

## UNDERGRADUATE PROJECT REPORT

Project Title:	Supervised Learning for Retinal Disease Classification using Fundus Images
Surname:	Gu
First Name:	Weixuan
Student Number:	202118020326
Supervisor Name:	Dr Grace Ugochi Nneji
Module Code:	CHC 6096
Module Name:	Project
Date Submitted:	May 6, 2024

**Chengdu University of Technology Oxford Brookes College**

**Chengdu University of Technology**

**BSc (Single Honours) Degree Project**

Programme Name: Software Engineering

Module No.: CHC 6096

Surname: Gu

First Name: Weixuan

Project Title: Supervised Learning for Retinal Disease Classification using Fundus Images

Student No.: 202118020326

Supervisor: Dr. Grace Ugochi Nneji

Date submitted: 6<sup>th</sup> May 2025

*A report submitted as part of the requirements for the degree of BSc (Hons) in Computer Science*

*At*

**Chengdu University of Technology Oxford Brookes College**

## Declaration

### Student Conduct Regulations:

Please ensure you are familiar with the regulations in relation to Academic Integrity. The University takes this issue very seriously and students have been expelled or had their degrees withheld for cheating in assessment. It is important that students having difficulties with their work should seek help from their tutors rather than be tempted to use unfair means to gain marks. Students should not risk losing their degree and undermining all the work they have done towards it. You are expected to have familiarised yourself with these regulations.

<https://www.brookes.ac.uk/regulations/current/appeals-complaints-and-conduct/c1-1/>

Guidance on the correct use of references can be found on [www.brookes.ac.uk/services/library](http://www.brookes.ac.uk/services/library), and also in a handout in the Library.

The full regulations may be accessed online at

<https://www.brookes.ac.uk/students/sirt/student-conduct/>

If you do not understand what any of these terms mean, you should ask your Project Supervisor to clarify them for you.

**I declare that I have read and understood Regulations C1.1.4 of the Regulations governing Academic Misconduct, and that the work I submit is fully in accordance with them.**

Signature



Date .....6<sup>th</sup> May 2025.....

REGULATIONS GOVERNING THE DEPOSIT AND USE OF OXFORD BROOKES UNIVERSITY MODULAR PROGRAMME PROJECTS AND DISSERTATIONS

Copies of projects/dissertations, submitted in fulfillment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Oxford Brookes University Library.

I agree that this dissertation may be available for reading and photocopying in accordance with the Regulations governing the use of the Oxford Brookes University Library.

Signature



Date .....6<sup>th</sup> May 2025.....

## **Acknowledgment**

The author would like to express heartfelt gratitude to his professors, friends, and family for their continuous and firm support, encouragement, and precious advice throughout the preparation of this project. Special thanks go to Tina, Jojo, and Grace, whose insightful suggestions and thoughtful feedbacks at every stage of the project were indispensable in guiding the direction and ensuring the successful completion. Furthermore, the author wishes to express gratitude to Grace particularly for her meticulous guidance and patience, which helped the author overcome numbers of difficulties and refine the project. Her expertise and thoughtful guidance provided clarity and direction during the most challenging moments. The author also extends his deepest gratitude to the members of the school council from both CDUT and Oxford Brooks for their constructive input, which provides a lot of resources to help refine and elevate the quality of this work. Furthermore, the encouragement and understanding from the author's family and friends have been a constant source of strength and motivation, allowing him to persevere through the rigorous demands of this project. Without their support, both emotional and intellectual, this project would not have been possible. The author is deeply grateful for their unwavering belief in his abilities and for being there through every step of this journey.

# Table of Contents

Declaration.....	ii
Acknowledgment.....	iii
Table of Contents .....	iv
List of Figures .....	vi
List of Tables.....	viii
Abstract .....	ix
Abbreviations .....	x
Glossary .....	xi
Chapter 1 Introduction.....	1
1.1    Background.....	1
1.1.1    Risk and Factor .....	1
1.1.2    Challenge .....	3
1.2    Aim .....	3
1.3    Objectives.....	4
1.4    Project Overview .....	5
1.4.1    Scope .....	5
1.4.2    Audience .....	5
1.4.3    Project Overview .....	6
Chapter 2 Background Review.....	7
2.1    Early DR detection means.....	7
2.2    Machine learning techniques for DR Detection .....	7
2.3    Deep learning methods for detecting DR .....	8
2.3.1    Supervised learning Techniques .....	8
2.3.2    Self-supervised learning Techniques.....	9
2.3.3    Other Researches on Generative Adversarial Network (GAN) and Attention Networks .....	10
Chapter 3 Methodology .....	14
3.1    Approach .....	14
3.2    VGG.....	14
3.3    ResNet.....	16
3.4    Self-Supervised Learning and Supervised Learning .....	18
3.5    Squeeze-and-Excitation attention .....	18

3.6	Evaluation Metrics .....	19
3.7	Dataset.....	20
3.8	Dataset preprocessing.....	21
3.8.1	Image Cropping and Enhancement .....	21
3.8.2	Dataset Balancing .....	22
Chapter 4 Implementation and Result Analyses .....		25
4.1	The implementation of the Gu_CNN model .....	25
4.2	Initial Convolution and Residual Feature Learning.....	25
4.3	Stage-wise Architecture Details .....	26
4.4	Global Pooling and Classification .....	28
4.5	Implementation Details.....	28
4.6	Result .....	30
4.7	Self-Supervised Model .....	34
4.7.1	Differences in Self-Supervised model with Supervised Model .....	34
4.7.2	Self-Supervised Model Results .....	35
4.8	Result comparation.....	38
4.9	Model Visualization .....	38
4.10	GUI.....	39
Chapter 5 Professional Issues .....		42
5.1	Project Management.....	42
5.1.1	Activities .....	42
5.1.2	Schedule.....	45
5.1.3	Project Data Management .....	45
5.1.4	Project Deliverables.....	45
5.2	Risk Analysis .....	45
5.3	Professional Issues .....	46
Chapter 6 Conclusion .....		48
References .....		49

## List of Figures

Figure 1 The neurovascular Unit of the Retina[4].....	2
Figure 2 The five different DR stages increase in severity from left to right: no DR, mild DR, moderate DR, severe DR, and proliferative DR.[5].....	2
Figure 3 Project overview.....	6
Figure 4 Natural history of diabetic retinopathy based on retinal microvascular disease progression and current treatment options.[12] .....	7
Figure 5 Architecture of a Typical method of ML (Random Forest) [14].....	8
Figure 6 Topology of Classical CNN-FC [15].....	8
Figure 7 Pre-trained Supervised Learning Model.....	9
Figure 8 Pre-trained Self-supervised learning model .....	10
Figure 9 Architecture of VGG19.....	15
Figure 10 Architecture of ResNet50 .....	17
Figure 11 Basic module of Residual learning [25].....	17
Figure 12 Structure of SE module[27].....	18
Figure 13 Example images from the APTOS 2019 dataset:.....	21
Figure 14 Comparison of image cropping and enhancement results.....	22
Figure 15 Comparison of dataset distribution before and after expansion.....	24
Figure 16 The general architecture diagram of the model.....	25
Figure 17 Architecture of the Residual Block with SE Attention .....	26
Figure 18 Confusion Matrix of Gu_CNN on Aptos2019 dataset .....	31
Figure 19 ROC curves of Gu_CNN on Aptos2019 dataset .....	32
Figure 20 Precision-Recall curves of Gu_CNN on Aptos2019 dataset.....	32
Figure 21 Training and Validation Loss(up) and Accuracy(down) curves of Gu_CNN on Aptos2019 dataset.....	33
Figure 22 Training and Validation Loss(up) and Accuracy(down) curves of Self-supervised model on Aptos2019 dataset.....	36
Figure 23 Confusion Matrix of Self-supervised model on Aptos2019 dataset.....	37
Figure 24 Precision-Recall curves of Self-supervised model on Aptos2019 dataset .....	37
Figure 25 ROC curves of Self-supervised model on Aptos2019 dataset .....	38

Figure 26 Grad-CAM Visualization .....	39
Figure 27 Home Page.....	40
Figure 28 Dataset shown in home page .....	40
Figure 29 DR Check page.....	41
Figure 30 Gantt Chart .....	45



## List of Tables

Table 1 Summary of Related Works.....	11
Table 2 Overview of images from the APTOS 2019 dataset.....	20
Table 3 Architecture Overview of the Gu_CNN Model Across Different Stages .....	27
Table 4 Summary of Hyperparameters .....	29
Table 5 Classification Report.....	30
Table 6 Classification Report for Self-Supervised Model.....	35
Table 7 Table of Activities .....	42

## **Abstract**

Diabetes is a chronic disease which can lead to serious complications, including the diabetic retinopathy (DR). DR is a severe illness and it is the main cause of blindness among people who are under the age of 50. DR happens when long-term high blood sugar level damaged the retinal blood vessels, and that will lead to swelling, fluid leakage, and finally vision impairment. In this situation, early detection and accurate classification of DR are important for early effective intervention. This project introduces Gu\_CNN model, a lightweight convolutional neural network (CNN) designed for the classification of the severity of DR's condition. The model combines VGG-style convolutional layers for early visual feature extraction, residual blocks of ResNet for deep feature learning, and Squeeze-and-Excitation (SE) attention modules to improve channel feature recalibration. The training was done on the APTOS 2019 Blindness Detection dataset, all the pictures have gone through structured preprocessing before use, such as image normalization, data augmentation and class rebalancing. As the result, the model achieved a test accuracy of 89.80% and a macro F1-score of 0.8988. Additional evaluation using confusion matrix analysis, ROC curves, and Precision-Recall curves confirmed the model's robustness and stable convergence. This work demonstrates the potential of combining classical feature extraction with attention mechanisms to build efficient and interpretable models for DR classification. Future research may focus on enhancing generalization performance under limited data conditions and further reducing loss values to improve accuracy.

**Keywords:** *Diabetic Retinopathy, Deep Learning, Convolutional Neural Networks, Residual Learning, Attention Mechanism, Medical Image Classification, Squeeze-and-Excitation*

## **Abbreviations**

**DR:** Diabetic Retinopathy

**CNN:** Convolutional Neural Network

**SE:** Squeeze-and-Excitation

**VGG:** Visual Geometry Group

**ResNet:** Residual Neural Network

**APTOS:** Asia Pacific Tele-Ophthalmology Society

**ML:** Machine Learning

**DL:** Deep Learning

**GAN:** Generative Adversarial Network

**OCT:** Optical Coherence Tomography

**ROC:** Receiver Operating Characteristic

**AUC:** Area Under the Curve

**PR:** Precision-Recall

**GAP:** Global Average Pooling

**ReLU:** Rectified Linear Unit

**BN:** Batch Normalization

**TP:** True Positive

**TN:** True Negative

**FP:** False Positive

**FN:** False Negative

## Glossary

This section defines the keywords and terminology from the Abstract and other relevant parts of the report:

**Diabetic Retinopathy (DR):** A diabetes complication affecting the retina, leading to vision impairment or blindness due to damaged blood vessels.

**Convolutional Neural Network (CNN):** A deep learning model designed for image processing, using convolutional layers to extract hierarchical features.

**Residual Learning:** A technique in deep learning where skip connections (residual blocks) mitigate gradient vanishing, enabling deeper networks (e.g., ResNet).

**Attention Mechanism:** A module (e.g., Squeeze-and-Excitation) that dynamically recalibrates channel-wise feature responses to enhance model focus on relevant patterns.

**Macro F1-Score:** The unweighted average of F1-scores across all classes, used to evaluate model performance on imbalanced datasets.

**Fundus Images:** Photographs of the interior surface of the eye, including the retina, used for diagnosing DR.

**Data Augmentation (DA):** Techniques (e.g., rotation, scaling) to artificially expand training data and improve model generalization.

**Gradient Clipping:** A method to prevent exploding gradients by capping gradient values during backpropagation.

**Class Rebalancing:** Adjusting sample weights or oversampling minority classes to address dataset imbalance.

**Note:** Common terms like "accuracy," "loss," or "dataset" are omitted as they are standard in computer science. Definitions align with the report's technical context and aim to clarify domain-specific jargon.

## **Chapter 1 Introduction**

### **1.1 Background**

Diabetes is a chronic disease. Its pathogenesis is that the pancreas cannot produce or secrete enough insulin to control blood sugar, which leads to elevated blood sugar levels. This disease is not only widespread in society, but also poses a high health risk.[1] One of the most severe complications is diabetic retinopathy (DR), which is the leading cause of blindness among people under the age of 50. In DR, consistently high blood sugar damage the retinal blood vessels, leading to swelling and leakage of blood or fluid. The most harmful vision loss typically occurs when the central retina swells, and this will seriously affect the patient's daily life.[2][3]

#### **1.1.1 Risk and Factor**

As far as the pathogenesis of DR is concerned, diabetes contributes directly to the development of DR by causing vascular damage in the retina. These damages mainly occur in the Retinal ganglion cells, Photoreceptor cells, Endothelial cells and Pericytes shown in Figure 1. Uncontrolled blood glucose is a major risk factor, i.e., chronic high blood glucose levels due to insufficient insulin production or secretion initiate this destructive process.[2] In short, diabetes-induced retinal dysfunction can be viewed as damage to the retinal neurovascular unit.[4][5].

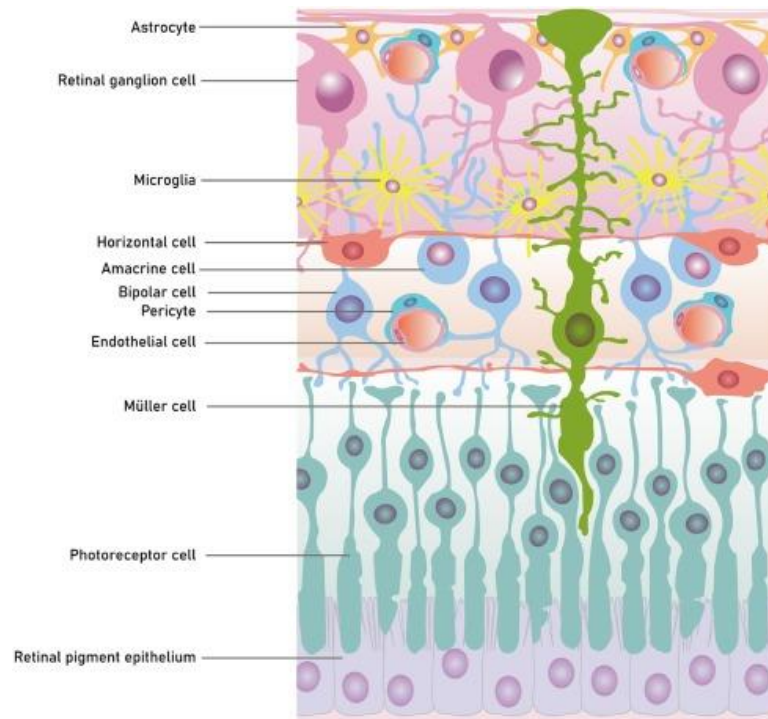


Figure 1 The neurovascular Unit of the Retina[4]

DR begins with retinal injury without visible microvascular abnormalities, small changes in the blood vessels (Non-proliferative retinal microvascular changes such as microaneurysms and punctate intraretinal hemorrhages), which is referred to as mild DR. At this stage, full recovery is possible. If proper care is not taken, after a few years, it progresses to moderate DR, at which point vascular leakage may begin. If not given adequate attention, pre-proliferative changes characterized by the presence of “cotton wool” spots will occur, and the disease will further progress to severe DR. Worse, it can evolve into the appearance of proliferative DR where abnormal neovascularization occurs, and can lead to irreversible visual impairment or even permanent blindness, especially if the swelling involves the central retina.[4] [5]

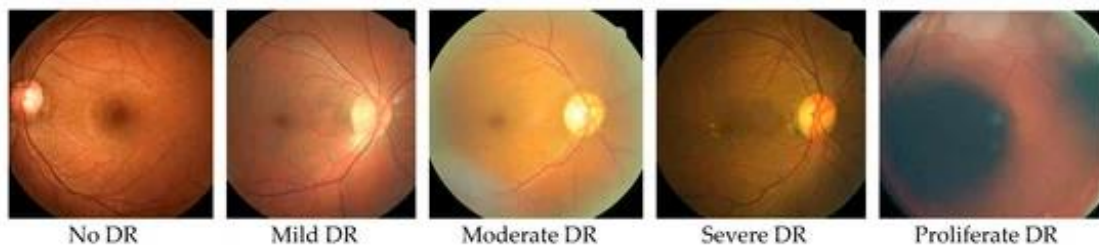


Figure 2 The five different DR stages increase in severity from left to right: no DR, mild DR, moderate DR, severe DR, and proliferative DR.[5]

Even more worrisome is the global burden of diabetes is increasing. According to Cho et al., there were about 451 million people aged 18–99 with diabetes in 2017, a number projected to increase to 693 million by 2045. This upward trend indicates that the number of patients at risk for DR will likewise grow, underscoring the importance of early detection and intervention.[5]

### **1.1.2 Challenge**

While machine learning (ML) and deep learning (DL) show potential in medical image analysis, their practical application in the identification of diabetic retinopathy (DR) still faces the following key challenges:

The development of DR Model is highly dependent on standardized retinal image data, but there are generally problems in the actual data set, such as unbalanced category distribution (such as the scarcity of advanced cases), inconsistent expert labeling (different physicians have significant differences in judging the stage of lesions), and feature variation caused by multi-source imaging equipment. These data limitations may lead to model training bias, reduce the detection sensitivity of early lesions and severe cases, and thus affect the reliability of clinical diagnosis.

In addition, technology deployment faces multiple real-world barriers. The mismatch between hardware computing power needs and the resources of primary care institutions, the compatibility of algorithms with hospital information systems (such as PACS, electronic medical records), and the lack of a standardized regulatory framework for AI-assisted DR Diagnosis. These systemic challenges keep most studies in the experimental stage and make it difficult to achieve large-scale clinical response.[2]

## **1.2 Aim**

As VGG has deeper depth and denser convolutional layers, however it is prone to the problem of gradient vanishing, while ResNet can effectively reduce the gradient vanishing problem through residual connections. In this study, it is tried to optimize the model by using the first few layers of VGG for extracting basic features, and then combining it with ResNet's residual structure for extracting advanced features to take advantage of their strengths and overcome the weaknesses. In addition, in the paper of S. Mishra et al.[7] it is mentioned that the pre-trained model has a greater improvement compared to the normal VGG16, in this study we will also try to pre-train the combined model and check the final result.[6]

### **1.3 Objectives**

The project will propose a supervised CNN model (combining VGG and ResNet) for the detection of DR and will be based on the dataset provided in the Asia Pacific Tele-Ophthalmology Society (APTOS) competition 'APTOS 2019 Blindness Detection' competition organized by the Asia Pacific Tele-Ophthalmology Society (APTOS) on Kaggle website.[a]

The author will segment, preprocess and enhance the dataset before training to improve the accuracy of the experimental results. After obtaining the best and stable experimental results, this project will compare and analyze the data from different directions. The evaluation of the models built in this project will be based on 'Accuracy', 'Loss', 'Precision', 'Recall' and 'F1-Score', and adhere to a rigorous scientific attitude to draw the corresponding conclusions.



## **1.4 Project Overview**

This subsection will explain what the significance of this project is and what benefits it can bring from both Scope and Audience perspectives. The general flow of the project will also be shown.

### **1.4.1 Scope**

The application of Convolutional Neural Networks (CNNs) in medical imaging, particularly for diabetic retinopathy (DR) detection, faces several challenges though it succeed in general image classification.[7] While VGG19 combined with GAN has shown great use in improving accuracy and reducing dataset requirements,[8] VGG19's depth often leads to gradient vanishing, hindering updates to shallow-layer features and overall learning efficiency. Additionally, the model's high parameter count and complexity can cause overfitting and make it computationally demanding. To solve these limitations, this project proposes a CNN model that introduces residual learning to VGG by combining ResNet to improve network performance.[9][10] This reduces parameters through depth wise convolutions and incorporates skip connections to enhance information flow, thereby improving efficiency and feature capture.

The significance of this study is:

- Contribute to the medical field
- Reduce training costs
- Help poor countries save money
- Improve efficiency in detecting whether a DR diagnosis is confirmed or not
- Reduce the rate of misdiagnosis
- Reduce stress for the healthcare professionals involved
- More efficient and accurate categorization of DR conditions

### **1.4.2 Audience**

The beneficiaries of the project are as follows:

- ◆ Healthcare professionals: Ophthalmologists will benefit from increased accuracy and efficiency in DR diagnosis, reducing repetitive works and significantly reducing their workload.
- ◆ Hospitals: Streamlining the diagnostic process, reducing overall overheads, improve diagnostic efficiency and reduce misdiagnosis rates, as a result to improve patient

care and outcomes, and indirectly reducing the likelihood of patient-physician conflict due to misdiagnosis.

- ◆ DR patients: Benefiting from the project's models, patients can be diagnosed faster and more accurately, reducing the incidence of letting their condition worsen and lowering the risk of blindness.
- ◆ Machine learning researchers: future researchers can continue to propose more valuable and better models based on this project, promoting the lasting development of machine learning.

### 1.4.3 Project Overview

The project primarily revolves around using a deep learning model to diagnose diabetic retinopathy (DR). First, the author retrieves data from online datasets and performs preprocessing, which includes data enhancement, sampling, and label reassignment. The processed data is then used to train the initial model, and it will be regularly updated as new and better datasets become available later. The trained model is deployed in a graphical user interface (GUI), where users can upload fundus images and get the prediction results from the system. Additionally, the system ensures user privacy by not saving uploaded images and focusing solely on analyzing and predicting results. Besides, it is important to let the user note that the prediction results provided by the system are for reference purposes only and cannot be the replacement of a real human doctor's diagnosis. The flowchart of the project overview is shown in Figure 3:

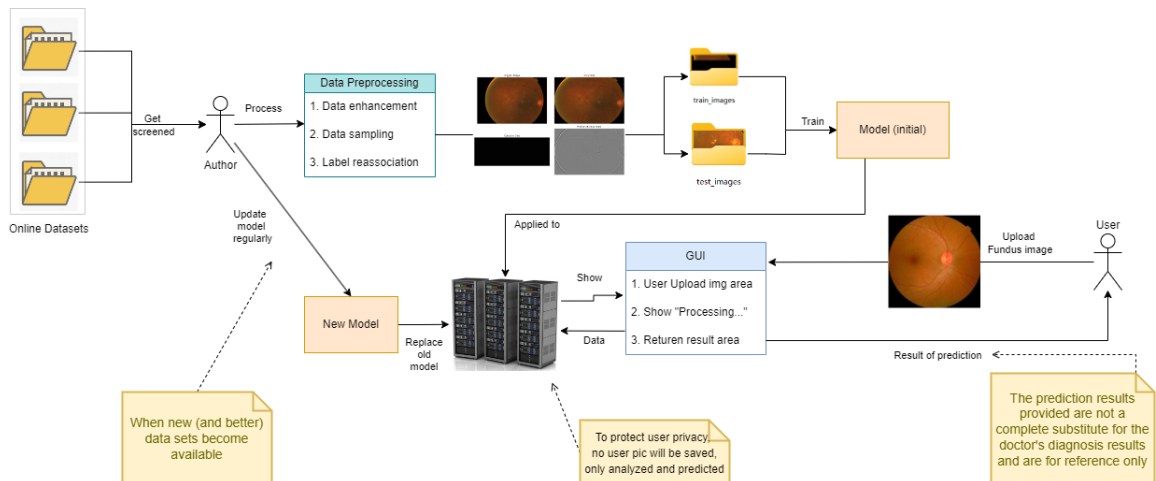


Figure 3 Project overview

## Chapter 2 Background Review

This section will discuss the contributions made by other researchers in identifying DF, and provide a list comparing their models.

### 2.1 Early DR detection means

Previously, detection of DR mostly relied on traditional ophthalmological methods, such as direct observation of the retina or the use of fundus photography, as well as focusing on the blood sugar levels. These methods allowed doctors to identify lesions by looking at physical changes in the retina, or changes in blood composition. As technology advanced, optical coherence tomography (OCT) was introduced, providing more detailed images of the retinal structure and helping to detect wounds earlier. However, these traditional methods have limitations, such as being highly dependent on the experience and skill of the doctor, and may not be sensitive enough to identify early lesions.[11]

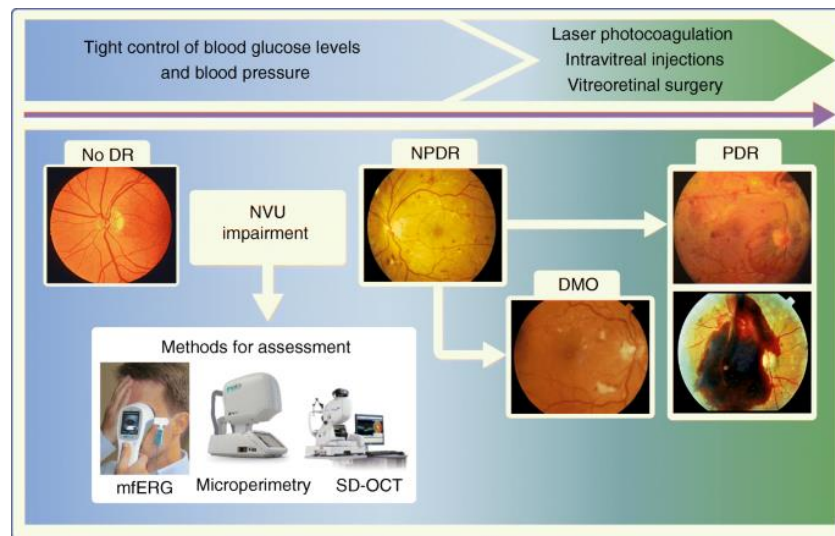


Figure 4 Natural history of diabetic retinopathy based on retinal microvascular disease progression and current treatment options.[12]

### 2.2 Machine learning techniques for DR Detection

In later years, machine learning techniques began to be applied to the detection of DR. The machine learning algorithms, for example like support vector machines (SVMs) and random forests, were used to analyze fundus images to check for any early signs of DR. These algorithms were trained on a large number of retinal images and were able to detect lesions with some sensitivity and specificity. However, while traditional machine learning methods can only just possible to reach parity with a human clinician, ML detection may

require more manual design for feature extraction and may have limitations in dealing with complex feature classes. [13]

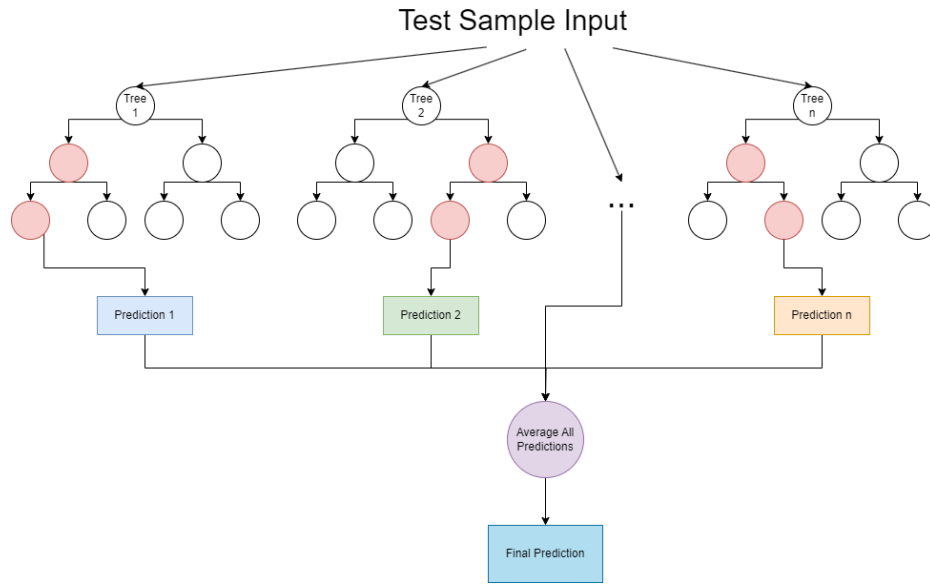


Figure 5 Architecture of a Typical method of ML (Random Forest) [14]

### 2.3 Deep learning methods for detecting DR

The advent of deep learning techniques, especially CNNs, has revolutionized DR detection. CNNs are able to automatically learn complex feature representations from images, improving the accuracy of detection.[14] As some of the models shown in Table 1 above, they all show considerable accuracy, sensitivity and specificity. However, it is worth noting book that the vast majority of DR's datasets currently have some imbalance problem, which may lead to the models encountering some problems during the training process, such as Loss is too high as well as the accuracy cannot be improved.

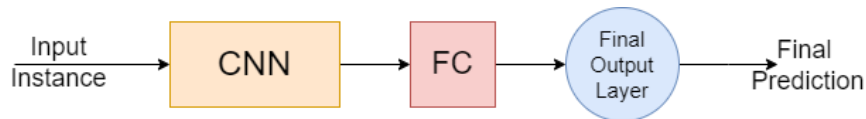


Figure 6 Topology of Classical CNN-FC [15]

#### 2.3.1 Supervised learning Techniques

In supervised learning, model learns a prediction or classification task from training data with known inputs and corresponding outputs. This approach relies on a large amount of labelled data so that the model can learn the mapping relationship between input features and output labels.

The application of supervised learning to DR detection includes the use of deep learning models such as CNNs that are trained on a large number of labelled fundus images to identify lesion features and perform classification. The advantage of this approach is that the models can learn accurate feature representations, but the disadvantage is that it requires a large amount of labelling work, which is particularly challenging in the field of medical images

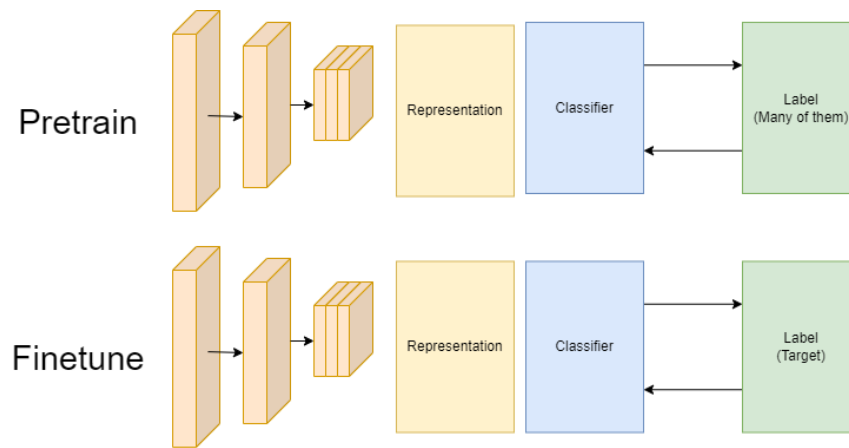


Figure 7 Pre-trained Supervised Learning Model

### 2.3.2 Self-supervised learning Techniques

Self-supervised learning is a special type of unsupervised learning in which models are trained by self-generating labels. In other words, the model uses the input data itself to create labels and then uses these labels to learn.

In DR detection, self-supervised learning can be used to generate internal labels by designing specific auxiliary tasks, such as pre-training an encoder with a contrast learning algorithm and then re-training the encoder and classifier on a small amount of labelled data. The advantage of this approach is that it can reduce the dependence on a large amount of labelled data, which helps to solve the problem of insufficient labelled data for medical images. Self-supervised learning models, such as SimCLR-DR, have shown the ability to overcome the problem of insufficient training data in DR detection and outperform traditional migration learning

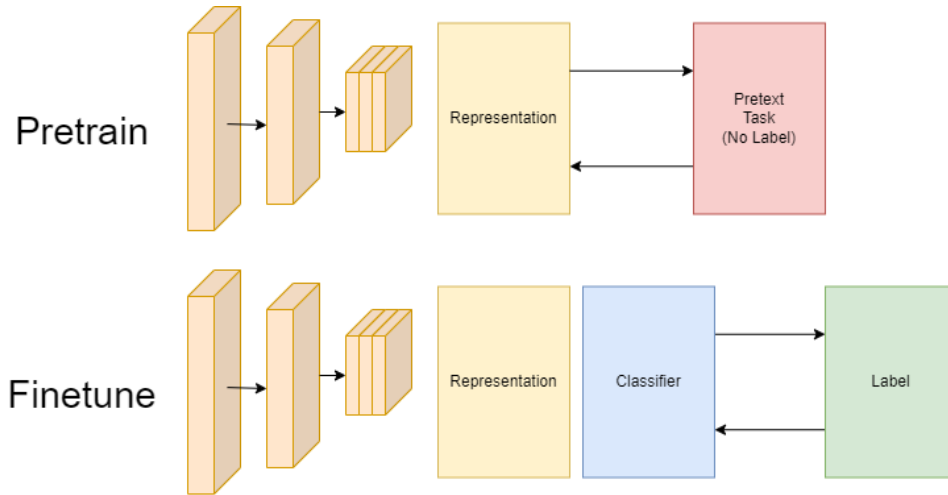


Figure 8 Pre-trained Self-supervised learning model

### 2.3.3 Other Researches on Generative Adversarial Network (GAN) and Attention Networks

Vaish et al.[16] used an improved ResNet-50 deep learning model to classify diabetic retinopathy. They used an improved pre-trained ResNet-50 model to develop a deep model and used the Adam optimizer to accelerate the convergence process. Their model achieved an accuracy of 74.18%, sensitivity of 85.08%, specificity of 87.50% and F1-score of 81.35% on the Kaggle APTOS dataset.

The approach adopted by Kabilan et al.[17] is to combine Deep Convolutional Generative Adversarial Network (DCGAN) with VGG19. Their idea is to perform preprocessing of retinal images through Contrast Constrained Adaptive Histogram Equalisation (CLAHE) technique where the contrast features of the image are designed to be identified from the original retinal image to identify clear vascular structures, which in turn helps the DCGAN to synthesise the image and finally allows VGG19 to produce accurate results. The results of their model on the unmentioned Dataset are 88.93% Accuracy, 89.68% Sensitivity, 88.31% Precision, 86.72% F1 Score, and 85.31% Recall, while the results obtained using SVM on the same Dataset were 84.19%, 83.1%, 85.41%, 86.0%, and 84.88%. Same, The result of CNN is 86.23%, 84.2%, 85.21%, 87.29, and 84.68%.

He et al.[18] proposed a new category attention block (CAB) for unbalanced DR data distributions, which explores more discriminative regional features for each DR class and treats each category equally. To capture more detailed information about small lesions, they also proposed the global attention block (GAB), which exploits detailed and category-

independent global attention feature maps of fundus images. They tested it on DDR dataset, Messidor dataset, and EyePACS dataset, and the best test results were in 85.69%, 86.68%, and 86.18% accuracy.

Deshpande et al.[19] used a pre-trained model, Inception V3. It was fine-tuned to add a custom classification layer consisting of a GlobalAveragePooling2D layer followed by two fully connected layers with ReLU activation and a SoftMax activation layer. Their model was trained and tested using two publicly available datasets EyePACS and APTOS 2019. Their model obtained 74.28% and 73.81% accuracy and F1 score on the EyePACS dataset and 81.61% and 80.21% accuracy and F1 score on the APTOS 19 dataset, respectively. [20]

The CNN model of Matthew et al.[21] was constructed using migration learning from an existing model, EfficientNet-B0, which had been pre-trained using the ImageNet dataset. The base model was then modified by adding tuning to the parameters and adding several new layers to the model. The results show that EfficientNet-B0 has an accuracy of 81.52%, but a relatively low F1 score of 67% due to data imbalance. Table 1 summarizes the findings and potential outcomes of the different researchers.

Table 1 Summary of Related Works

Author	Datasets	Methods & Models	Results	Limitations
Vaish et al. [16]	Kaggle APTOS	pre-trained ResNet-50, Adam optimizer	accuracy of 74.18%  sensitivity of 85.08%  specificity of 87.50%  F1-score of 81.35%	The model's dependence on input image quality, unevenness of data set categories may affect model generalization, and the improved ResNet-50 model, although excellent, may

				still be limited by the optimizer and hyperparameter selection on the training effect.
Kabilan et al.[9]	Data is obtained from kaggle and been processed. Dataset name is not mentioned.	DCGAN+VGG19	88.93% Accuracy 89.68% Sensitivity 88.31% Precision 86.72% F1 Score 85.31% Recall	Relying on GAN to generate data may affect authenticity; VGG19 model is old and has low computational efficiency. Actual clinical applicability has not been verified.
He et al. [17]	DDR, Messidor, EyePACS	CABNet on mainstream CNN Architecture	Accuracy: 85.69% on DDR, 86.68% on Messidor, 86.18% on EtePACS	Only image-level supervision resulted in inaccurate localization of small lesions. Inability to identify the specific lesion type (e.g. exudation type); The generalization of the model on a broader dataset was not verified.



Deshpande et al. [18]	EyePACS. APTOS 19	pre-trained model Inception V3	EyePACS: Accuracy 74.28%  F1 score 73.81 %  APTOS 19: Accuracy 81.61 %  F1 score 80.21 %	Sensitivity to input image quality (such as resolution, lighting conditions, and distortion), and the impact of class imbalances and labeling errors in the EyePACS dataset on model accuracy and reliability of results.
Matthew et al. [19]	APTOS 2019 Blindness Detection	EfficientNet-B0	Accuracy 81.52%  F1 score 67%	Data imbalance leads to poor identification of a few classes; Relies on additional hardware (20D lens); Merging categories sacrifices diagnostic details.

## **Chapter 3 Methodology**

### **3.1 Approach**

The CNN model proposed in this study has a combination of VGG (Similar convolution layer) and ResNet (Use the residual layer). The general idea is to achieve the purpose of combining the two parts by sharing shallow convolutional and branching feature extractors.

This new model Gu\_CNN has two main advantages:

- ◆ Reducing the number of parameters: Shared shallow convolution reduces the amount of redundant computation in the initial convolutional layers, resulting in a model with fewer parameters than the fully parallel approach.
- ◆ Extracting multi-level features: VGG is good at details and ResNet is good at global, through this approach it can achieve the complementary fusion of different features.

### **3.2 VGG**

The VGG neural network model is a relatively complete deep neural network model based on CNN. It was constructed by the Computer Vision Group at the University of Oxford.[22] As an example, VGG19 is a deep convolutional neural network proposed by the team Visual Geometry Group from the University of Oxford in 2014, which improves the accuracy of image categorization by increasing the number of network layers (a total of 19) and using smaller convolutional kernels (3x3). VGG19, with its simple structure and high performance, has become a classic architecture in deep learning and is especially suited to handle large-scale image classification tasks.[23]

The network model of VGG-19 is shown below:

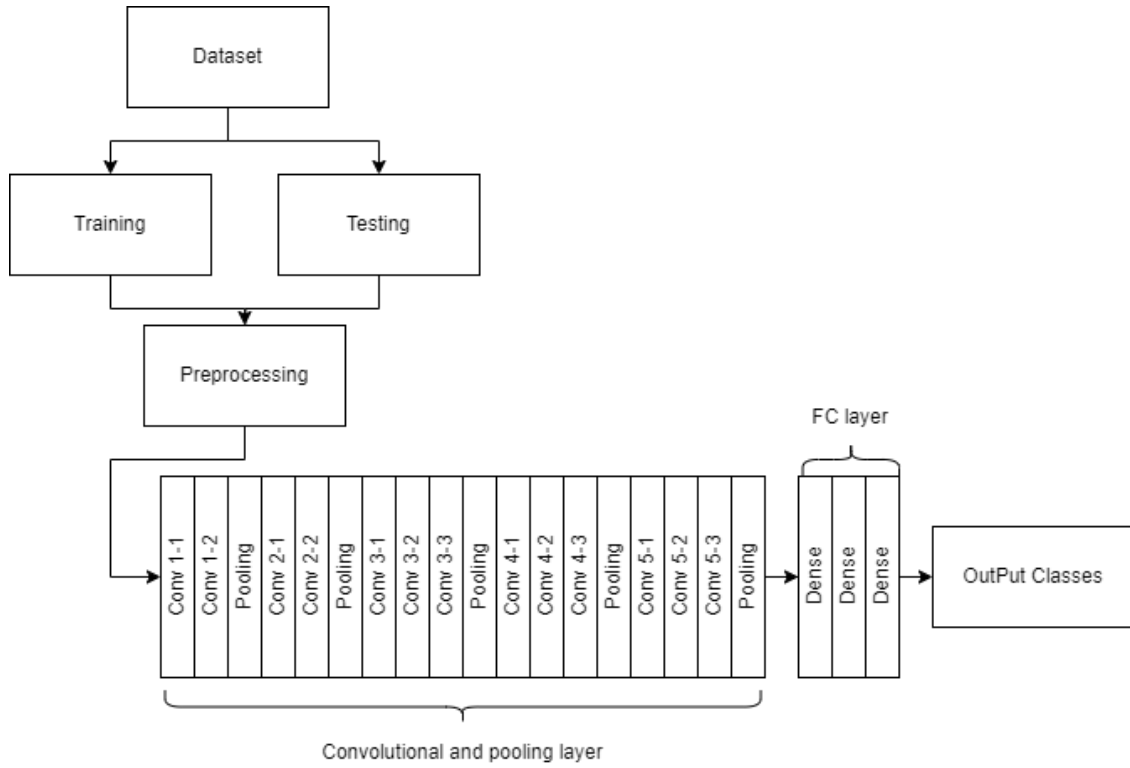


Figure 9 Architecture of VGG19

VGG19 consists of 16 Convolutional Layers (Conv), 5 Max Pooling Layers (Max Pooling), 3 Fully Connected Layers (Fully Connected) and Activation Functions (ReLU). The main features are as follows:

### 3.2.1 Convolutional layer:

Convolutional layer is used to extract the spatial features of the image. Assuming that the input feature map size is  $H \times W$ , the convolution kernel size is  $K \times K$ , the output feature map size is  $H' \times W'$ , the calculation formula is:

$$H' = \frac{H - K + 2P}{S} + 1 \quad (1)$$

$$W' = \frac{W - K + 2P}{S} + 1 \quad (2)$$

In the context of convolutional operations,  $H$  and  $W$  represent the height and width of the input feature map, respectively.  $K$  denotes the size of the convolutional kernel, such as  $3 \times 3$ .  $P$  refers to the padding applied to the input, and  $S$  indicates the stride, which determines the step size of the convolution across the input feature map.

### 3.2.2 Activation function (ReLU):

VGG19 Each convolutional layer is followed by a ReLU activation function, which serves to introduce nonlinearity. the ReLU is calculated as:

$$ReLU(x) = \max(0, x) \quad (3)$$

### 3.2.3 Pooling layer (Max Pooling):

The maximum pooling layer is used to downsample the feature map and is usually added after every two convolutional layers. Assuming a pooling window size of  $P \times P$  and a step size of  $S$ , the output size after pooling is:

$$H' = \frac{H - P}{S} + 1 \quad (4)$$

$$W' = \frac{W - P}{S} + 1 \quad (5)$$

In the pooling operation, the parameters  $H$ ,  $W$ ,  $P$ , and  $S$  keep the same definitions as in the convolutional layer, referring to the height and width of the input, the pooling window size, and the step size, respectively.

## 3.3 ResNet

ResNet (Residual Neural Network) is a deep neural network proposed by a team from Microsoft Research in 2015 that solves the gradient vanishing problem by introducing Residual Blocks and Skip Connections. The Residual Block allows information to be passed directly between network layers, allowing gradients to flow smoothly to the shallow layers when backpropagating.[24] This innovative design not only improves the training effect of the model, but also allows ResNet to build very deep network structures (e.g., 152 layers) without affecting the performance due to gradient vanishing, and thus performs well in tasks such as image recognition and detection.

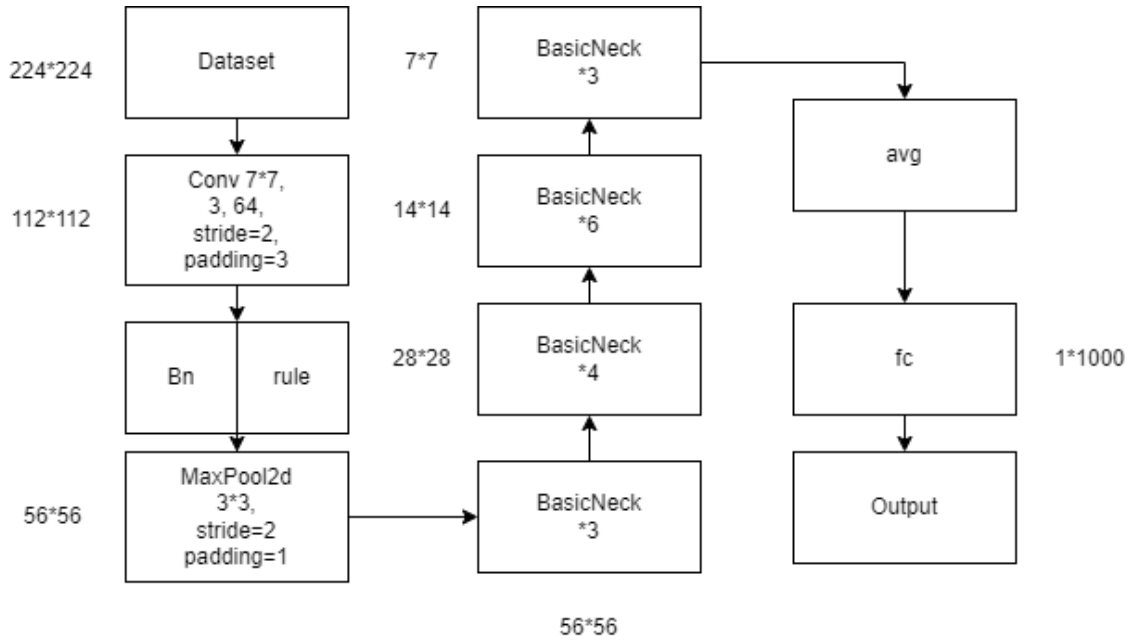


Figure 10 Architecture of ResNet50

### 3.3.1 Residual block (ResBlock)

The 'Residual Blocks' are the core components of ResNet. The formula for each ResBlock is as follows:

$$y = \text{ReLU}(F(x) + I(x)) \quad (6)$$

where  $x$  and  $y$  are the input and output of ResBlock, respectively. The function  $F$  is a combination of a  $3 \times 3$  convolution with step size  $(s, s)$ , batch normalization (BN), ReLU, another  $3 \times 3$  convolution with step size  $(1, 1)$ , and BN. The jump connection  $I$  depends on the dimension of  $F(x)$ ; if the dimension of  $x$  coincides with that of  $F(x)$ , then  $I$  is the identity; otherwise,  $I$  is a  $1 \times 1$  convolution followed by BN, which adjusts the dimension of  $x$  to coincide with that of  $F(x)$ .

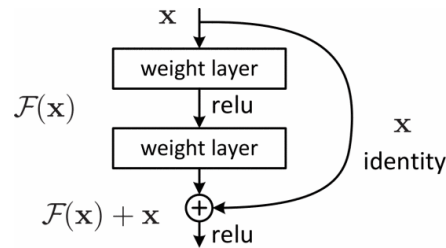


Figure 11 Basic module of Residual learning [25]

### 3.4 Self-Supervised Learning and Supervised Learning

Supervised learning is a traditional machine learning approach in which models are trained using labeled datasets. Each input sample is associated with a corresponding target label, thus allowing the model to learn a direct mapping from input to output. Supervised learning is widely used in tasks such as image classification and regression, where its performance depends heavily on the availability of sufficient high-quality labeled data.

Self-supervised learning, on the other hand, is an approach that eliminates the need for manual labeling by generating supervised signals directly from the data itself. Models are trained to solve predefined pretexting tasks with the goal of learning meaningful feature representations from unlabeled data. Self-supervised learning is increasingly used in areas such as computer vision and natural language processing, especially when labeled data is scarce or expensive to acquire.[26]

### 3.5 Squeeze-and-Excitation attention

The Squeeze-and-Excitation (SE) attention mechanism is a lightweight and efficient attention module characterized by a small number of parameters and a simple structure. It operates by first applying a pooling operation to compress the spatial dimensions of the input feature map, capturing channel-wise global information. The resulting descriptor is then passed through a fully connected layer to perform dimensionality reduction and expansion, enabling the model to learn inter-channel dependencies. At last, an activation function is used to generate a weight vector, which is applied to recalibrate the original feature maps.[27] The structure of SE module is shown as below:

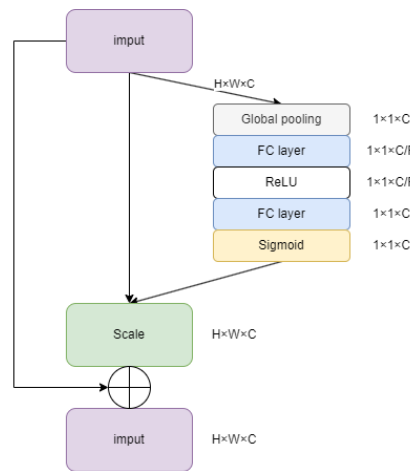


Figure 12 Structure of SE module[27]

### 3.6 Evaluation Metrics

In this section, the key evaluation metrics used to assess the accuracy and reliability of the model will be described. These metrics include: accuracy, loss (cross-entropy loss), recall, precision, and F1-score, each of them provides a different perspective on the model's predictive capabilities.

#### 3.6.1 Accuracy:

The accuracy rate indicates the proportion of samples correctly predicted by the model and is given by the formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

**TP:** True Positives (true classes, number of positive classes correctly classified)

**TN:** True Negatives (True Negatives, number of negative classes correctly classified)

**FP:** False Positives (False Positive Classes, the number of negative classes that are incorrectly classified as positive)

**FN:** False Negatives (False Negative Classes, number of positive classes misclassified as negative)

#### 3.6.2 Loss(Cross-Entropy Loss):

The loss value is a measure of the difference between the model predictions and the true labels, and is given by the formula:

$$Loss = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{ij} \log(p_{ij}) \quad (2)$$

**N** is the number of samples in the batch

**C** is the number of classes

$y_{ij}$  is the true label of the  $i$  th sample in the  $j$  th class (One-hot code)

$p_{ij}$  is the probability that the  $i$  th sample predicted by the model belongs to class  $j$

#### 3.6.3 Recall:

Recall indicates the proportion of all actual positive samples that the model correctly predicts as positive, with the formula:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Recall measures how many of the model's actual positive class samples are correctly classified as positive.

#### 3.6.4 Precision:

Precision indicates the proportion of all samples predicted to be positive that are positive. The formula:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

The higher the Precision, the better the accuracy of the model in predicting the positive class, i.e., fewer false positives (FPs).

#### 3.6.5 F1 Score:

The F1 Score is the reconciled average of Precision and Recall and is commonly used in the evaluation of unbalanced datasets. The formula is:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

The value of F1 Score is between 0 and 1(or 0% to 100%), and the closer it is to 1, the better the model performance. F1 Score is a composite indicator, which is suitable for evaluating the model performance when the distribution of positive and negative samples is not balanced.

### 3.7 Dataset

The dataset[a] contains a large number of retinal images taken using fundus photography under a variety of imaging conditions, and clinicians rated the severity of diabetic retinopathy for each image on a scale of 0 to 4:

0 - no DR, 1 - mild, 2 - moderate, 3 - severe, and 4 - proliferative DR.

The dataset is provided in the form of a compressed package containing train.csv - the training labels, test.csv - the training labels, test.csv - the training labels, and test.csv - the training labels. train.csv - training labels, test.csv - test set, and train.zip - training set images (3662 PNG images), test.zip - public test set images (1928 PNG images).

Table 2 Overview of images from the APTOS 2019 dataset

Class	Data
-------	------



No DR	1805
Mild DR	370
Moderate DR	999
Severe DR	193
Proliferative DR	295
Total	3662

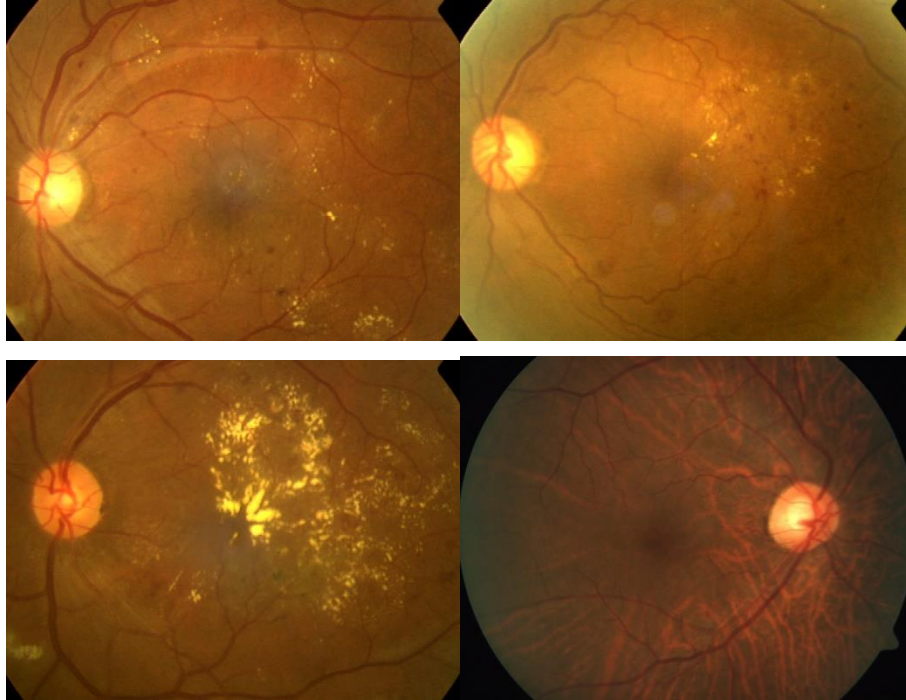


Figure 13 Example images from the APTOS 2019 dataset:

### 3.8 Dataset preprocessing

In this project, systematic preprocessing is performed for APTOS 2019 Blindness Detection dataset to improve the model training effect and generalization ability. The preprocessing is mainly divided into two parts: image cropping and enhancement processing, and dataset expansion.

#### 3.8.1 Image Cropping and Enhancement

Raw retinal images often have a large amount of extraneous background (e.g., black regions), and the direct use of these images may lead to increased training noise and slower model convergence. Therefore, this study designs an image cropping method based on gray-level thresholding and contour detection to retain the effective regions and standardize the image size.

The specific process is as follows:

**Gray-level Cropping:** after converting the image into a grayscale map, a mask is generated based on the luminance threshold (set to 7), and only the regions with luminance higher than the threshold are retained, and then re-sampled according to the original size.

**Contour-based Cropping:** Through binarization and contour extraction techniques, locate the largest outer rectangular region in the image and crop it accordingly.

**Universal Cropping Interface:** In order to be compatible with different strategies, a unified cropping function `universal_crop` is designed, which can be used to select grayscale cropping or contour cropping methods according to the settings.

The cropped image is further enhanced to simulate the multiple variations that may occur during the imaging process. The enhancement method adopts the Ben Graham style (Ben's preprocessing), which improves the local contrast of the image through Gaussian blurring and weighted superposition, and enhances the recognizability of key information such as blood vessels and exudates.

The entire cropping and enhancement process is integrated in the batch processing function `enhance_and_save_images`, which ensures that all training samples are processed under a uniform specification.

The comparison of the effects before and after preprocessing is presented in Figure 14:

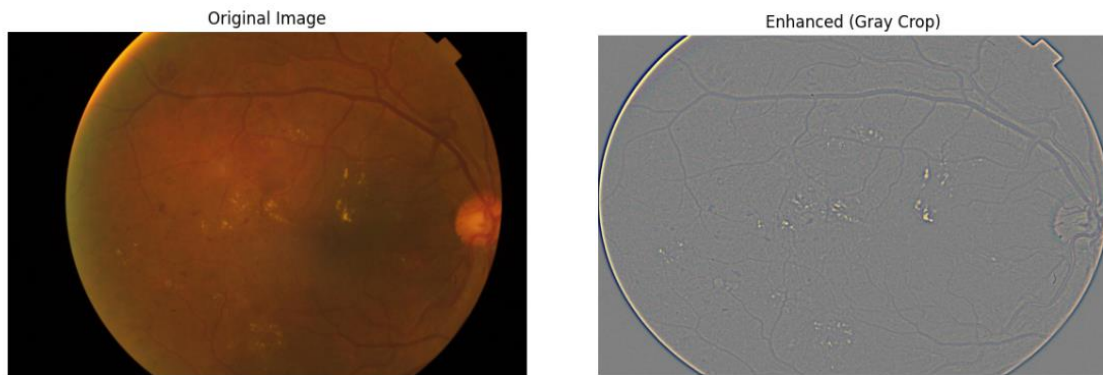


Figure 14 Comparison of image cropping and enhancement results

### 3.8.2 Dataset Balancing

As the sample size of different lesion classes (Label 0-4) in the APTOS 2019 dataset is extremely unbalanced, the details can be seen in the dataset introduction. In order to

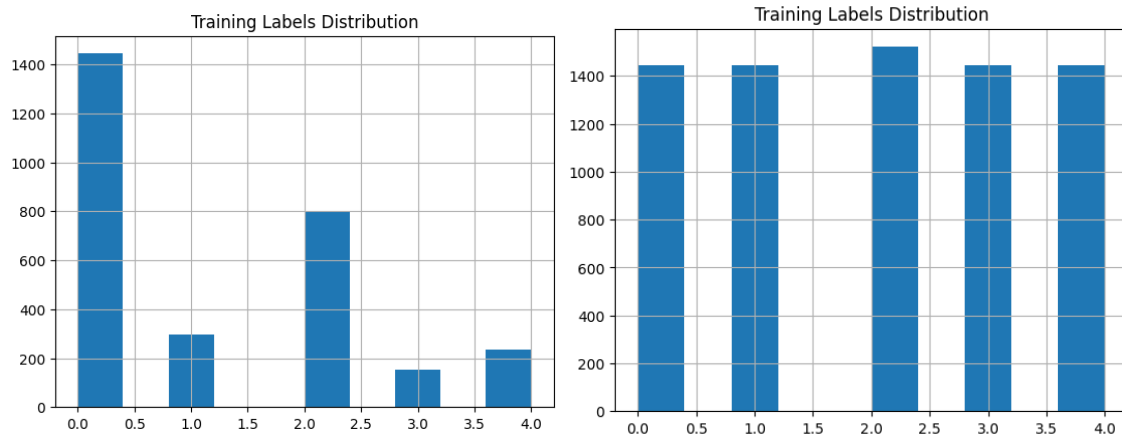
alleviate the problem of category skewness, this study introduces the Data Augmentation (DA) technique to expand the categories with less sample size.

The augmentation process is as follows:

Firstly, the image files that have been processed (after cropping + augmentation) are read. Then, using ImageDataGenerator under the Keras framework, transformations such as rotation ( $\pm 25^\circ$ ), random scaling ( $\pm 15\%$ ), horizontal flipping, and brightness adjustment (0.8-1.2 times) are applied to generate diverse images. Then for each category, the target number of samples is set to be the number of samples of the category with the most samples in the current dataset. Finally, new images are dynamically generated on the basis of the original samples, saved to a specified directory, and the label file (CSV format) is synchronously updated.

In addition, in order to avoid repeated enhancement, the system checks the existing enhanced images before each expansion to ensure that the generation process is efficient and redundancy-free.

The following is the distribution of the expanded dataset:



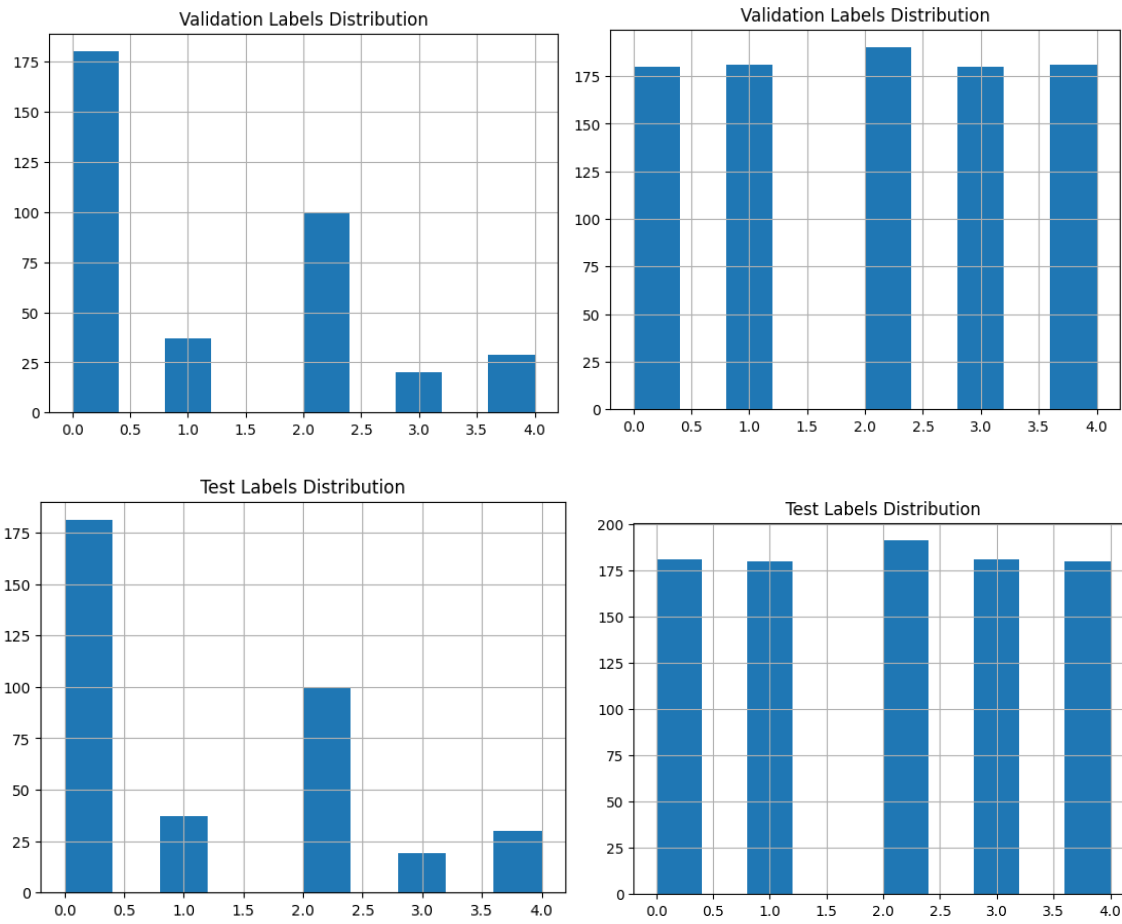


Figure 15 Comparison of dataset distribution before and after expansion

## Chapter 4 Implementation and Result Analyses

### 4.1 The implementation of the Gu\_CNN model

In this project, the core model architecture combines VGG convolutional feature extraction with residual layer of ResNet, while integrating a Squeeze-and-Excitation (SE) attention mechanism to enhance channel feature recalibration.

The overall structure starts with a series of convolutional and residual layers, followed by global pooling and a fully connected classification layer. This hybrid design aims to preserve fine-grained visual patterns in early stages while allowing deeper abstraction and stable gradient flow in later stages. By exploiting local feature sensitivity and global feature generalization, the architecture seeks to achieve high accuracy in diabetic retinopathy classification.

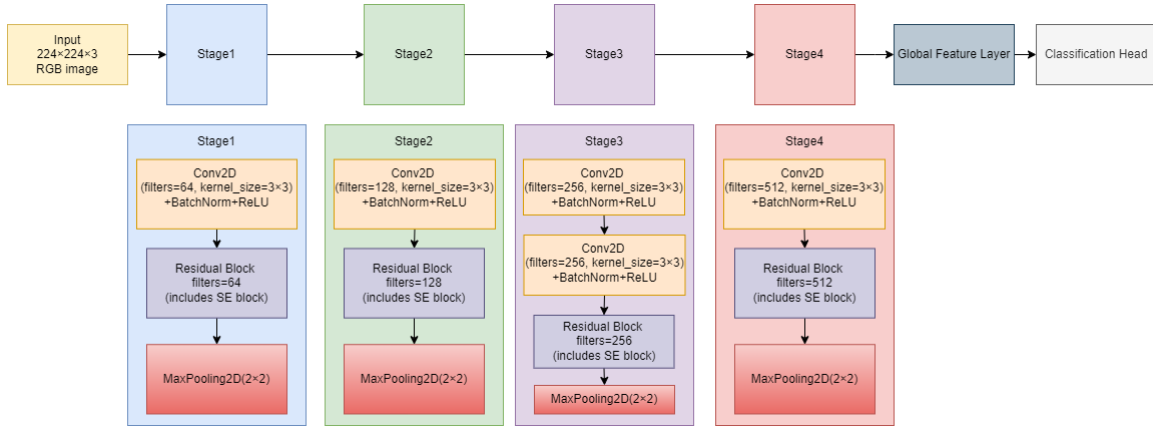


Figure 16 The general architecture diagram of the model

### 4.2 Initial Convolution and Residual Feature Learning

The model starts with a 3x3 convolutional layer and applies 64 filters on the input image followed by batch normalization and ReLU activation function. This initial convolution is used to extract basic low-level features such as edges and texture.

After that, the model enters multiple stages, each consisting of a standard convolution block, a custom residual block, and a MaxPooling layer. In each residual block, which architecture is shown in figure 17, two convolutional layers are superimposed, and each convolutional layer is followed by batch normalization. Residual concatenation is achieved by shortcuts that convert inputs directly to outputs, with the option of 1x1 convolution if the number of filters changes. In addition, an SE attention module is inserted between the

convolution and addition operations, allowing the model to adaptively recalibrate the feature channel response.

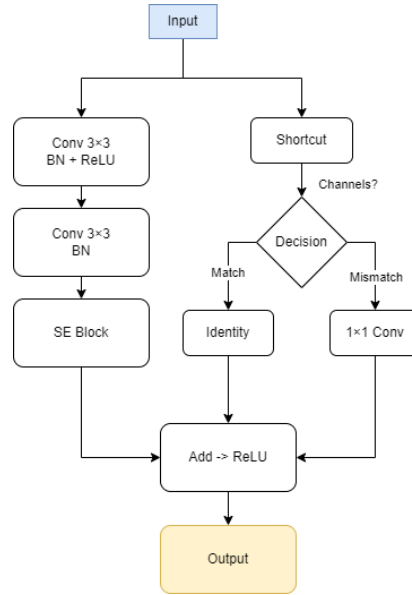


Figure 17 Architecture of the Residual Block with SE Attention

### 4.3 Stage-wise Architecture Details

The detailed progression through the network is divided into four main phases.

In the first phase, a residual block with 64 filters is applied after the initial convolutional block, followed by a MaxPooling operation to reduce the spatial dimension by half.

The second phase continues with a convolutional block containing 128 filters, followed by another residual block containing 128 filters, and a subsequent MaxPooling layer.

The third stage involves two consecutive 3x3 convolutions with 256 filters, followed by a residual block with 256 filters. The MaxPooling layer is again used to downsample the feature mapping.

In the final stage, a convolution with 512 filters is applied, followed by a residual block with 512 filters. After this, MaxPooling is performed to further compress the spatial information.

The gradual increase in the number of filters and the decrease in spatial resolution allows the model to efficiently capture the complex and hierarchical features required for fine-grained diabetic retinopathy classification.

The table of the architecture of the proposed model is shown as below:

Table 3 Architecture Overview of the Gu\_CNN Model Across Different Stages

Stage	Input Size	Operation	Filters	Output Size
0	224×224×3	Conv 3×3 + BN + ReLU	64	224×224×64
0	224×224×64	Residual Block + SE	64	224×224×64
0	224×224×64	MaxPooling 2×2	-	112×112×64
1	112×112×64	Conv 3×3 + BN + ReLU	128	112×112×128
1	112×112×128	Residual Block + SE	128	112×112×128
1	112×112×128	MaxPooling 2×2	-	56×56×128
2	56×56×128	2× (Conv 3×3 + BN + ReLU)	256	56×56×256
2	56×56×256	Residual Block + SE	256	56×56×256
2	56×56×256	MaxPooling 2×2	-	28×28×256
3	28×28×256	Conv 3×3 + BN + ReLU	512	28×28×512
3	28×28×512	Residual Block + SE	512	28×28×512
3	28×28×512	MaxPooling 2×2	-	14×14×512

#### **4.4 Global Pooling and Classification**

After feature extraction, a Global Average Pooling (GAP) layer is applied to reduce each feature mapping to a single value by computing the average across spatial dimensions. This step not only reduces the number of parameters, but also strengthens the spatial invariance, which is required for classification tasks.

The output of the GAP layer is then passed through a fully connected (Dense) layer followed by a Softmax activation function. The final layer outputs five predicted probability distributions for diabetic retinopathy severity (ranging from 0 to 4).

#### **4.5 Implementation Details**

The original dataset is divided into training, validation and testing subsets in the ratio of 8:1:1. Initially, 80% of the samples were assigned to the training set and the remaining 20% was further split evenly into the validation and test sets. Stratified sampling was used during splitting to preserve the original class distribution of all subsets. The data size obtained was 7297 samples for training, 912 samples for validation and 913 samples for testing.

The input dataset is processed before starting the training by normalizing all the pixel values to the range [0,1]. For the training set, various data enhancement techniques were applied using the `imagedatagenerator` class in TensorFlow. The enhancement operations include random rotation up to 20 degrees, horizontal and vertical shifts up to 20% of the image width or height, random clipping transformations, scaling changes in the range of 20%, and horizontal flipping. For both the validation and test sets, only the pixel values are rescaled to ensure that the evaluation results accurately reflect real-world generalizations without artificial variations.

The model was trained for up to 100 epochs with a batch size of 16 and an input image size of 224×224 pixels. The initial learning rate was set to  $5 \times 10^{-4}$ . An early stopping mechanism is also incorporated to stop training if the validation loss does not improve for 8 consecutive validations, and restores the model weights from the epoch that achieved the best validation performance.

To dynamically adjust the learning rate during training, a `CosineDecayRestarts` scheduler was employed. In this schedule, the first decay cycle lasts 5 epochs, with each subsequent cycle doubling in length. After each cycle, the maximum learning rate



decreases by a factor of 0.8. A small minimum learning rate ( $\alpha=1e-6$ ) was maintained to ensure continued learning during the later stages of training.

The optimizer used was Adam, enhanced with gradient clipping (clip norm of 1.0) to prevent gradient explosion, particularly in the presence of highly non-convex loss landscapes. The loss function applied was sparse categorical crossentropy, which is suitable for multi-class classification tasks with integer-encoded labels. Besides, the accuracy was used as the primary evaluation metric throughout training.

All the hyperparameters of this model are summarized in the following table:

Table 4 Summary of Hyperparameters

Hyperparameter	Value/Description
Input Image Size	224 × 224 pixels
Batch Size	16
Optimizer	Adam with gradient clipping (clipnorm=1.0)
Initial Learning Rate	$5 \times 10^{-4}$
Learning Rate Scheduler	Cosine Decay Restarts
First Decay Steps	10 epochs
Multiplicative Factor (t_mul)	2.0
Learning Rate Decay Factor (m_mul)	0.8
Minimum Learning Rate (alpha)	$1 \times 10^{-6}$
Loss Function	Sparse Categorical Crossentropy
Metric	Accuracy
Early Stopping Patience	8 epochs without improvement in validation loss

Class Weights	Computed using Scikit-learn's <code>compute_class_weight</code> based on training set
Training/Validation/Test Split	80% / 10% / 10%, stratified sampling
Data Augmentation (Training)	Rotation (20°), width/height shift (20%), shear (0.2), zoom (0.2), horizontal flip
Data Augmentation (Validation/Test)	Only rescaling to [0, 1] range
Number of Epochs	100

#### 4.6 Result

The Gu\_CNN model was evaluated on the test set, achieving a test loss of 0.4642 and a test accuracy of 89.80%. During training, the model reached a training loss of 0.4624 with an accuracy of 91.38%, while on the validation set, the validation loss and accuracy were 0.4662 and 90.68%, respectively. The small gap between training and validation results indicates a consistent model behavior across different data splits.

The model's F1-score on the test set was 0.8988, with a recall of 0.8980 and a precision of 0.9026. The detailed classification report shows class-wise performance, where classes 0 and 4 exhibited relatively higher precision and recall, while classes 2 and 3 showed comparatively lower but stable metrics.

Table 5 Classification Report

	precision	recall	f1-score	support
0	0.93	0.97	0.95	180
1	0.84	0.93	0.88	180
2	0.79	0.81	0.80	191
3	0.99	0.87	0.93	181
4	0.96	0.92	0.94	180
Accuracy			0.90	912
Macro avg	0.90	0.90	0.90	912

Weighted avg	0.90	0.90	0.90	912
--------------	------	------	------	-----

The confusion matrix is presented to further illustrate the model's prediction distribution across all classes:

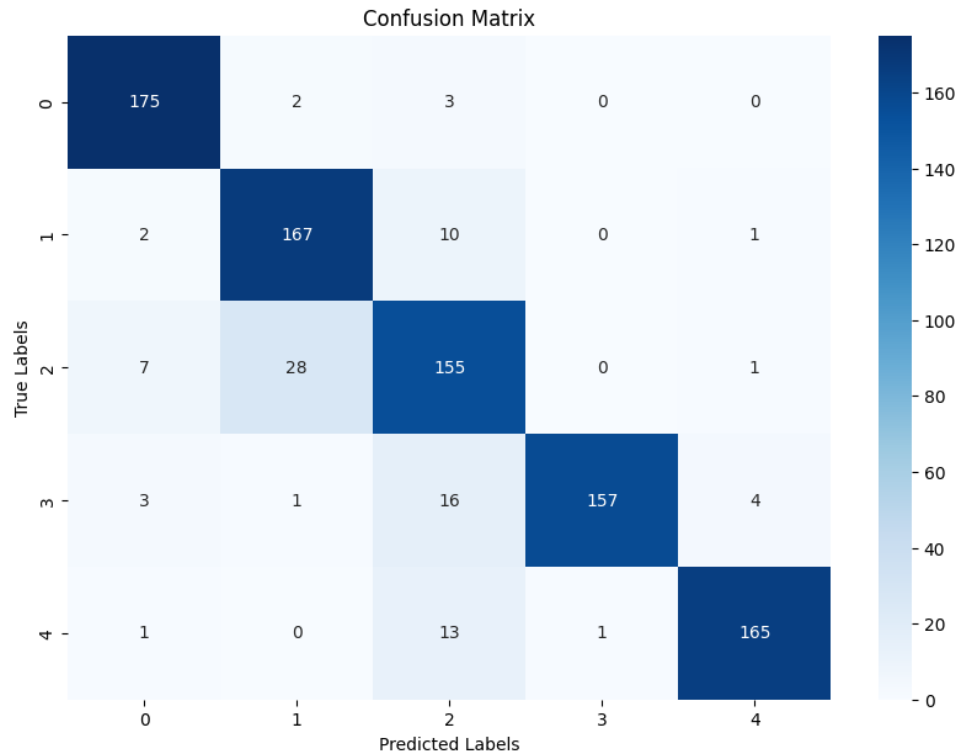


Figure 18 Confusion Matrix of Gu\_CNN on Aptos2019 dataset

From the confusion matrix, the model's prediction performance is excellent on most of the categories, and most of the samples are accurately categorized. Category 0 (normal), category 1 (mild DR), category 3 (severe DR), and category 4 (proliferative DR), the number of correct classifications dominated absolutely. It should be noted that there is a certain degree of confusion in category 2 (moderate DR), with more misclassifications mainly with category 1, probably because these two categories are closer in terms of feature performance, which leads to an increase in the difficulty of model discrimination.

Receiver Operating Characteristic (ROC) curves for each class were plotted to analyze the sensitivity and specificity of the model at various thresholds:

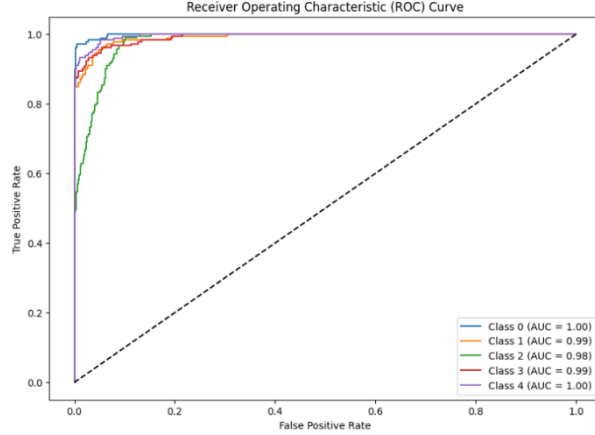


Figure 19 ROC curves of Gu\_CNN on Aptos2019 dataset

In the ROC curve analysis, the model shows very high differentiation ability for all categories, with AUC values higher than 0.98, among which the AUC of category 0 and category 4 reaches 1.00. The overall curves are close to the upper left corner, which indicates that the model has excellent sensitivity and specificity, and is able to differentiate between samples of different categories effectively.

In addition, Precision-Recall curves were generated to assess model performance under class imbalance conditions:

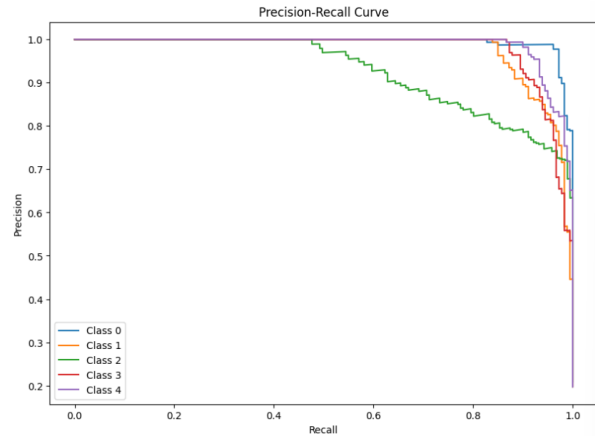


Figure 20 Precision-Recall curves of Gu\_CNN on Aptos2019 dataset

The stability of the model under positive and negative sample imbalance can be further verified by Precision-Recall (PR) curve analysis. The PR curves for category 0, category 1 and category 4 are close to the upper right corner, showing a good balance between high precision and high recall. Category 2 shows a certain magnitude of Precision decrease in the high recall interval, indicating an increase in the false positive rate when recalling moderate DR samples, which is consistent with the performance in the confusion

matrix, suggesting that category 2 can be optimized specifically for category 2 in the future, e.g., sample enhancement, category weight adjustment, etc.

Finally, the training process was monitored by recording the loss and accuracy curves over epochs, which provide insight into the model's convergence behavior and training dynamics.

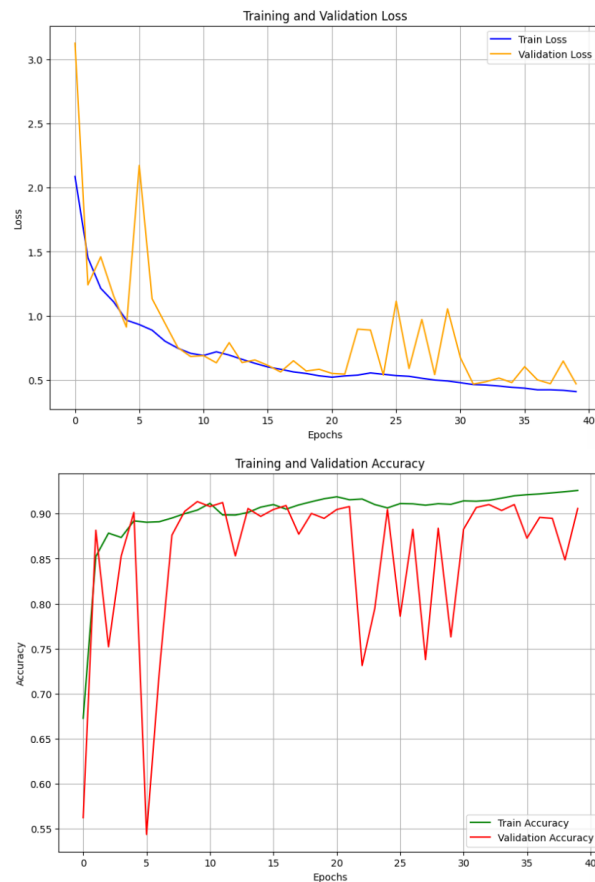


Figure 21 Training and Validation Loss(up) and Accuracy(down) curves of Gu\_CNN on Aptos2019 dataset

The loss curves figure shows that the training loss (blue curve) starts at about 2.0 and gradually decreases and levels off as the number of training cycles increases, eventually stabilizing at about 0.5. The validation loss (yellow curve) is similarly high at first, at about 3.0, but declines more rapidly, decreasing rapidly over the first few cycles and then fluctuating between about 0.5 and 1.5, showing greater instability. A significant peak in validation loss occurs at about cycle 5, after which it declines but still shows large fluctuations throughout the training process.

The Accuracy curves figure on the right side shows the trend of the model's training accuracy and validation accuracy over 40 training cycles. It can be seen that the training accuracy (green curve) starts at about 0.70 and gradually rises and stabilizes as the training progresses, eventually approaching 0.95, showing that the model's performance on the training set continues to improve. The validation accuracy (red curve) starts low at about 0.55, but rises rapidly in the first few cycles and then fluctuates between 0.85 and 0.95, showing some instability. Despite the overall upward trend in validation accuracy, there are several significant decreases during training, especially around cycles 5, 25, and 30.

## **4.7 Self-Supervised Model**

To compare with the supervised model, a self-supervised model with a similar architecture is introduced. Although the self-supervised model shares many architectural similarities with the supervised model, it differs in the training approach, as the self-supervised model uses Masked Autoencoding (MAE) instead of the traditional label-based learning process.

### **4.7.1 Differences in Self-Supervised model with Supervised Model**

Despite many similarities in the architectures of the self-supervised and supervised models, there are significant differences in their training methods and objectives. First, the self-supervised model utilizes the Masked Autoencoding (MAE) architecture, which involves randomly masking parts of the input image, forcing the model to learn how to reconstruct these missing parts and thereby learn useful image features. This process is fundamentally different from traditional supervised learning, where the model learns from labeled data.

In the design of the self-supervised model, the input image is randomly masked, and the masked image is fed into the network for encoding. The encoder extracts features from the image, and the decoder attempts to reconstruct the original image. The training process optimizes the model by minimizing the reconstruction error (such as Mean Squared Error, MSE), rather than using traditional classification loss functions (such as Cross-Entropy). Unlike supervised models that focus on classification tasks, the self-supervised model's objective is to extract features from unlabeled data through self-learning.

Additionally, the decoder structure in the self-supervised model differs from that of the supervised model. The output layer in supervised models is typically a fully connected

layer that maps the features to the final class labels, whereas the output layer in the self-supervised model is a decoder that progressively restores the image details through transposed convolutions, performing an image reconstruction task instead of classification.

#### 4.7.2 Self-Supervised Model Results

The self-supervised model achieved an overall accuracy of 91%, demonstrating robust performance across various severity levels of diabetic retinopathy. It showed strong precision and recall in detecting both mild (Class 1) and severe (Class 3) diabetic retinopathy. The model's precision and recall ranged from 0.93 to 0.99, with macro-average precision, recall, and F1-score all reaching 0.94. While the model performed well overall, its performance was slightly weaker in detecting Class 2 (moderate DR), where precision was lower compared to other classes. The specific classification result report is presented in Table 6:

Table 6 Classification Report for Self-Supervised Model

	precision	recall	f1-score	support
0	0.93	0.97	0.95	361
1	0.92	0.90	0.91	361
2	0.78	0.92	0.84	381
3	0.99	0.90	0.94	361
4	0.98	0.86	0.92	361
Accuracy			0.91	1825
Macro avg	0.92	0.91	0.91	1825
Weighted avg	0.92	0.91	0.91	1825

The accuracy and loss curves (figure 22) demonstrate the model's effective learning process. The accuracy curve shows a steady increase during training, reaching a peak around 91%, with minimal fluctuation between training and validation accuracy. The validation accuracy closely mirrors the training accuracy, indicating that the model generalizes well and avoids overfitting. In contrast, the loss curve steadily decreases over epochs, indicating that the model is successfully minimizing reconstruction error and

learning the relevant features from the data. But unfortunately, the loss rate eventually stabilized above 0.6

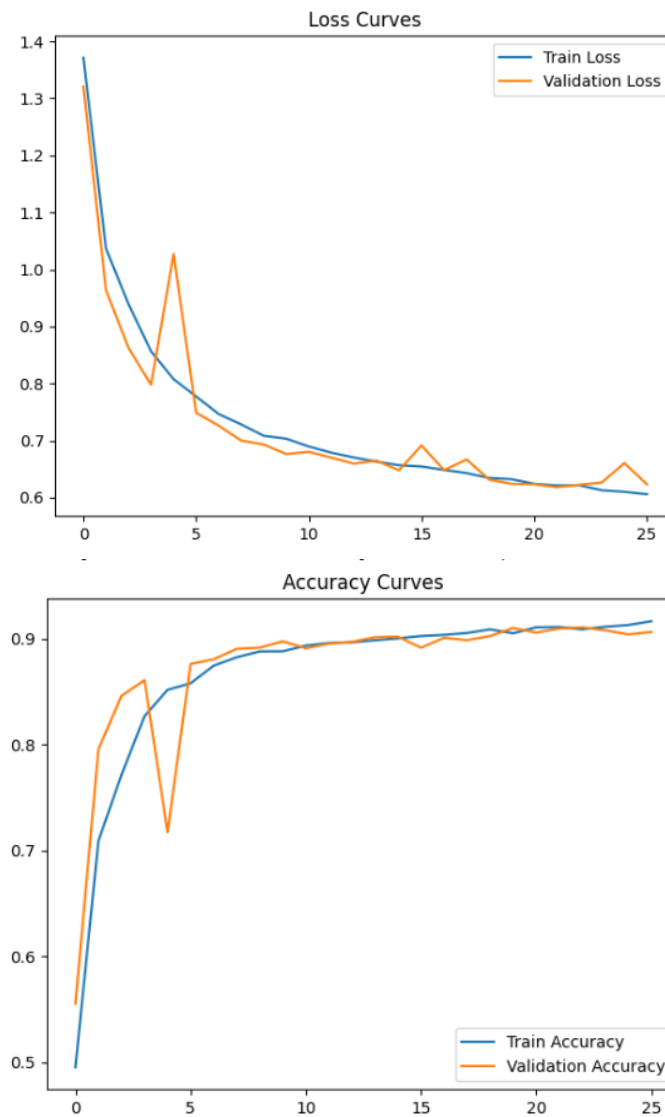


Figure 22 Training and Validation Loss(up) and Accuracy(down) curves of Self-supervised model on Aptos2019 dataset

The confusion matrix (Figure 23) indicates that the self-supervised model performs excellently in identifying Class 0 and Class 3, with a few misclassifications occurring mainly between Classes 1, 2, and 4. The majority of predictions fall on the diagonal, showing that the model correctly classifies most of the instances, though some errors remain due to the complexities of differentiating between similar classes.



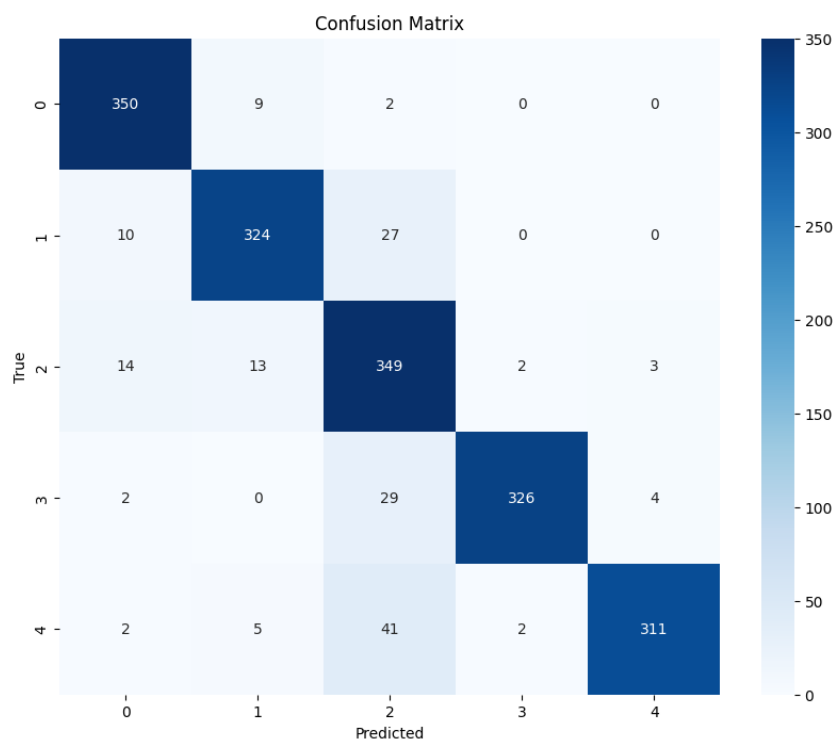


Figure 23 Confusion Matrix of Self-supervised model on Aptos2019 dataset

The precision-recall curves (figure 24) shows that the self-supervised model performs well in maintaining high precision while ensuring a relatively high recall, especially for Class 0 and Class 3. The curves show that the model is particularly effective in detecting the absence of DR and severe DR, with area under the curve (AP) values above 0.90 for all classes.

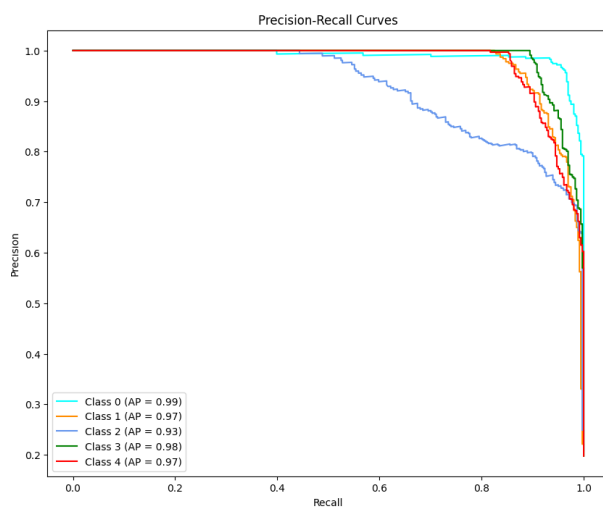


Figure 24 Precision-Recall curves of Self-supervised model on Aptos2019 dataset

The multiclass ROC (figure 25) curve shows the model exhibits high performance for all classes, with an Area Under the Curve (AUC) close to 1.0 for all categories, especially for Class 0 (no DR) and Class 1 (mild DR).

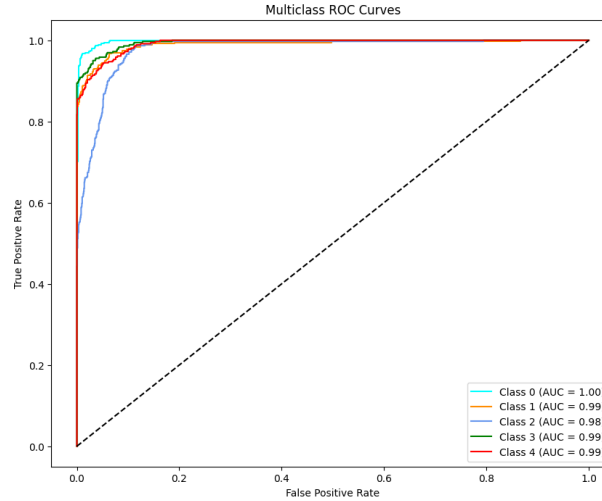


Figure 25 ROC curves of Self-supervised model on Aptos2019 dataset

#### 4.8 Result comparison

Besides of all the result shown in the first few sections, the training time of the self-supervised model was much longer than the supervised model. While the supervised model required approximately 6 to 8 hours for training, the self-supervised model took more than 12 hours to complete its training. When come in to performance, the self-supervised model achieved a slightly higher accuracy, with a difference of about 0.02, but a much higher loss rate of above 0.6, which was 0.1 higher than the supervised model's loss. Despite this, the self-supervised model was still a viable approach but less efficient in this project.

As short, due to the faster training time and an acceptable performance, the supervised model was selected to proceed with the classification task for diabetic retinopathy, ensuring a balance between efficiency and accuracy for real-world application.

#### 4.9 Model Visualization

Grad-CAM (Gradient-weighted Class Activation Mapping) is a technique used to visualize the focus areas of Convolutional Neural Networks (CNNs) during classification, improving the interpretability of the model. It generates heatmaps that highlight regions in an image that the model considers important for making its prediction. In these heatmaps, blue areas represent regions with less focus, while red areas indicate where the model is

paying the most attention.[28] In the provided figure 22 of diabetic retinopathy (DR), Grad-CAM shows how the model focuses on different areas depending on the severity of DR. For normal and mild DR images, the model's attention is concentrated on the central and outer regions of the retina, where abnormalities are less visible. However, in moderate and severe DR images, the heatmaps reveal that the model focuses more on areas with significant lesions, such as exudates and hemorrhages, reflecting the model's ability to identify more prominent signs of the disease. This technique provides valuable insights into the model's decision-making process, increasing its transparency and helping users understand which features are most influential in the diagnosis.

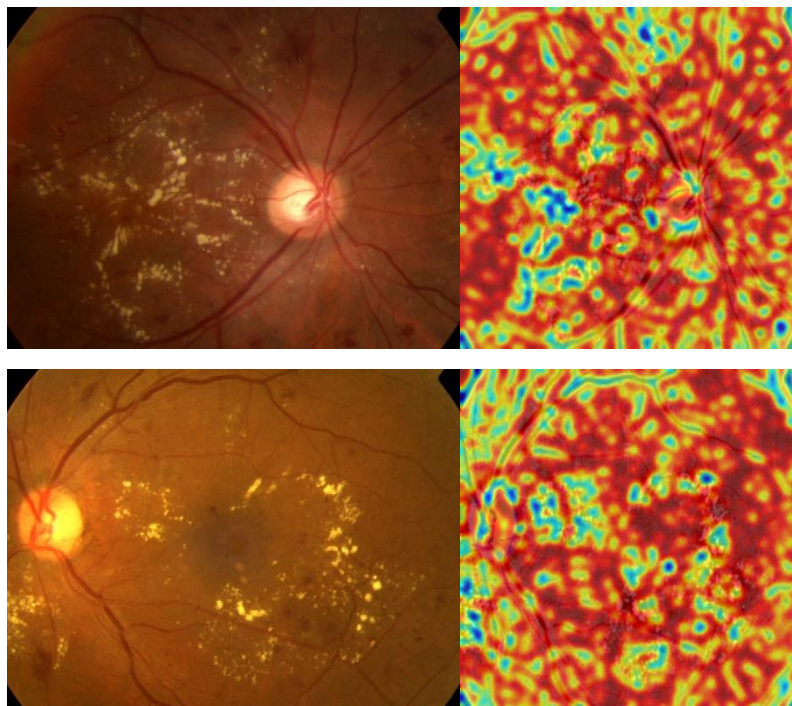


Figure 26 Grad-CAM Visualization

#### 4.10 GUI

Building upon the robust model architecture discussed earlier, a graphical user interface (GUI) was developed to facilitate user interaction with the diabetic retinopathy detection system. The GUI, which is shown in figure 23-25, serves as an intuitive interface for healthcare professionals and other users to upload retinal images, initiate the classification process, and view results in real-time. By leveraging the trained Gu\_CNN model, the GUI provides an easy-to-use platform for predicting the severity of diabetic retinopathy, allowing users to understand and visualize the model's output.

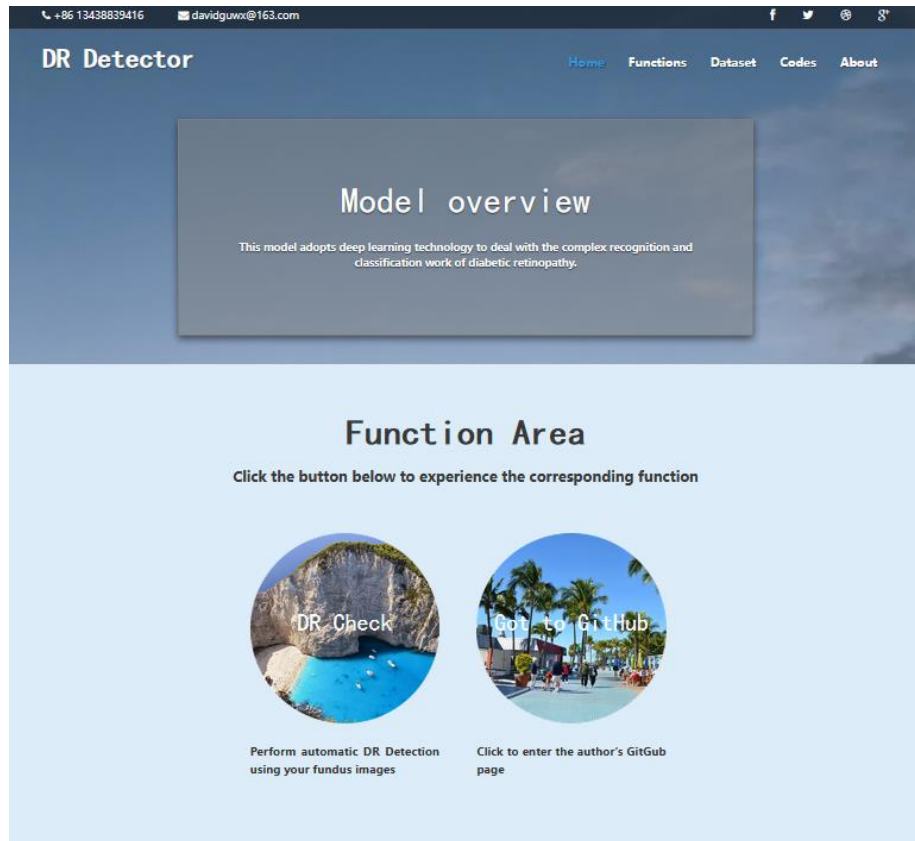


Figure 27 Home Page



Figure 28 Dataset shown in home page

By clicking on the “DR Check” button in the home page(Figure 23), the user can entre the DR check page. In the DR check page which is shown in figure25, the user can simply upload the fundus image by clicking on the green button “Chose File” and select. After that, with one click on the “Start Checking” button, the classification results will be displayed on the right side. Users can compare the results given on the web page to determine whether they have DR.

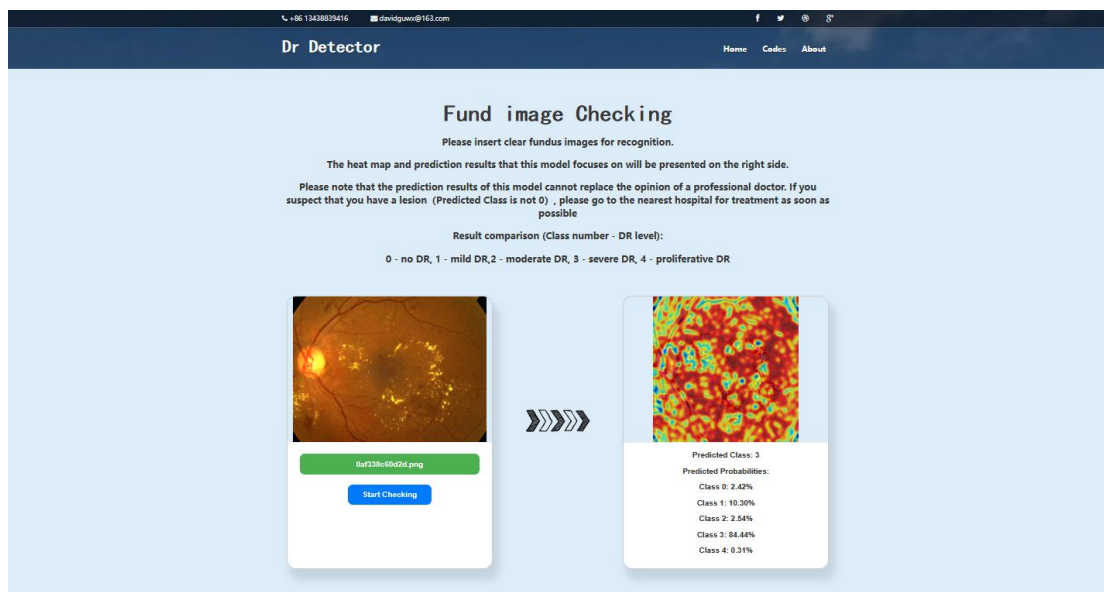


Figure 29 DR Check page

## Chapter 5 Professional Issues

### 5.1 Project Management

This section outlines the various phases and activities that were undertaken to manage the project efficiently. From the initial preparation, where foundational knowledge was accumulated, to the final testing and reporting stages, the project was systematically organized to meet its objectives. The following subsections will detail the specific activities, schedule, and data management practices that guided the project to its successful completion.

#### 5.1.1 Activities

Table 7 Table of Activities

Phase	Objectives	Status
<b>1. Preparation</b>	<ol style="list-style-type: none"><li>1. Review DR deep learning.</li><li>2. Learn matrix operations, vector spaces, eigenvalues and eigenvectors, and basic probability and statistics</li><li>3. Study classification methods, Understanding the difference between supervised and unsupervised learning.</li><li>4. Understand the principles of neurons, activation functions, loss functions, forward and backward propagation.</li><li>5. Learn the principles of convolutional, pooling, and fully connected layers, and the application of CNNs in image processing.</li></ol>	Done
<b>2. Accumulation of further knowledge</b>	<ol style="list-style-type: none"><li>1. Research how VGG19 and ResNet worked in image classification.</li></ol>	Done

	<p>2. Research how DR conditions are classified at various levels.</p> <p>3. Find and delve into at least four CNN models (e.g., VGG16 and 19, variants of ResNet) and related programming libraries that can be leveraged.</p> <p>4. Learn the definitions, principles and modifications of loss functions, optimizers, model building and optimization.</p> <p>5. Research into measures such as GANs to assist in training models.</p>	
<b>3. Data collection</b>	<p>1. Gather at least 3 DR datasets from Kaggle.</p> <p>2. The dataset was classified into 5 categories: No DR, Mild, Moderate, Severe, Proliferative DR.</p> <p>3. Setting multiple test set and training set ratios</p>	Done
<b>4. Actual development and implementation</b>	<p>1. Build VGG19&amp;ResNet mix model.</p> <p>2. Trying to deepen the model</p> <p>3. Repeat training, analyze, compare results and improve.</p> <p>4. Optimize the selected model and continuously adjust the hyperparameters until optimal.</p>	Done
<b>5. Final testing and other work</b>	<p>1. Testing with other datasets to detect generalization ability.</p>	Done

	<p>2. Summarize all results, comparatively analyze and record.</p> <p>3. Finish up Project Report and Prepare for the presentation</p>	
--	--	--



### 5.1.2 Schedule

The time schedule is shown below

In the table3 below, 1-1 stands for Phase 1 Objective1, 1-2 stands for Phase 1 Objective 2 et.al. All tasks are shown in table 6.

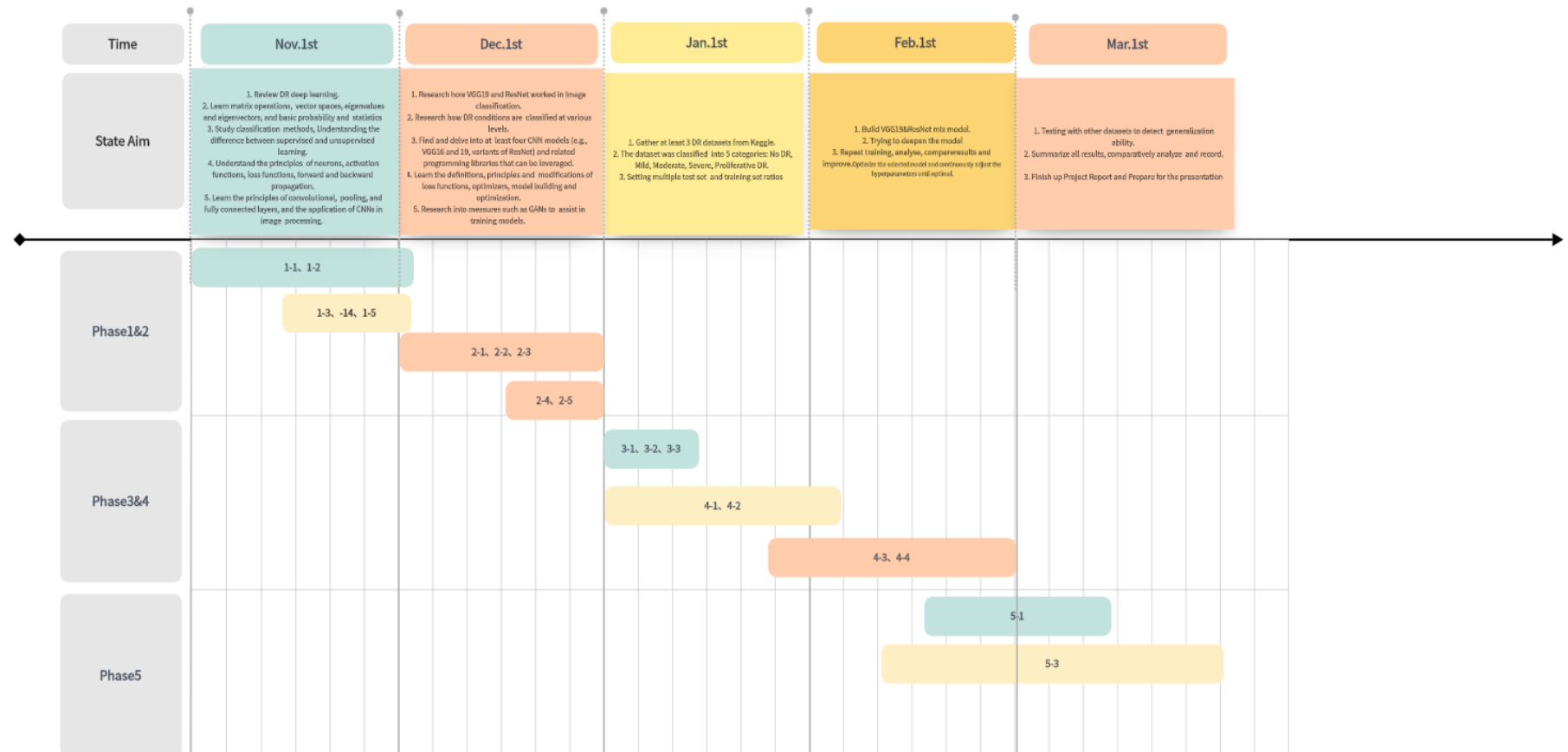


Figure 30 Gantt Chart

### 5.1.3 Project Data Management

Save all files (including: datasets, various reports {proposal, weekly report, progress report, final report}, thesis, drawn images as well as source files, different versions of the model code, test logs) on a laptop, desktop computer, GitHub, and USB stick respectively. Synchronize data on GitHub once a day, on the hard disk every two days, and sync all files to the desktop computer every Friday.

The GitHub link is: <https://github.com/Blacknomizi/DR-detect>

### 5.1.4 Project Deliverables

- A. Project proposal
- B. Progress Report
- C. Final Project Report
- D. Project codes
- E. Project presentation slides
- F. Poster presentation file

## 5.2 Risk Analysis

In the process of developing and deploying the self-supervised learning model for retinal disease classification, several risks must be considered. These risks span across technical, resource, and schedule aspects:

**Technical Risks:** Deep learning models for retinal disease classification, particularly those based on self-supervised learning, face substantial technical challenges. These include the difficulty of achieving high accuracy with limited labeled data, the risk of overfitting, and the need for fine-tuning model architectures to handle complex medical image data. Additionally, training such models requires significant computational resources, which may lead to long training times and high costs.

**Resource Risks:** Access to high-quality, annotated fundus images is crucial for training accurate models. However, medical datasets can be scarce and expensive to acquire. Moreover, there may be challenges in securing necessary computational resources, such as GPUs for training the model, which could affect both the progress and performance of the project.

**Schedule Risk:** The inherent uncertainty in model development and the potential delays in obtaining data may lead to schedule risks. Adjustments may be required in case of

unforeseen difficulties, such as slow model convergence, hardware failures, or challenges in obtaining sufficient labeled data.

**Mitigation Strategies:** To mitigate technical risks, one approach could be to adopt pre-trained models and fine-tune them with the available data, which may reduce training time and improve model robustness. Resource risks can be addressed through collaboration with medical institutions or by leveraging publicly available datasets. To manage schedule risks, regular progress reviews and a flexible timeline should be implemented, ensuring the project can adapt to changes without significant delays.

**Impact of Addressed Risks:** Addressing these risks might lead to an optimized approach, including the use of data augmentation techniques to reduce the dependency on large datasets, as well as the exploration of lightweight models to reduce computational demands.

**Future Risks:** Potential future risks include the emergence of new technologies that may require revisiting the model architecture, the need for continuous data collection to adapt to evolving disease patterns, and the possibility of research staff turnover. Establishing long-term collaborations and ensuring ongoing access to necessary resources will help mitigate these risks.

### 5.3 Professional Issues

**Legal Issues:** In the development of a self-supervised learning model for retinal disease classification, it is essential to comply with regulations regarding data privacy and security. Medical data, especially images of patients' eyes, is sensitive and must adhere to laws such as GDPR (General Data Protection Regulation) or HIPAA (Health Insurance Portability and Accountability Act) to protect personal privacy. Furthermore, intellectual property issues related to the use of pre-trained models and datasets must be carefully managed.

**Social Issues:** The deployment of retinal disease classification systems powered by deep learning raises concerns about the ethical use of technology in medical diagnoses. Public perception of AI in healthcare can lead to concerns about job displacement for medical professionals and the reliability of AI-based decisions. These issues must be addressed by demonstrating the transparency and accountability of the AI system.

**Ethical Issues:** Ethical concerns are paramount when working with medical data. Informed consent from patients for the use of their medical images must be obtained, and data should only be used for the intended purposes. Additionally, fairness and equity must be considered to avoid biased model predictions, particularly in diverse populations. Steps should be taken to ensure that the model does not perpetuate or amplify existing healthcare disparities.

**Environmental Issues:** Training deep learning models requires substantial computational power, which can contribute to high energy consumption and environmental impact. To minimize the ecological footprint, the model development should prioritize energy-efficient computing resources, and consider offloading some computational tasks to cloud services that utilize renewable energy sources. Furthermore, regular evaluations of the environmental impact of training and model deployment should be conducted to ensure sustainability.

## Chapter 6 Conclusion

This work proposed a CNN model for diabetic retinopathy (DR) classification, which combined VGG and ResNet architectures to extract low-level and high-level features, and feature recalibration using squeeze and excite (SE) attention. The model was trained on the preprocessed APTOS 2019 dataset (3,662 samples). The preprocessing techniques including image cropping, enhancement, class rebalancing, and data enhancement (rotation, scaling, and flipping). The model has a test accuracy of 89.80%, an F1 score of 0.8988, a precision of 0.9026, a recall of 0.8980, and a macro-mean of all metrics of 0.90. The training loss is 0.4624 and the validation accuracy is 90.68%. The confusion matrix showed strong categorization results, especially for severe and proliferative DR (grades 3 and 4), while the AUC values of the ROC curves were above 0.98 for all categories. The precision recall curves confirmed the balanced performance of the model, especially for the less represented categories. These results highlight the robustness of the model and reliable generalization across data subsets.

Although some achievements have been made, this study still has some limitations. A key issue is data imbalance, especially in severe and diffuse DR cases. Although data augmentation was used to solve this issue, the model showed lower performance in the moderate DR category, possibly due to subtle feature differences that make it difficult to distinguish between adjacent stages of DR, or overfitting due to a large number of expansions of the same fundus image. Since only one dataset was used, the reliance on the APTOS 2019 dataset may limit the model's ability to handle various imaging variations such as illumination, resolution, or other environmental factors. Further research is needed to evaluate the robustness of the model in different environments and with different data sources.

Future work will be focused on improving the performance and adaptability of the model. Continue to improve the self-supervised learning models with better results and shorter training times, in order to achieve a further reduction in training costs by utilizing large amounts of unlabeled data. Additionally stronger regularization methods and optimized learning rate strategies will be considered to reduce losses. In addition, improved interpretability through explainable artificial intelligence (XAI) approaches to facilitate clinical deployment efforts, as healthcare professionals and general users need to trust and understand AI-driven predictions.

## References

Papers:

- [1] Amin, Javeria, Sharif, Muhammad, Yasmin, Mussarat, A Review on Recent Developments for Detection of Diabetic Retinopathy, Scientifica, 2016, 6838976, 20 pages, 2016. <https://doi.org/10.1155/2016/6838976>.
- [2] M. Z. Atwany, A. H. Sahyoun and M. Yaqub, "Deep Learning Techniques for Diabetic Retinopathy Classification: A Survey," in IEEE Access, vol. 10, pp. 28642-28655, 2022, doi: 10.1109/ACCESS.2022.3157632. <https://doi.org/10.1109/ACCESS.2022.3157632>.  
M. Z. Atwany, A. H. Sahyoun and M. Yaqub, "Deep Learning Techniques for Diabetic Retinopathy Classification: A Survey," in *IEEE Access*, vol. 10, pp. 28642-28655, 2022, doi: 10.1109/ACCESS.2022.3157632. <https://ieeexplore.ieee.org/document/9729867>.
- [3] Y. Niu, L. Gu, Y. Zhao and F. Lu, "Explainable Diabetic Retinopathy Detection and Retinal Image Generation," in IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 1, pp. 44-55, Jan. 2022, doi: 10.1109/JBHI.2021.3110593. <https://ieeexplore.ieee.org/document/9531456>.
- [4] S. Yang, J. Zhang, L. Chen. The cells involved in the pathological process of diabetic retinopathy, *Biomedicine & Pharmacotherapy*, Volume 132, 2020, 110818, ISSN 0753-3322, <https://doi.org/10.1016/j.biopha.2020.110818>.
- [5] Vora, P.; Shrestha, S. Detecting Diabetic Retinopathy Using Embedded Computer Vision. *Appl. Sci.* 2020, 10, 7274. <https://doi.org/10.3390/app10207274>
- [6] N.H. Cho, J.E. Shaw, S. Karuranga, Y. Huang, J.D. da Rocha Fernandes, A.W. Ohlrogge, B. Malanda, IDF Diabetes Atlas: Global estimates of diabetes prevalence for 2017 and projections for 2045, *Diabetes Research and Clinical Practice*, Volume 138, 2018, Pages 271-281, ISSN 0168-8227, <https://www.sciencedirect.com/science/article/pii/S0168822718302031>.
- [7] S. Mishra, S. Hanchate and Z. Saquib, "Diabetic Retinopathy Detection using Deep Learning," 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 2020, pp. 515-520, doi: 10.1109/ICSTCEE49637.2020.9277506. <https://ieeexplore.ieee.org/document/9277506>

- [8] S. Agarwal, A. Bhat, A survey on recent developments in diabetic retinopathy detection through integration of deep learning. *Multimed Tools Appl* 82, 17321–17351 (2023). <https://doi.org/10.1007/s11042-022-13837-5>
- [9] C. Kabilan, S. Madhesh Kumar and G. Latha Selvi, "The Role of Fundus Imaging and GAN in Diabetic Retinopathy Classification using VGG19," 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2024, pp. 1-5, doi: 10.1109/ACCAI61061.2024.10601890. <https://ieeexplore.ieee.org/document/10601890>
- [10] R. S. Rajkumar, T. Jagathishkumar, D. Ragul and A. G. Selvarani, "Transfer Learning Approach for Diabetic Retinopathy Detection using Residual Network," 2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2021, pp. 1189-1193, doi: 10.1109/ICICT50816.2021.9358468. <https://ieeexplore.ieee.org/document/9358468>.
- [11] Dai, L., Wu, L., Li, H. et al. A deep learning system for detecting diabetic retinopathy across the disease spectrum. *Nat Commun* 12, 3242 (2021). <https://doi.org/10.1038/s41467-021-23458-5>
- [12] Simó, R., Stitt, A.W. & Gardner, T.W. Neurodegeneration in diabetic retinopathy: does it really matter?. *Diabetologia* 61, 1902–1912 (2018). <https://doi.org/10.1007/s00125-018-4692-1>
- [13] Jo-Hsuan Wu, T Y Alvin Liu, Wan-Ting Hsu, Jennifer Hui-Chun Ho, Chien-Chang Lee, Performance and Limitation of Machine Learning Algorithms for Diabetic Retinopathy Screening: Meta-analysis, *Journal of Medical Internet Research*, Volume 23, Issue 7, 2021, ISSN 1438-8871, <https://doi.org/10.2196/23863>.
- [14] D. Yuan, J. Huang, X. Yang and J. Cui, "Improved random forest classification approach based on hybrid clustering selection," 2020 Chinese Automation Congress (CAC), Shanghai, China, 2020, pp. 1559-1563, doi: 10.1109/CAC51589.2020.9326711. <https://ieeexplore.ieee.org/document/9326711>
- [15] G. Priyadharshini and D. R. Judie Dolly, "Comparative Investigations on Tomato Leaf Disease Detection and Classification Using CNN, R-CNN, Fast R-CNN and Faster R-CNN," 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023, pp. 1540-1545, doi: 10.1109/ICACCS57279.2023.10112860. <https://ieeexplore.ieee.org/document/10112860>

- [16] A. Vaish, D. Singh, A. Garg, A. Tiwari and A. Rathore, "Classification of Diabetic Retinopathy Using Improved ResNet-50 Deep Learning Model," 2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 2024, pp. 1-5, doi: 10.1109/ICDCECE60827.2024.10549184.  
<https://ieeexplore.ieee.org/document/10549184>
- [17] A. He, T. Li, N. Li, K. Wang and H. Fu, "CABNet: Category Attention Block for Imbalanced Diabetic Retinopathy Grading," in IEEE Transactions on Medical Imaging, vol. 40, no. 1, pp. 143-153, Jan. 2021, doi: 10.1109/TMI.2020.3023463.  
<https://ieeexplore.ieee.org/document/9195035>.
- [18] G. Deshpande, Y. Govardhan and A. Jain, "Machine Learning-Based Diabetic Retinopathy Detection: A Comprehensive Study Using InceptionV3 Model," 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), Manama, Bahrain, 2024, pp. 994-999, doi: 10.1109/ICETSIS61505.2024.10459541.  
<https://ieeexplore.ieee.org/document/10459541>.
- [19] A. Matthew, A. A. S. Gunawan and F. I. Kurniadi, "Diabetic Retinopathy Diagnosis System Based on Retinal Biomarkers Using EfficientNet-B0 for Android Devices," 2023 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT), Malang, Indonesia, 2023, pp. 207-212, doi: 10.1109/COMNETSAT59769.2023.10420736.  
<https://ieeexplore.ieee.org/document/10420736>.
- [20] G. Deshpande, Y. Govardhan and A. Jain, "Machine Learning-Based Diabetic Retinopathy Detection: A Comprehensive Study Using InceptionV3 Model," 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), Manama, Bahrain, 2024, pp. 994-999, doi: 10.1109/ICETSIS61505.2024.10459541.  
<https://ieeexplore.ieee.org/document/10459541>
- [21] S. Dasari, B. Poonguzhali and M. Rayudu, "Transfer Learning Approach for Classification of Diabetic Retinopathy using Fine-Tuned ResNet50 Deep Learning Model," 2023 International Conference on Sustainable Communication Networks and Application (ICSCNA), Theni, India, 2023, pp. 1361-1367, doi: 10.1109/ICSCNA58489.2023.10370255.  
<https://ieeexplore.ieee.org/document/10370255>.



- [22] Y. Tao, "Image Style Transfer Based on VGG Neural Network Model," *2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, Dalian, China, 2022, pp. 1475-1482, doi: 10.1109/AEECA55500.2022.9918891.  
<https://ieeexplore.ieee.org/document/9918891>
- [23] C. Kabilan, S. Madhesh Kumar and G. Latha Selvi, "The Role of Fundus Imaging and GAN in Diabetic Retinopathy Classification using VGG19," *2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, Chennai, India, 2024, pp. 1-5, doi: 10.1109/ACCAI61061.2024.10601890.  
<https://ieeexplore.ieee.org/document/10601890>
- [24] S. Dasari, B. Poonguzhali and M. Rayudu, "Transfer Learning Approach for Classification of Diabetic Retinopathy using Fine-Tuned ResNet50 Deep Learning Model," *2023 International Conference on Sustainable Communication Networks and Application (ICSCNA)*, Theni, India, 2023, pp. 1361-1367, doi: 10.1109/ICSCNA58489.2023.10370255.  
<https://ieeexplore.ieee.org/document/10370255>.
- [25] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.  
<https://ieeexplore.ieee.org/document/7780459>
- [26] V. Deshpande, A. Sivanantham, I. Kapila, N. Singh, J. M and P. R, "Investigating The Potential of Self-Supervised Learning in Adversarial Machine Learning," *2024 IEEE 2nd International Conference on Innovations in High Speed Communication and Signal Processing (IHCSP)*, Bhopal, India, 2024, pp. 1-6, doi: 10.1109/IHCSP63227.2024.10959888.  
<https://ieeexplore.ieee.org/document/10959888>
- [27] W. Xia, F. Yu, H. Wang and R. Hong, "A high-precision lightweight smoke detection model based on SE attention mechanism," *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, Guangzhou, China, 2022, pp. 941-944, doi: 10.1109/ICCECE54139.2022.9712739.  
<https://ieeexplore.ieee.org/document/9712739>
- [28] T. He *et al.*, "MediMLP: Using Grad-CAM to Extract Crucial Variables for Lung Cancer Postoperative Complication Prediction," in *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 6, pp. 1762-1771, June 2020, doi: 10.1109/JBHI.2019.2949601. <https://ieeexplore.ieee.org/document/8883038>

Dataset:

[a]. Asia Pacific Tele-Ophthalmology Society (APTOS). APTOS 2019 Blindness Detection. Available at: <https://www.kaggle.com/c/aptos2019-blindness-detection/data>