table below.

Table 4.27 The settings of networks training hyperparameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Network - 1**  **(DenseNet169 - 1, EffifcientNetB5 - 1, ResNet50 - 1)** | **Network - 2**  **(DenseNet169 - 2, EffifcientNetB5 - 2, ResNet50 - 2)** |
| Loss function | Categorical cross entropy | Mean squared error |
| Optimizer | Adam | Adam |
| Batch size | 16 | 16 |
| Learning rate warm up step | 1e-3 | 1e-3 |
| Learning rate fine tune step | 1e-4 | 1e-4 |
| Epoch warm up step | 5 | 5 |
| Epoch warm up step | 30 | 30 |

#### d. Result

Table 4.28 DenseNet169 with imbalanced dataset evaluation based on accuracy and loss analysis

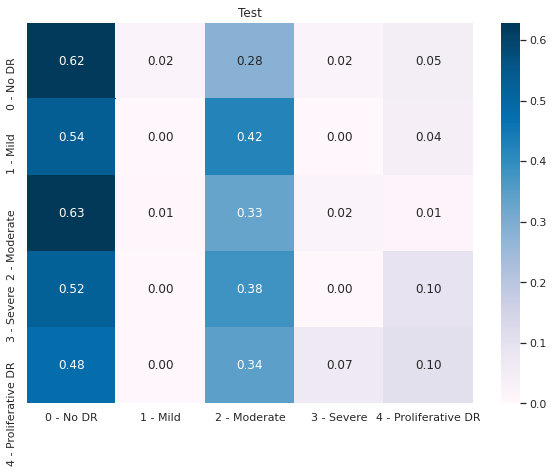
|  |  |  |
| --- | --- | --- |
| **Parameter** | **DenseNet169 - 1** | **DenseNet169 - 2** |
| Best train accuracy | 0.6852 | 0.7179 |
| Best train loss | 0.7862 | 0.2876 |
| Best validation accuracy | 0.6771 | 0.6291 |
| Best validation loss | 0.8379 | 0.5442 |
| Test accuracy | 0.6899 | 0.6294 |

Table 4.29 EfficientNetB5 with imbalanced dataset evaluation based on accuracy and loss analysis

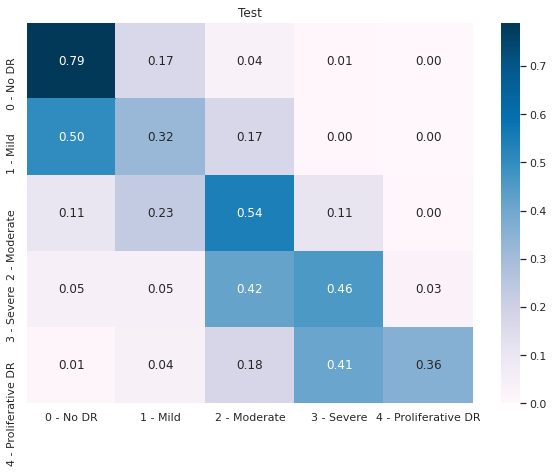
|  |  |  |
| --- | --- | --- |
| **Parameter** | **EfficientNetB5 - 1** | **EfficientNetB5 - 2** |
| Best train accuracy | 0.7106 | 0.7757 |
| Best train loss | 0.5879 | 0.2149 |
| Best validation accuracy | 0.6687 | 0.6218 |
| Best validation loss | 0.8779 | 0.5503 |
| Test accuracy | 0.6860 | 0.6540 |

Table 4.30 ResNet50 with imbalanced dataset evaluation based on accuracy and loss analysis

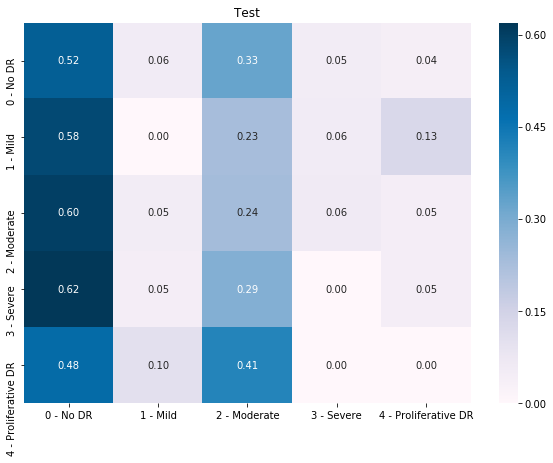
|  |  |  |
| --- | --- | --- |
| **Parameter** | **ResNet50 - 1** | **ResNet50 - 2** |
| Best train accuracy | 0.7120 | 0.5274 |
| Best train loss | 0.7120 | 0.1758 |
| Best validation accuracy | 0.6652 | 0.4360 |
| Best validation loss | 0.8410 | 0.5976 |
| Test accuracy | 0.6480 | 0.5940 |

****

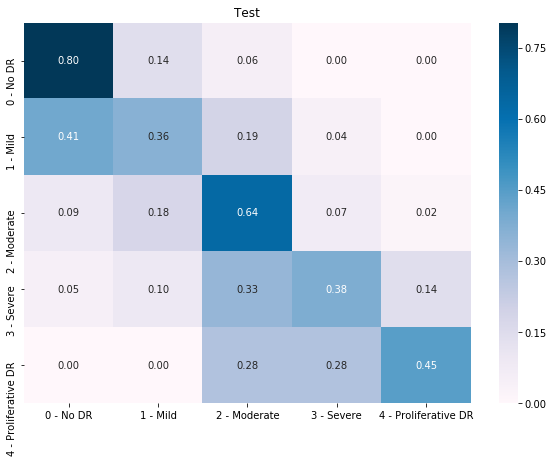
Picture 4.12 Confusion matrix of DenseNet169 - 1 evaluated on test dataset

****

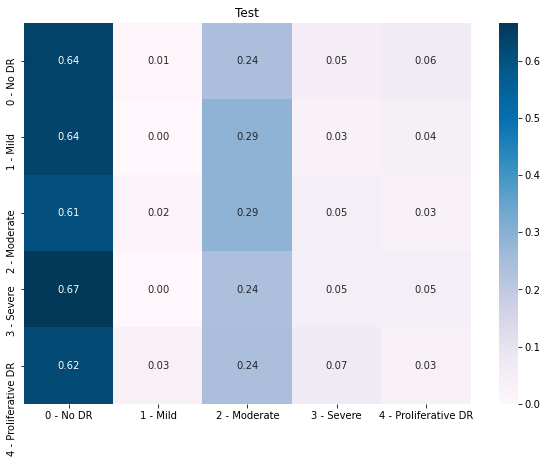
Picture 4.13 Confusion matrix of DenseNet169 - 2 evaluated on test dataset

****

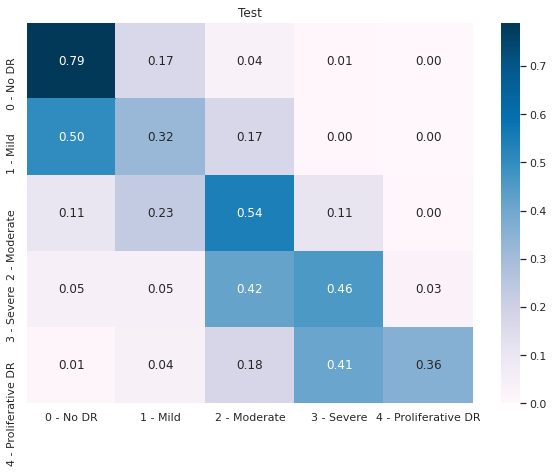
Picture 4.14 Confusion matrix of EfficientNetB5 - 1 evaluated on test dataset

****

Picture 4.15 Confusion matrix of EfficientNetB5 - 2 evaluated on test dataset

****

Picture 4.16 Confusion matrix of ResNet50 - 1 evaluated on test dataset

****

Picture 4.17 Confusion matrix of ResNet50 - 2 evaluated on test dataset

Solving the imbalanced dataset problem section b, both networks-1 (DenseNet169 - 1, EfficientNetB5 - 2, ResNet50 - 1) networks-2 (DenseNet169 - 2, EfficientNetB5 - 2, ResNet50 - 2) gave the same accuracy in training and testing. However, evaluated model performance on the confusion matrix, it can be seen that networks-2 gave better results. Whereas networks-1 can only predict images of 2 labels with the highest quantities (label 0 and label 2), networks-2 was able to distribute images to almost all labels.

### 4.2.4. Improving accuracy

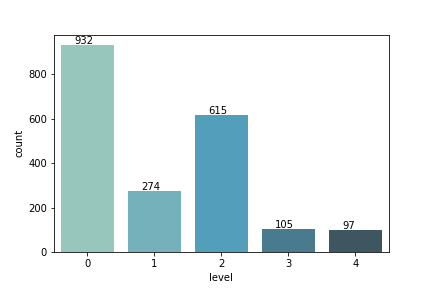
#### a. Data

K-fold cross-validation is where a given data set is split into a k number of sections/folds where each fold is used as a testing set at some point. In this work, the data set was split into 5 folds. In the first iteration, the first fold is used to test the model and the rest are used to train the model. In the second iteration, the second fold was used as the testing set while the rest serve as the training set. This process is repeated until each fold of the 5 folds has been used as the testing set.

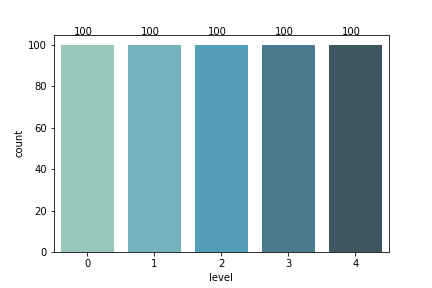
A dataset with 20,721 fundoscopy images was split into 5 folds. The ratio between the validation dataset and the training dataset for each fold was 0.1. Model showed performance on a test dataset with 500 images in balanced and imbalanced data type.

Table 4.31 Data distribution in each fold

|  |  |  |  |
| --- | --- | --- | --- |
| **Label** | **Fold** | **Training dataset** | **Validation dataset** |
| 0 | 0 | 8383 | 932 |
| 1 | 2470 | 274 |
| 2 | 5528 | 615 |
| 3 | 940 | 105 |
| 4 | 877 | 97 |
| 1 | 0 | 8383 | 932 |
| 1 | 2470 | 274 |
| 2 | 5528 | 615 |
| 3 | 940 | 105 |
| 4 | 877 | 97 |
| 2 | 0 | 8383 | 932 |
| 1 | 2470 | 274 |
| 2 | 5528 | 615 |
| 3 | 940 | 105 |
| 4 | 877 | 97 |
| 3 | 0 | 8383 | 932 |
| 1 | 2470 | 274 |
| 2 | 5528 | 615 |
| 3 | 940 | 105 |
| 4 | 877 | 97 |
| 4 | 0 | 8383 | 932 |
| 1 | 2470 | 274 |
| 2 | 5528 | 615 |
| 3 | 940 | 105 |
| 4 | 877 | 97 |



Picture 4.18 Class distribution in imbalanced test dataset



Picture 4.19 Class distribution in balanced test dataset

#### b. Model architecture

Here, the classification head outputs a one-hot encoded vector, where the presence of each stage is represented as 1. Regression head outputs real numbers in the range [0,4.5), which is then rounded to an integer that represents the disease stage. Briefly, if the data point falls into category k, it automatically falls into all categories from 0 to k −1. So, this head aims to predict all categories up to the target. The final prediction is obtained by fitting a linear regression model to outputs of three heads. The neural network structure of EfficientNetB5 is shown in Figure 5

Table 4.32 EfficientNetB5 architecture

|  |  |  |
| --- | --- | --- |
| **Layer type** | **Output shape** | **Param** |
| GlobalAveragePooling2D | (2048) | 0 |
| Dense | (1) | 2049 |

#### c. Training

Table 4.33 The setting of hyperparameters during training

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Loss function | Mean squared error |
| Optimizer | Adam |
| Batch size | 16 |
| Learning rate warm up step | 1e-3 |
| Learning rate fine tune step | 1e-4 |
| Epoch warm up step | 5 |
| Epoch fine tune step | 30 |

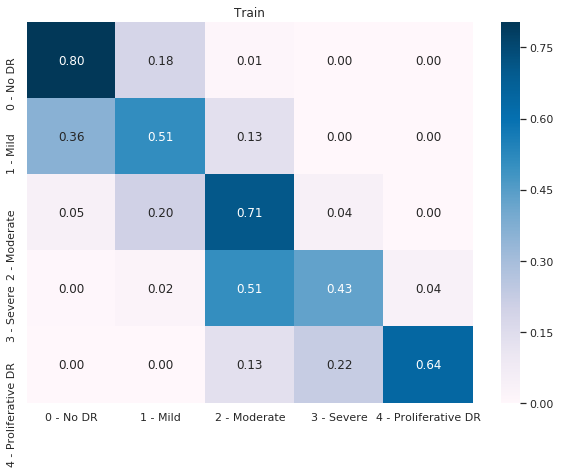
#### d. Result

* Accuracy and loss analysis

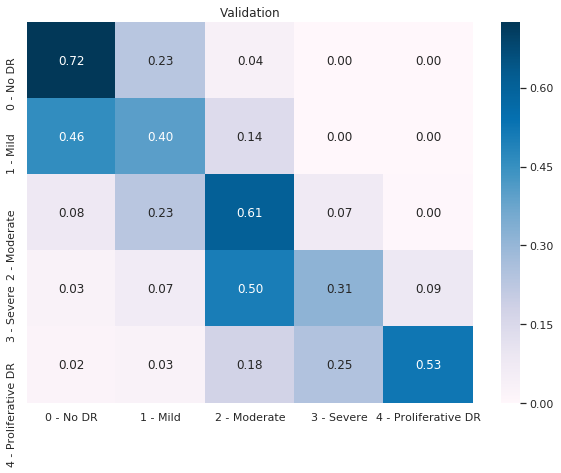
Table 4.34 5-fold evaluation based on accuracy and loss analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | **Fold 0** | **Fold 1** | **Fold 2** | **Fold 3** | **Fold 4** |
| Train accuracy | 0.7066 | 0.6721 | 0.6917 | 0.6452 | 0.6402 |
| Validation accuracy | 0.6218 | 0.6233 | 0.6512 | 0.6173 | 0.6243 |
| Test with imbalanced dataset accuracy | 0.6360 | 0.6440 | 0.6180 | 0.6200 | 0.5820 |
| Best train loss | 0.3384 | 0.3532 | 0.3249 | 0.3959 | 0.4022 |
| Best validation loss | 0.4954 | 0.5708 | 0.4701 | 4986 | 0.4822 |

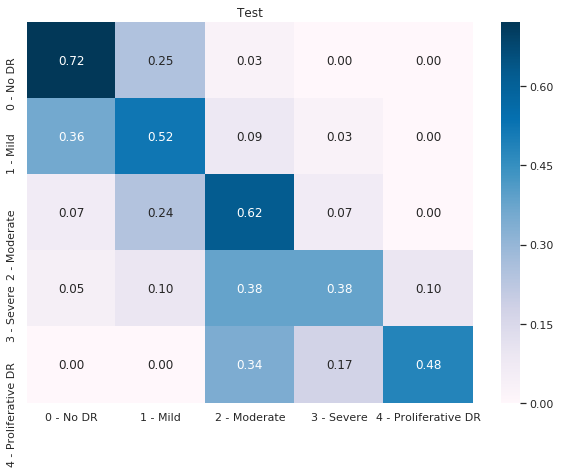
* Confusion matrix



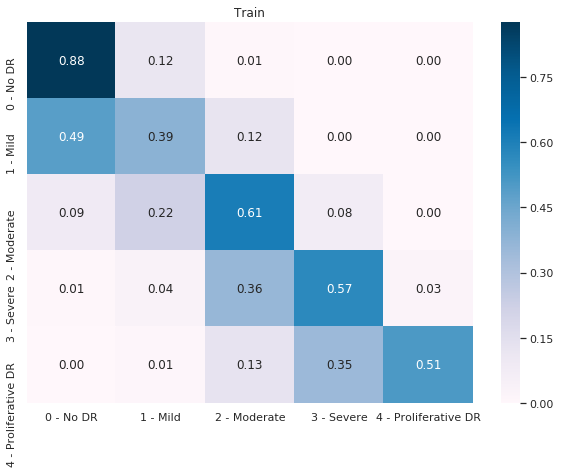
Picture 4.20 Confusion matrix of fold-0 model on train dataset



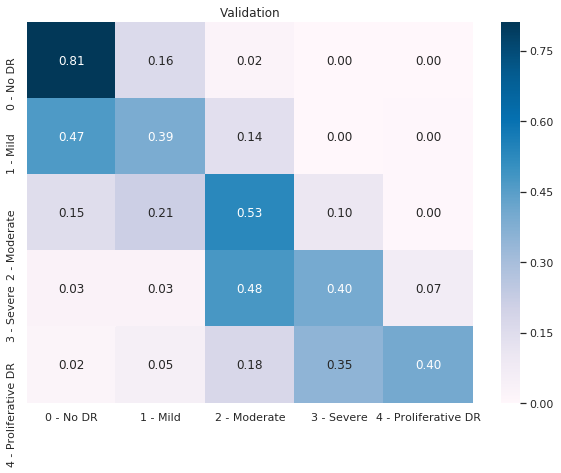
Picture 4.21 Confusion matrix of fold-0 model on validation dataset



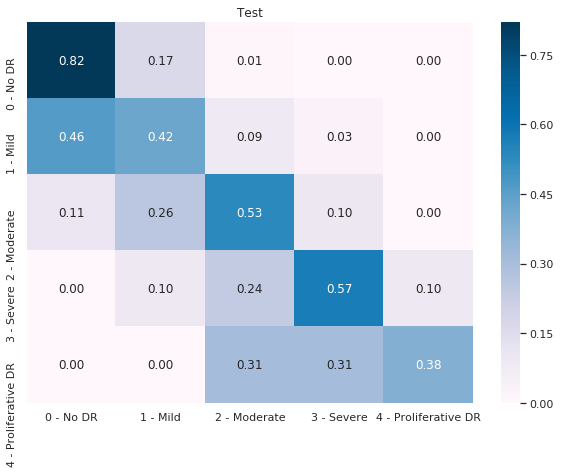
Picture 4.22 Confusion matrix of fold-0 model on imbalanced test dataset



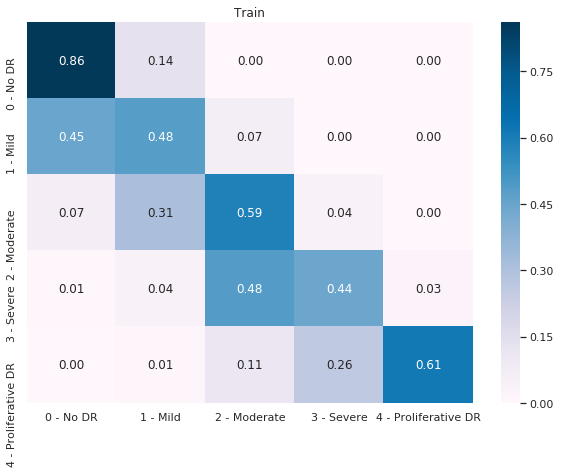
Picture 4.23 Confusion matrix of fold-1 model on train dataset



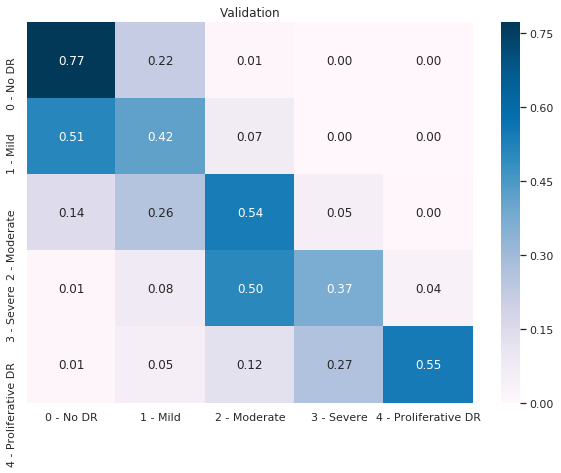
Picture 4.24 Confusion matrix of fold-1 model on validation dataset



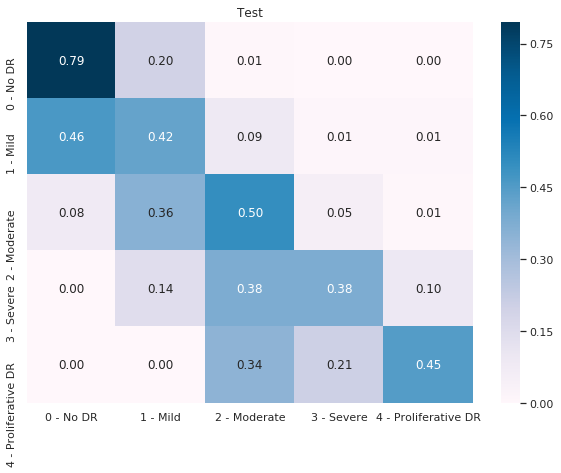
Picture 4.25 Confusion matrix of fold-1 model on imbalanced test dataset



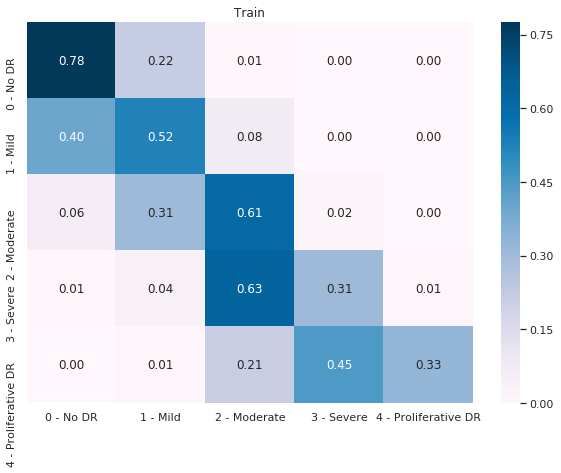
Picture 4.26 Confusion matrix of fold-2 model on train dataset



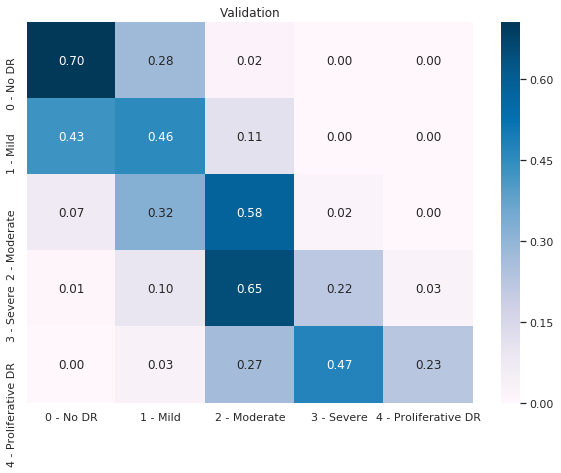
Picture 4.27 Confusion matrix of fold-2 model on validation dataset



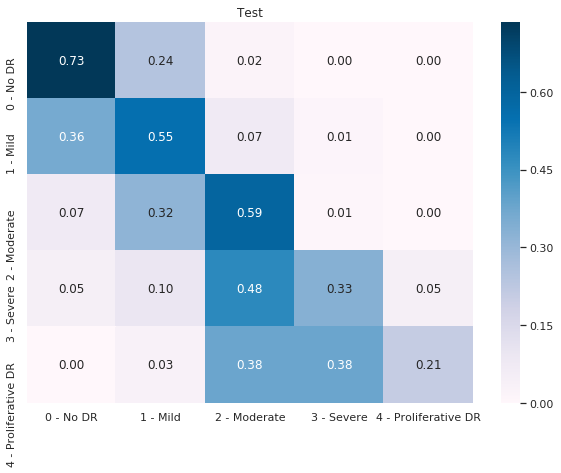
Picture 4.28 Confusion matrix of fold-2 model on imbalanced test dataset



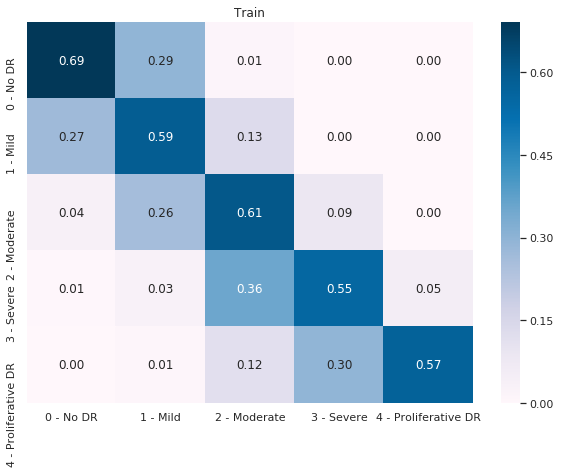
Picture 4.29 Confusion matrix of fold-3 model on train dataset



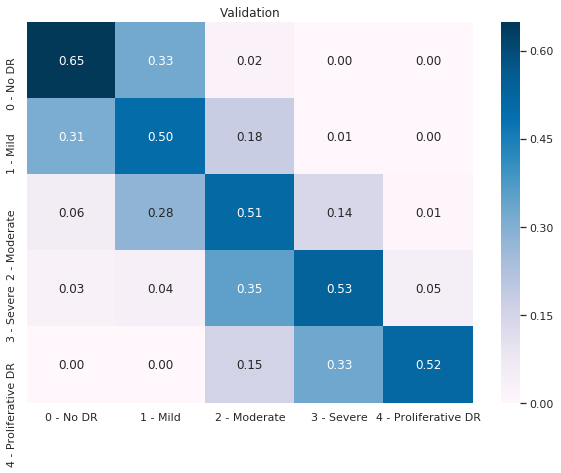
Picture 4.30 Confusion matrix of fold-3 model on validation dataset



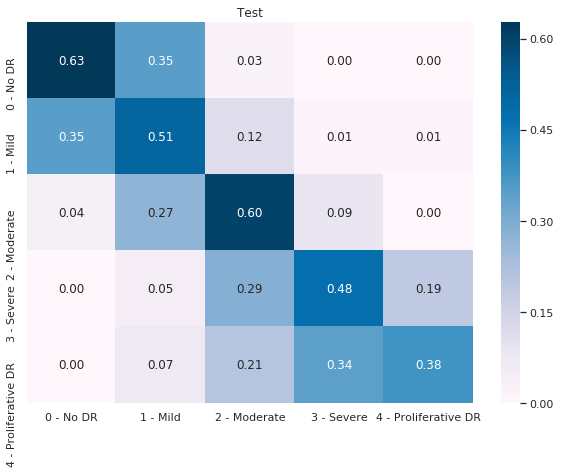
Picture 4.31 Confusion matrix of fold-3 model on imbalanced test dataset



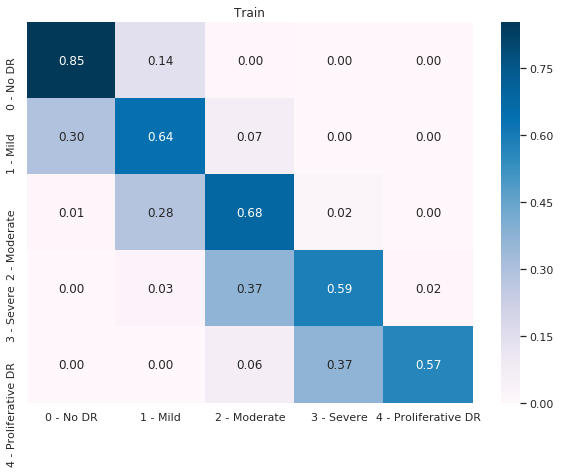
Picture 4.32 Confusion matrix of fold-4 model on train dataset



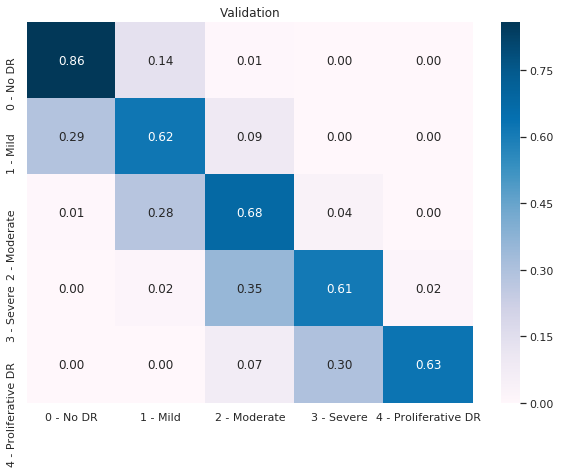
Picture 4.33 Confusion matrix of fold-4 model on validation dataset



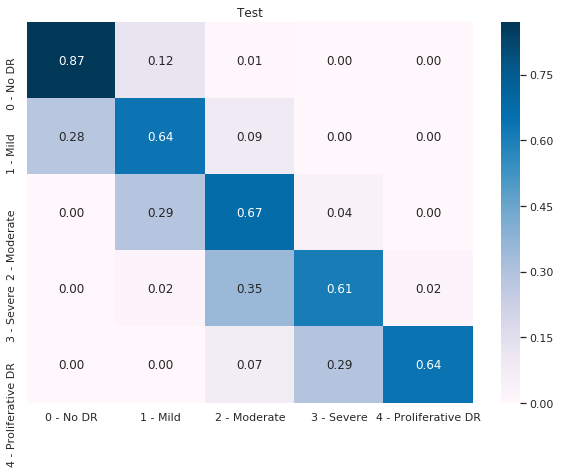
Picture 4.34 Confusion matrix of fold-4 model on imbalanced test dataset



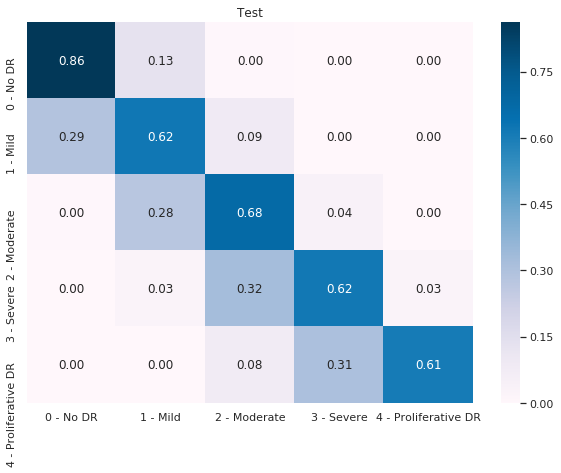
Picture 4.35 Confusion matrix of median model on train dataset



Picture 4.36 Confusion matrix of median model on validation dataset



Picture 4.37 Confusion matrix of median model on imbalanced test dataset



Picture 4.38 Confusion matrix of median model on balanced test dataset

* The quadratic weighted kappa

Table 4.35 5-fold evaluation based on the quadratic weighted kappa

|  |  |  |
| --- | --- | --- |
| **Fold** | **Dataset type** | **Quadratic weighted kappa value** |
| Fold 0 | Train dataset | 0.852 |
| Validation dataset | 0.764 |
| Imbalanced test dataset | 0.781 |
| Fold 1 | Train dataset | 0.837 |
| Validation dataset | 0.755 |
| Imbalanced test dataset | 0.796 |
| Fold 2 | Train dataset | 0.844 |
| Validation dataset | 0.769 |
| Imbalanced test dataset | 0.781 |
| Fold 3 | Train dataset | 0.807 |
| Validation dataset | 0.754 |
| Imbalanced test dataset | 0.754 |
| Fold 4 | Train dataset | 0.824 |
| Validation dataset | 0.781 |
| Imbalanced test dataset | 0.772 |
| Avg. Fold | Train dataset | 0.885 |
| Validation dataset | 0.891 |
| Imbalanced test dataset | 0.891 |

### 4.2.5. Conclusion

The k-fold cross-validation technique that combines transfer learning and treats classification output as a linear output on the EfficientNetB5 network gave optimal results, up to 74% accuracy for imbalanced datasets and 76% for balanced datasets. The results also confirm the feasibility of diabetic retinopathy detection.

Besides that, the results also compare the performance of some networks based on CNN architectures (DenseNet, ResNet, EfficientNet). This is the basic background for approaching and select networks for the same problems.

## 4.3. Server development

### 4.3.1. Dependency

In order to easily implement the system, the development environment installs the following libraries:

* Flask: is the main library to build APIs for the application.
* Flask OIDC: an extension to Flask that allows adding OpenID Connect based authentication to the website in a matter of minutes. It depends on Flask and oauth2client.
* NumPy: help convert weights and resulting parameters in the prediction process.
* Open-CV: helps to process images before prediction.
* Flask-SQLAlchemy, psycopg2: connect to database.
* Keras, TensorFlow: Frameworks for building models.

### 4.3.2. Implementation

#### a. Keycloak

Keycloak is an open-source Identity and Access Management solution aimed at modern applications and services. It makes it easy to secure applications and services with little to no code. By using Keycloak, developers can add authentication to applications and secure services with minimum efforts. No need to deal with storing users or authenticating users. It's all available out of the box. Developers even get advanced features such as User Federation, Identity Brokering, and Social Login.

Keycloak server can be created and run in Docker container:

* Build Keycloak server by latest images in docker
* Custom themes for user login
* Run Keycloak server by docker-compose

#### b. Application

Flask is the most policed and feature-rich micro-framework. Flask comes with all its benefits of the fast template, strong WSGI features, and extensive documentation. Flask gives lots of good features, vast no of extension facilities for a new project. Flask has two dependencies, they are Werkzeug, a WSGI utility library, and Jinja2, a template engine. Werkzeug is a utility library used for the Python programming language. And it is also a toolkit of WSGI (web server gateway interface). It has licensed under the BSD License. Werkzeug can register software for the request, response, and utility function. Jinja2 is a template engine for the Python programming language and is licensed under a BSD license. You can build the basic layout of your page and can mention which will be changed with a template. In this way, you can change your header of the page by upgrading it in one place only. Flask for Python based on Werkzeug, jinja2, and good intention.

Flask uses the Jinja template library to render templates, called flask-template. Templates are files that contain static data as well as placeholders for dynamic data. A template is rendered with specific data to produce a final document. In Flask, Jinja is configured to auto escape any data that is rendered in HTML templates. This means that it’s safe to render user input; any characters they’ve entered that could mess with the HTML.

An extension to Flask - Flask OIDC that allows adding OpenID Connect based authentication to the website in a matter of minutes. It depends on Flask and oauth2client. Flask OIDC helps to register and authenticate accounts simply with Keycloak.

List APIs implemented in application:

Table 4.36 List APIs implemented in application

|  |  |  |
| --- | --- | --- |
| **No.** | **Function** | **Method** |
| 1 | Access home page | GET |
| 2 | Create account | POST |
| 3 | Login | POST |
| 4 | Logout | POST |
| 5 | Update account profile | POST |
| 6 | Access application guidance | GET |
| 7 | Upload fundus photography images | POST |
| 8 | Retrieve prediction result | GET |
| 9 | Track the rate of patients | GET |

## 4.4. Evaluation result

## 4.5. Conclusion

Chapter 4 presents some requirements for the deployment environment, model training, application building, and application operation. It also helps readers have a more overview and specific information about the application that supports the diagnosis of diabetic retinopathy.

# CONCLUSION

## 1. Achievement

* Have a better understanding of learning the basic machine learning and know-how to apply them to solve real problems.
* Implementing the Back-End system using Flask python,
* Understand professional knowledge about symptoms, causes, and prevention for diseases selected for investigation.
* Understand deep neural network structure through a trained model.
* Know how to solve problems encountered during training model to predict as well as the techniques of image data enhancement and transformation.
* Take advantage of user upload data to enhance your app's data set by sorting labels and preprocessing.
* Build an application with an easy-to-see and easy-to-use interface for users.
* This is a project that follows the correct process of developing a technology product.
* Compliance with the schedule and high self-discipline work.

## 2. Disadvantage

* The interface is not eye-catching and attractive.
* The application is quite simple and has few functions
* The number of stages supported by diseases is not numerous and limited in quantity.
* The application has not been evaluated by a large number of users yet.

## 3. Future development

Although the project solved the initial requirements, it is still possible to develop more features as well as solve some outstanding issues, such as:

* Enhance image database to support increased accuracy when predicting.
* Support for medical centers, hospitals in the examination, and diagnosis of diabetic retinopathy.
* Optimize the interface, increase user experience.
* Integrate many features such as news, scientific research related to medicine learning to give users more experience about the application.
* Incorporate more factors to increase the predictive rate of the model as well as description from the user, information about the medical history.
* Apply and improve training models with new algorithms to achieve high precision.

# REFERENCES

[1] University of Birmingham, University Hospitals Birmingham NHS Foundation Trust., The Lancet Digital Health, 2019

[2] Chen XW, Lin X. Big data deep learning: challenges and perspectives. IEEE Access, 2014

[3] Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS. Deep learning for visual understanding: a review. Neurocomputing, 2016

[4] Pires R, Avila S, Wainer J, Valle E, Abramoff MD, Rocha A. A data-driven approach to referable diabetic retinopathy detection. Artif Intell Med, 2019

[5] Jiang H, Yang K, Gao M, Zhang D, Ma H, Qian W. An interpretable ensemble deep learning model for diabetic retinopathy disease classification. In: 2019 41st Annual International conference of the IEEE engineering in medicine and biology society (EMBC), 2019

[6] Wang X, Lu Y, Wang Y, Chen WB. Diabetic retinopathy stage classification using convolutional neural networks. In: International Conference on information Reuse and Integration for data science, 2018

[7] Lu J, Xu Y, Chen M, Luo Y. A coarse-to-fine fully convolutional neural network for fundus vessel segmentation. Symmetry, 2018

[8] Zheng S, et al. Conditional random fields as recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, 2015

[9] Hoover A, Kouznetsova V, Goldbaum M. Locating blood vessels in retinal images by piece-wise threshold probing of a matched filter response. IEEE Trans Med Imag, 2000

[10] A. Budai, R. Bock, A. Maier, J. Hornegger, and G. Michelson, “Robust vessel segmentation in fundus images,” Int J Biomed Imag, vol. 2013, 2013

[11] Staal JJ, Abramoff MD, Niemeijer M, Viergever MA, van Ginneken B. Ridge-based vessel segmentation in color images of the retina. IEEE Trans Med Imag, 2004

[12] Owen CG, et al. Measuring retinal vessel tortuosity in 10-year-old children: validation of the computer-assisted image analysis of the retina (CAIAR) program. Investing Ophthalmol Vis Sci, 2009

[13] Mirjana Karanovic, Srdjan Sladojevic, Andras Anderla, Marko Arsenovic. Credit Card Fraud Detection – Machine Learning methods, 2019

# APPENDIX1

**1. Instruction of diabetic retinopathy**

Diabetic retinopathy is an eye disease that can affect people with diabetes. It occurs due to structural damage in the blood vessels that supply the retina, which is the layer of the eye that detects light and then sends this information to the brain where it is converted into an image. If undiagnosed and untreated, advanced diabetic retinopathy has the potential to cause severe vision loss. It usually takes several years for the condition to progress to the point where you are at risk of losing your sight: blood vessels become increasingly damaged as diabetic retinopathy progresses, which leads vision to worsen.

Having elevated blood sugar levels over long periods of time can pose a threat to our eyesight. High blood glucose levels in people with diabetes cause the blood vessels in the retina to become ‘leaky’, which causes hemorrhaging of blood and fluids. This causes our vision to become distorted. In advanced stages of the condition, there is an increase in the number of new, abnormal blood vessels. These new blood vessels are fragile and can cause scarring of the retinal surface, leading to further complications. The stages of diabetic retinopathy are categorized in many different ways, but put simply they are:

* Mild nonproliferative diabetic retinopathy
* Moderate nonproliferative diabetic retinopathy
* Severe nonproliferative diabetic retinopathy
* Proliferative diabetic retinopathy

**2. Diabetic retinopathy stages**

**2.1. Mild non-proliferative diabetic retinopathy**

This is the earliest stage of diabetic retinopathy – ‘non-proliferative’ means that the eye is not making new blood vessels, a process known as ‘proliferation’. The blood vessels are weakened and there are one or more microaneurysms present — tiny balloon-like bulges in the vessel walls from where blood and fluid leak into the retina. The changes are subtle and at this stage, carry no threat to your vision.

Because symptoms are either mild or non-existent at this stage, people with diabetes are advised to undergo yearly eye checks. This check is called a diabetic retinopathy screening, where your eyes will be dilated and images (known as digital retinal photography). This will reveal any changes in the retina.

No specific treatment is recommended for background diabetic retinopathy. However, diligent blood sugar control can prevent the condition from progressing. Also, keeping your blood pressure within the target range (with lifestyle changes or medications, if necessary) can delay or prevent the progression of retinopathy and reduce the risk of vision deteriorating. A yearly follow-up is usually recommended for people with background diabetic retinopathy.



Appendix Picture 1.1 Illustrative image for mild NDR

**2.2. Moderate non-proliferative diabetic retinopathy**

As diabetic retinopathy progresses, blood vessels that bring nourishing blood to the retina swell and distort. These vessels are blocked from transporting blood normally, resulting in hypoxia (lack of oxygen), while multiple microaneurysms (tiny swellings) are present in the walls of the blood vessels. When these microaneurysms rupture, it causes hemorrhages. Fluid may then begin collecting in the macula, which is the central area of the retina. This complication is known as ‘diabetic macular oedema’6. The macula is responsible for ensuring our vision is ‘sharp’: that we can see and read in great detail and have the best possible color vision. Macular oedema may cause blurred vision. Other symptoms of moderate non-proliferative retinopathy can include floaters (spots or strings in our vision), dark areas in our vision, problems differentiating colors, and vision that fluctuates in quality.

Specific treatment is usually not necessary for non-proliferative diabetic retinopathy. At this stage of the disease, controlling modifiable risk factors is recommended to prevent further damage from occurring. Regular monitoring with dilated eye examinations is very important because early diagnosis and treatment can help prevent blindness in over 90% of cases.

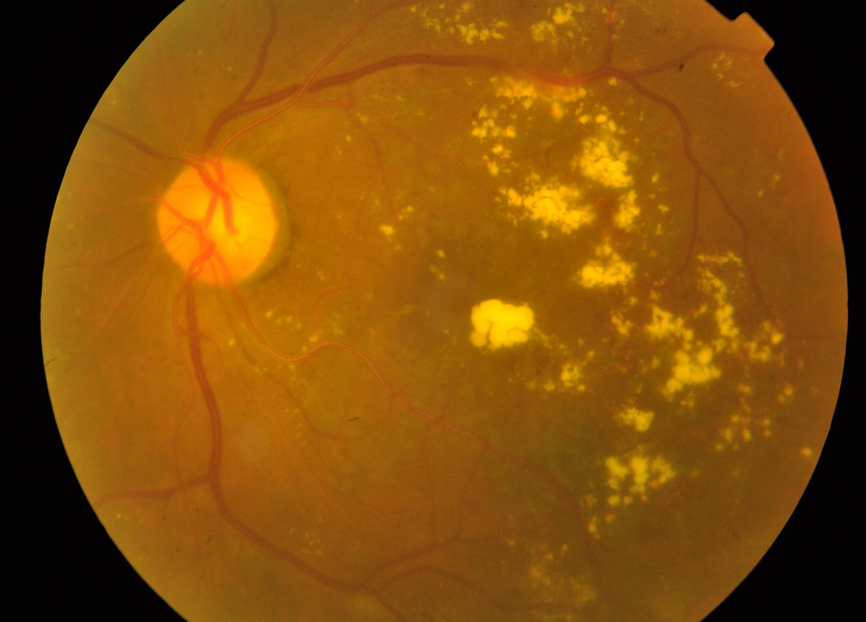
****

Appendix Picture 1.2 Illustrative image for moderate NDR

**2.3. Severe non-proliferative diabetic retinopathy**

This stage represents the most severe form of non-proliferative diabetic retinopathy. It is characterized by the presence of cotton wool spots on the retina due to damage to the nerve fibers. These are severe abnormalities in the retina vessels that appear as fluffy white patches on the retina. At this stage, more blood vessels are blocked, depriving extensive areas of the retina of blood and oxygen. Microaneurysms are present in all four quadrants of the eye, which are sections of the eye that each work to interpret what you see within your total field of vision. Venous beading also occurs, which is where the veins in the eye appear like strings of sausages.

Approximately 50 percent of people with severe non-proliferative diabetic retinopathy will progress to proliferative diabetic retinopathy within one year if the disease takes its natural course. People at this stage of diabetic retinopathy may require dilated eye examinations as often as every two to four months.

****

Appendix Picture 1.3 Illustrative image for severe NDR

**2.4. Proliferative diabetic retinopathy**

At the most advanced stage of diabetic retinopathy, circulatory problems deprive large parts of the retina of oxygen. This can lead to the development of new blood vessels. However, these new blood vessels are abnormally fragile, tend to leak, and can bleed (this is known as proliferation). They grow into the vitreous humor, the gel-like substance present in the eye, and cause hemorrhages. Scar tissue formation may then occur, leading to contraction, retinal detachment, and permanent vision loss.

To preserve vision, this stage of diabetic retinopathy requires specific treatment. There are several treatments available for proliferative diabetic retinopathy, including laser treatments, eye injections, and surgery. Laser treatment works by cauterizing areas of the retina to shrink the abnormal blood vessels. Eye injections with anti-VEGF drugs are effective in treating macular edema and slowing the progression of diabetic retinopathy. Vitrectomy surgery may also be performed at this stage to treat severe bleeding.

****

Appendix Picture 1.4 Illustrative image for severe Proliferative DR